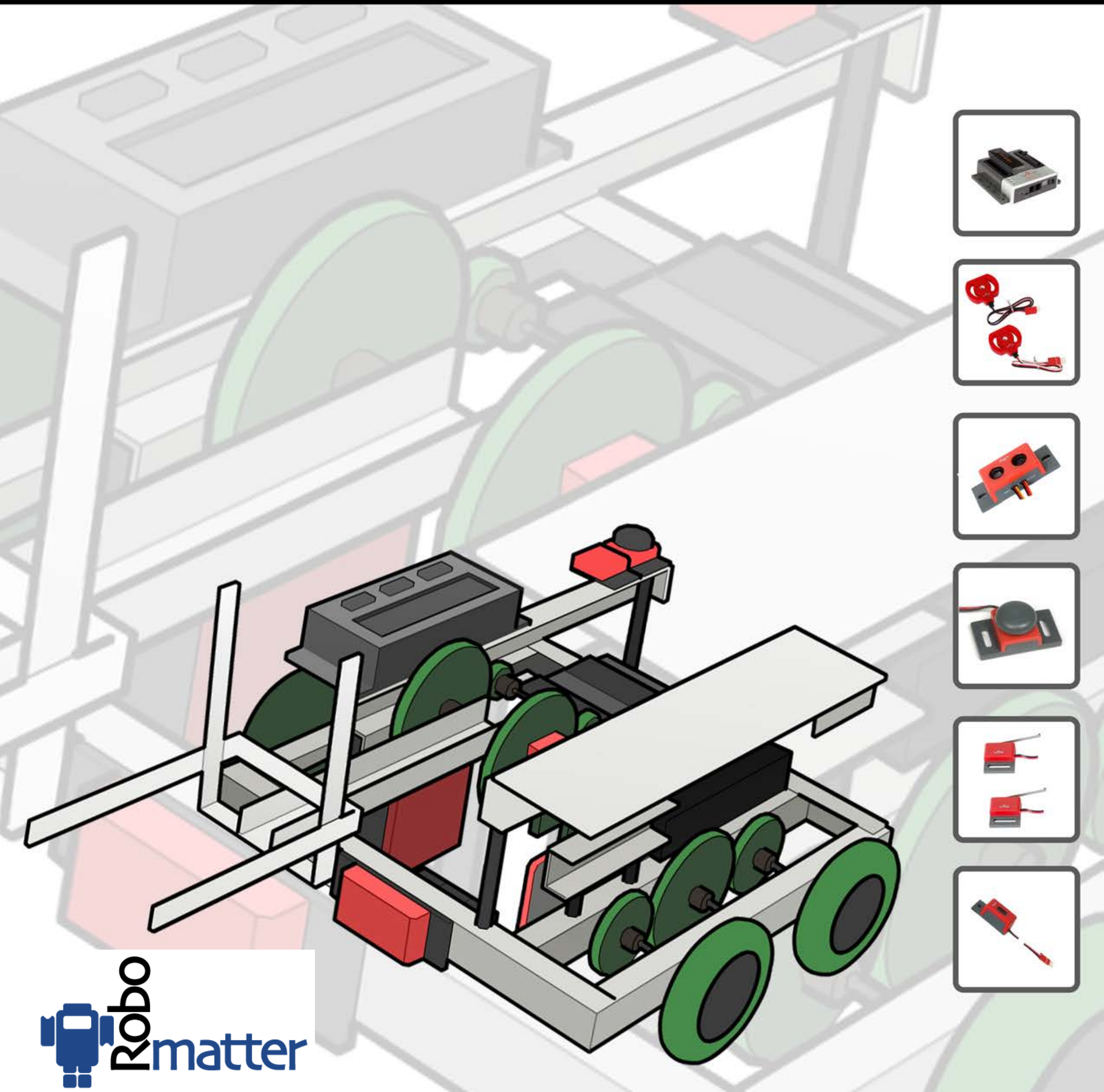
ROBOTC VEX® Cortex®

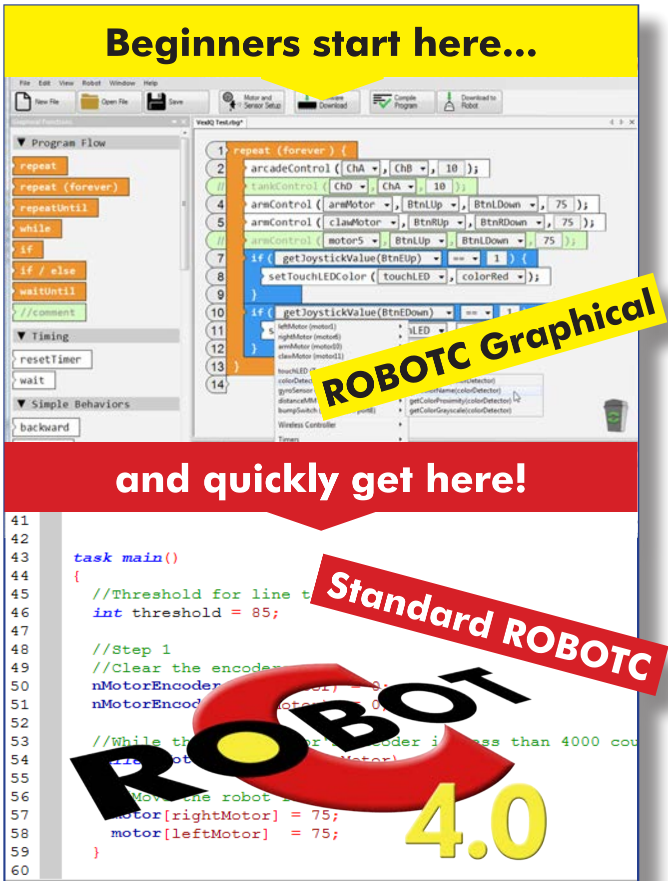
video trainer using ROBOTC

Teacher's Guide



The ROBOTC VEX Cortex Curriculum was produced by Carnegie Mellon's Robotics Academy

ROBOTC Graphical/ROBOTC Text



ROBOTC 4

The Language of Robot Innovation!

ROBOTC 4.0 includes a drag and drop graphical interface that provides a quick way for teachers to introduce students to basic programming logic.

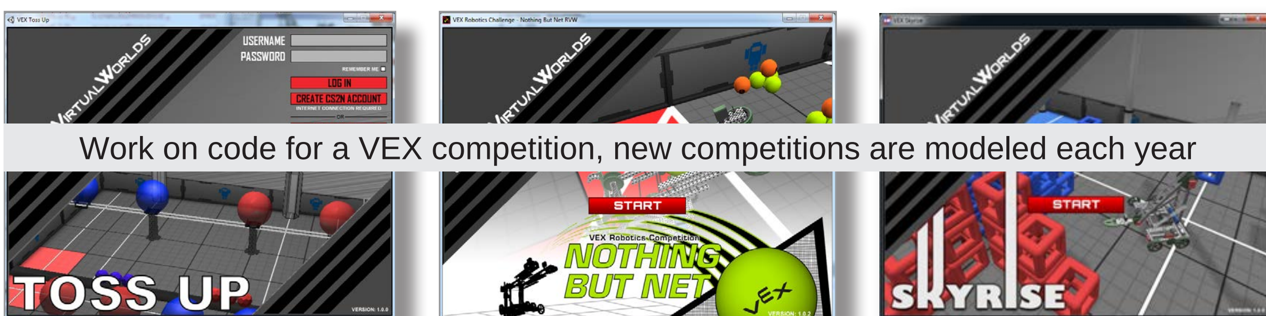
ROBOTC graphical is not intended to replace ROBOTC, but to provide a scaffolded on-ramp to begin to teach programming.

Robot Virtual World Software

Simulation Software that makes a difference!

Robot Virtual Worlds (RVW) has evolved into a game changer in the world of robotics education. RVW enables every student to have their own robot to learn and practice programming. Students build their program in simulation and then test it on a physical robot. Homework packs are priced at less than $5 per student.

Test your code with a virtual robot then use the same code on your physical robot!



Work on code for a VEX competition, new competitions are modeled each year

www.robotvirtualworlds.com

or have kids learn to write code using these fantasy challenges.

Teaching Robotics! What are you going to teach?

If you are new to robotics, or are designing your robotics course it is important to think about what you want to teach. Yes, you are going to teach robotics, but what is it that your course will emphasize? Robotics Academy research shows that any one robotics lesson can cover five or more different concepts, *and* unless you foreground what you want students to learn, that little or no "new measurable learning" will take place. Think about the following:

- Robotics is an eclectic mix of applied science, mathematics, computer science, mechanics, physics, engineering design, sensors, problem solving... what will your course emphasize?
- When kids study robotics they can learn to develop 21st century skills like: cooperation and collaboration, teamwork and problem solving, critical thinking and creativity... Are these skills important to you?
- When students engage in robotics engineering activities the can learn important lessons like: the first step to solving a problem is to conduct research, how to allocate and manage time, the need to develop a set of plans before you start building, the iterative nature of design, how to work with and manage people... What activities will use in your course to teach these lessons?
- Each student that comes into your classroom has a different world experience that they draw from. It is important to scaffold instruction but you don't want to limit creativity. How can your course support students with varying technical backgrounds?

General topics that robotics can teach

Robotics is a relatively new to education, and teachers and administrators have varying reasons why they want to implement a robotics program; we've learned that there is no "one size fits all". As you are building your robotics program, you will want to carefully consider what it is that you want to teach via robotics. Here are a couple of ideas:

- Grade level mathematics - there are many ways to mathematize robots via programming; computing distances, angle, and unit rates, passing parameters in the form of formulas, and applying geometry trigonometry, and physics to a student robot design.
- Reading and writing with understanding - There are many ways to incorporate research, writing, and presentations into robotics courses. When students engage in an engineering design problem they should begin by researching how others have solved the problem in the past. From there they write up what they've learned and if time allows make a presentation. All of these skills tie into the English and Language Arts College and Career Readiness Standards.
- Introductory through advanced programming concepts - Computational thinking and programming are new basics that students must understand. See the K-12 Computer Science Teacher Association Standards and you will find that robotics addresses many of those standards.
- Introduction to engineering - Robotics provides many opportunities for students to apply engineering design principles to solve complex problems that require them to consider trade-offs such as costs, safety, reliability, and aesthetics; they are learning engineering. These skills can be taught at the elementary through college level and aligns with the new Science and Technology standards.
- Teaching technological literacy - Robotic systems require student to engage with computers, sensors, the Internet, electrical mechanical actuators, and a plethora of evolving technologies.
- The development of 21st Century Skills - There are over 35,000 US school based robotics teams. Robotics competitions are purposefully designed to place students in situations where they have to work in teams and problem solve. Properly designed activities develop 21st century skills in students.

Table of Contents

Table of Contents

Frequently Asked Questions

Before starting

- Will *the VEX Cortex Video Trainer* help me teach to Standards?

  Yes! See Appendix

- What do I need to prepare for class?

  See Checklist, page 6; Lesson Structure, page 7; and Classroom Setup page 8.

- What general topics are covered with the curriculum?

  See Topics Covered, page 9 and 11 and the Chapter Overview on pages 17-34.

- How do I use the VEX Cortex Video Trainer in the classroom?

  See Using the VEX Cortex Video Trainer in Class, page 10.

- I want to know what's in each Chapter and Unit, where to I go?

  See Chapter Overview Fundamentals page 17; Setup pages 18-19; movement, pages 19 - 24; Remote Control, pages 25 - 26; Sensing pages 27 - 31; and Engineering pages 32 - 34.

During class

- How do I teach kids to think about programming?

  See "Breaking Programs into Behaviors" and "Sense Plan Act" pages 112 - 113.

- What should I teach and when should I teach it?

  See the Scope and Sequence section, pages 17 - 34.

- What do I do about students who go faster/slower than the others?

  See Differentiated Instruction, page 9.

After class

- How do I prepare my kids for competitions?

  See Robotics Competitions: see the Engineering Section pages 32 - 34 and pages 98 - 101.

- Are there quizzes or homework?

  Each sub unit page includes a series of "Check your understanding" questions that can be used to develop a unit quiz. See Rubrics, pages 102 - 103. Robots are hard to take home, but there are many ways to incorporate research assignments or Robot Virtual World assignments into homework assignments.

Checklist/How do I use this teacher's guide?

The curriculum guide is organized in four sections:

Introduction - pages 7-13 - background information to get started
Scope and Sequence - pages 14-16 - a short description of the curriculum
Chapter Overview - pages 17-34 - a detailed description of the curriculum
Navigating the Curriculum - pages 35-102 - a pictorial description of the curriculum

☐ **Identify the Goals of your Robotics Course**
Robotics can be used to teach to lots of standards. This curriculum is designed to introduce students to how to program, an important part of robotics, but not the only thing that you can teach through robotics. Determine what you want students to learn in your class. See page 3.

☐ **Set up the student workstations**
See page 8, Workstation Setup.

☐ **(Recommended) Build the Clawbot with Sensors**
VEX kit configuration has changed multiple times over the years. There are multiple robot build plans in the Setup section of the curriculum, we recommend that your standard build is Clawbot. We also recommend that if your budget allows it, that you have one set of robots to teach programming and another set of robot parts to teach engineering and for competitions.

☐ **Become familiar with the lessons**
See page 7 to become familiar with the lesson flow. Read the Scope and Sequence, pages 14-16; the Chapter Overview, pages 17 - 34, where each chapter is described in a shortened format, and Navigating the Curriculum, pages 35 - 103, where you will find a pictorial view of what you will find on each page of the VEX Cortex Video Trainer.

☐ **Determine overall pacing for the module**
Identify key dates that you would like to have each project due by; make these clear to students in your syllabus or assignment sheets.

☐ **(Highly Recommended) Read the following lessons on how to teach students about how to think about writing code for robots.**
See Breaking Programs into Behaviors, page 112; Sense Plan Act, page 113; Introduction to Pseudocode, page 114; and Introduction to Flowcharts, page 120.

☐ **(Highly Recommended)** Get involved with the RECF Foundation's annual VEX competition challenge. The challenge changes every year and provides teachers with a brilliant engineering problem for their class to solve each year.

VEX Video Trainer Lesson Structure

Guided Robot Programming Activities, Extension Activities for Advanced Students, and Engineering Investigations

The VEX Cortex Video Trainer Curriculum is designed to teach introductory level robot programming, logic and reasoning skills, and engineering process using robotics at the context. The core curriculum consists of four chapters (Movement, Remote Control, Sensing, and Engineering) and each chapter is broken into units that teach key robotics programming and engineering concepts. Additionally, there is a huge amount of support for teachers coaching teams in Robotics Competitions for the first time; see the engineering section of the curriculum.

Each unit comprises a unit level programming challenge that students will to solve by the end of the unit.

- **ROBOTC RBC Files** RBC files are starter programs that will automatically open once your software is installed and configured, click the file and the starter program will open. Some browsers will require uses to save and then open the file using ROBOTC software.
- **Additional Unit Level Robotic Programming Challenges** and setup guides
- **Step-by-step guided video instruction** that introduces key lesson concepts (e.g. Loops). Students are taught programming using a step-by-step process; foundational programming concepts are integrated into each unit and repeated in subsequent units.
- **Built-in "Check your understanding" questions** designed to provide students with instant feedback on whether they understood the big ideas in each lesson.
- **Reference Guides** that are designed to support the lesson (e.g. white space, comments, loops, conditional statements, Boolean logic, etc.)
- **Engineering Investigations** that guide students through an engineering experiment, the activities are designed to deepen students' understanding of that concept.
- **Robot Virtual World** extension activities. The RVW activities are designed to significantly enhance student's programming opportunities allowing them to program robots underwater, on an island, in outer space, and via a VEX Cortex international competition. Students are able to use the same programming commands on their virtual solution as they do on their physical robot.

Classroom Setup

What is the best setup for student workstations?

Ideally, pairs of students will work together at one computer, with one VEX robot.

Set up each workstation with:

a. **ROBOTC 4 for VEX Software** installed on each computer.
   - Check each computer to see that the software works
   - Check each computer to see that Robot Virtual World software works

b. Access to the **The VEX Cortex Video Trainer Curriculum** software
   - This can be installed locally or on a local network server with proper licensing
   - This may also be accessed remotely via Internet, if your school's network infrastructure and policies allow

c. Two pairs of **headphones** with **headphone splitters**
   - One pair for each student
   - Avoid using speakers, as multiple workstations in the same classroom will generate too much overlapping noise

d. One **VEX Cortex robot kit per work station**

e. **ROBOT Virtual World Software** This software is not required to use the curriculum and complete the lessons, but research shows that it is a very effective tool to teach programming.

What are the System Requirements for the VEX Cortex Video Trainer Curriculum?

Introduction to Programming VEX Curriculum

- HTML5-compatible browser (Firefox, Chrome, Internet Explorer 10+)
- Tablets (iPad, Android, Windows) with HTML5 browsers should work as well when accessing the curriculum from the Internet

Robot Virtual World Software

- PC Compatible OS: with an Intel Core 2 processor family or better
- Memory: 2 GB RAM
- Graphics: NVIDIA® 8800GTS or better, ATI Radeon™ HD 3850 or better
- Hard Drive: 1.5 GB free hard drive space to install all virtual worlds

What does the VEX Cortex Video Trainer Teach?

- How to control basic robot movements
  a. Robot math
  b. Sequences of commands, structures, computational thinking
- Sensors and how they work
- Intermediate concepts of programming
  a. Program Flow Model
  b. Programming Remote Controls
  c. Decision-Making Structures
    - Loops
    - Conditionals
    - Functions
- Teach troubleshooting strategies and engineering practices
  a. Problem-solving strategies
  b. Teamwork
- The iterative nature of engineering design and process

Differentiated Instruction

One of the biggest challenges facing teachers today is meeting the needs of each individual student in their classroom; that is the core of differentiated instruction. Differentiated instruction asks teachers to approach students at their instructional level, and requires students to show evidence of growth from their instructional level. Differentiated instruction encompasses more than just assessment. It involves all aspects of instruction: classroom delivery, overall learning environment, learning content, and assessment. The VEX Cortex Video Trainer provides many opportunities for students of all abilities:

- Programming - the unit challenges are supported by step-by-step instructional videos that students can work through at their own pace.
- Solving the open-ended programming challenges embedded into the units that make up the Movement, Remote Control, Sensing, and Engineering Units.
- Completing the virtual programming challenges found in the Robot Virtual World games (Ruins of Atlantis, Palm Island, Operation Reset, Highrise): attempt to complete the entire world, or choose to program different robots within a Virtual World.
- Challenging gifted students to iteratively improve their engineering and programming solutions using ROBOTC.
- Working cooperatively with students having difficulty grasping some concepts.
- Engaging in engineering challenges that are found in robotics competitions.

Using the VEX Cortex Video Trainer in Class

The VEX Cortex Video Trainer Curriculum is designed for student self-pacing in small groups, preferably pairs that are working together at one computer, with one VEX Cortex robot. It can also be used in “virtual mode” where students are learning programming using a virtual robot that is programmed using the exact same commands that they will use on their actual VEX Cortex robot.

Programming tasks are designed to involve some – but not extensive – mechanical consideration, so that hands-on design tasks may remain authentic without becoming logistically difficult. The Engineering section enables larger teams and requires more building.

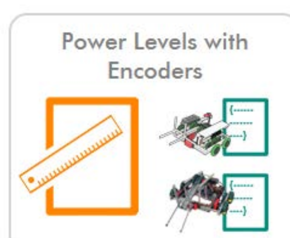
Solutions will not need parts in excess of those included in the VEX Cortex core set, so it is sufficient to leave each team with one kit (although access to additional parts may allow students to construct more creative solutions to problems).

A typical plan for a programming unit (movement, remote control, and sensing) is:

1. Review the unit challenge which is always the first lesson on the page.
2. Groups proceed through the video trainer materials at their own pace, watching the video and then answering the Check Your Understanding questions.
3. At the bottom of many of the pages students will find Engineering Investigations and/or Programming Challenges (see pictures below). They should complete the challenges in the order that they are presented.
4. Each group constructs its own solution to the Unit Challenge
    - Groups may be asked to document their solutions in journals or logs, and especially to explain how they overcame the key problems identified at the start of the unit
5. Assign the Additional assignments based on the focus of the class
    - There are many handouts and challenges that allow teachers to extend the use of the curriculum to teach engineering practices, prepare for robotic competitions, or teach computational thinking generally
    - Complete teacher assigned work.

Engineering Investigations and Programming Challenges

At the bottom of many of the lessons students will find programming challenges and engineering investigations. The Basketball Drills picture shows the icon that indicates a programming challenge, the Power Levels with Encoders icon represents and Engineering Investigation

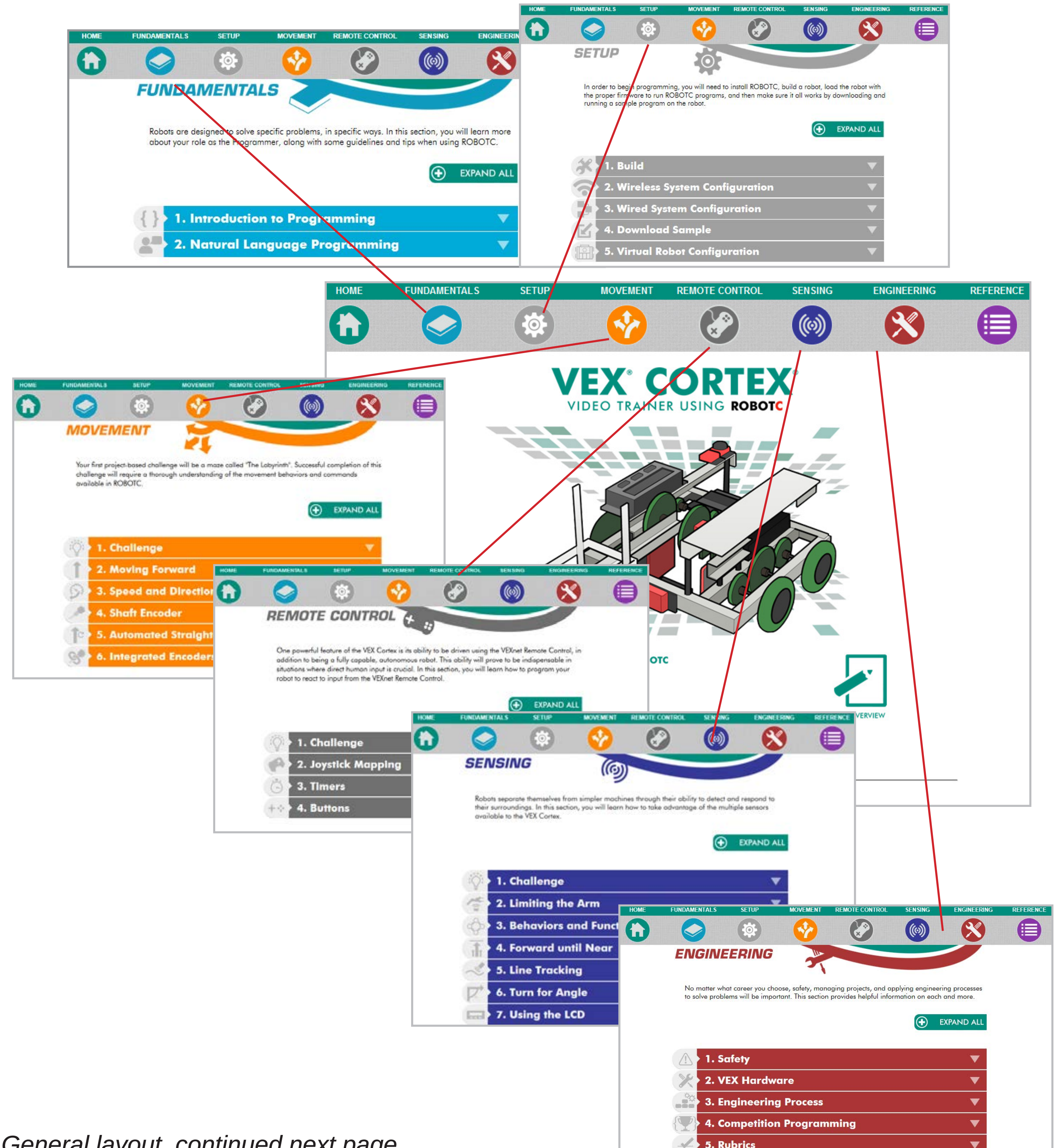


Topics are Covered in Each Unit

Unit Name	Main Topics
Fundamentals - 10 days	
Introduction to Programming	Basic rules on how to think about programming and syntax
Natural Language Programming	A set of reference documents on Natural Language programming
Setup - 10 days	
Build	Robot Building Instructions
Wireless System Configuration	How to setup and configure VEX Cortex wireless communications
Wired System Configuration	How to configure a wired system
Downloading Sample Programs	How to download a sample program, there are nearly 200 sample programs!
Virtual Robot Configuration	How to use Robot Virtual Worlds
Movement Unit - 20 days	
Moving Forward	Level one programming
Speed and Direction	How to change speed and direction
Shaft Encoders	Loops, configure and use shaft encoders
Automated Straightening	If/Else, variables, and values
Integrated Encoders	How to use IMEs, PID, and precise movement
Remote Control - 15 days	
Joystick Mapping	How to program the VEXnet Joystick
Timers	How to program and use Timers
Buttons	How to program buttons on the VEX Joystick
Sensing - 25 days	
Limit the Arm	The Touch Sensor
Behaviors and Function	Functions and Parameters
Forward Until Near	The Ultrasonic Sensor
Line Tracking	Line Tracking Sensors
Turn for Angle	The Gyro Sensor
Using the LCD	Displaying Messages on the LCD
Engineering - semester	
Safety	Handouts and quizzes
VEX Hardware	Handouts and guides
Engineering Process	Instructional videos and handouts
Competition Programming	Instructional videos and handouts
Rubrics	Assessment rubrics

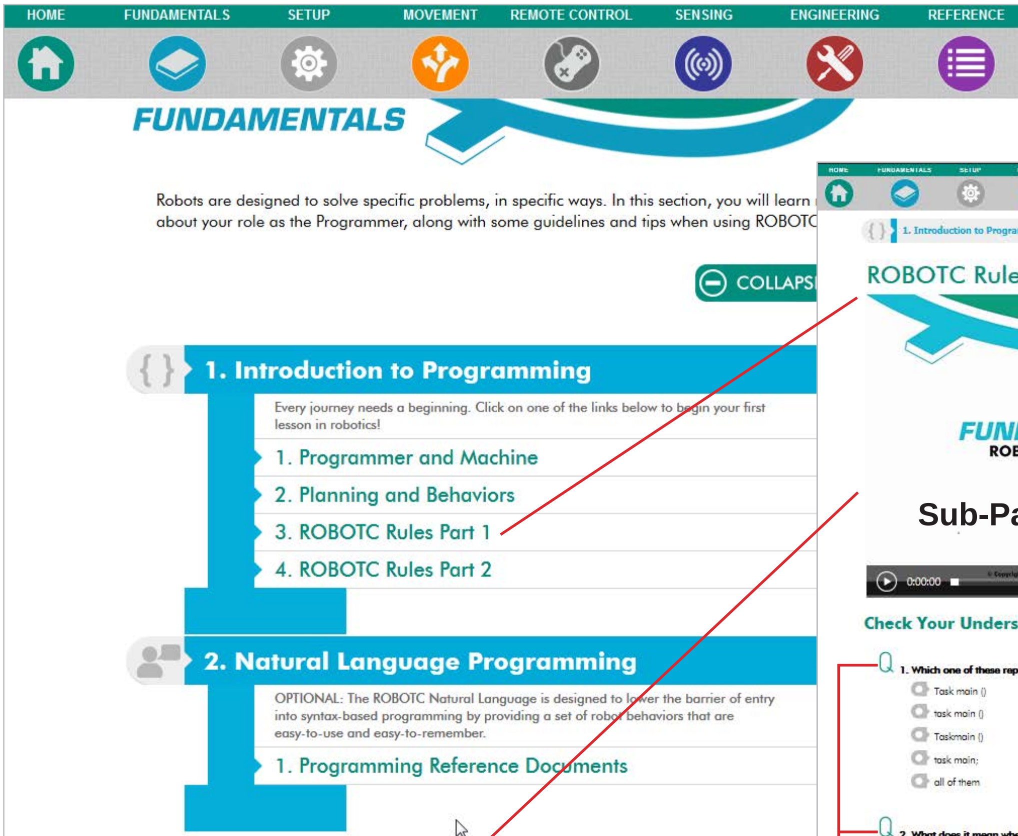
General Layout of the VEX Cortex Video Trainer Using ROBOTC

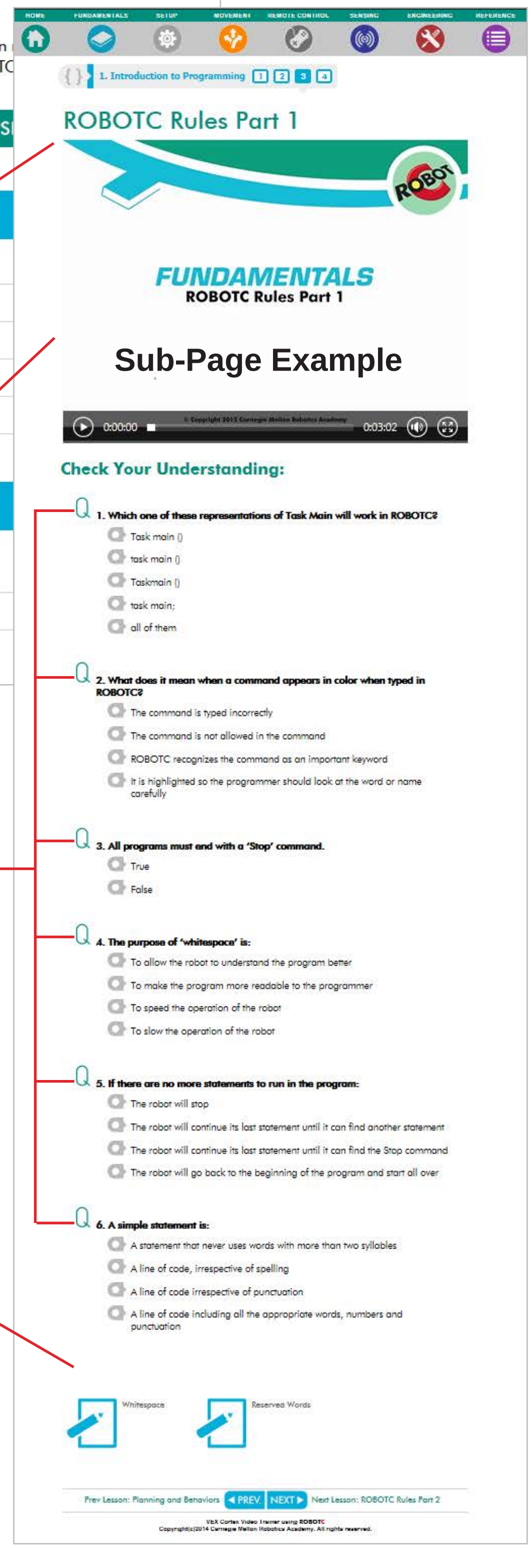
The VEX Cortex Video Trainer uses the buttons at the top of the page to navigate the Fundamentals, Setup, Movement, Remote Control, Sensing, and Engineering sections of the curriculum. The last button, Reference, is designed to allow the user to navigate the whole curriculum to find a handout or video lesson.



General layout continued next page.

General Layout continued





ROBOTC Fundamentals Video
Each lesson has an individual video to support students

Check your Understandings
Questions designed to provide students time to reflect on what they just learned

Reference Materials
to support the lesson

Sub-Pages

Each sub page within the main navigation opens to a sub-page. Pictured at the right is the ROBOTC Rules Part 1 sub-page. The majority of the pages contain one short video. As set of "check your understandings" questions that asks students questions about the video that they just watched, and additional reference materials if they are important to the lesson.

Scope and Sequence

This scope and sequence is a short three page version of what a year long class might look like. For a more detailed description of what is in each chapter go to the Chapter Overview section starting on page 17. The outline below uses the following format:

Approximate Class Days	A brief description of the lesson and where to find it in the curriculum.

Introduction to Robotics Course Outline

Note: there are many ways to teach this course. This set of general guidelines assumes that you want to teach a robotics engineering course and want to begin by teaching programming.

3-5 days	Class Rules and Organization/Assessment/Engineering Journal The first week of class covers classroom organization and an overview of what students will learn in your class. A key concept that students should have when they leave this course is: "What a robot?". In today's world, robots are everywhere, we just don't call them robots. Begin with the handout on page 15, Sense Plan Act. Have the students identify as many technologies that fit that description as possible and sort them into industry sectors (i.e. banking, manufacturing, entertainment, healthcare...). Have students select an industry sector and answer the following question: How has robotics affected that industry sector? Or assign them the general topic: What is a robot? First Assignment - "What is a robot?"
3-5 days	Safety Lesson - page 32 and pages 94-95 Safety is an integral part of an robotics course and should be covered at the beginning of the class and then strictly enforced throughout the year. You will find many handouts, worksheets, and quizzes in the Safety Unit found in the Engineering Chapter. Every class setting is different, it is the responsibility of the teacher to select the appropriate handouts and tools to teach safety in their class. It is paramount that students understand that a clean lab is a safe lab and that they are responsible to cleanup at the end of each day.
10 days	Introduction to Engineering (Rube Goldberg Machine™) - Internet Students will be working on engineering design problems throughout their robotics class. A good first assignment is to randomly break students into teams and have them complete a "Rube Goldberg Machine" (RGM). RBM contests are designed to encourage teamwork and out-of-the-box problem solving, it is also a fun activity that allows you to introduce the engineering process activities that are presented in the Engineering Chapter: Engineering Process, Engineering Design Journals, Teamwork, Project Planning, PERT or Gantt Charts, etc. It will be the up to the teacher to determine what parts students will use for their RGMs (e.g. VEX parts, random parts from home, or a combination of parts) Be sure to set a limit on the number of days that you give students to complete their project. For RGM rules go to www.rubegoldberg.com The RGM project also provides the teacher with the opportunity to introduce students to the assessment tools that they will use throughout the course. Rubrics - Introduce students to your evaluation tools • Engineering Journal Rubric - pages 98-101 • Work Habits Rubric - page 102 • Writing and Proposal Rubrics - page 102

Scope and Sequence Continued

5-10 days

Setup Unit

- Build Robot - pages 45-46

If you do not have robots built, then now is the time to build them. There are an unlimited number of robot builds that you can design using VEX kits, we suggest that you use the “Clawbot with Sensors” build. It is important that whatever *teaching robot* that you choose for your class to build that all of the robots have the same configuration (motors are in the same ports, sensors are in the same ports, all robots have the same gripper). If every student’s robot is different, uses different mechanics, has sensors plugged into different ports, and has different types of grippers, then it will take you much longer to help them to troubleshoot their robot’s problems.

Once the Robots are Built Test Them

- Robot Configuration - see pages 47-48
- Download a sample program - see page 49

5 days

Teaching Robot Math - Expedition Atlantis Robot Math Game

Expedition Atlantis is a free math programming game that has proven to teach robot math to students. It is easier to teach the math without having students worry about the programming. You can download the software from the Robot Virtual World website. The software includes an easy to follow teacher guide.

Teaching Programming Using Simulation Software - pages 53-54

Many teachers are turning to Robot Virtual World software to teach the programming portion of the curriculum. Research shows that when students are trying to learn multiple concepts at the same time (programming, mechanics, sensors, and math) that they become confused. It is much easier to foreground and teach one concept at a time rather than try to teach multiple concepts at a time. The Robot Virtual World software eliminates students having to learn the mechanics of the system (How do I configure my robot to talk with VEXnet? What port am I plugged into? Is my robot battery charged? Did someone in another class change things on my robot?). The curriculum can ge taught without RVWs, but RVWs is a more efficient way to teach robot programming.

20 days

The Movement Chapter - pages 19-24 and pages 56-67

The Movement Chapter provides a scaffolded way to introduce students to basic programming using ROBOTC and VEX robots. Each Chapter in the curriculum includes a Chapter Challenge; in the Movement Chapter the challenge is the Labyrinth Challenge.

IMPORTANT

Begin the chapter by introducing the challenge and explain to students that they will learn to program the Labyrinth Challenge multiple ways, from using very simple timing to using feedback from encoders, to developing their own automated straightening algorithm, to using PID. The curriculum uses this “simple to efficient” way to introduce programming concepts to students in all of the chapters. For students to truly learn programming, they should complete all methods of solving the challenge.

FUNDAMENTALS CHAPTER

Begin with the “Moving Forward” unit in the Movement Chapter before you spend time in the Fundamentals Chapter. Once students complete the Moving Forward unit they will have context that they can relate to when the learn what is taught in the Fundamentals Chapter. The Fundamental Chapter teaches students: how machines think, what behaviors are, and what comments, whitespace , reserved words, compiler errors, and general rules around syntax are. Students can begin by using pseudocode in the Movement Chapter and flowcharts in the Remote Control and Sensor’s Chapter when the programming logic becomes more complex.

Scope and Sequence Continued

15 days

The Remote Control Chapter - pages 25-26 and pages 68-73

The Remote control unit begins with the Minefield Challenge. Once again, the chapter is structured showing simple methods and then advanced methods of programming. This chapter also introduces students to Loops, Boolean Logic, Timers, and programming the VEX remote control buttons to automatically elicit specific behaviors (i.e. press this button and do a 90 degree turn).

Pseudocode and Flowcharts

It is common for beginning programmers to write programs that compile and that they believe that the logic is correct, but then the robot doesn't do what they think that it should do. Starting with the Remote Control chapter, require students write their program using pseudocode and then develop a flowchart that illustrates the robot's decision making. You will find classroom resources to teach these important processes on pages 17 - 25 of this teacher's guide.

Minefield Competition

The Minefield Competition is and end of chapter activity that provides a game like environment that is motivating to many students. Feel free to modify the game. Provide opportunities for student to write the rules for the game.

25 days

The Grand Challenge!

The Sensing Chapter - pages 27-31 and pages 74-92

The Sensing Chapter begins with the Grand Challenge. The Grand Challenge provides an opportunity for students to demonstrate the ability to use all of the sensors. What is shown on the Grand Challenge video is an example of what the challenge could look like. We recommend that you provide students with a set of behaviors (not the actual code) that their robot will need to complete and that you don't actually share the challenge with them until two days before the actual challenge. For example, you robot will need to be able to complete the following behaviors: Stop at a black line, turn accurately using feedback from a gyro sensor, identify the distance from an object using feedback from an ultrasonic sensor, and identify the location of an object and pick it up and come home.

The Sensing Chapter continues to build on a student's basic understanding of programming and logic. There are many ways to program a robot to complete a task. It is important for the teacher to not only watch the robot perform the task, but also to look at the student's code to make sure that they are applying all of the programming concepts that each individual unit attempts to teach.

The programming unit teaches students how to write functions and pass parameters, about variable types, if/else statements, switch cases, and all of the VEX Cortex sensors. The curriculum continues to use many challenges that the students have seen before, but requires them to solve the challenge using a different, more advanced programming strategy. Feel free to allow the students to modify the programming challenges as long as they teach the same foundational concepts.

Second Semester 90 Days

The Engineering Chapter - pages 32-34 and pages 93-103

The VEX Cortex Video Trainer is intended to be used to teach students introductory programming. To truly learn engineering, students must be engaged in multiple engineering problems. We strongly suggest that you enroll your students in a Robotics Education and Competition Foundation (RECF) competition. The RECF sponsors a new competition each year and provides teachers with a great tool to teach engineering with. When students are given a problem have them begin by conducting research to see how others solved the problem. Require them to use all of the engineering tools found in the Engineering Chapter: the Engineering Journal, time management tools like PERT and Gantt Charts, and Design Reviews.

Chapter Overview

What topics are covered in each Unit?

Chapter: Fundamentals - 10 days

1. Introduction to Programming - 5 days, then ongoing

a. Programmer and Machine - an instructional video that introduces the new programmer to how they need to think to translate their ideas into a language that a machine can understand.

b. Planning and Behaviors - an instructional video that introduces the new programmer to the idea of robot behaviors. This sub-unit also includes three reference PDFs: Behaviors, Pseudocode and Flowchards, and Thinking about Programming.

c. ROBOTC Rules Part 1 - an instructional video that introduces students to C programming syntax and how ROBOTC uses color to indicate reserved words. The sub-unit also includes two reference PDFs: Whitespace and Reserved Words.

d. ROBOTC Rules Part 2 - a followup instructional video that continues to teach students about syntax, comments, and error messages. This sub-unit also includes three reference PDFs: Comments, ROBOTC Error Messages, and ROBOTC Rules.

e. In this teacher guide you will find several introduction to programming guides:

- Breaking Programs into behaviors, page 14
- Sense Plan Act, page 15
- Introduction to pseudocode, pages 16 - 21
- Introduction to flowcharts, pages 22 - 25

2. Natural Language Programming

a. Natural Language programming places code segments into functions and is intended to make it easier to enable new programmers to program simple code. Natural Language provides a bridge between ROBOTC Graphical and full ROBOTC.

It is our recommendation that teachers and students learn to program using full ROBOTC and use the Movement, Remote Control, and Sensing units as they are currently written and supported to teach and learn VEX robot programming.

The Natural Language programming unit contains a large number of reference guides that have been added as a resource for teachers.

Chapter Overview Continued

Chapter: Setup

The Setup Unit contains resources that you and your students can use the first time that they are starting with their robots.

1. Lesson: Build

The build section includes building instructions for four robot types.

a. Recbot Building Instructions

b. Squarebot 4 Building Instructions for use with ProtoBot kits

c. Clawbot with Sensors Building Instructions (***recommended build for this curriculum***)

d. Swervebot Building Instructions

2. Wireless System Configuration

a. Update Cortex Firmware (Wireless) - an instructional video that guides the new user how to update the Cortex firmware with the Master CPU Firmware and the ROBOTC Firmware. This unit also includes Check your Understanding questions and a VEX Cortex Driver Installation guide produced by VEX robotics.

b. Updating VEXnet Joystick Firmware - an instructional video that shows the new user how to update the VEXnet Joystick with the latest firmware using ROBOTC. The lesson also includes Check Your Understanding questions and two PDFs: Establishing a VEXnet Link and VEXnet Joystick Calibration

3. Wired System Configuration

a. Update Cortex Firmware (Wired) - an instructional video that guides the new user how to update the Cortex firmware with the Master CPU Firmware and the ROBOTC Firmware. This unit also includes Check Your Understanding questions and a VEX Cortex Driver Installation guide produced by VEX robotics.

4. Download Sample Programs

a. Download a Sample Program Part 1 - An instructional video that shows how to download a sample program over VEXnet. The lesson also includes a USB-to-Serial Cable Driver Installation instructional PDF.

b. Download a Sample Program Part 2 - The second part of the Download Sample Program video. This lesson also includes Check Your Understanding questions.

c. Download a Sample Program over USB - and instructional PDF that takes the new programmer step-by-ste through the download sample program over USB procedure.

Continued next page

Chapter Overview Continued

Chapter: Setup Continued

5. Virtual Robot Configuration

a. Download your First Program (Virtual) - Robot Virtual Worlds (RVW) enable all kids with computers to have access to a robot. This lesson includes an instructional video and Check Your Understanding questions. You can find out more about RVWs at: www.robotvirtualworlds.com

b. Camera Operation in RVWs - This lesson shows how to use some of the additional controls in the RVW environment. This lesson includes an instructional video and Check Your Understanding questions.

c. The Measurement Toolkit - This lesson shows a new RVW user how to use the tools found in the measurement toolkit which are embedded into all RVWs. This lesson includes an instructional video and Check Your Understanding questions.

Chapter: Movement Chapter - 20 days

1. Labyrinth Challenge Unit/Movement Chapter

a. The Labyrinth Challenge - This lesson set is designed to show the challenge that students will work on for their first problem. The Labyrinth Challenge requires students to program their robot to travel a specific distance, turn accurately, and then repeat these behaviors multiple times.

Note: RBC files are used throughout the curriculum. They are designed to automatically start a program when you are ready to begin programming. At the left is the virtual robot icon and you will need to have a working copy of RVWs in order to open that file type. At the right (below) is a physical robot icon. When you select those files and click them ROBOTC will automatically open.

2. Moving Forward Unit/Movement Chapter - 5 days

a. Program Dissection - This lesson explains what the introductory lines of code do. It also begins to teach some of the syntax related to ROBOTC. This lesson set includes: an instructional video, RBC files, Check Your Understanding questions, and an instructional PDF that walks students through the steps to run a program.

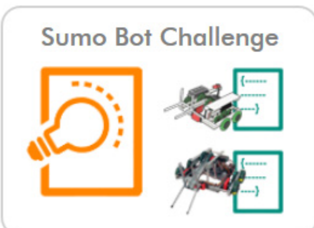
Continued next page

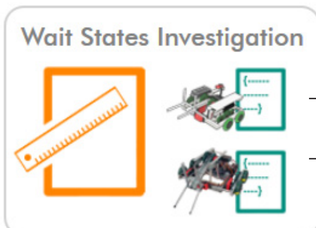
Chapter Overview Continued

b. Reversing Motor Polarity - This lesson shows the student how to program their robot to move backward by changing the polarity of their motors. The lesson set includes: an instructional video, RBC files, and Check Your Understanding questions.

c. Renaming Motors - This lesson teaches students how to use the Motors and Sensors setup window to give custom names to individual robot motors. The lesson set includes: an instructional video, RBC files, and Check Your Understanding questions.

d. Timing - This lesson teaches students how to adjust how long a motor is turned on and off using timing. The lesson set includes: an instructional video, RBC files, and Check Your Understanding questions.





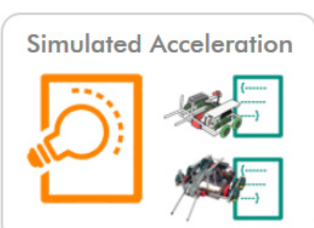
RVW RBC File

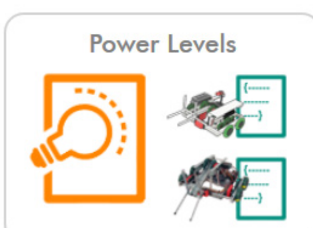
Physical Robot RBC File
Double click the file and ROBOTC opens

Moving Forward Programming and Engineering Investigation - The Moving Forward lesson includes two simple programming and engineering investigations. The SumoBot Challenge requires student to change power levels and to investigate what that does to rotational torque. The Wait State Investigation requires students to change the amount of time that the robot moves and to record the distance that it travels. Both of these investigations can be completed in either a physical or virtual environment.

3. Speed and Direction Unit/Movement Chapter - 5 days

a. Motor Power Levels - This lesson explains how to change the power levels programmatically. The lesson includes: an instructional video, RBC files, Check Your Understanding questions, and two programming investigations: Simulated Acceleration and Power Levels.





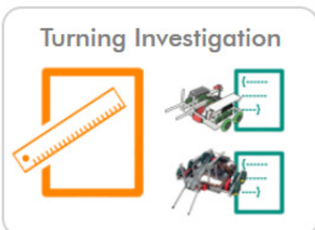
Continued next page

Chapter Overview Continued

Speed and Direction Unit/Movement Chapter Continued

The Motor Power Levels Unit includes to programming challenges. Simulated Acceleration is a very simple excise that requires students to simulate a robot accelerating. At this point students are still learning code and this is providing them practice writing code and understanding how pseudocode works. The Power Levels Engineering Investigation provides students with an opportunity to find out if there is a proportional relationship between Power Levels and distance travelled.

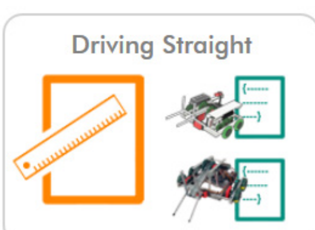
b. Turn and Reverse - This lesson is designed to show students how to make different types of turns. The lesson includes an instructional video, RBC files, Check Your Understanding questions, and two programming investigations: the Turning Investigation and the Sentry Simulation Level 1.



The Turning Investigation is designed to provide students with the opportunity to practice with swing turns and point turns. They are asked to conduct an experiment that involves programming their robot to turn and then change a value and predict how far the robot will turn given the change in the value.

The Sentry Simulation requires students to program a robot using timing to march around a square. Students are required to answer questions about what happens when they change values in their programs.

a. Manual Straightening - This lesson requires students to make adjustments to their robot to make sure that it goes perfectly straight. This is the most primitive way to program their robot and will not be used once they begin using sensor feedback. The lesson includes a video, RBC files, Check Your Understanding questions, and an additional programming challenge named: Drive Straight.



Continued next page

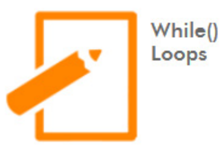
Chapter Overview Continued

3. Shaft Encoders - 5 days

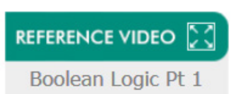
a. Shaft Encoders - This lesson explains how shaft encoders work. The lesson includes: an instructional video, RBC files, Check Your Understanding questions, and a PDF reference guide that explains how the shaft encoders work.

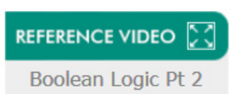
b. Forward for Distance Part 1 - Explains how to program shaft encoders and how to use the Motors and Sensors Setup window. The lesson includes: an instructional video, RBC files, Check Your Understanding questions.

c. Forward for Distance Part 2 - Continues to explain how to use and program shaft encoders and why it is important to clear the encoders and how a while loop works. This lesson includes an instructional video, Check for Understanding questions, RBC files, a reference PDF that explains how While Loops work, and two reference videos that explain Boolean Logic.



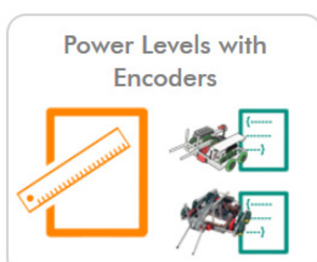
While Loop reference handout - this handout explains how while loops are used in programming.





Boolean Logic reference videos - Boolean Logic is used to help the robot to make decisions. It is critical that students understand this concept and is sometimes difficult for students to understand. These videos will be found in multiple places in the curriculum.

d. The Sensor Debug Window - The debugger is an incredible tool that is included within ROBOTC. MAKE SURE YOUR STUDENTS KNOW HOW TO USE THE DEBUGGER! This lesson includes an instructional video, RBC files, Check Your Understanding questions, and two programming challenges: Basketball Drills and Power Levels with Encoders.



e. Forward and Turning - This lesson completes the introduction to shaft encoders lesson and includes a summary video, Check Your Understanding questions, RBC files, and an additional Engineer Investigation called Turning with Encoders.

Continued next page

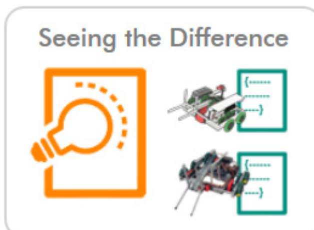
Chapter Overview Continued

4. Automated Straightening Unit/Movement Chapter - 5 days

a. Automated Straightening Part 1 - This lesson teaches students to use a combination of encoders and conditional statements to have the robot self correct its forward movements. The lesson includes an instructional video, RBC files, Check Your Understanding questions, and an “if-else” reference guide.

The if-else reference guide provides multiple examples with comments on how the if-else structure can be used.

b. Automated Straightening Part 2 - Completes the video instruction. The lesson includes an instructional video, Check Your Understanding questions, RBC files, and to programming challenges: Driving Straight 2 and Seeing the Difference.



Driving Straight 2 challenges students to solve the same course that they solved earlier, but this time they are using feedback from sensors and an algorithm that they just developed. Hopefully they see that using sensors and programming is much better than motors and timing.

Seeing the Difference requires students to use the built in debugger. The debugger is a programmer’s best friend and this investigation is designed to give students practice using the debugger.

c. Values and Variables Part 1- This teaches students about variables and the power of using variables when writing code. This lesson includes an instructional video, RBC files, Check Your Understanding questions, and two reference PDFs: Variables and Global Variables.

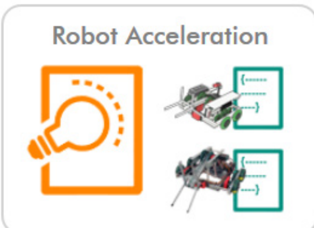
Variables are very important in programming. These two reference guides teach students about variable types, how to declare variables, and provide commented examples of variables being used in code.

Continued next page

Chapter Overview Continued

Automated Straightening Unit/Movement Chapter Continued

d. Values and Variables Part 2 - This lesson is part two of values and variables. Students are required to create two variable in their program and use them to make programming more efficient. This lesson set includes an instructional video, RBC files, Check Your Understanding questions, and the Robot Acceleration programming challenge.



Students completed this challenge several units ago. This time, they are able to create a variable and manipulate it mathematically.

5. Integrated Encoders Unit/Movement Chapter - 5 days

a. Forward for Distance IME - This lesson teachers students how to use Integrated Motor Encoders. This lesson set includes an instructional video, RBC files, Check Your Understanding questions, and an Integrated Motor Module reference guide produced by VEX Robotics.

b. Principles of PID - This lesson teaches students about how a Proportional Integral Derivative controller works. This lesson set includes an instructional video and Check Your Understanding questions.

c. Forward for Distance PID - Teaches students how to control the distance that the robot moves using IMEs and PID. The lesson set includes an instructional video, Check Your Understanding questions, and RBC files.

d. Forward for Target Distance - This lesson teaches students how to stop in an exact location using the MoveMotorTarget command. They will also learn about idle loops and logical operators. They will need to review logical operators using the Boolean Logic 2 reference video. This lesson set includes the instructional video, RBC files, Check Your Understanding questions, and two programming challenges: Basketball Drills and Sentry Simulation.

Students completed this challenge several units ago using only timing, now they can use motor encoders and PID and they should see that they can program their robot to move more accurately this time.

Continued next page

Chapter Overview Continued

Chapter: Remote Control - 15 days

1. The Minefield Challenge/Remote Control Chapter

a. The Minefield Challenge - this video and PDF show students what the culminating activity for the chapter will look like. In this chapter, students learn how to program their VEXnet Joystick.

Pseudocode and Flowcharts - 3 days then ongoing

Note: It is important to require students to use pseudocode and flowcharts now! Students will develop their initial code using pseudocode and then develop a flowchart that shows the robot's decision making.
Find lesson resources on pages 17 - 25 in this teacher's guide.

2. Joystick Mapping Unit/Remote Control Chapter - 3 days

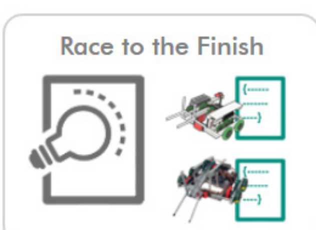
a. Introduction to Remote Control - this video shows how remote control works with the VEX Cortex system. This lesson set includes a video, Check Your Understanding questions, and a VEXnet Joystick Calibration PDF.

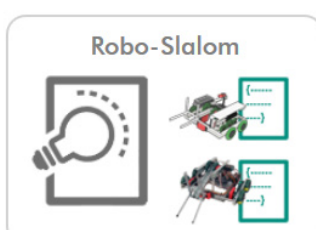
b. Real-Time Control - In this lesson students experiment with an existing sample program and attempt to optimize their real-time remote control. They will learn how the while loop works and Boolean Logic. This lesson set includes an instructional video, RBC files, Check Your Understanding questions, and two reference handouts: While Loops, and Boolean Logic.

Students will use While Loops and Boolean Logic for most programming activities moving forward. These are important reference guides for them.

c. Mapping Values Part 1 - This lesson set teaches students which joystick button and stick maps to which values.

d. Mapping Values Part 2 - This video completes the Joystick Mapping lesson and teaches students how to adjust the speed of the motors. The lesson includes: instructional videos, RBC files, Check Your Understanding questions, and several programming challenges: Robo-Slalom and Race to the Finish.





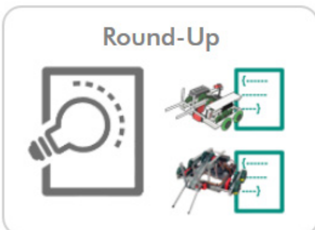
Continued next page

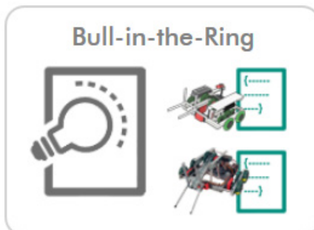
Chapter Overview Continued

3. Timers Unit/Remote Control Chapter - 2 days

a. Time and Timers- In the first lesson in this lesson set, students learn about Timers and how they can be used in programming. This lesson includes an instructional video, Check Your Understanding questions, and a PDF reference guide on using Timers with ROBOTC.

b. Using Timers - In this lesson students learn to implement timers in their programs. This lesson includes an instructional video, RBC file, Check Your Understanding questions, and two programming challenges: Round Up, and Bull In the Ring.



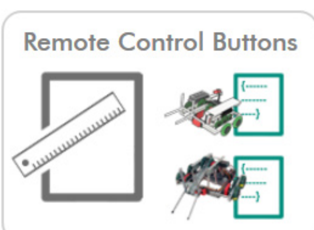


Round Up and Bull In the Ring are two simple remote control programming challenges. Encourage students to modify the challenges to make them more engaging.

4. Buttons Unit/Remote Control Chapter - 5 days

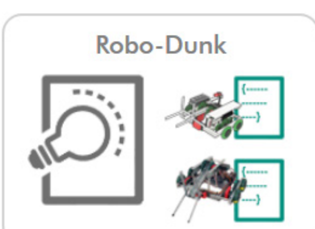
a. Remote Control Buttons - in this video lesson students learn how to identify and program the buttons on the top and front of the remote control.

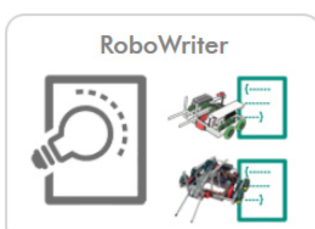
b. Remote Start - In this lesson students will learn how to program the robot to start using a button on the remote control. This lesson set includes an instructional video, RBC files, Check Your Understanding questions, and a programming challenge named: Remote Control Buttons.

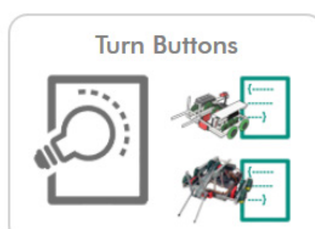


The Remote Control Buttons challenge is designed to guide students as they program the buttons on their VEX remote control. The programmable remote control is a very powerful tool in robotics competitions. Encourage students to program robot behaviors to their buttons. For example, BtnX makes the robot turn right.

d. Controlling the Arm Part 1, Part 2, & Part 3 - This lesson provides a step-by-step set of instructions to program their robot's arm to be controlled by the remote control. The lesson set includes three instructional videos, RBC files, Check Your Understanding questions, and three practice programming challenges.







5. Conduct a Minefield Challenge In-class Competition! - 2 days

Continued next page

Chapter Overview Continued

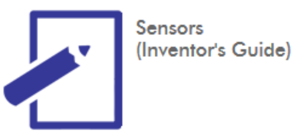
Chapter: Sensing - 25 days

1. The Grand Challenge/Sensing Chapter

a. The Grand Challenge is a teacher designed challenge that provides an opportunity for the student to demonstrate that they know all of the sensors, how to write functions and pass parameters, and can write reusable code. Tell students in advance by the end of the chapter that they will need to write functions that enable their robot to use feedback from sensors to solve multiple programmable tasks. The video and PDF on the Grand Challenge page show and example of what the challenge might look like.

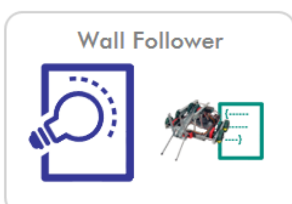
2. Limiting the Arm Unit/Sensing Chapter - 5 days

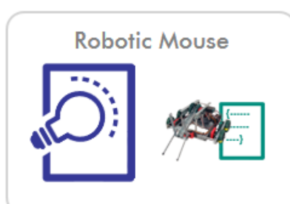
a. Configuring Sensors - this lesson teaches students how to use the Motors and Sensor Setup window to configure the robot's sensors. This lesson set includes an instructional video, RBC files, Check Your Understanding questions, and two reference guides: Sensors and Servo Modules.

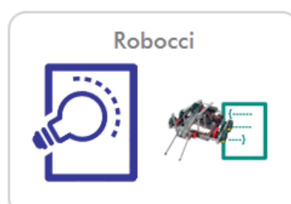


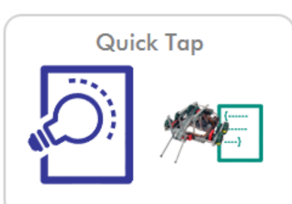
The Sensors and the Servo Modules reference guides explain how VEX sensors and servo modules work.

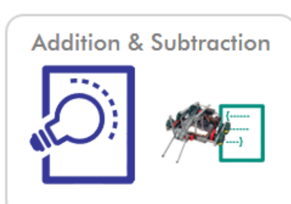
b. Limiting the Arm Part 1 and Part 2 - In this lesson students will learn how to use both the touch sensor and the potentiometer to control the arm on the robot. This lesson set includes an instructional video, RBC files, Check Your Understanding questions, several additional programming challenges and reference handouts.



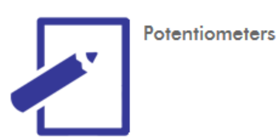








These five challenges can only be completed using physical robots. The if-else, Switch Cases, and Potentiometer reference guides provide the student with sample code that helps them to understand how these structures work.

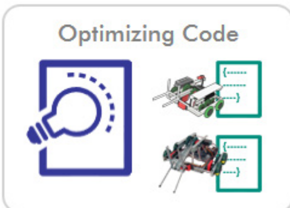


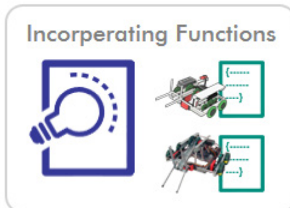
Continued next page

Chapter Overview Continued

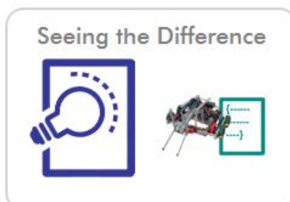
3. Behaviors and Functions Unit/Sensing Chapter - 7-8 days

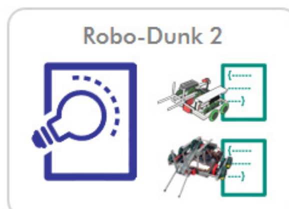
a. Behaviors and Functions Part 1 and Part 2 - In this lesson set students will learn how functions can make their programs shorter and easier to follow. The lesson set includes: two instructional videos, RBC files, Check Your Understanding questions, and two programming challenges.

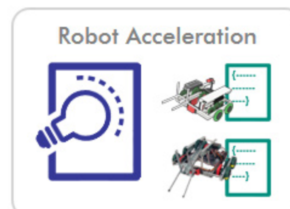




b. Passing Parameters Part 1 and Part 2 - In this lesson set students learn about the power of using parameters in functions. The lesson set includes: two instructional videos, RBC files, Check Your Understanding questions, and four programming challenges.







Seeing the difference can only be completed using a physical robot.

4. Forward Until Near Unit/Sensing Chapter - 5 days

a. The Ultrasonic Rangefinder - In this lesson students will learn what an Ultrasonic Rangefinder is and how it works. The lesson includes: an instructional video, Check Your Understanding questions, and the Ultrasonic Rangefinder reference guide.

Ultrasonic Rangefinder (Inventor's Guide)

b. Forward until Near - In this lesson student will learn how to write a program that uses the ultrasonic rangefinder. This lesson set includes an instructional video, RBC files, Check Your Understanding questions, several additional programming challenges and a Thresholds reference handout.

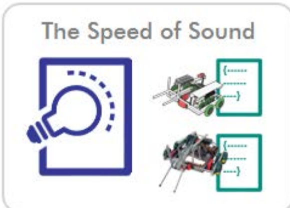
Thresholds

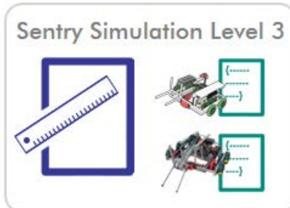
Continued next page

Chapter Overview Continued

4. Forward Until Near Unit/Sensing Chapter Continued

d. Straight Until Near - In this lesson students will learn how to implement the straight until near behavior. The lesson includes: an instructional video, RBC files, Check Your Understanding questions, and two programming challenges.



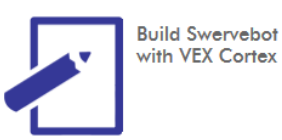


e. Straight Until Near (Fine Tuning) - In this lesson student will complete their Forward Until Near coding. This lesson set includes an instructional video, RBC files, Check Your Understanding questions, an additional programming challenge and the Boolean Logic reference guide.

Note: Logical operators can be tricky for students to understand. This may be a good time to review this concept with your class.

5. Line Tracking Sensors Unit/Sensing Chapter - 10 days

a. Line Tracking Sensors - this lesson reviews the sensors students have used up to this point in the curriculum and then teaches students how VEX Cortex Line Tracking sensors work. This lesson set includes an instructional video, RBC files, Check Your Understanding questions, and two reference guides: Build instructions for Swervebot and a Line Follower reference guide.



Note: Linetracking is difficult with some robot types. You may want to have a specially built robot specifically for this lesson because it takes a very long time for students to take apart and build a new robot. Another option is to complete the lesson using the Robot Virtual World simulation software.

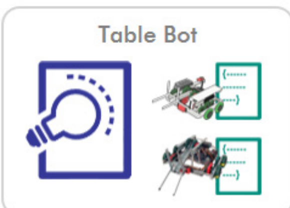
Continued next page

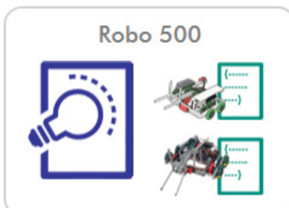
Chapter Overview Continued

Line Tracking Sensors Unit/Sensing Chapter Continued

b. Calculating Thresholds - In this lesson students will learn how to calculate the a threshold value using feedback from the ROBOTC debugger. The lesson includes: an instructional video, RBC files, Check Your Understanding questions, a programming challenge, and two reference guides.

c. Basic Line Tracking - This lesson teaches students how to write a program that will complete a basic line tracking behavior. The lesson includes an instructional video, RBC files, Check Your Understanding questions, and two programming challenges.





d. Line Tracking for Distance - In this lesson students will learn how to use feedback from multiple sensors to track a line and stop at a specific distance. The lesson includes: an instructional video, RBC files, Check Your Understanding questions, a programming challenge.

e. Optimized Line Tracking - In this lesson students will learn how to optimize their motor speeds in order to track lines more efficiently. The lesson includes: an instructional video, RBC files, Check Your Understanding questions, and two programming challenges.

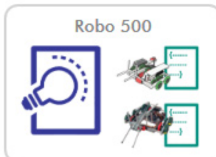
Continued next page

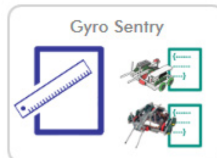
Chapter Overview Continued

6. Turn for Angle Unit/Sensing Chapter - 5 days

a. The Gyro Sensor - In this lesson students will learn how the Gyro Sensor works. The lesson includes: an instructional video and Check Your Understanding questions.

b. Turn for Angle Part 1 and Part 2 - In this lesson set students will learn how to program their robot to use feedback from a gyro sensor to perform a measured turn. This lesson set includes two instructional videos, RBC files, Check Your Understanding questions, and three programming challenges.





Note: Students have seen variants of these robot challenges before. The difference now is that they have more programming tools to work with. Encourage students to modify the programming challenges to make them more interesting.

7. Intro to the LCD - 5 days

a. The VEX LCD - In this lesson students will learn about the many uses of the VEX LCD. This lesson set includes an instructional video, Check Your Understanding questions, and the VEX LCD reference guide.

b. Displaying Text - In this lesson teaches students how to use the VEX LCD as an output device. The lesson includes an instructional video, RBC files, and Check Your Understanding questions.

c. Displaying Sensor Values - In this lesson teaches students how to continually update the values on the LCD and to display current robot sensor values. The lesson includes an instructional video, RBC files, and Check Your Understanding questions.

Note: Have students go back to several previous programming challenges and to modify the programming challenges so that students can see real time sensing feedback on the LCD.

8. Compete in the Grand Challenge Competition - 5 days

The Grand Challenge is a teacher designed programming competition where the students don't get the actual challenge until two days before the competition. One week before the competition they will learn about the behaviors that their robot need to perform. For example, you robot will need to be able to complete the following behaviors: Stop at a black line, turn accurately using feedback from a gyro sensor, identify the distance from an object using feedback from an ultrasonic sensor, and identify the location of an object and pick it up and come home.

Continued next page

Chapter Overview Continued

Chapter: Engineering - Semester

1. Safety Unit/Engineering Chapter

a. Safety - The safety reference guides and tests are to be used at the beginning of the semester to establish a safe work environment. The safety resources include the following handouts:

- Safety is an Attitude
- Safety (from the Inventor's Guide)
- General Lab Safety
- Electrical Safety
- Power Tool Safety
- The Safety Checklist

b. Safety Tests - It is important to administer a safety test to establish the importance of safety in your classroom. The safety resource unit includes the following safety tests:

- Safety Aptitude Test
- General Safety Test
- Safety Quiz
- Robotics Lab Inspection Sheet

2. VEX Hardware Guides

a. Motors and Building Materials - This section includes great resources to teach new builders about Structural systems and how to build with VEX hardware. The resources include the following handouts:

- Motion - a 26 page guide that shows how VEX motion systems are designed. This guide uses a large amount of pictures that show how parts connect.
- Structure - a 17 page guide that shows new builders how parts go together and important considerations about weight distribution and balance.
- Motor Handouts - VEX has multiple motor types, these handouts describe the technical use and capability of the various motors.

b. Sensors and Displays - This section includes technical guides for the following VEX hardware:

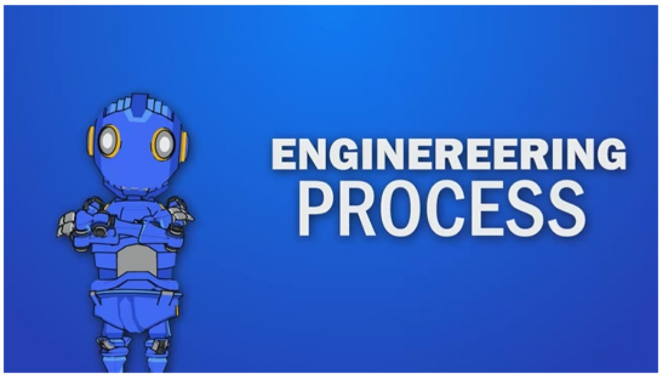
- Shaft Encoders
- Potentiometers
- Ultrasonic Rangfinder
- Light Sensors
- VEX LCD

Continued next page

Chapter Overview Continued

3. Engineering Process Unit/Engineering Chapter - Ongoing

a. Engineering Process - This lesson is a foundational lesson designed to help students to think about engineering process generally. This lesson includes an instructional video and key handouts that describe process that students will use throughout the course.



The Engineering Process video at the left uses a combination of levity and graphics to teach students about how to solve and engineering challenge.

Keeping and Engineering Journal is a guide for students on what they should keep in the journal.

Engineering Process Reference

Keeping an Engineering Journal

Design Reviews

Understanding the Problem

Brainstorming

Engineering Definitions

.

b. Project Planning - In this instructional video students learn the importance of project planning. The handouts provide insight on how to build a team, plan time, organize ideas, record project progress, and prepare for a competition.

Team Building

Gantt Chart

PERT Chart

Organizational Matrix Ideas

Recording Progress

Preperation for a Competition

Planning Your Time

Continued next page

Chapter Overview Continued

3. Engineering Process Unit/Engineering Chapter Continued

c. Engineering Automated Workcell - There are many engineering projects that your students can solve using VEX Robotics parts. This video shows an automated workcell made out of VEX robot parts.

4. Competition Programming

a. Competition Programming Part 1 and Part 2 - Robot Competitions have strict rules about how programs should be written to compete in the competition. This lesson includes two instructional videos, RBC files, Check Your Understanding questions, and a step by step guide.

ROBOTC Software Inspection Guide

b. The VEXnet Competition Switch - When you get to the competition, you will be using a VEXnet Competition Switch, not all schools have one. This instructional video and Check Your Understanding questions are designed to prepare you to use the VEXnet competition switch.

c. The Programming Hardware Kit - This lesson will show you how you can test your competition program using the Programming Hardware kit.

5. Rubrics

a. Assessment rubrics are common tools used for project based courses. These rubrics may not fit everything that you are doing in your classroom, but can be used as a guide to develop your own rubrics, or can be used as they are.

Writing Criteria

Workplace Competencies

Robotics Exploration Rubrics

Presentation Rubrics

Proposal Assesment Rubric

Internal Design Reviews

Working Habits

Engineering Journal Rubric

External Design Reviews

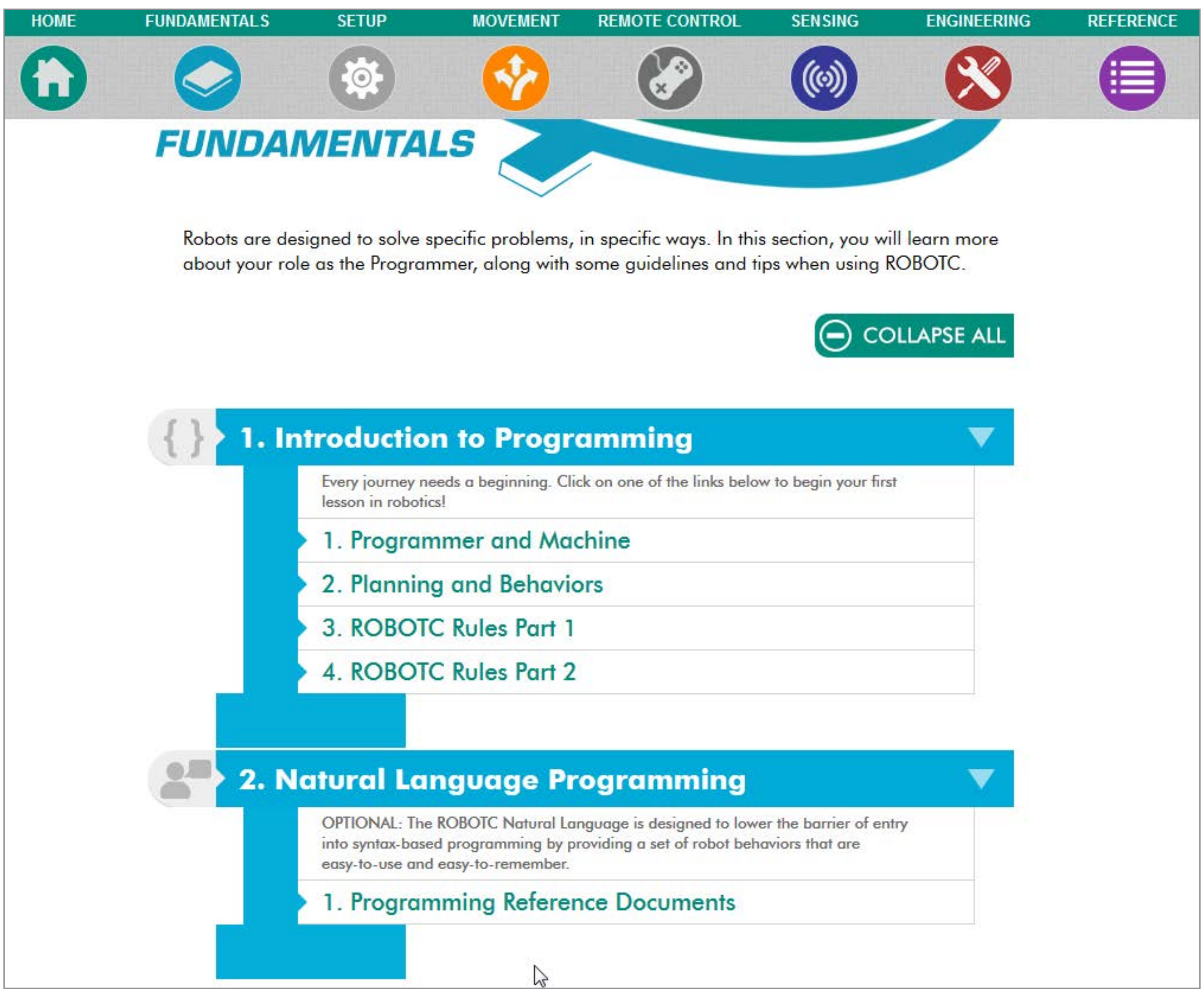
Navigating the Curriculum

The curriculum is divided into two sections: "Getting Started" and "Programming and Engineering."

The ***Fundamentals and Setup*** chapters provide resources that will be helpful to new VEX/ROBOTC students.

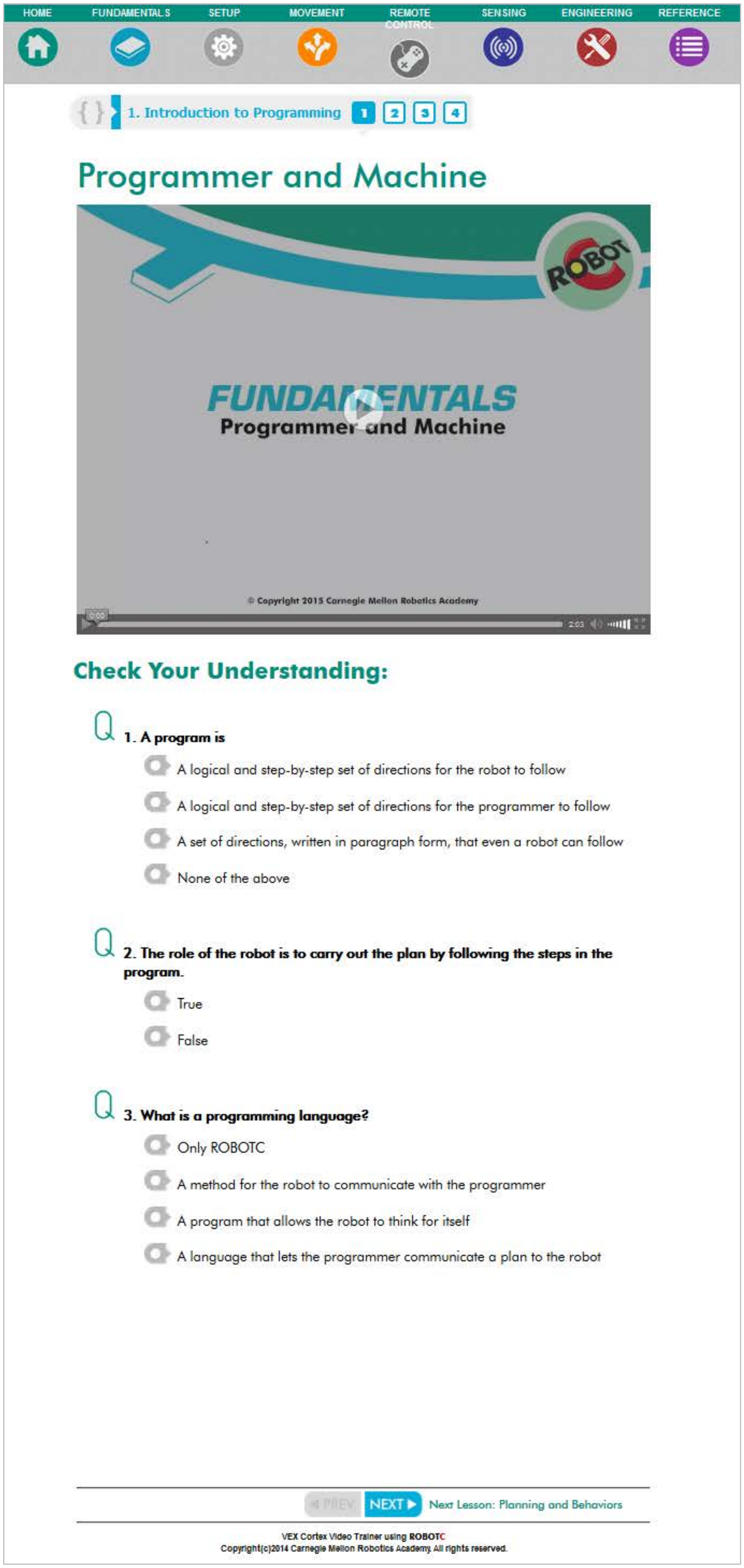
The ***Programming and Engineering Lessons*** are found in the Movement, Remote Control, Sensing, and Engineering sections.

Fundamentals/Introduction to Programming



The Fundamentals Unit is divided into two Lesson Sets: Introduction to Programming and Natural Language Programming

Fundamentals/Introduction to Programming

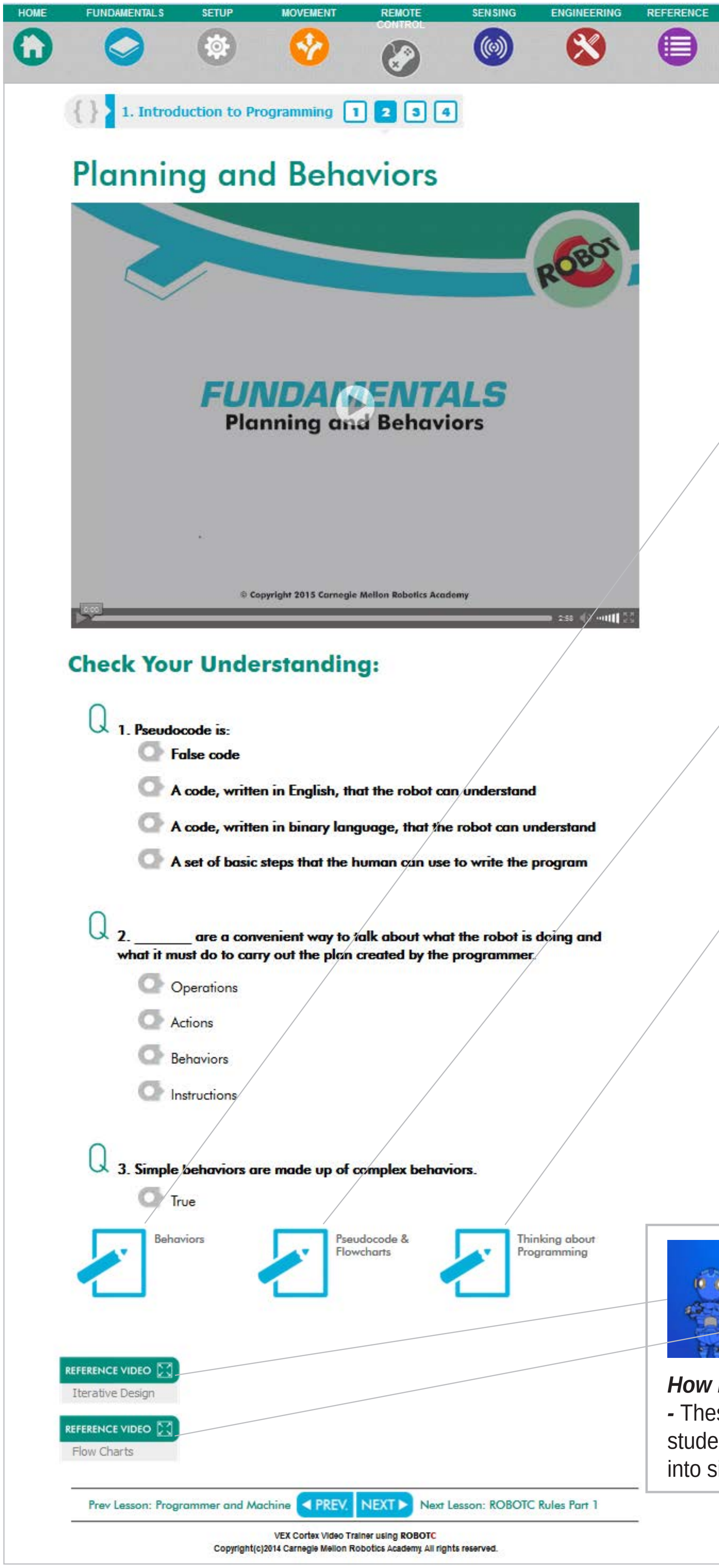


The Programmer and the Machine

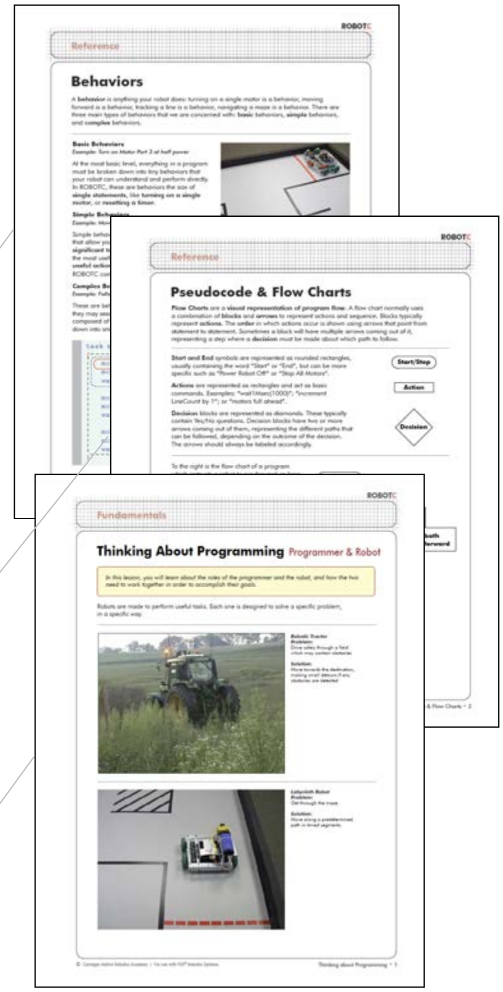
The “Programmer and Machine” video explains to students the role of the programmer and the machine, and how the programmer must learn to think like a machine in order to program robots. The video is are designed to explain programming concepts to beginners.

Each page video includes a set of “check your understanding” questions that cover the main topics covered in the video.

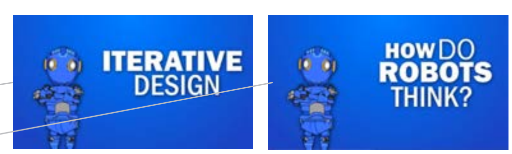
Fundamentals/Introduction to Programming



The Planning and Behaviors Video
The Planning and Behaviors introductory video explains behavior based programming to students. The lesson also includes: check your understandings questions, three handouts, and two additional videos which are shown below.

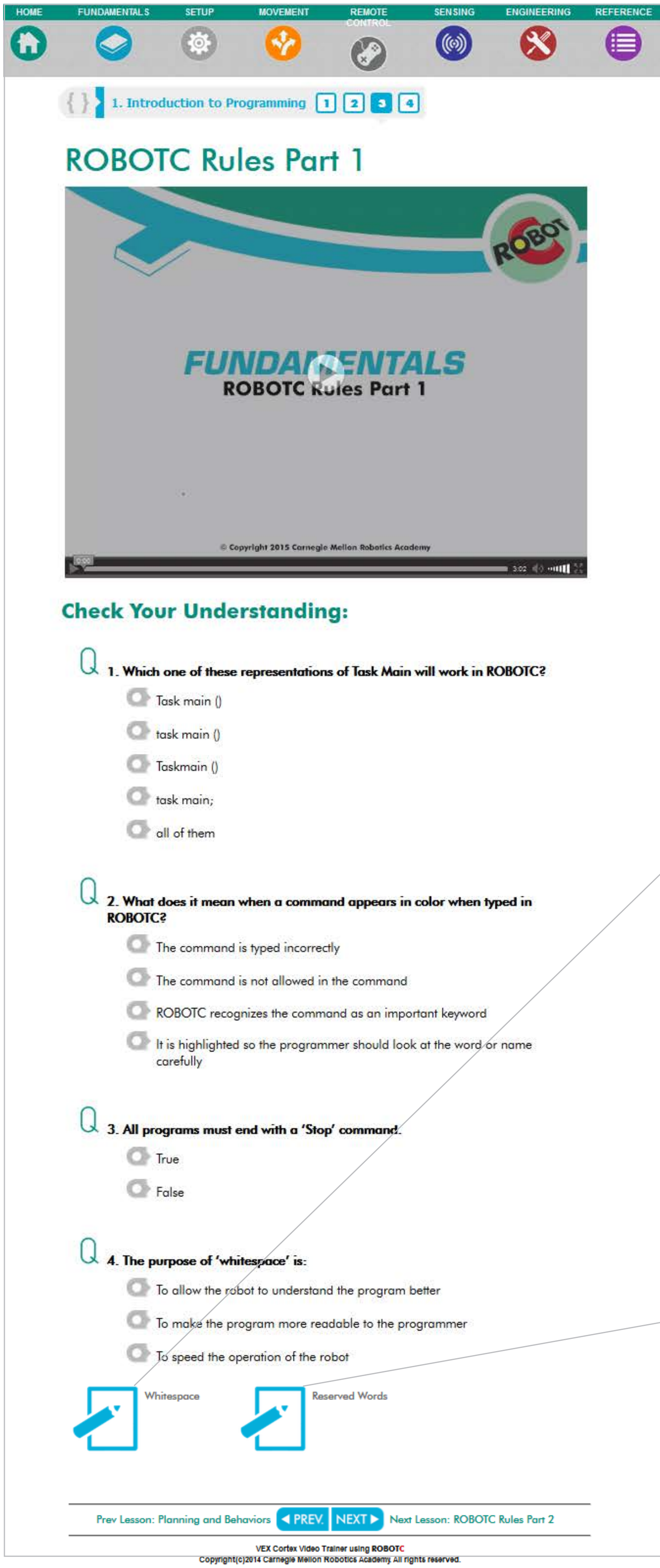


Behaviors and Flowcharts & Pseudocode Helper Pages - These are reference PDFs that students can use as study guides. The helper pages are designed to be guides to the topics.

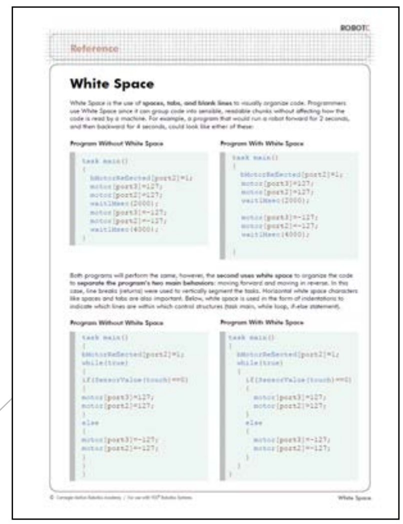


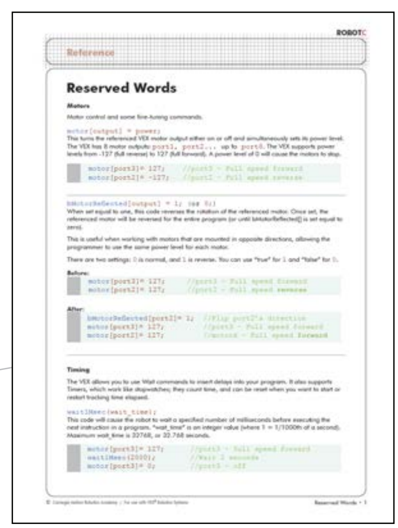
How Do Robots Think and Iterative Design **-** These two new videos are designed to help students understand how to break their programs into simple logic and behaviors.

Fundamentals/Introduction to Programming



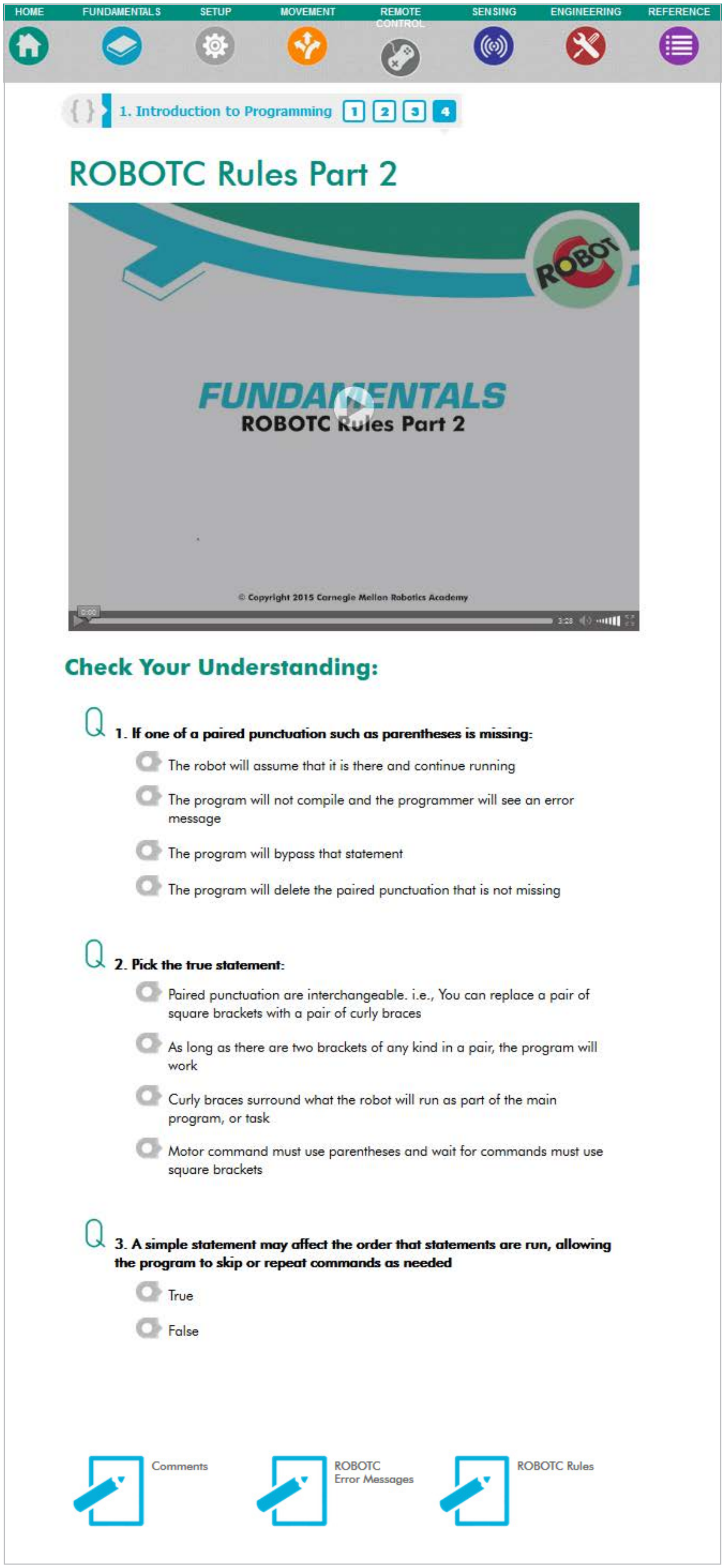
The ROBOTC Rules Part One includes: The introductory video, check your understanding questions, and the two Whitespace and Reserved Words handouts shown below.



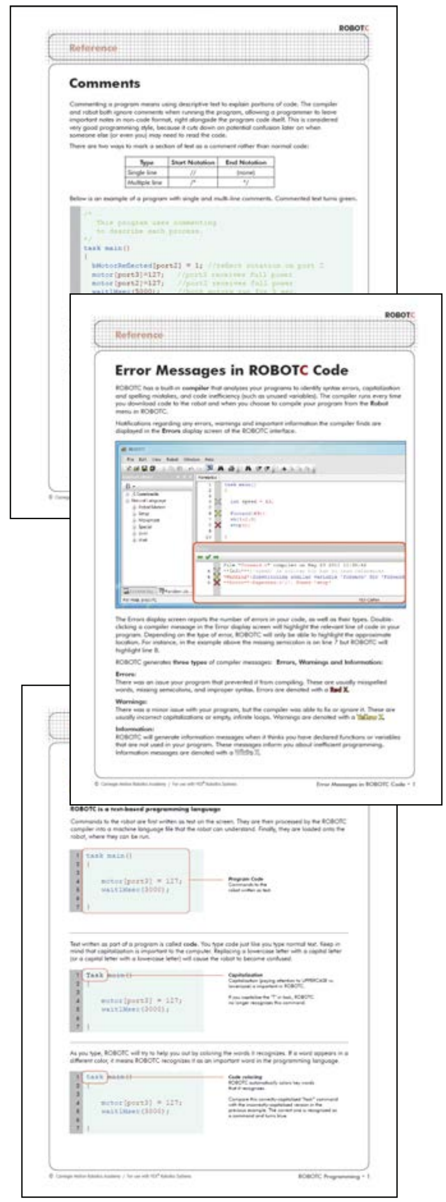


Whitespace and Reserved Words Helper Pages - These are reference PDFs that students can use for reference or as study guides.

Fundamentals/Introduction to Programming

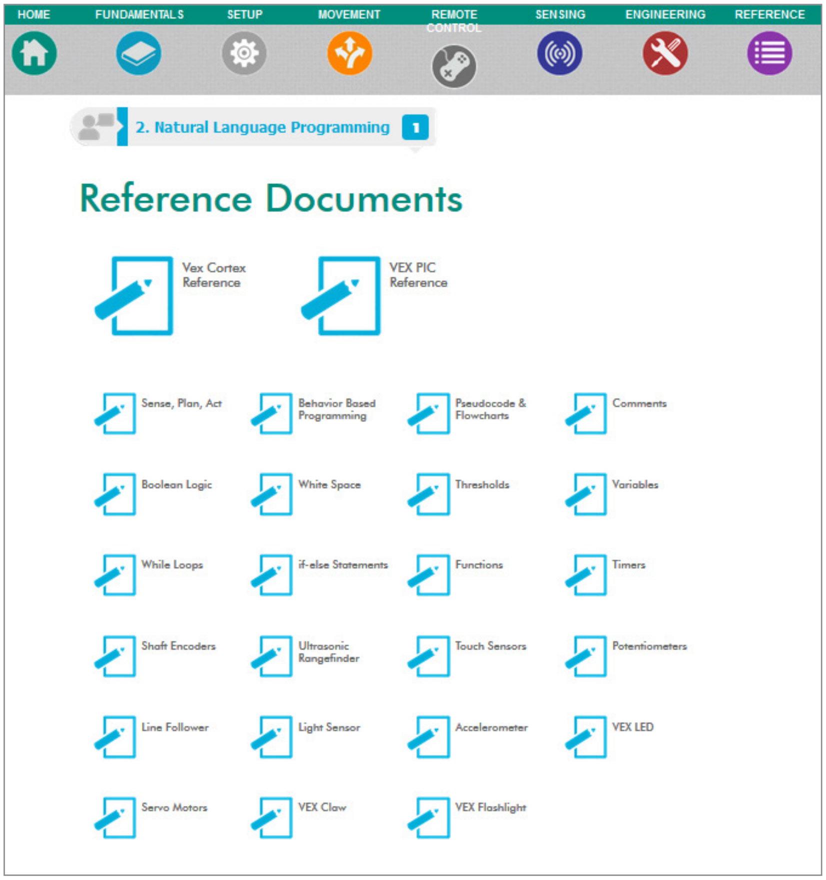


The ROBOTC Rules Part Two includes: The introductory video, check your understanding questions, and the three handouts shown below.



Comments, ROBOTC Error Messages, and ROBOTC Rules - These handouts are designed to complement the ROBOTC Rules programming videos above.

Fundamentals/Natural Language Programming



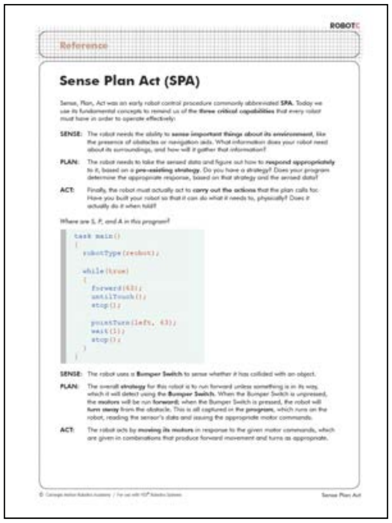
What is Natural Language?

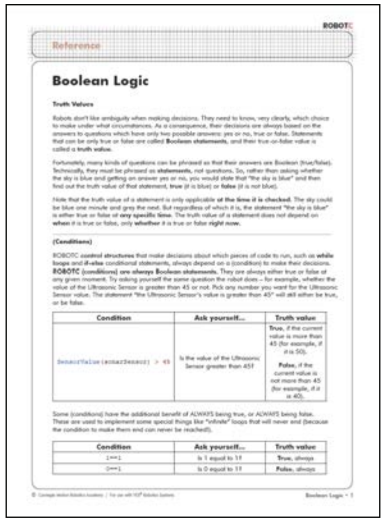
Natural Language provides an intuitive, easy to use, English language version of ROBOTC. Natural Language is for beginner programmers and is designed as a stepping stone to full ROBOTC programming The Natural Language Library is filled with commands that are both easy to use and easy to remember. Natural Language commands encompass entire robot behaviors into a single command. The documents provided with the curriculum were current with the date the curriculum was developed. If you are looking for the latest features of ROBOTC's Natural Language consult the ROBOTC Wiki at: http://www.robotc.net/wiki/VEX2_Natural_Language.

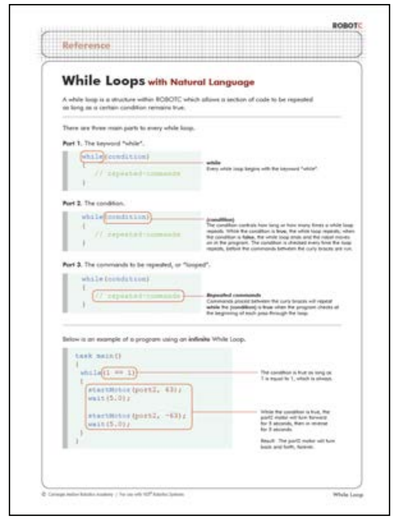
VEX PIC and VEX Cortex Natural Language reference guides. These two PDFs contain examples of the code and explanations of how the code works.

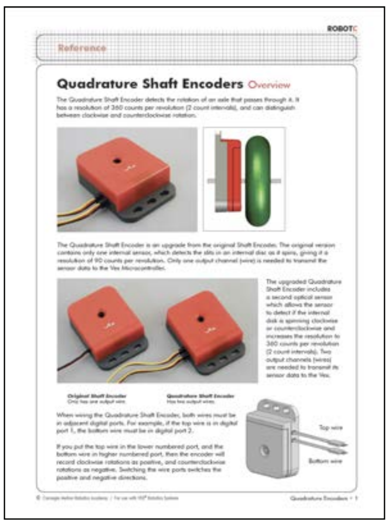
Note: Natural Language is a good choice if you have a very short amount of time to dedicate to teaching programming. BUT, when students begin to write complex programs Natural Language has limitations. If you are teaching a semester or a full year course it is recommended that you begin by teaching traditional ROBOTC.

Fundamentals/Natural Language Programming (NL) - Additional handouts that explain ROBOTC NL programming. The handouts can be used a study guides or quick reference sheets. They are available online at www.ROBOTC.net under VEX Curriculum so that they can be used for homework.


Reference
Sense Plan Act (SPA)


Reference
Boolean Logic

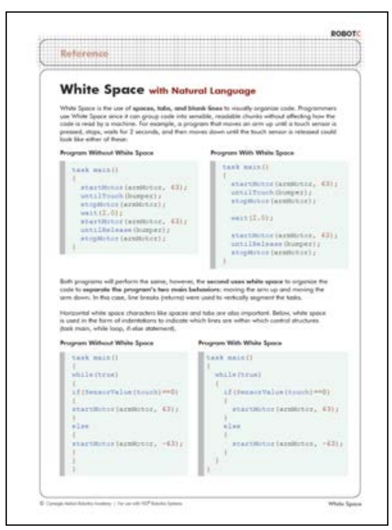

Reference
While Loops with Natural Language

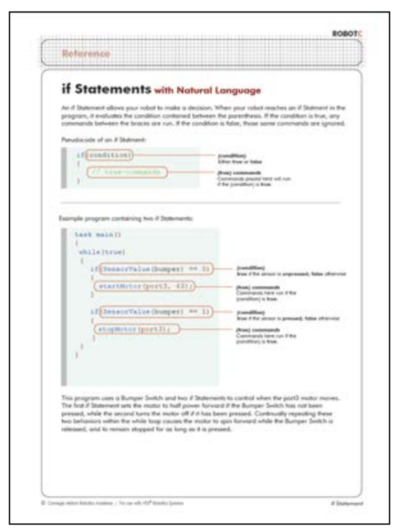

Reference
Quadrature Shaft Encoders Overview


Reference
Line Tracking Sensor Overview

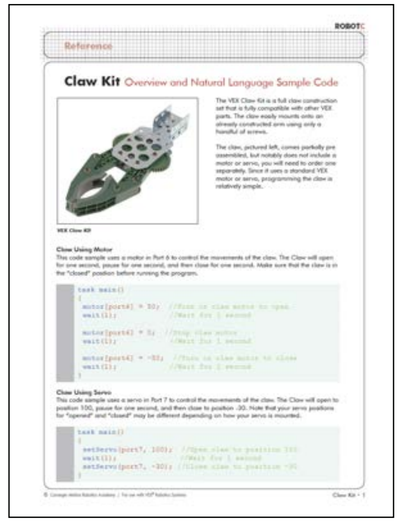

Reference
Servo Module Overview

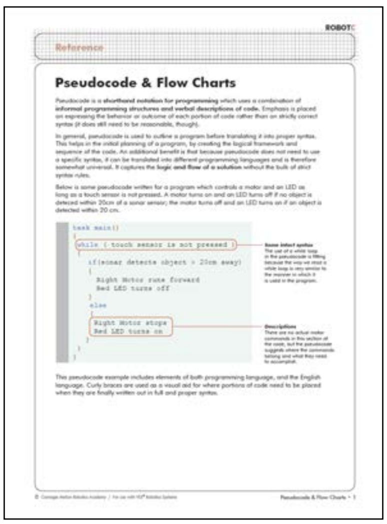

Reference
Behaviors

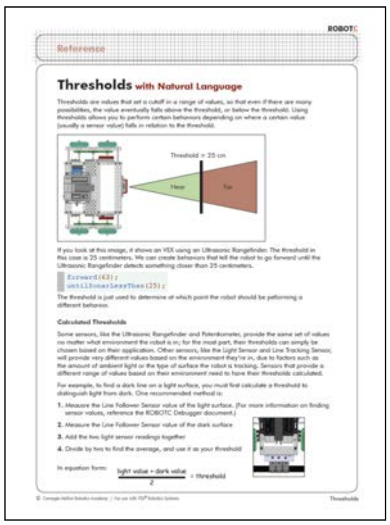

Reference
White Space with Natural Language

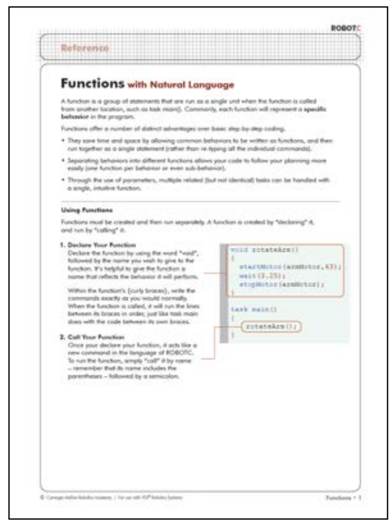

Reference
if Statements with Natural Language

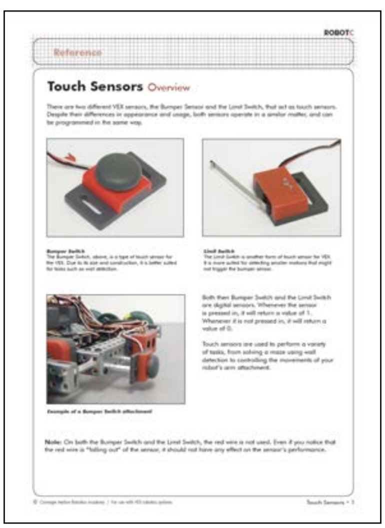
Fundamentals/Natural Language Programming - Additional handouts that explain ROBOTC NL programming. The handouts can be used a study guides or quick reference sheets. They are available online so that they can be used for homework.

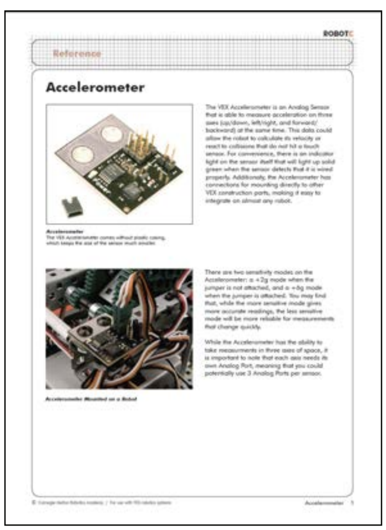




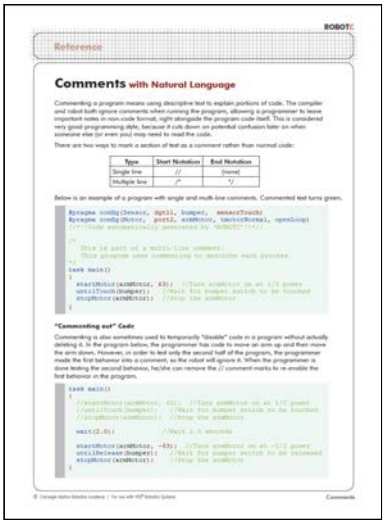


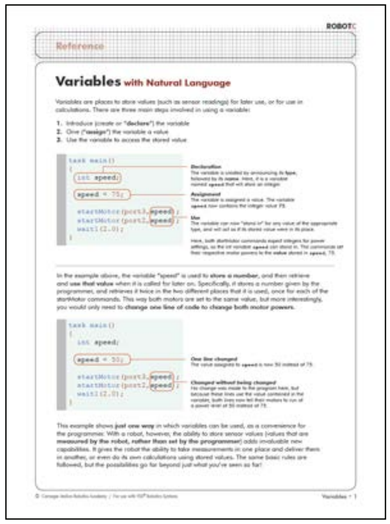


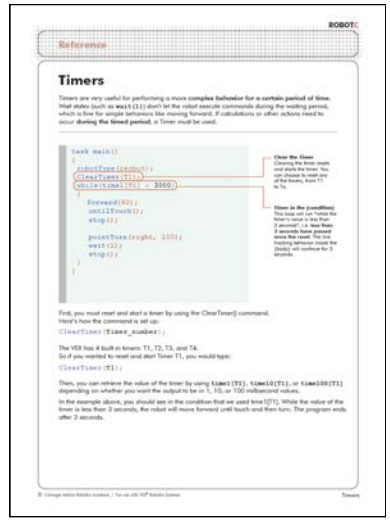


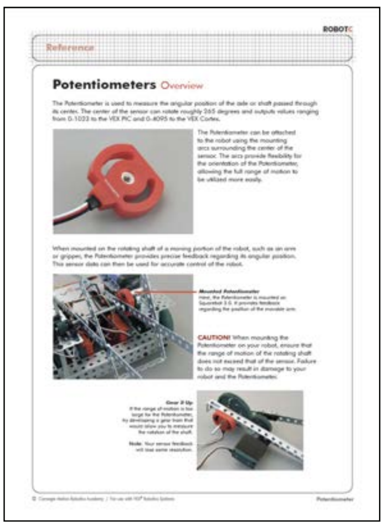


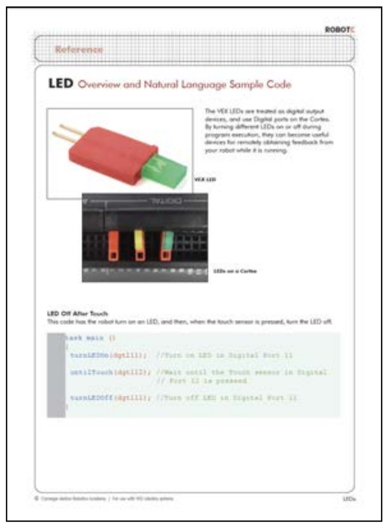
Fundamentals/Natural Language Programming - Additional handouts that explain ROBOTC NL programming. The handouts can be used a study guides or quick reference sheets. They are available online so that they can be used for homework.


ROBOTC
Reference
Comments with Natural Language


ROBOTC
Reference
Variables with Natural Language

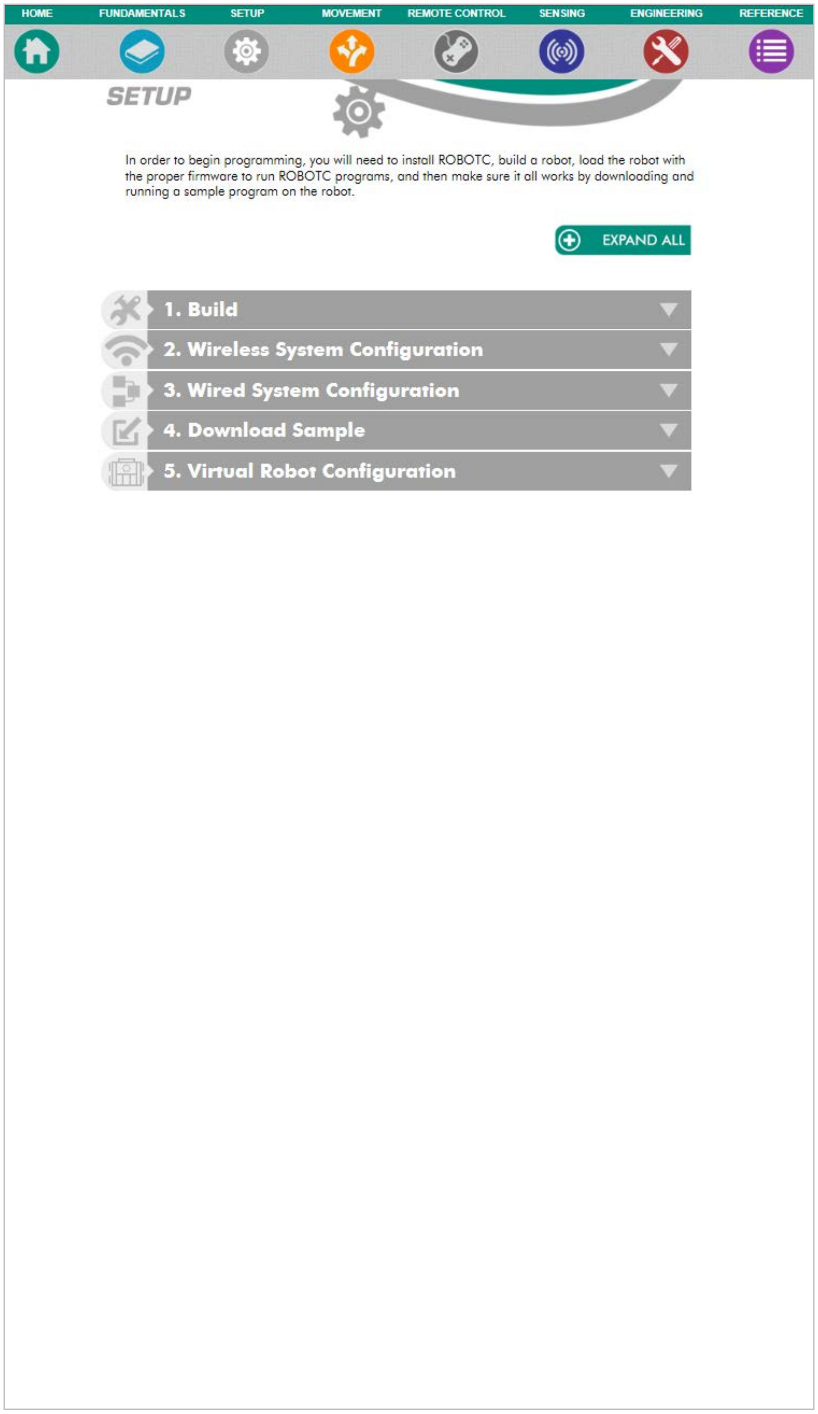

ROBOTC
Reference
Timers


ROBOTC
Reference
Potentiometers Overview


ROBOTC
Reference
LED Overview and Natural Language Sample Code

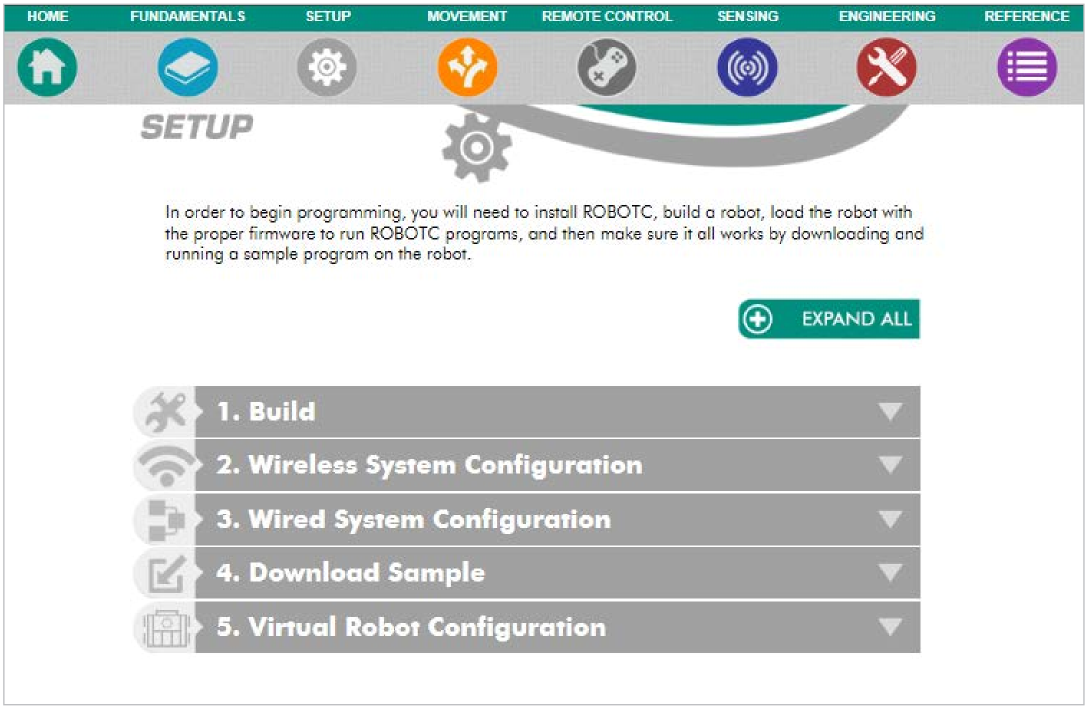
Natural Language Introductory Video -The Video at the left and all of the handouts are posted for free at: www.robotc.net/NaturalLanguage/

VEX Cortex Setup



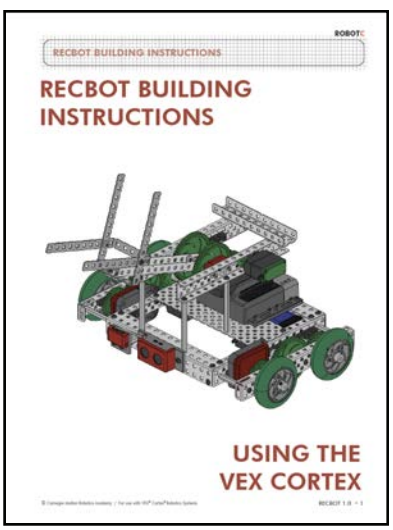
`Setup` - In order to begin programming, you will need to build a robot, use ROBOTC to update the firmware on the VEX Cortex and VEXnet Joystick, and then make sure that it works by downloading and running a sample program. The SETUP section guides a user step-by-step through this process. This section is divided into three sections: Build, System Configuration, and Download Sample.

Setup/Build

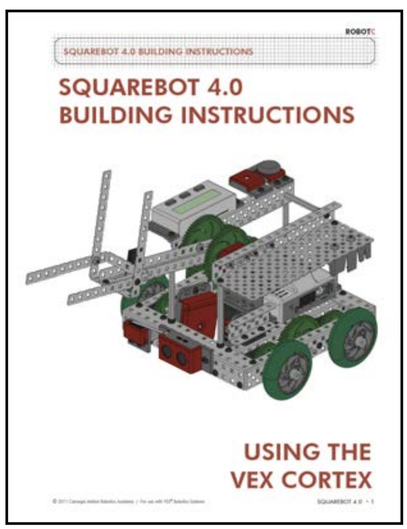


Note: the Clawbot with Sensors build on the next page is the recommended build for schools that purchased Clawbot kits.

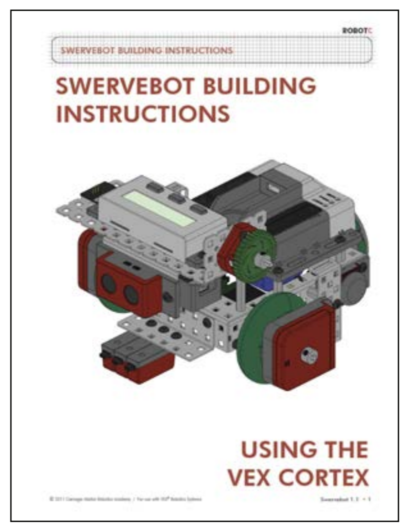
Robot Building instructions come in PDF format allowing you to either print them or use them directly from the screen. Additional instructions and updates will always live on the VEX Building Link at the Robotics Academy website.



RECBOT is the robot used most in the curriculum. It can be built out of the VEX Protobot kits with plus sensor kits.

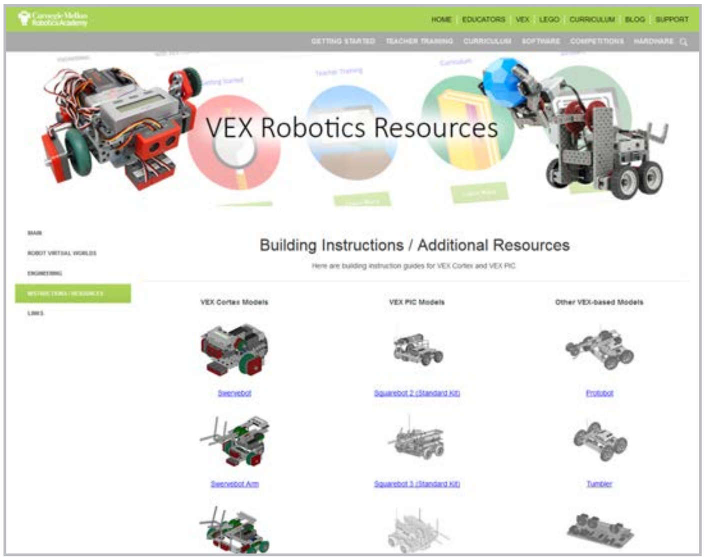


Squarebot 4.0 is an excellent alternative to the RECBOT, and takes up a little less space. It requires very little adjustment to make it work with the videos, and is also buildable using Protobot kits plus sensor kits.



Swervebot is a significantly smaller robot model that requires significant cutting to the VEX metal to create. It's smaller form factor make it ideal for storage and for robot applications that require sharp turns such as line following.

Additional Building Instructions for VEX Robots

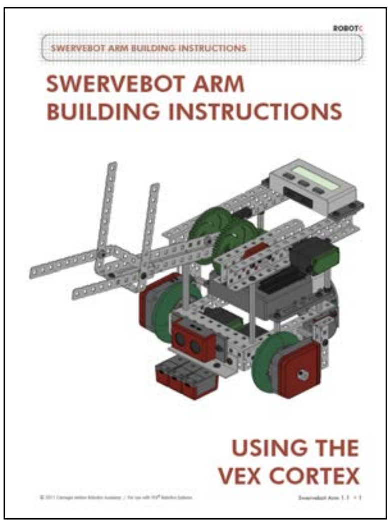


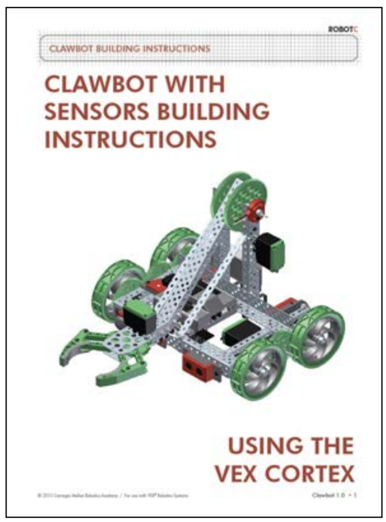
Choosing a robot model is a big choice because you may not have the time to have kids build robots multiple times during the year.

Read all of the notes below each set of plans before you begin.

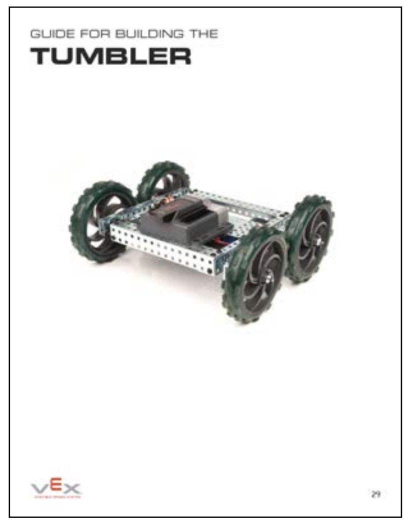
All of the Robot Plans plans are available via the VEX Robotics link at the Robotics Academy website.

www.vexteacher.com





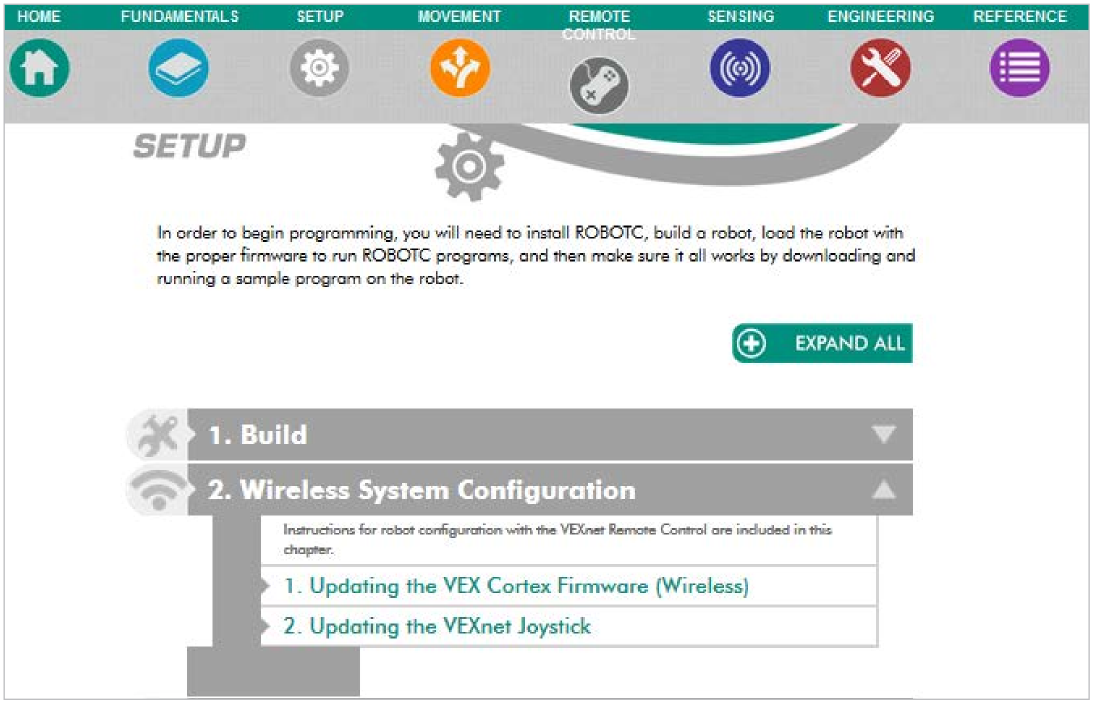
The Clawbot instructions are the best choice if you have the VEX Clawbot kits. Some adaptation will be needed on your part to account for the differences between it and the other builds, such as which motors are used to move the robot forward.



The Test Bed works well with Natural Language programming, and allows the system to be learned in a controlled environment.

Tumbler and Protobot are excellent robots but are not recommended for use with the curriculum.

Setup/Wireless System Configuration



Instructions for Wireless Programming

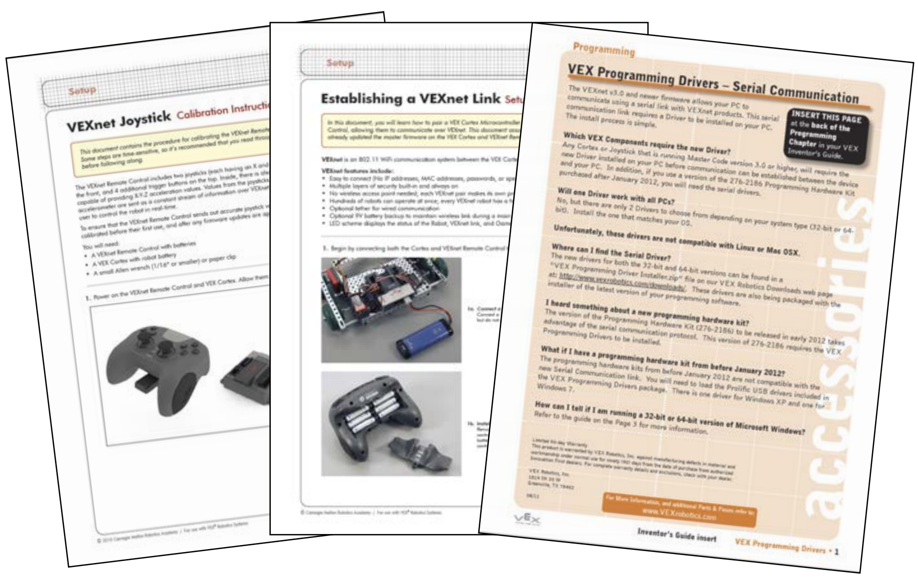
Updating the Cortex Firmware Video - Teaches students about the VEX Master firmware and the ROBOTC user firmware. The video teaches students how to download the latest firmware and software before they begin programming their robots.

Updating the VEX Joystick Firmware Video - Guides students step-by-step as they update the VEXnet Joystick firmware.

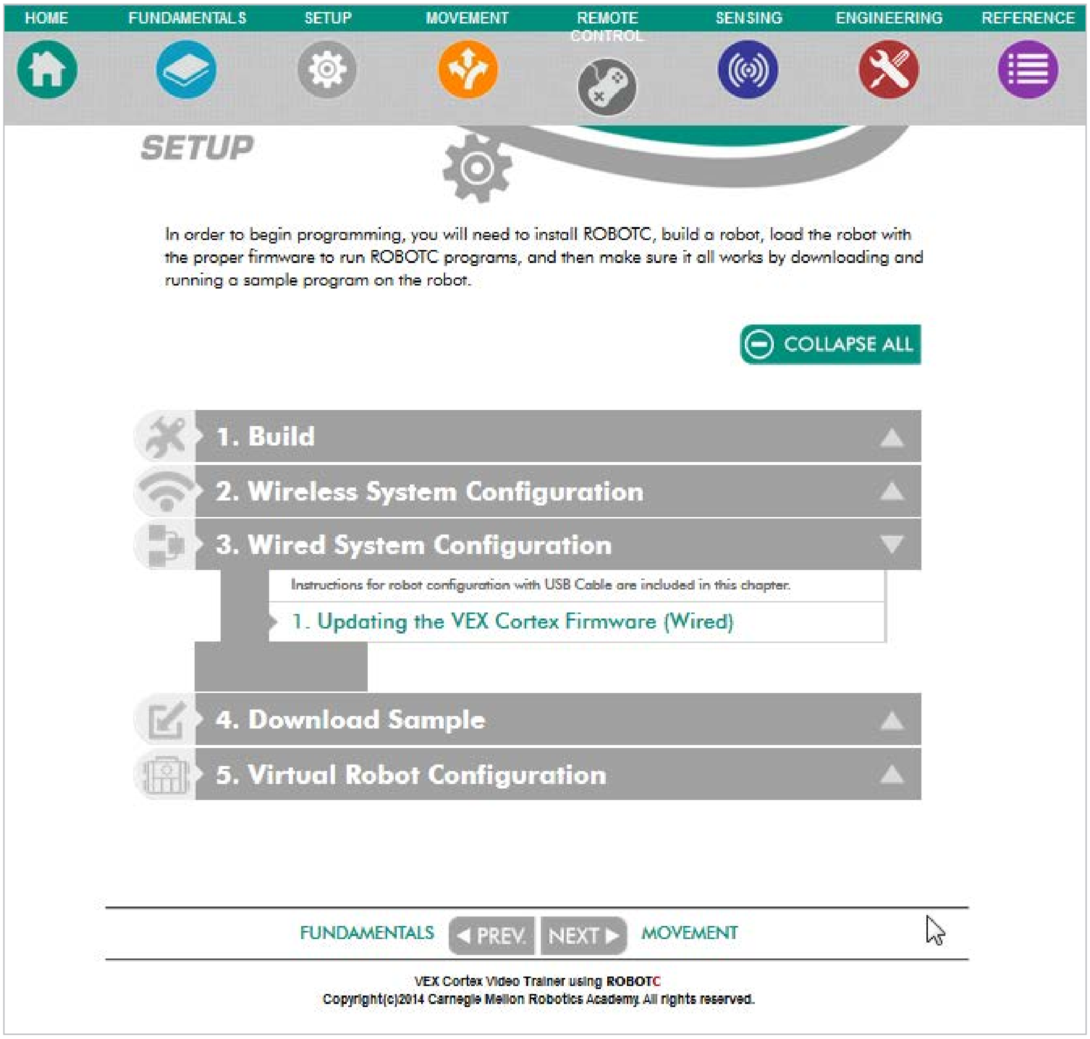
Establishing a VEXnet Link, VEXnet Joystick Calibration, and VEX Programming Drivers Reference Guides - Step-by-step instructions that guide students through the VEXnet setup and configuration process.

Update Cortex Firmware (Wireless)

Update VEXnet Joystick Firmware



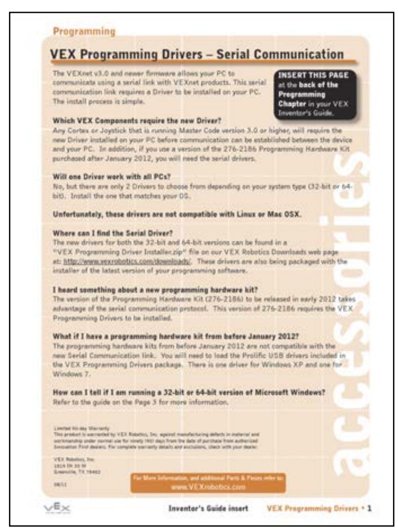
Setup/Wired System Configuration



Instructions for Wired System Configuration

Updating the Cortex Firmware Video - Teaches students about the VEX Master firmware and the ROBOTC user firmware. The video teaches students how to download the latest firmware and software before they begin programming their robots.

VEX Programming Drivers Reference Guides - Step-by-step instructions that guide students through the VEXnet setup and configuration process.



Programming

VEX Programming Drivers – Serial Communication

The VEXnet v3.0 and newer firmware allows your PC to communicate using a serial link with VEXnet products. This serial communication link requires a Driver to be installed on your PC. The install process is simple.

INSERT THIS PAGE at the back of the Programming Chapter in your VEX Inventor's Guide.

Which VEX Components require the new Driver?
Any Cortex or Joystick that is running Master Code version 3.0 or higher, will require the new Driver installed on your PC before communication can be established between the device and your PC. In addition, if you use a version of the 276-2186 Programming Hardware Kit purchased after January 2012, you will need the serial drivers.

Will one Driver work with all PCs?
No, but there are only 2 Drivers to choose from depending on your system type (32-bit or 64-bit). Install the one that matches your OS.

Unfortunately, these drivers are not compatible with Linux or Mac OSX.

Where can I find the Serial Driver?
The new drivers for both the 32-bit and 64-bit versions can be found in a "VEX Programming Driver Installer.zip" file on our VEX Robotics Downloads web page at: http://www.vexrobotics.com/downloads/. These drivers are also being packaged with the installer of the latest version of your programming software.

I heard something about a new programming hardware kit?
The version of the Programming Hardware Kit (276-2186) to be released in early 2012 takes advantage of the serial communication protocol. This version of 276-2186 requires the VEX Programming Drivers to be installed.

What if I have a programming hardware kit from before January 2012?
The programming hardware kits from before January 2012 are not compatible with the new Serial Communication link. You will need to load the Prolific USB drivers included in the VEX Programming Drivers package. There is one driver for Windows XP and one for Windows 7.

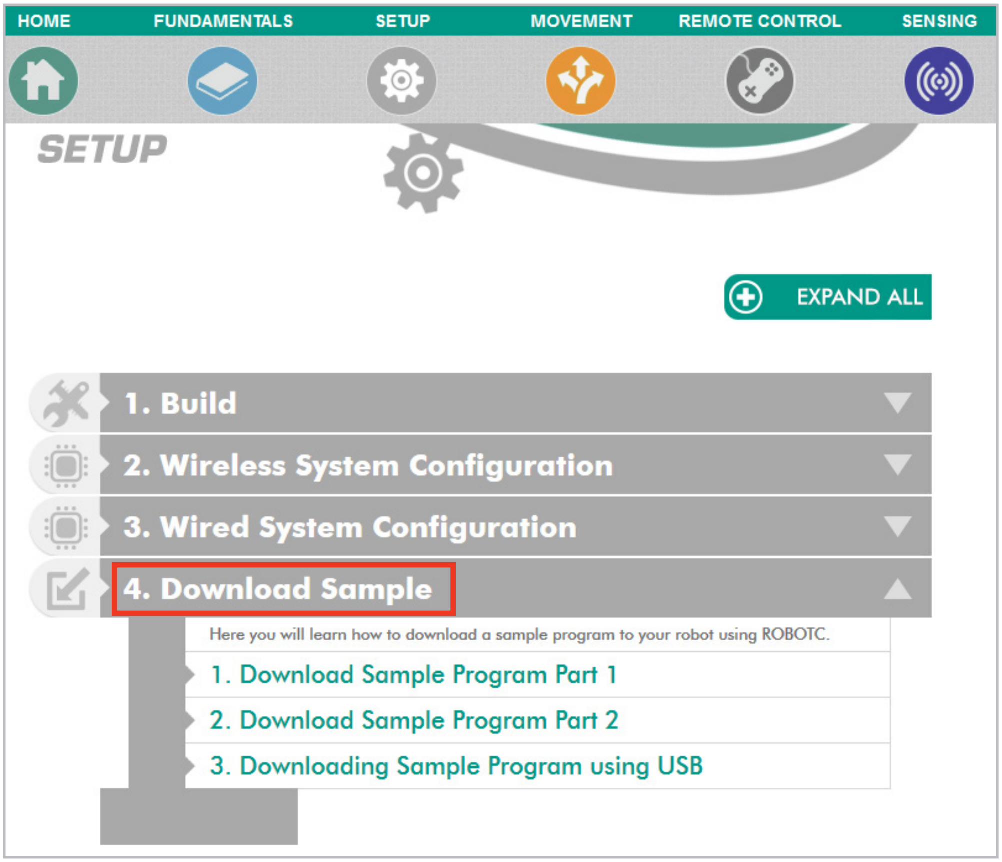
How can I tell if I am running a 32-bit or 64-bit version of Microsoft Windows?
Refer to the guide on the Page 3 for more information.

For More Information, and additional Parts & Pieces refer to www.VEXrobotics.com

accessories

Inventor's Guide insert VEX Programming Drivers • 1

Setup/Downloading Sample Program



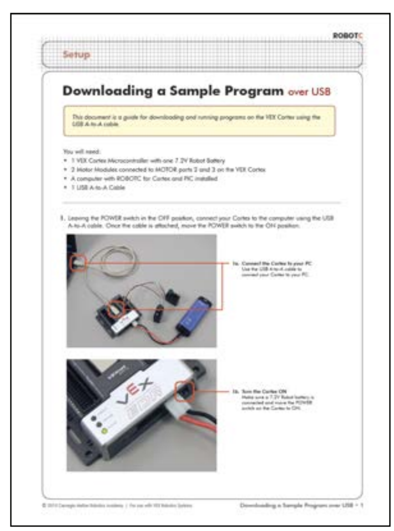
Instructions for Wireless Programming

Downloading a Sample Program over VEXnet Videos - Guides students step-by-step through the process of downloading a ROBOTC program. The first video tells them how to setup their hardware. The second video shows them how to powerup their VEX Cortex and VEXnet Joystick, confirm their settings, and then download their program. Once this process is complete the Cortex remembers all of the settings and so setup only need to be done once.

USB to Serial Driver Installation and Downloading a Sample Program over USB Reference Guides - Step-by-step instructions that guide students through installing drivers.

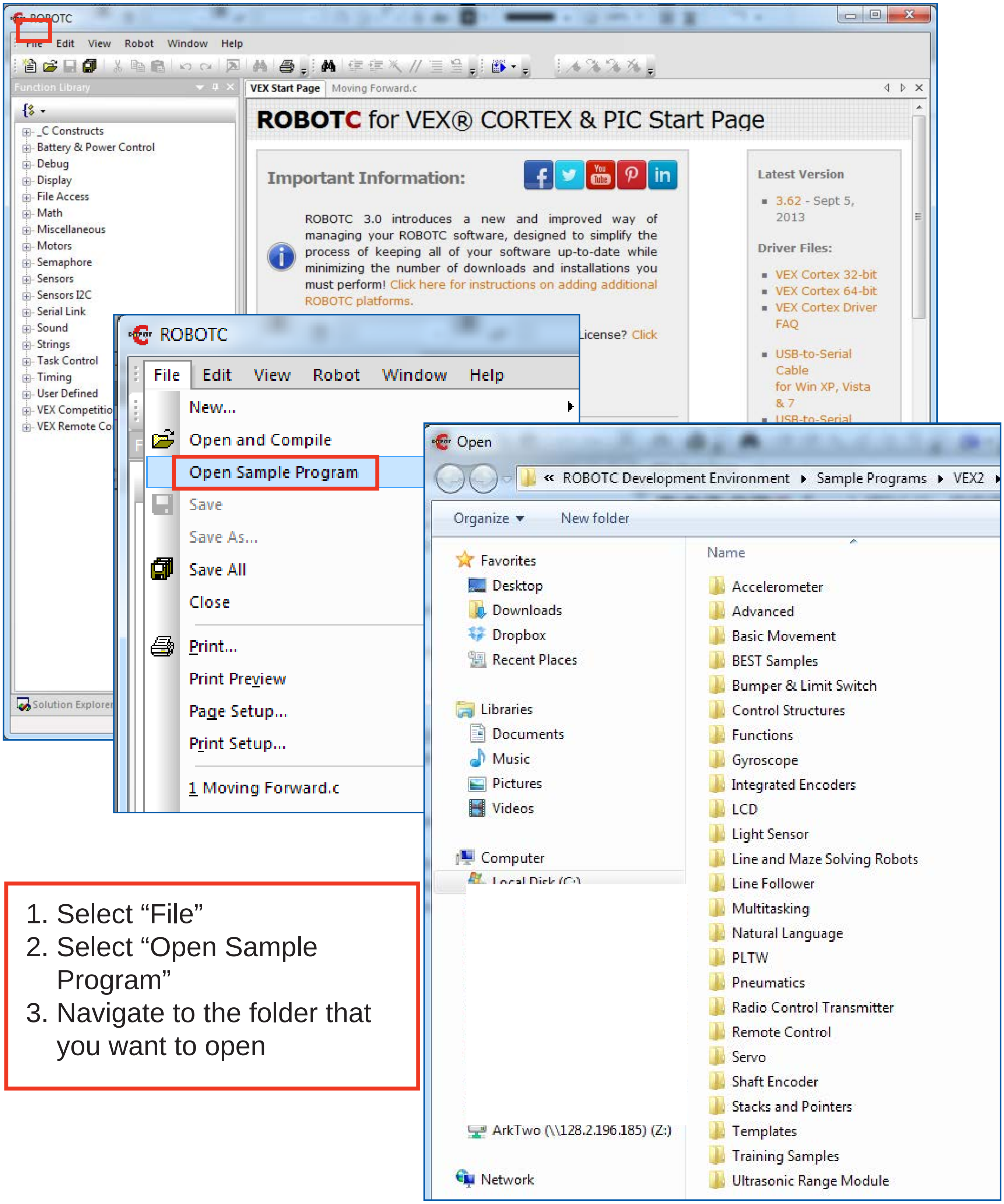
Download Sample Program over VEXnet Part 1

Download Sample Program over VEXnet Part 2



ROBOTC Has over 250 Working Sample Programs

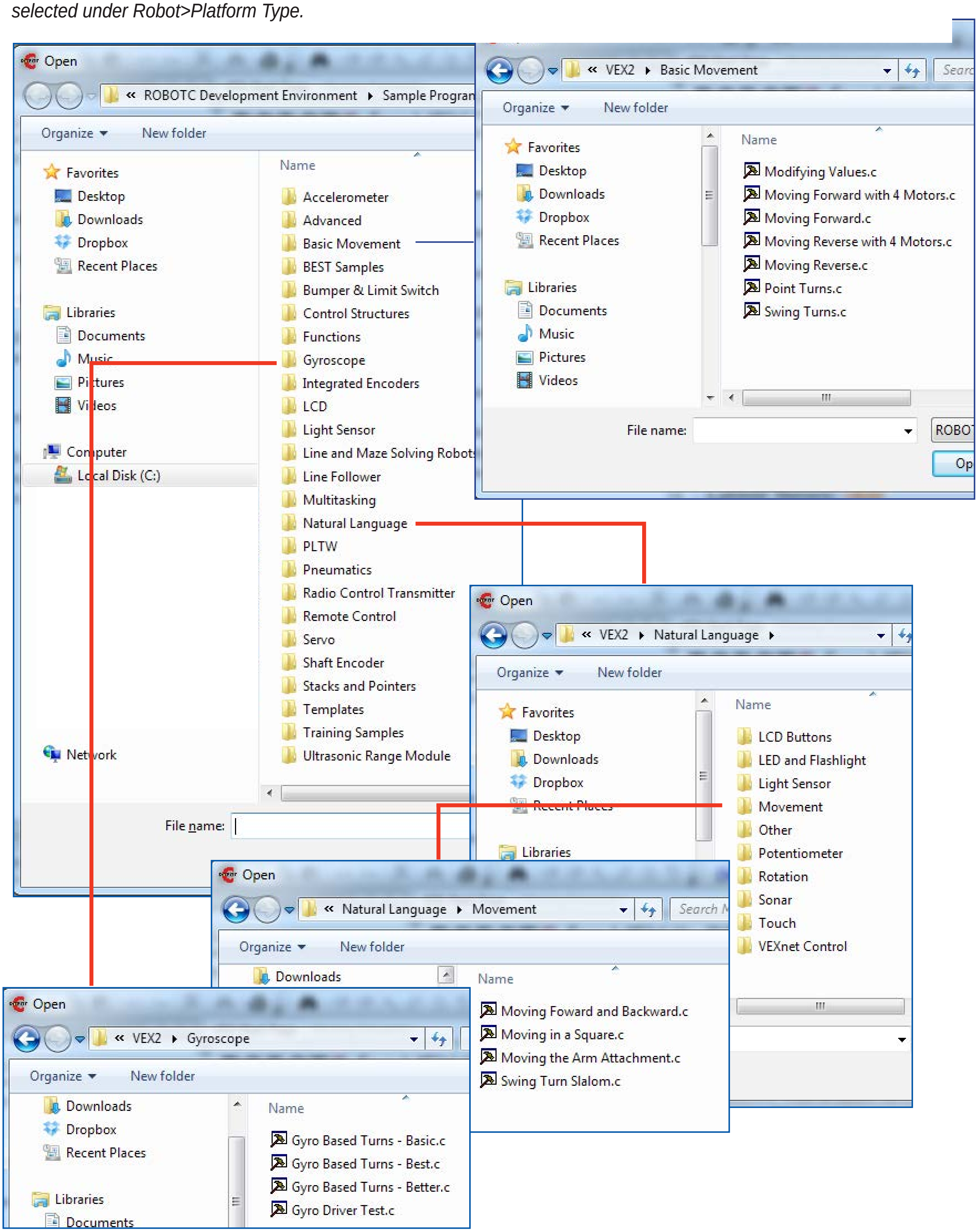
Often it is easier for new programmers to get a feel for programming by editing working programs. To access ROBOTC's extensive library of examples select "File" and then "Open Sample Program". Each folder contains examples of working code that can be loaded onto a student's robot and tested. *Note: the code in those sample programs was written for a specific robot and adjustments may need to be made in ROBOTC's Motors and Sensors Setup.*



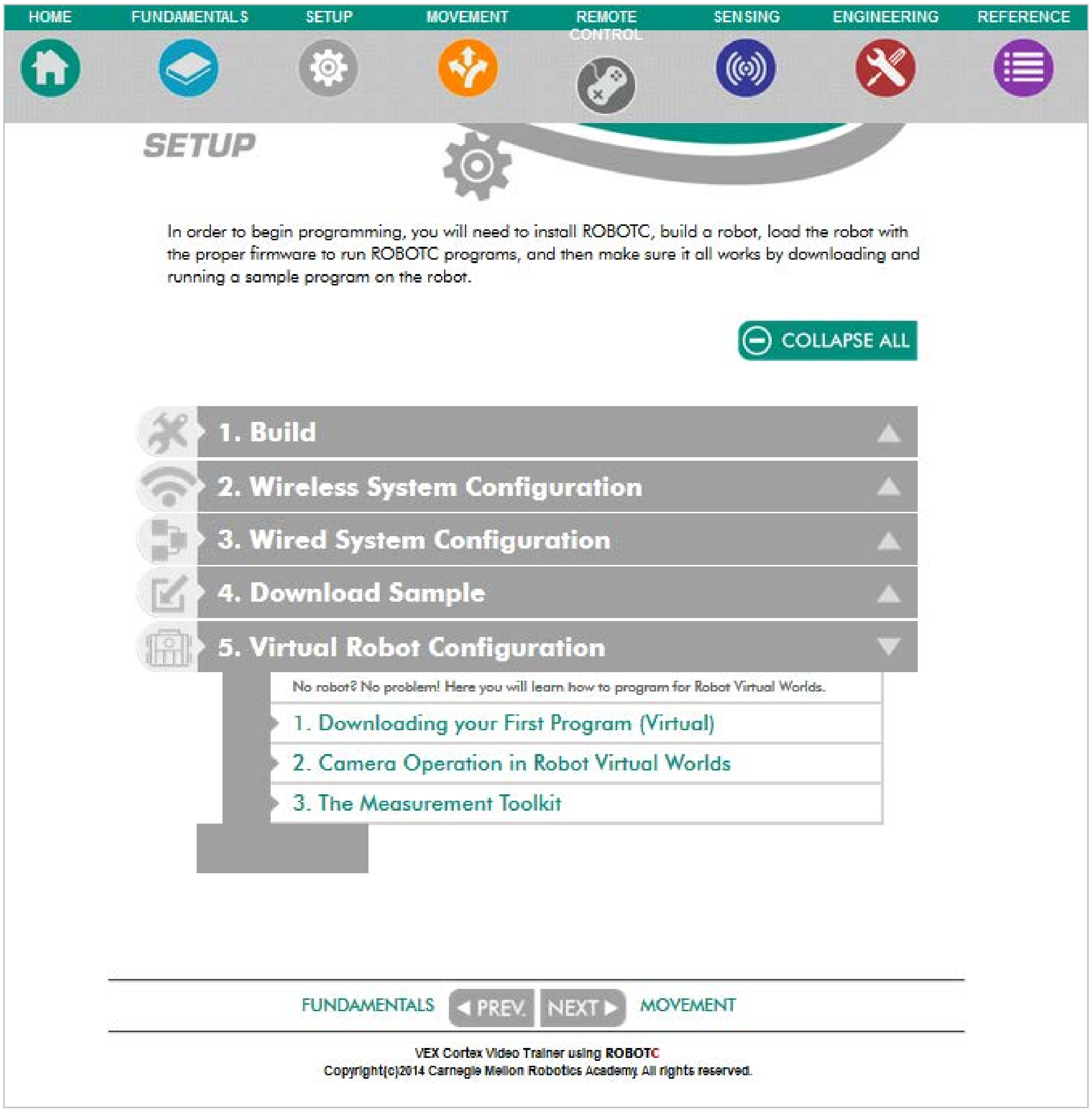
1. Select "File"
2. Select "Open Sample Program"
3. Navigate to the folder that you want to open

ROBOTC Has over 250 Working Sample Programs

Pictured below are the types of programs that a student will find. The programs range from Basic Movement using standard ROBOTC to Basic Movement using ROBOTC's Natural Language Library to using the VEX Gyro Sensor and everything in between. *Note: when programming using Natural Language Commands, Natural Language must be selected under Robot>Platform Type.*

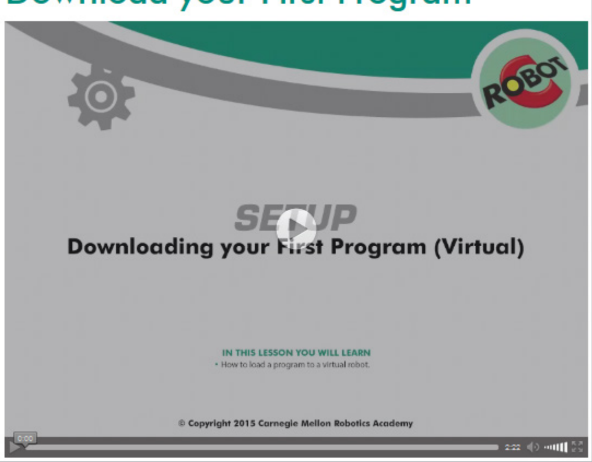


Setup/Virtual Robot Configuration



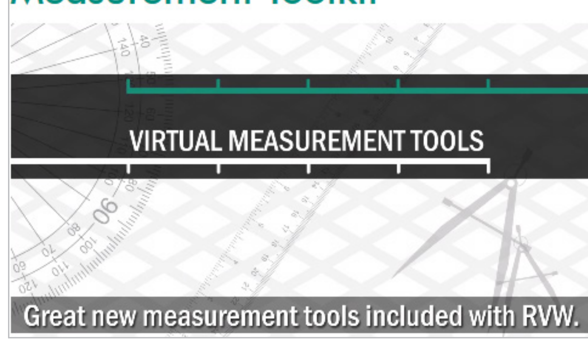
This lesson set includes three instructional videos designed to help the new user to program using Robot Virtual Worlds.

Download your First Program



Camera Operation in RVW

Measurement Toolkit



Robot Virtual World Software - the Curriculum Companion

Teaching Programming via Robot Virtual Worlds (RVW)

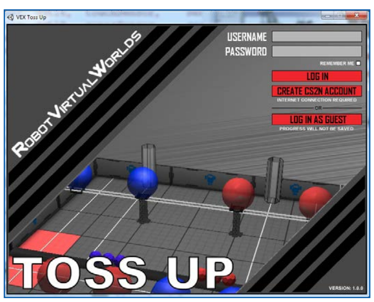
The Curriculum Companion is designed to mirror the VEX Cortex Video Trainer using ROBOTC. The RVW software includes over 40 programming challenges that are found in the VEX Cortex Video Trainer Curriculum.

Throughout the Introduction to Programming section you will find icons that show which challenges are included.

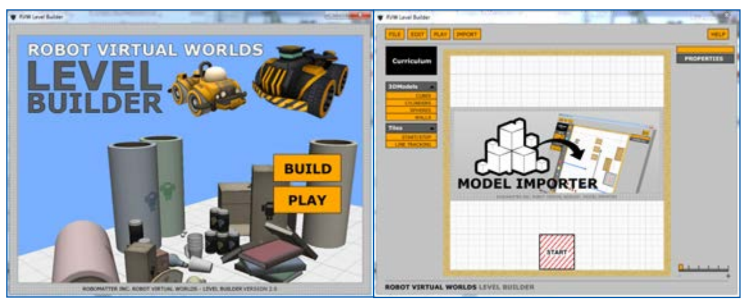
Additionally, students can pick from five robot configurations to choose to program, they are shown at the left of this page; VEX Squarebot, VEX Clawbot, VEX Swervebot, BuggyBot, and Mammal Bot.

Additional VEX Robotics Virtual Worlds

A ROBOTC RVW License gives you access to the Curriculum Companion, *PLUS* all of the RVWs below.



The current VEX Robotics Engineering Challenge. Students can begin developing code and strategies before they build their robots



The Level Builder and Model Importer allow student to build their own levels and import their own models into their new worlds. RVW is compatible with any modeling software that can generate a .STL file, including Autodesk, SolidWorks, and Google Sketchup.

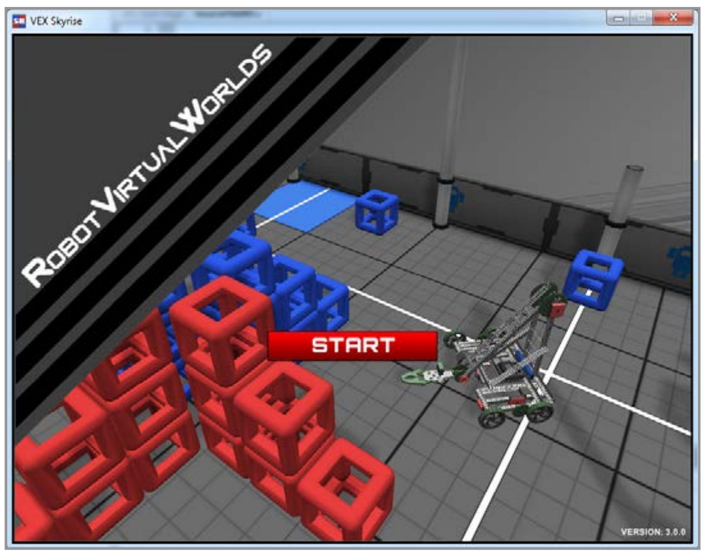
The Palm Island, Operation Reset, and Atlantis Robot-To-The-Rescue programming games take kids to yet to be discovered worlds where their robots need to be programmed to solve challenges!

Virtual Robot Competitions

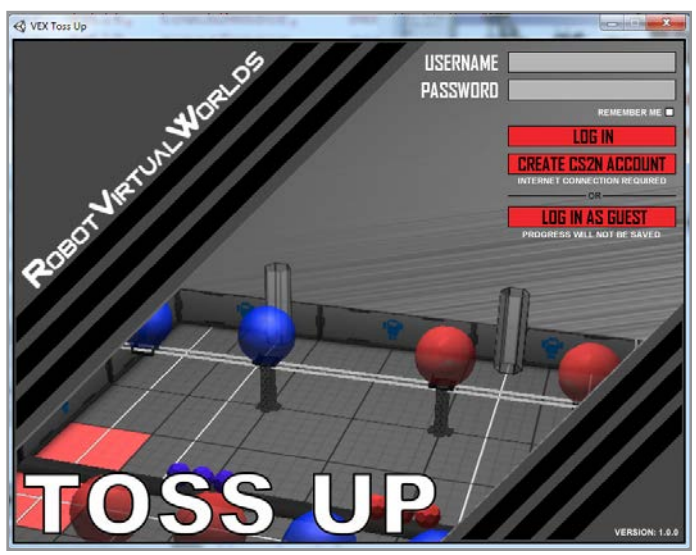
Each year, VEX robot competitions are modeled and available for use in your class. Pictured below are prior year example competitions.

Host your own in school virtual competition! All of these games are available at the Robot Virtual World website.

Pictured above is Nothing But Net, the VEX 2015 Game

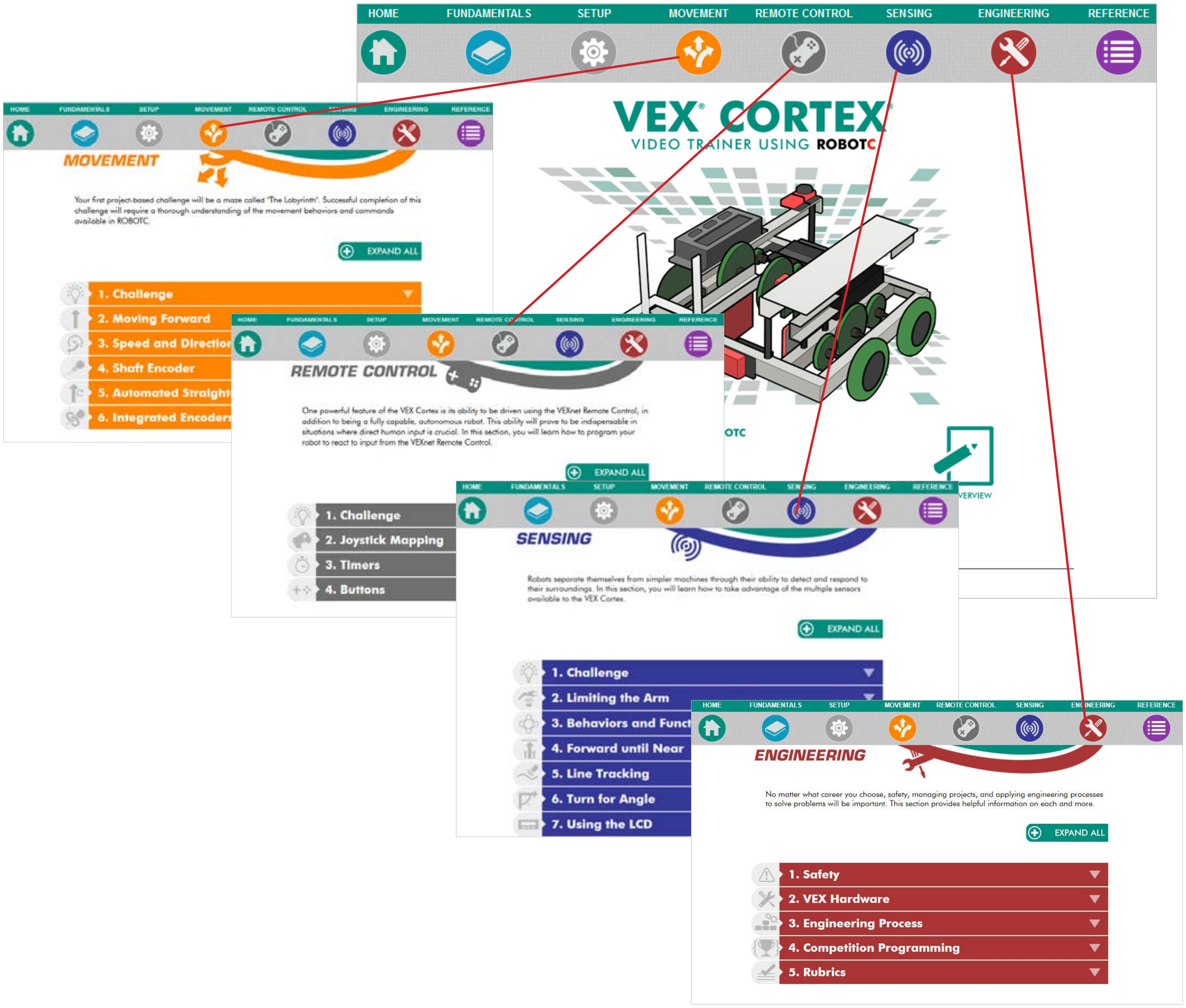


Pictured at the right is Skyrise, the VEX 2014 Game



Pictured at the right is Toss UP, the VEX 2013 Game

Guided Programming and Engineering Challenges

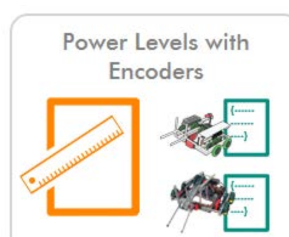


The Guided Programming and Engineering Lessons are designed to be “learner centered”.

Students work through the lessons in a step-by-step fashion. The first three units: Movement, Remote Control, and Sensing contain Lesson Sets teach a particular programming concept and include several programming and engineering challenges (see below). The last lesson set is “Engineering”, This section includes resources that students will use as they solve their engineering design challenges.

Engineering Investigations and Programming Challenges

At the bottom of many of the lessons students will find programming challenges and engineering investigations. The Basketball Drills picture shows the icon that indicates a programming challenge, the Power Levels with Encoders icon represents and Engineering Investigation.



The Movement Unit

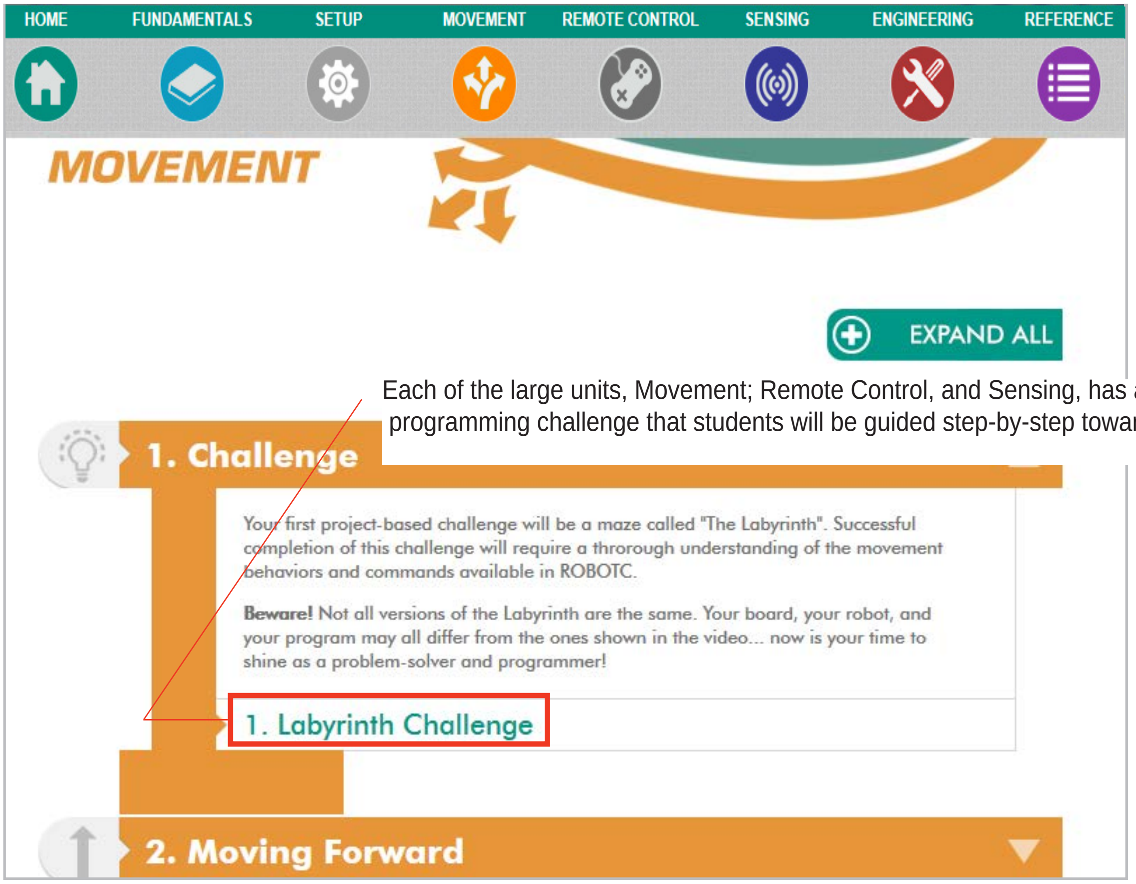
The Movement Unit

The Movement Unit is taught using four Lesson Sets and a programming challenge. The Lesson Sets begin using sample code that is already included in ROBOTC. The first Lesson Set, Moving Forward, teaches students in a very lockstep manner what each line of code does while introducing them to moving motors for specific amounts of time. The second Lesson Set, Speed and Direction, explains motor power levels and how to reverse polarity. The second Lesson Set includes an “engineering lab” that the students will complete. The engineering labs place students in the role of engineer where they run their robots, measure results, iteratively test the results to determine reliability, and then extrapolate from their data set to predict new robot behaviors. The third Lesson Set, Shaft Encoders, begins to introduce students to Boolean Logic and While Loops. Automated Straightening introduces students to if-else Statements and Variables and teaches them to develop their own automated straightening algorithm. The Integrated Encoders lesson teaches how to use the Integrated Motor Encoders.

The Movement Unit also includes fourteen programming challenges where students are challenged to solve simple movement programming challenges. Although some of the challenges appear to be repetitive, the extra challenges enable the teacher to differentiate instruction, it will be up to the teacher to decide which students do which challenges. The extra challenges give the teacher to differentiate the instruction based on student’s ability.

Note: It will be important to remind students that although the initial work may seam easy, that the skills that they learn in the movement unit are foundational pieces that they must understand before they move to the Remote Control and Sensing Units.

Movement/Challenge Description



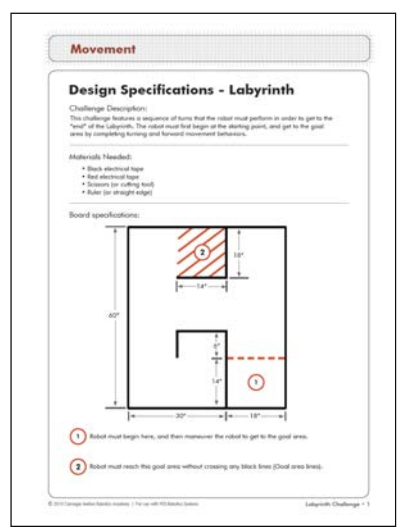
Each of the large units, Movement; Remote Control, and Sensing, has a unit programming challenge that students will be guided step-by-step toward a solution.

Labyrinth Programming Challenge

Each Programming Unit (Movement, Remote Control, Sensing, and Resources) contains a Unit Programming challenge that is designed to place the learning into an interesting context. In the "Movement" Lesson Set the programming challenge is the "Labyrinth Challenge".

In this challenge, students will learn:

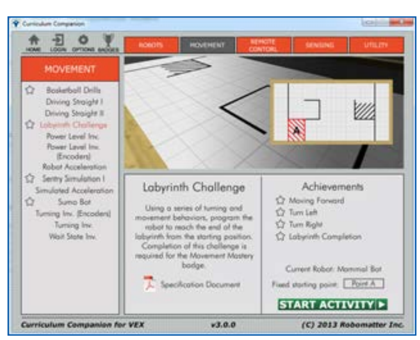
- Behavior based programming logic
- How to program their robots to accurately move forward, backward and turn
- The syntax rules related to programming using ROBOTC



Labyrinth Challenge Teacher Resources

The Movement Unit programming challenge comes with a PDF that explains the rules to the challenge, an example video solution.

The Labyrinth Challenge is also available in a Robot Virtual World simulation format.



Movement/Moving Forward

The Moving Forward Lesson Set

Each Lesson Set is designed to teach a related set of programming concepts and is designed to give students confidence opening a program and understanding what the individual lines of code mean. Every set is supported with a combination of video and print resources:

Program Dissection Video - Students are given a line by line description of the code used in the first sample program.

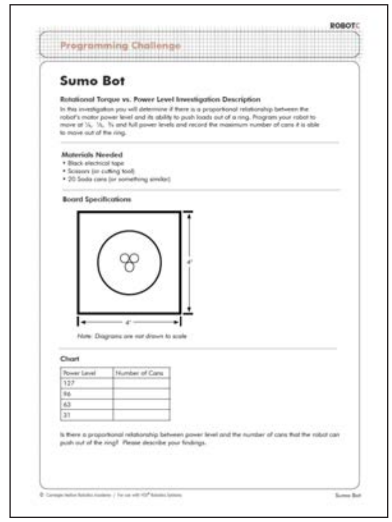
Reversing Polarity Video - This video shows students how to make their robot change directions and turn.

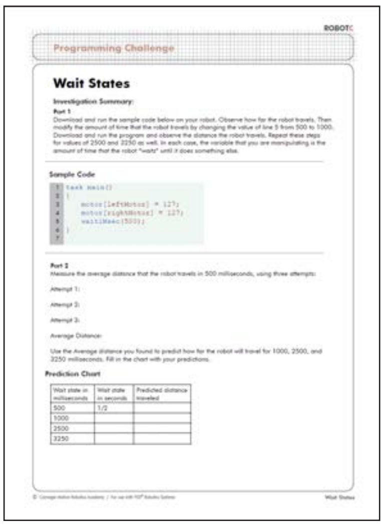
Renaming Motors Video - ROBOTC provides a very handy utility that allows the programmer to rename their motors using names that make sense to them (i.e. leftMotorArm, or RightWheel). This video shows how it works.

The Moving Forward Lesson Set is continued on the next page.

Movement/Moving Forward (continued)


ROBOTC
Reference
Running a Program


ROBOTC
Programming Challenge
Sumo Bot

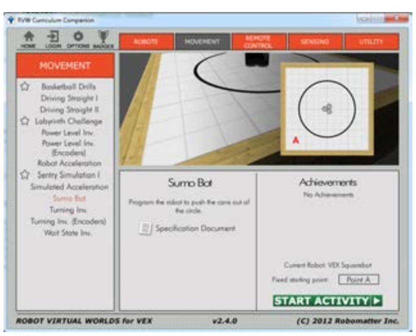

ROBOTC
Programming Challenge
Wait States

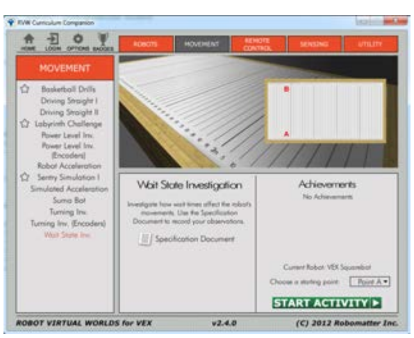
The Moving Forward Lesson Set

Moving Forward Timing Video - Timing is the least accurate way to control a robot, but it is also the simplest method.

Wait States Engineering Investigation - This PDF contains an Engineering Investigation that requires students to take an initial measurement and then to make predictions of how far their robot will travel based on changing the amount of time the robot is set to run.

SumoBot Challenge- At this point in a student's programming career all they can do is move forward. This is a nice engineering challenge that allows students to focus on weight distribution, traction, and design.



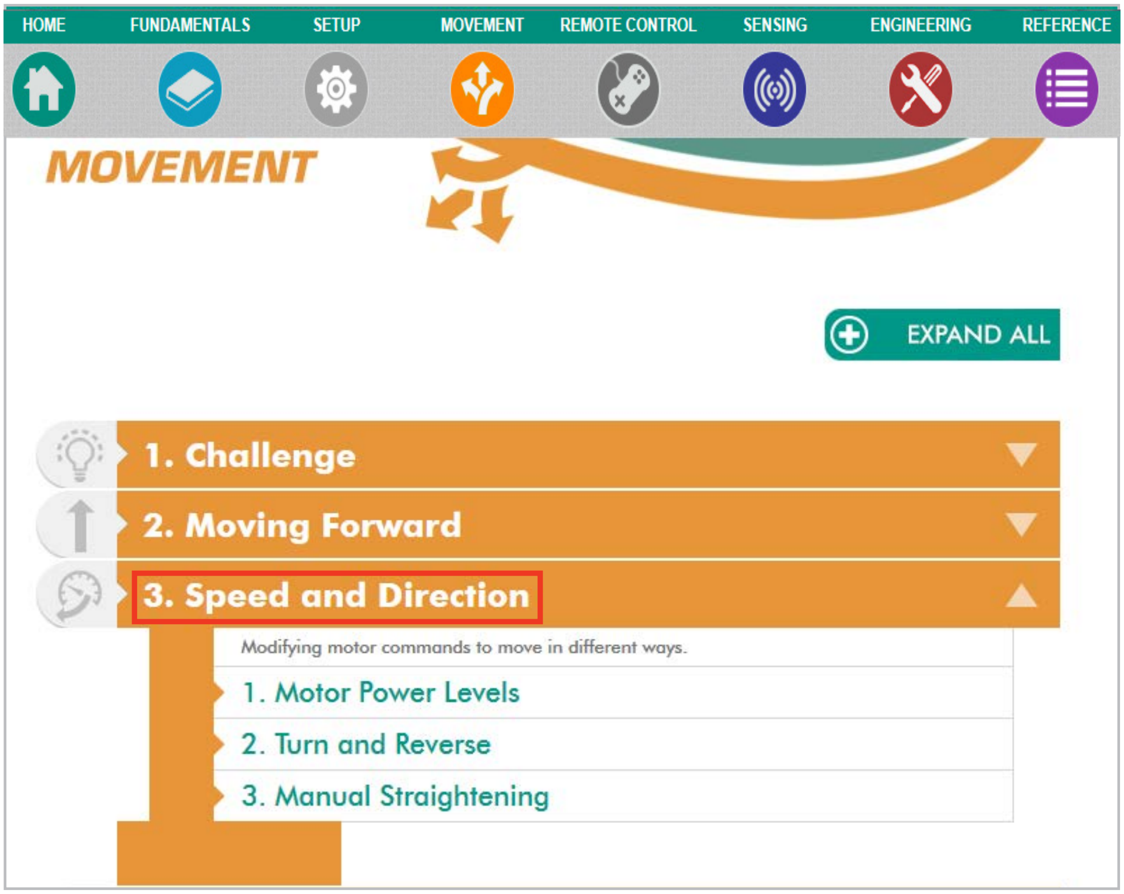


Robot Virtual World Simulations

Wait States Engineering Investigation - Students are required to run their robots, take measurements, and then predict and record their measurements for their next run.

Ths SumoBot Challenge - Enables students to simulate the programming challenge.

Movement/Speed and Direction



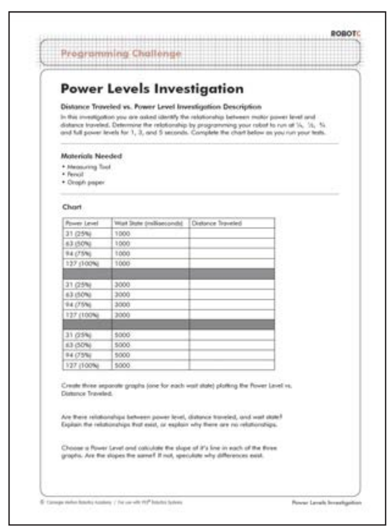
The Speed and Direction Lesson Set

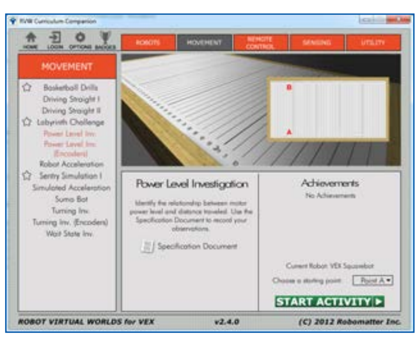
In this lesson set students continue to build confidence in their ability to program simple movements. The lesson set consists of three videos, three Engineering Challenges, and two Programming Challenges.

Motor Power Lesson

The Motor Power Level Video - teaches students how to change the power level on the robot. Students learn that as they change the power level, they are in effect changing the robot's speed. They will also complete an Engineering Lab named "Power Level Investigation". In this investigation, students program their robots at a variety of power levels and keep the amount of time the robot runs constant. In the investigation students will investigate if there is a proportional relationship between power levels and speed.

The Power Level Investigation is also available as a Robot Virtual World Simulation.





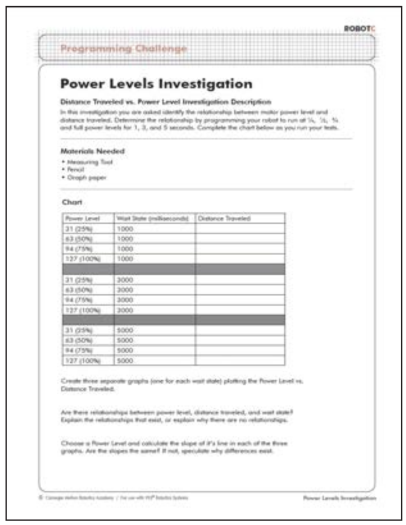
Movement/Speed and Direction (continued)

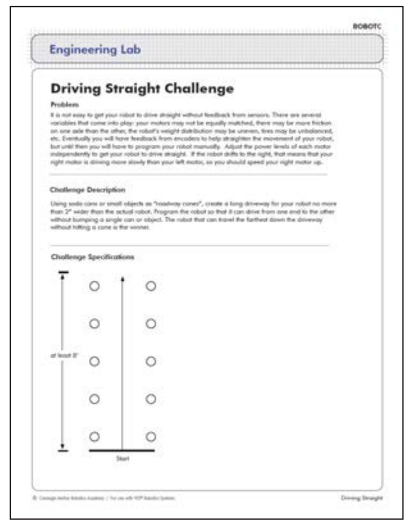
Speed and Direction

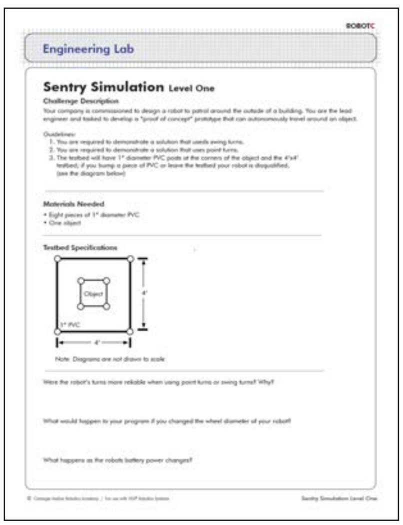
Turn and Reverse Video - teaches students how changing motor polarity enables the robot to turn and change direction.

Manual Straightening Video - Challenges the student to use power levels to control the robot to go straight (students will learn a better method soon).

Programming Challenge PDFs - The Power Level Investigation, Driving Straight Challenge, and Sentry Simulation provide students with opportunities to practice programming and learn about the VEX robot system.

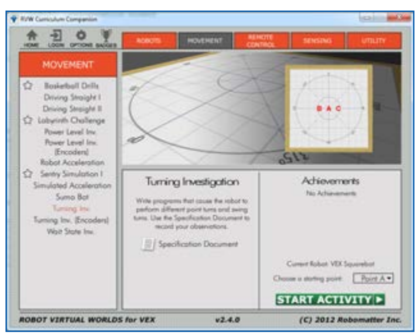


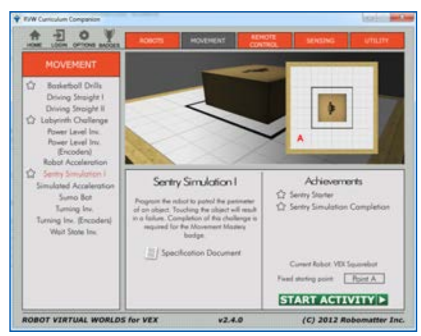


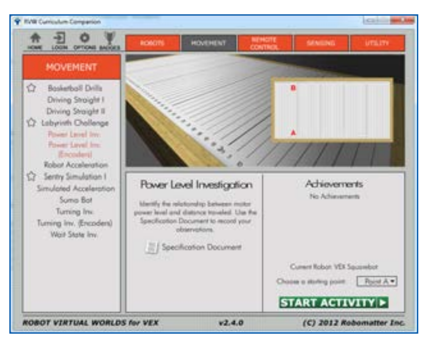


Robot Virtual World Simulations

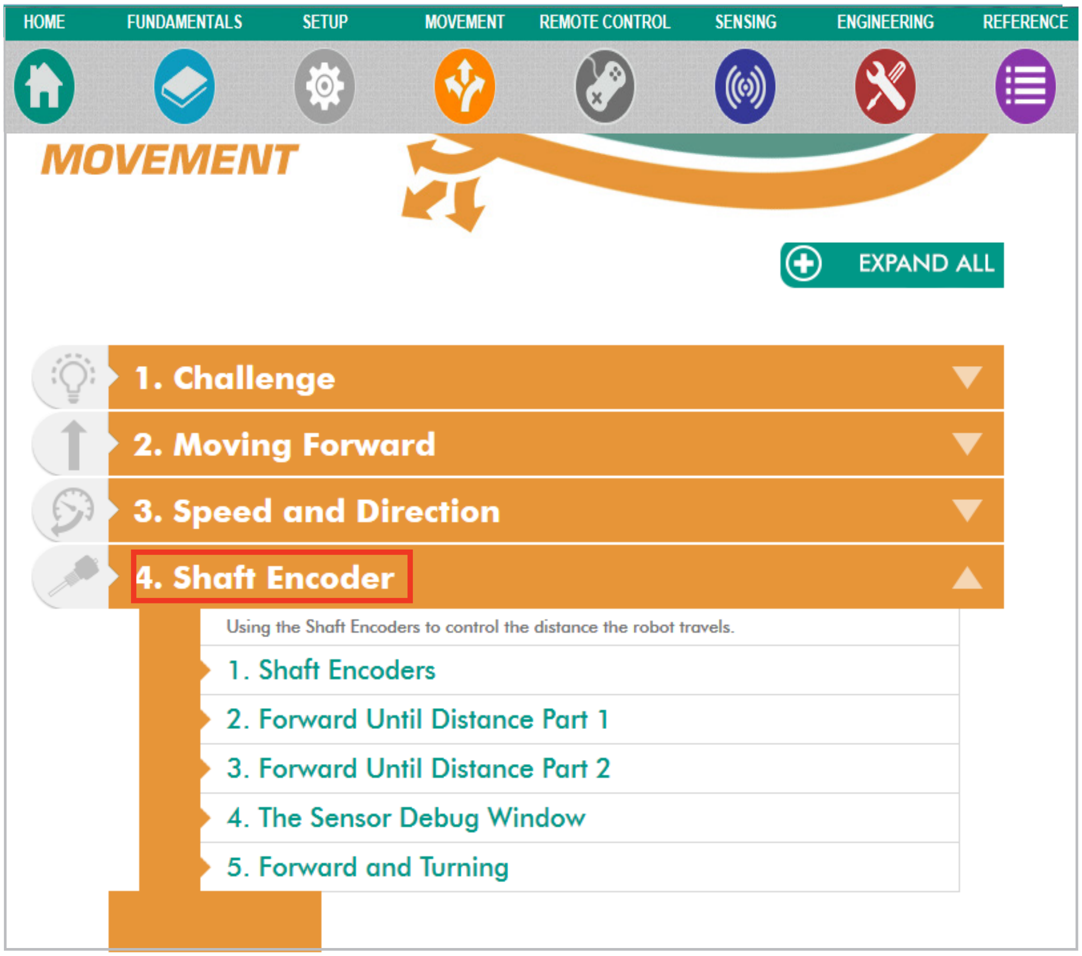
Motor Power Levels, Sentry Simulation, Turning, and the Driving Straight challenges are available via the ROBOTC RVW simulation software.







Movement/Shaft Encoders



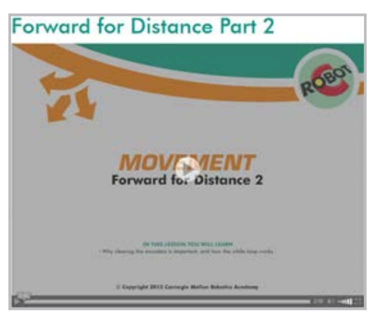
Shaft Encoders Lesson Set

Shaft encoders allow you to accurately control the distance your robot travels. Since the release of this curriculum VEX Robotics released smart motors where the encoder is built into the motor. You can find information about the Integrated Motor Encoders: http://www.robotc.net/wiki/Tutorials/Programming_with_the_new_VEX_Integrated_Encoder_Modules

The Shaft Encoder Video - This video teaches students how shaft encoders work and how they are used to control distance.

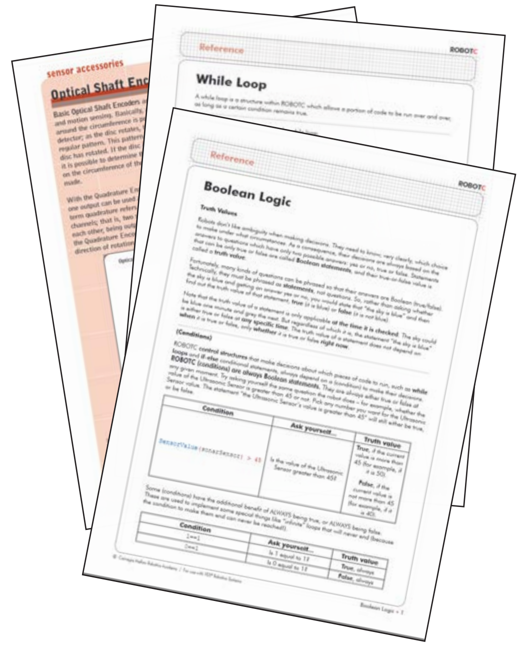
The Motor and Sensors Setup, Forward for Distance 1 Video - This video teaches students how to configure the shaft encoders using the Motors and Setup wizard.

The Encoder, Forward for Distance 2 Video - This video teaches students why clearing the values in the encoder is important and introduces them to how "While Loops" work.



Movement/Shaft Encoders (continued)

Boolean Logic Part 1 & Part 2 Videos - Boolean Logic enables robots to make decisions. The first video teaches students how conditional statements work. The second teaches them about logical operators. There is also a Boolean Logic 3 page PDF that complements the videos.



Boolean Logic PDF - Covers the same topics as the Boolean Logic videos, but in a print format.

Shaft Encoders PDF - Teaches students how shaft encoders work.

The While Loop Handout - Shows students the code that controls a while loop.

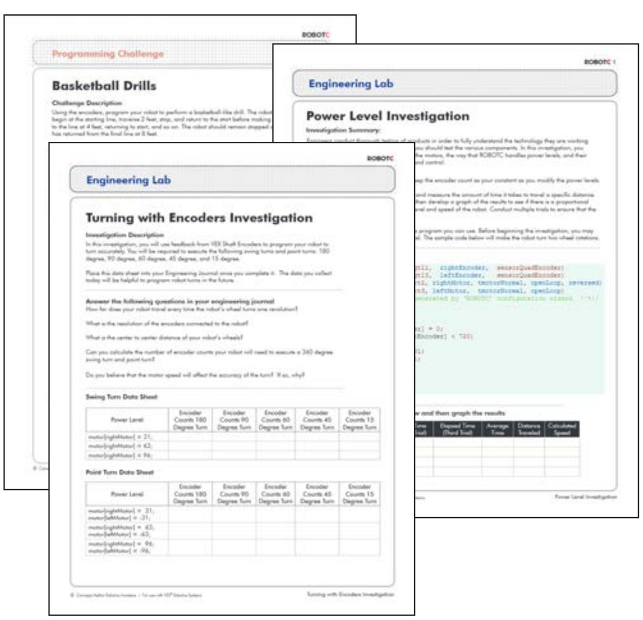
Additional Shaft Encoder Lesson Resources

The Sensor Debug Window Video - ROBOTC's debug window is one of its most powerful tools. This video introduces students to how they can see all of the robot's input and output values.

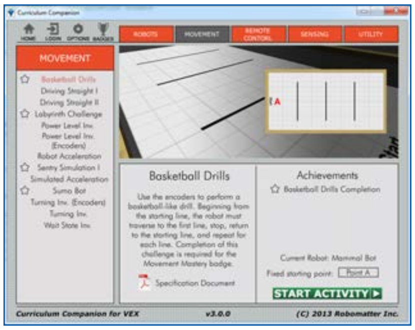
Shaft Encoders, Forward and Turning Video- Teaches students how to use encoders in path planning.

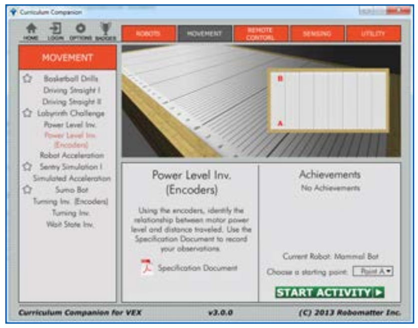
The BasketBall Challenge - A programming challenge designed to give student's practice with the shaft encoders.

The Power Level and Turning with Encoder's Engineering Investigations - students use feedback from the "Debug" window to gather data to complete these two Engineering Investigations.



Movement/Shaft Encoders (continued)





Basketball Drills, Power Levels with Encoders, and Turning with Encoders are available as RVW Challenges.

Movement/Automated Straightening

The Automated Straightening Lesson Set

This lesson set continues to introduce students to programming concepts like if-else statements, variable types, and global variables. Students will get more practice using Boolean Logic as they create their own self-straightening algorithm.

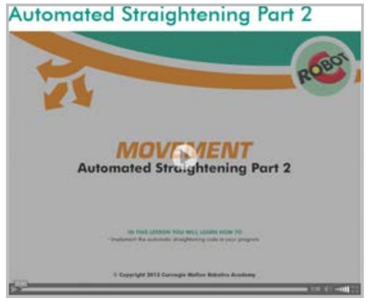
Automated Straightening Part 1 Video - This video teaches students will learn how to use feedback from encoders and conditional statements to develop an algorithm that allows the robot to self-correct its forward movements.

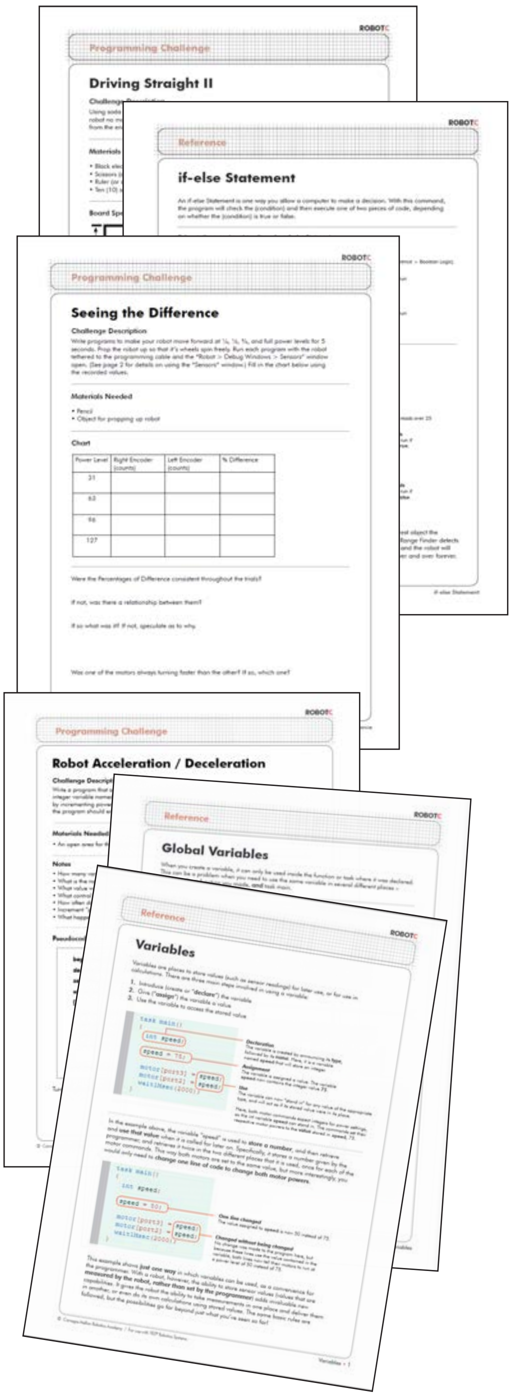
Automated Straightening Part 2 Video - This video teaches students how to implement the automated straightening algorithm on their robot.

if-else Statement Handout - This handout can be used as a study guide that shows students how the if-else Statement can be used with a while loop to help a robot make a decision.

Movement/Automated Straightening resources continued next page.

Movement/The Automated Straightening (continued)





The Driving Straight Programming Challenge - This challenge requires students to write a program that controls their robot to go perfectly straight. The challenge is also available using the RVW simulation software.

Seeing the Difference Challenge - This challenge requires students to use feedback from ROBOTC's real time debugger to track the robot's wheel movement.

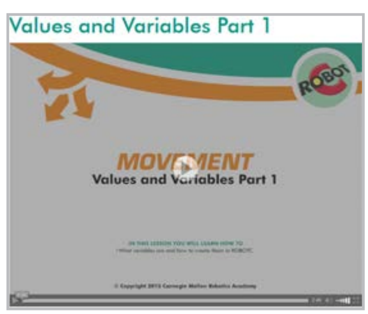
Boolean Logic Part 1 & Part 2 Videos - Boolean Logic enables robots to make decisions. The first video teaches students how conditional statements work. The second teaches them about logical operators. There is also a Boolean Logic 3 page PDF that complements the videos.

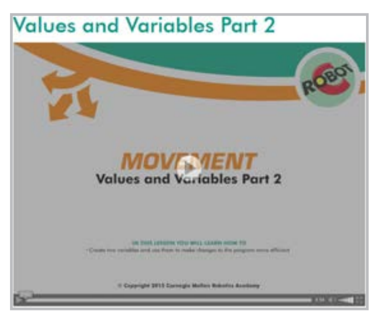
Variables and Values

Variables and Values Video Part 1 - This video introduces students to the power of variables. This lesson teaches students about how to specify variables and variable types..

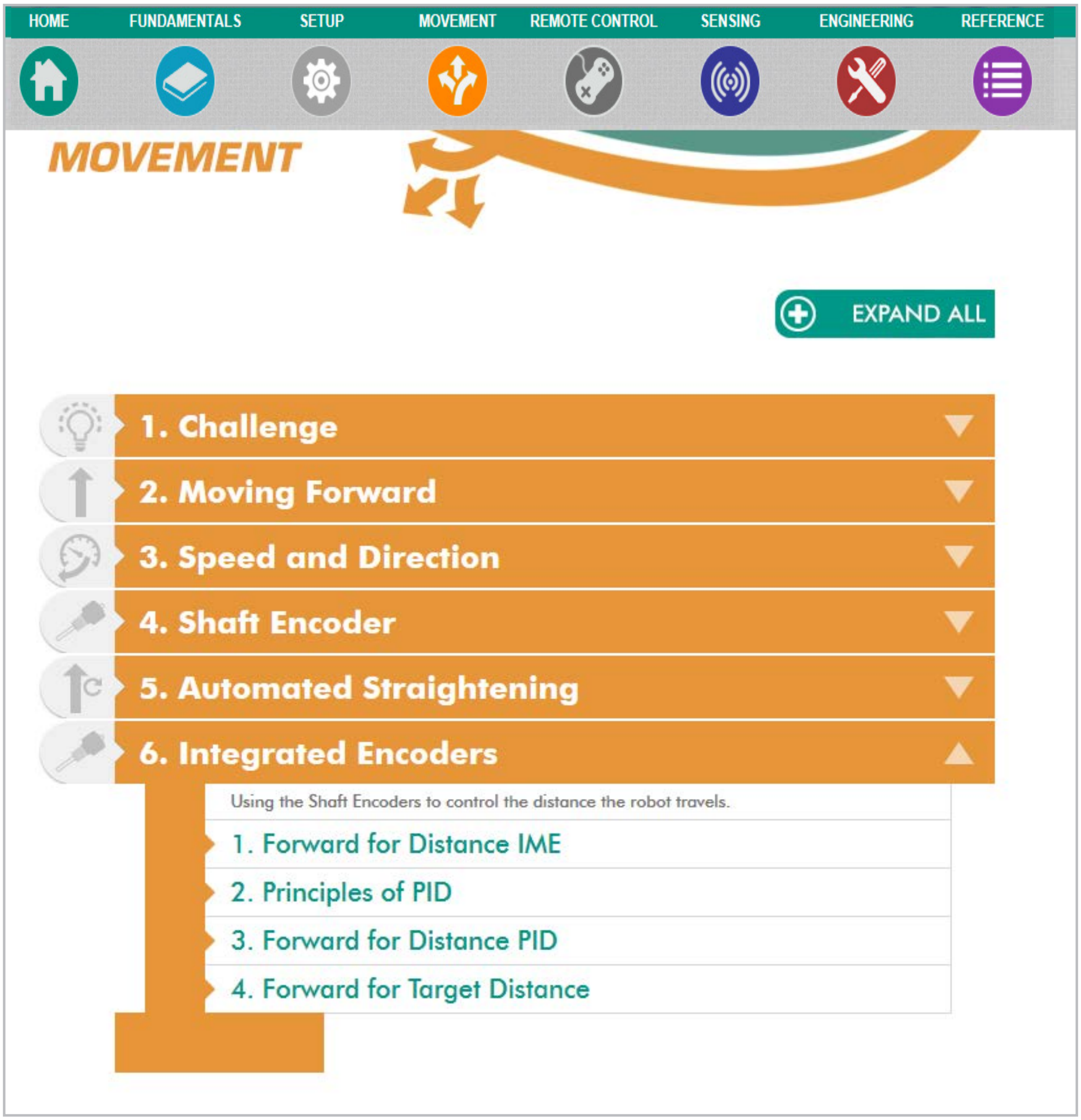
Variables and Values Video Part 2 - This teaches students about variable names and types, how to initialize a variable, and how to use variables in their program.

Variable and Global Variables Reference Guide - This PDF can be used as a study guide and covers everything taught in the two variable videos.





Integrated Encoders



The Integrated Encoders Lesson Set

This lesson set contains four videos designed to introduce students to the Integrated Motor Encoders and PID.

Forward for Distance IME Video - This video teaches students will learn how to use feedback from encoders and conditional statements to develop an algorithm that allows the robot to self-correct its forward movements.

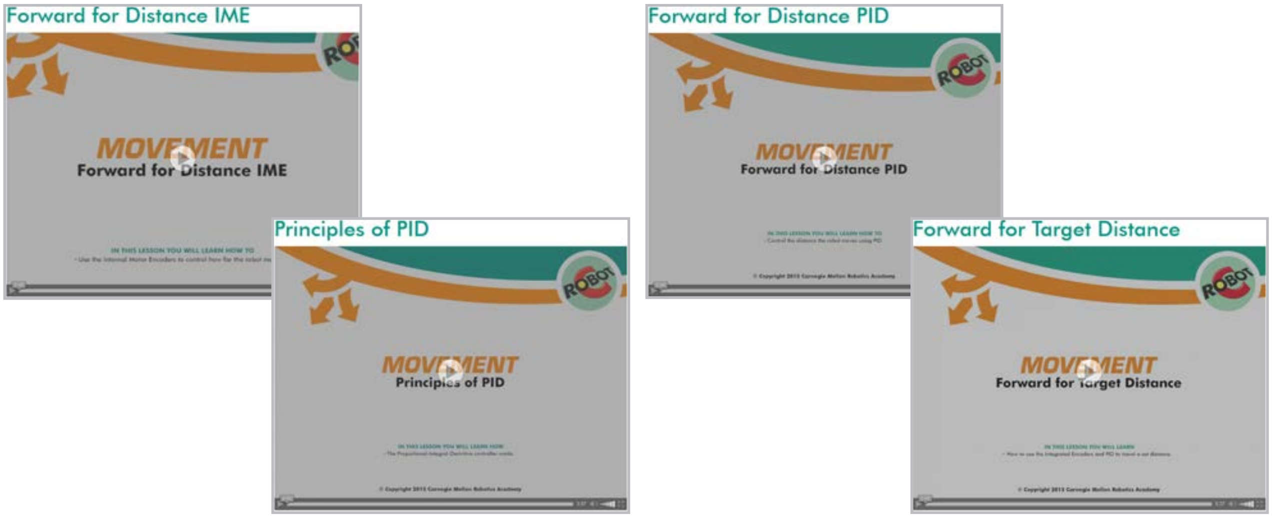
Principles of PID Video - This video teaches students how PID enabled robots can automate the ability to track how far each motor spins and make automated adjustments as the robot moves.

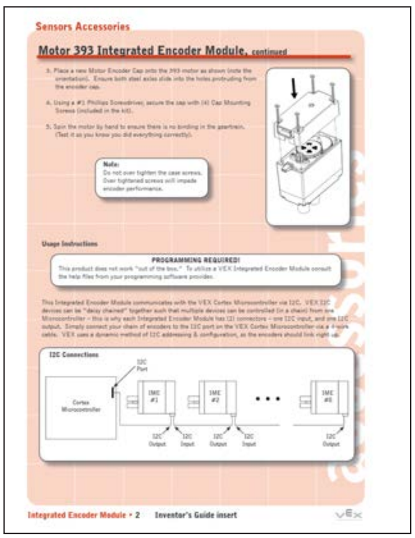
Forward for Distance PID Video - This video shows how to setup your motors programmatically to use PID and how to observe the values of the motors using the debug window.

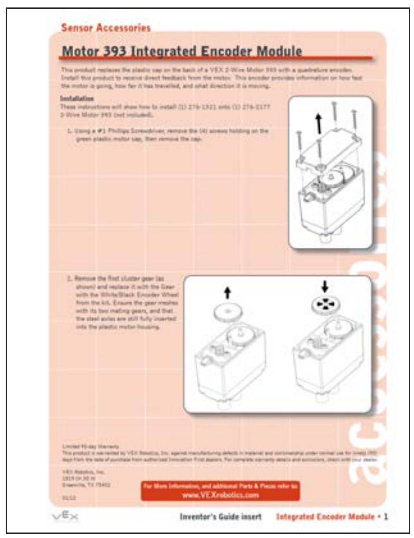
Forward for Target Distance PID Video - This video teaches how to use the moveMotorTarget and getMotorEncoder commands eliminating the robot drifting past the target location.

Integrated Encoder resources continued next page.

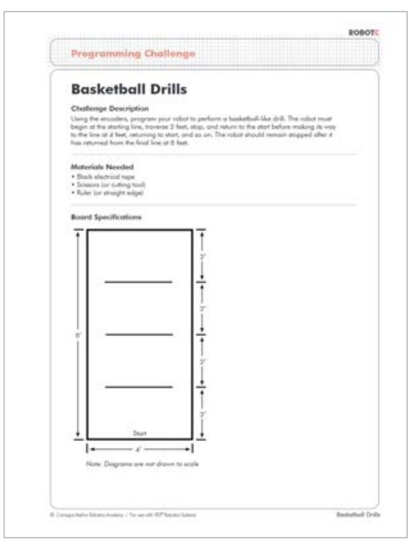
Integrated Encoders (continued)

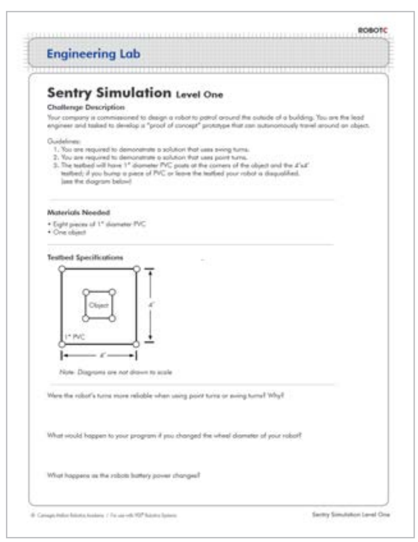






The VEX Cortex Integrated Motor Encoders need to be installed. This lesson set includes a step-by-step set of instructions that you will need to install your IMEs.

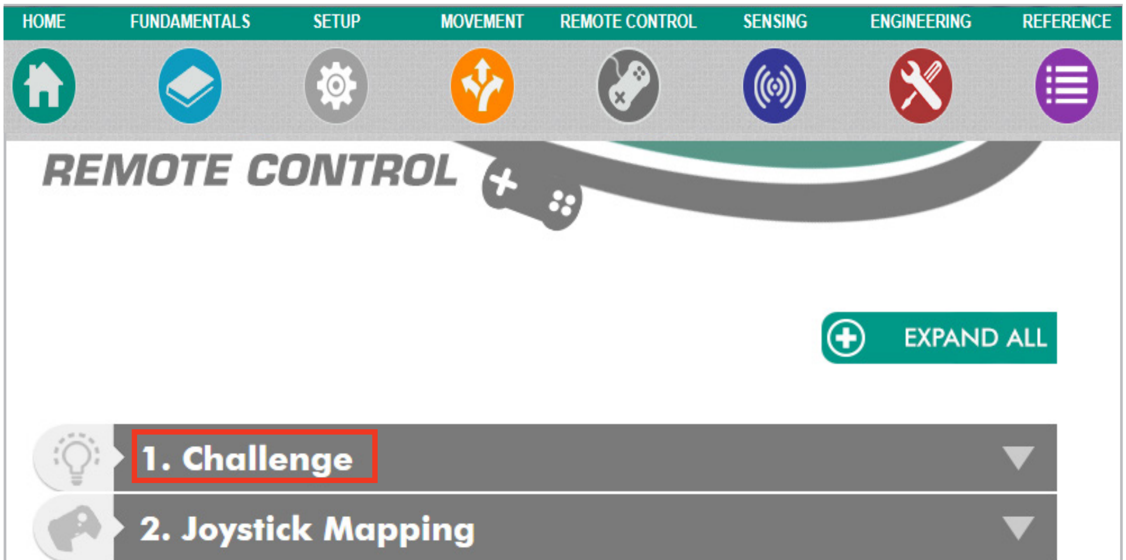




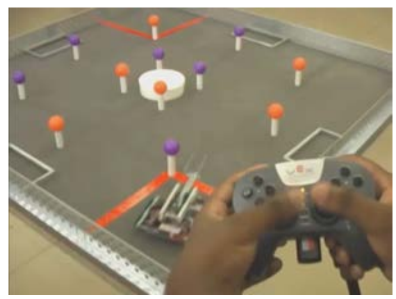
Note: Any time a student needs to run code in the virtual world to follow along with a video, they can use the Utility tables (huge table, metric or imperial distance table, turn table).

Now that you know how to use IMEs these lessons are designed to provide practice.

Remote Control/Challenge Description



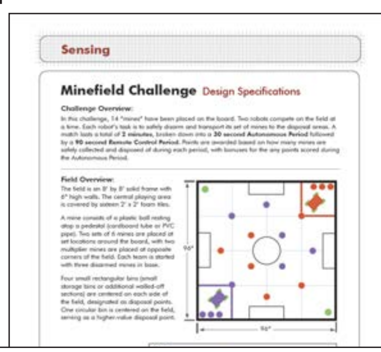
Note: Pictured at the right is the culminating challenge for the Remote Control Unit. Begin the unit by showing students the Minefield RC challenge. Students are not expected to know how to solve the challenge until they complete the Joystick Mapping, Timers, and Buttons Lessons.

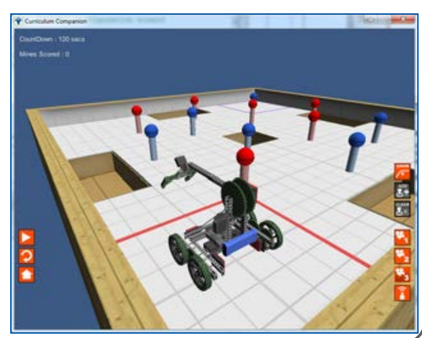


Using Operator Input to Control Your Robot/The Minefield Challenge

A powerful feature of the VEX Cortex is its ability to be driven using the VEXnet Remote Control. This ability proves to be indispensable in the competition environment. In this unit you will learn how to optimize and program your VEX remote control. You will learn to program buttons on the Joystick controller to enable robot behaviors.

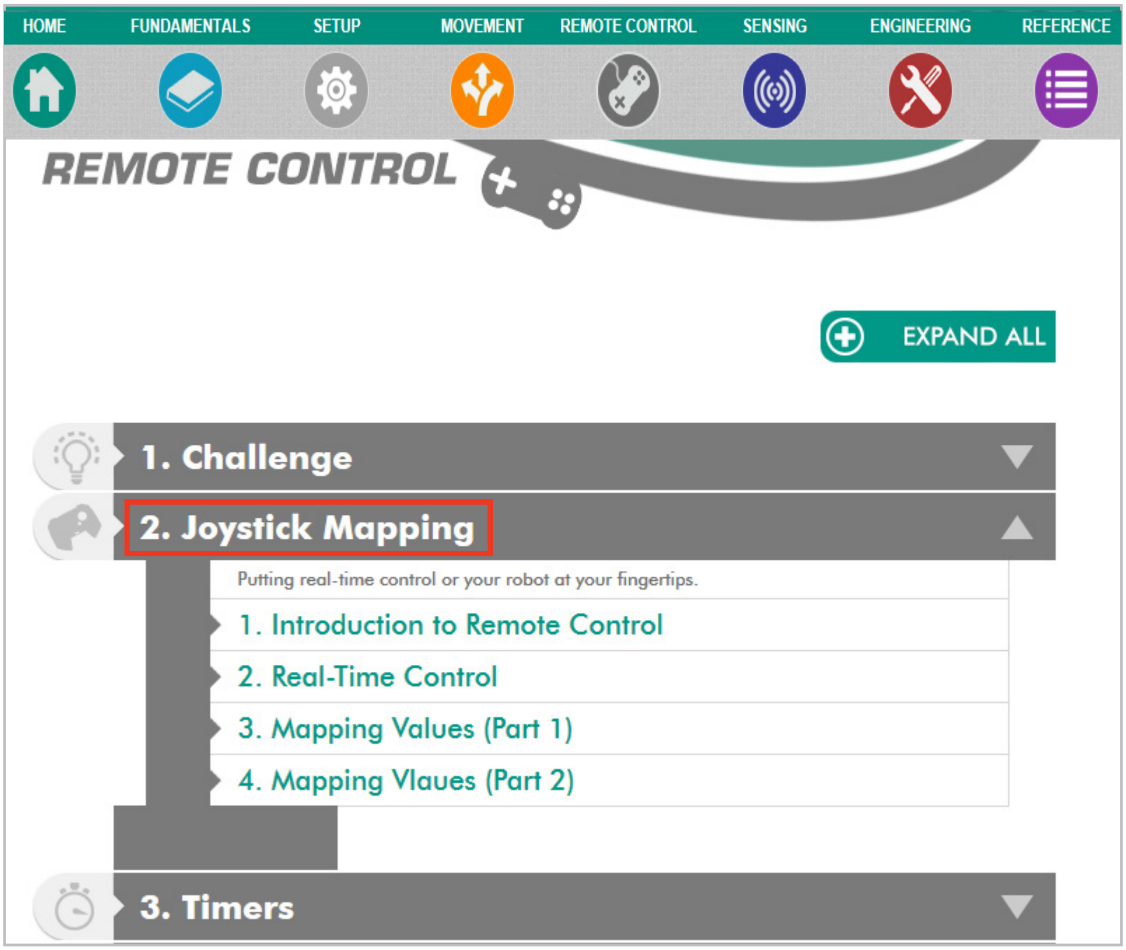
The Minefield Challenge Resources - Remote Control Video Solution, Minefield Challenge PDF that describes the rules, and a Robot Virtual World simulation environment that allows students to develop and test code.





Note: the VEXnet Joystick does not communicate its values to the PC, only to a VEX Cortex, so it cannot be used to control the Virtual Robot. A USB Logitech Joystick is recommended for use with the virtual worlds. There are resources at the ROBOTC Wiki for using any USB joystick and also directly in ROBOTC's Help Documentation.

Remote Control/Joystick Mapping



Introduction to Remote Control Video - Students will learn how the VEXnet Remote Control works and be able to describe the range of values that the joystick provides. Additionally, they will learn which ROBOTC commands allow them to access the Remote Control.

Real Time Control Video - Students will learn how loops work and the difference between and infinite loop and a loop controlled by a conditional statement, they will also learn new ROBOTC reserved words that allow them to control the different remote controller channels.

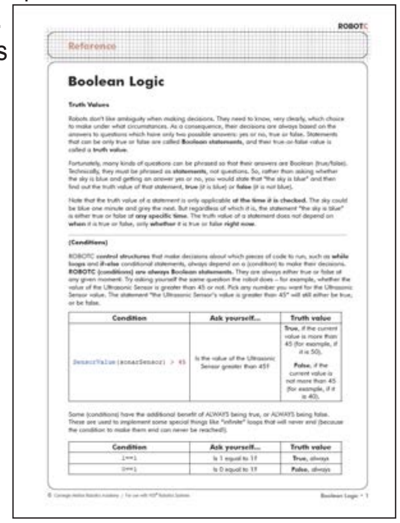
VEXnet Joystick Calibration Guide - Initially, calibrating the Joystick may appear complicated to students. This is a four page guide with lots of pictures that takes them step-by-step through Joystick calibration.

Joystick Mapping Resources are continued on the next page.

Remote Control/Joystick Mapping (continued)

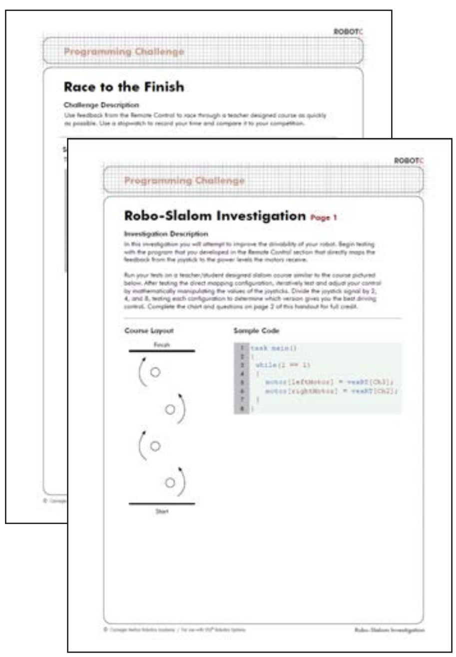
While Loops PDF - A simple example that shows how While Loops work.

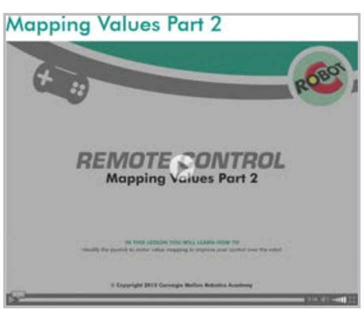
Boolean Logic Part 1 & Part 2 Videos - The Boolean Logic videos are included in multiple lessons and are intended for review. These videos teach foundational principles that all programmers need to know. The first video teaches students how conditional statements work. The second teaches them about logical operators. There is also a Boolean Logic 3 page PDF that complements the videos.

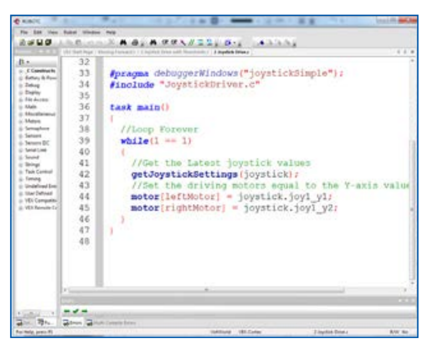


Mapping Values Videos Part 1 & 2 - These videos teach students how to map motor ports to the VEX joystick. By the end of the videos students will be able to change the motor speeds assigned by the remote control and will be able to optimize the joystick values to suit their needs.

Race to the Finish and Robo-Slalom Challenges - These are two relatively simple challenges that students will have fun with. They are designed to give the student practice optimizing the value and configuration of the VEX joystick.

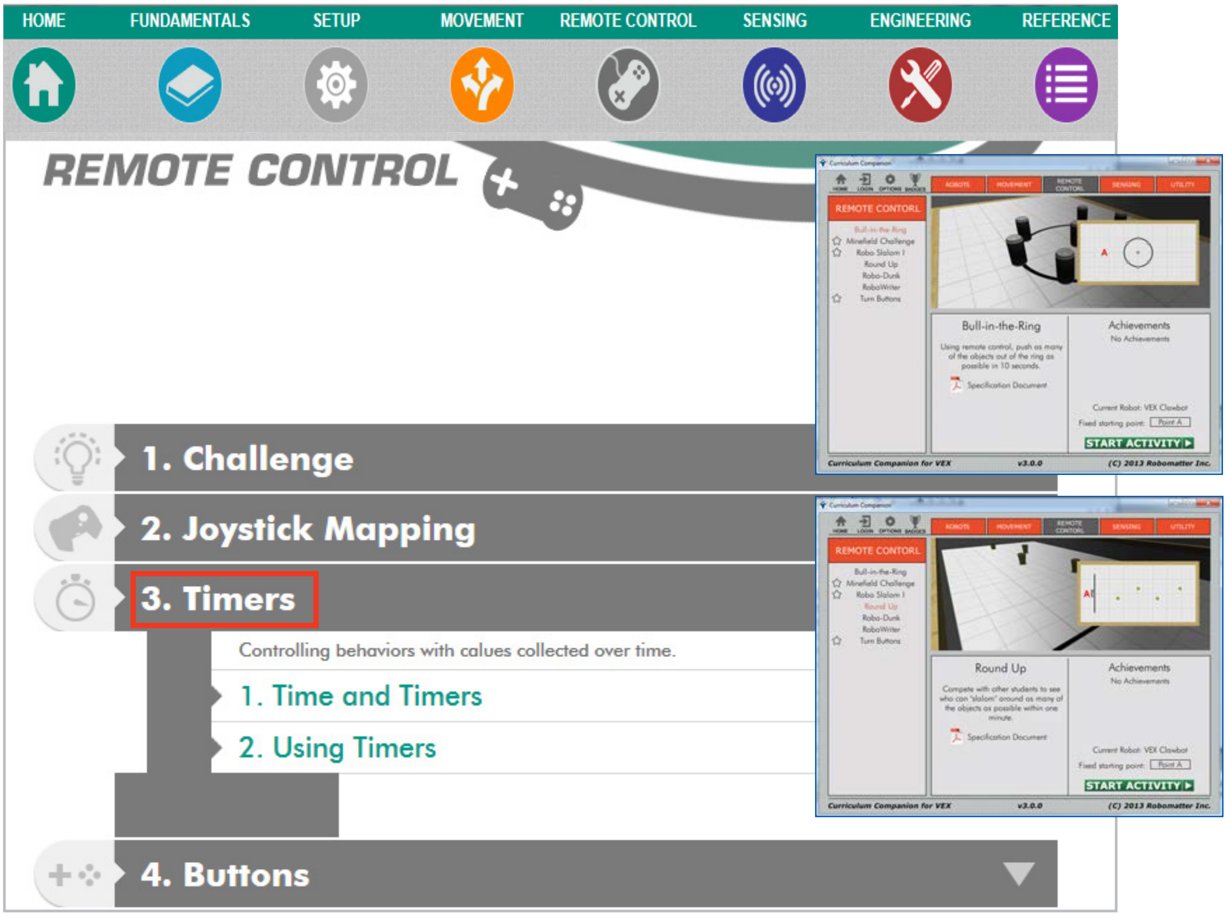




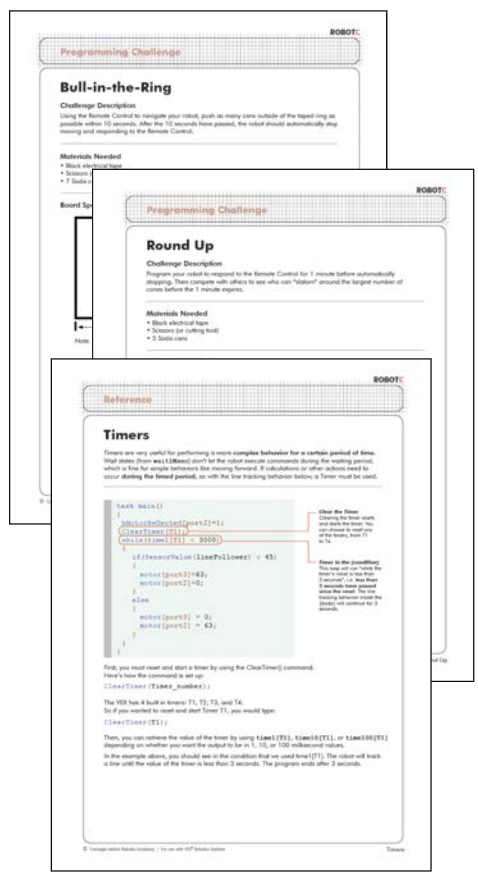


ROBOTC Sample Programs - ROBOTC contains three sample programs in the Remote Control Folder, 2 Joystick Drive with Thresholds, 2 Joystick Drive, and Top Hat Drive. Students are encouraged to begin by opening the sample programs and modifying them.

Remote Control/Timers



Time and Timers Video - In this video students will learn the difference between using the Wait1Msec command and using Timers to control behaviors, they will also learn how many Timers are available, how to reset Timers, and how to access the value of a timer programmatically.

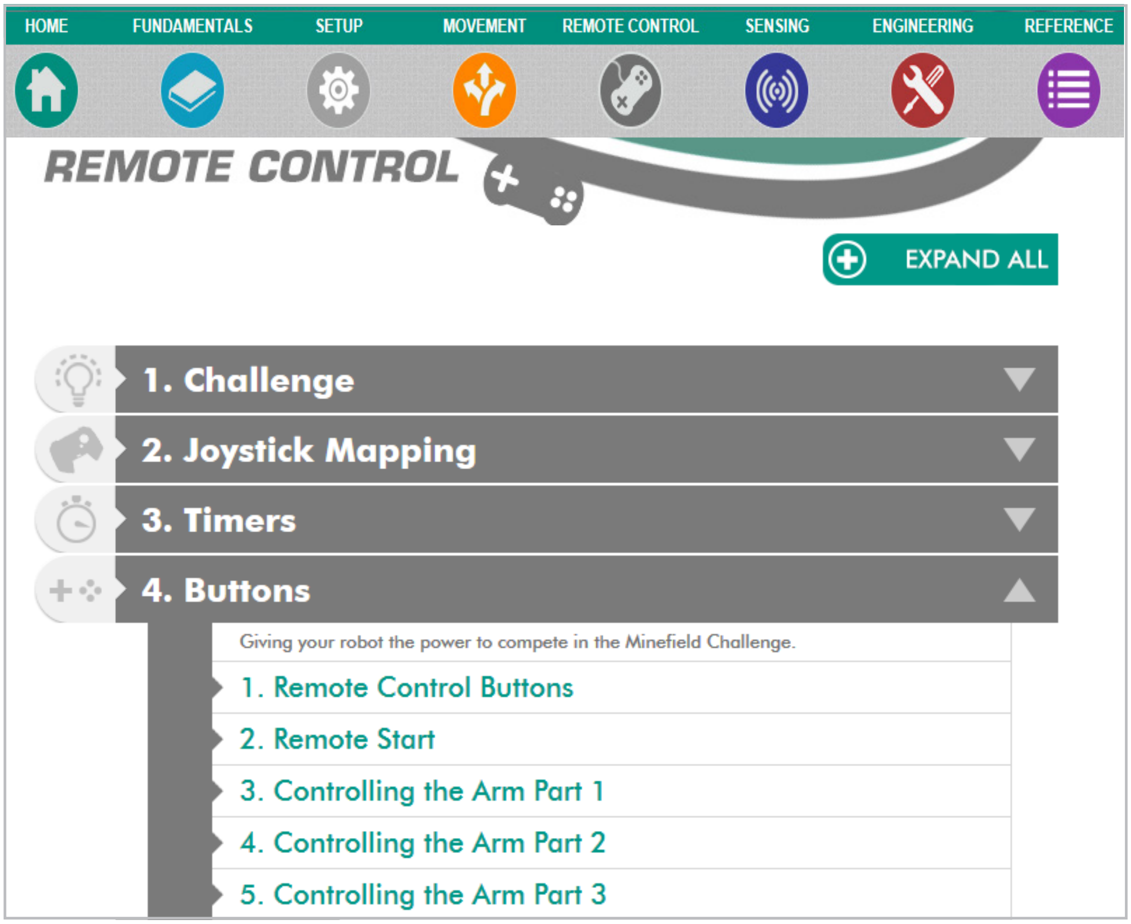


The Using Timers Video - This video teaches students how to use the reserved words built into ROBOTC to control timers.

The Timers PDF Reference Guide - is a one page handout that can be used as a study guide for using Timers.

The Round-Up and Bull-In-The-Ring Challenges - Designed to give students the opportunity to apply Timers. Both challenges are available as Robot Virtual World Simulations.

Remote Control/Buttons



Buttons provide a very powerful input enabling students to control their robot's behaviors. This unit teaches students how to develop code that uses feedback from the remote control's buttons to control their robot's movements.

The Remote Control Buttons Video - This video teaches students the numbering systems for buttons on the VEX remote control. It also shows them how to access the value of a button, and the values that buttons send to the controller.

The Remote Start Video - This video teaches students how to develop competition ready code and how an idle loop can be used to control the start of a program.

The Controlling the Arm Video Part 1 - This video introduces students to programming an arm for the Minefield Challenge, how to use the Motors and Sensors Setup Wizard to rename motors, and how they can use an if statements to control the up and down motion for the arm.

Controlling the Arm Video Part 2 - This video teaches students how the else branch can be used to complete their arm control algorithm.

Controlling the Arm Video Part 3 - The final video shows students how to integrate a while loop and several if-else statements to control the robot's arm.

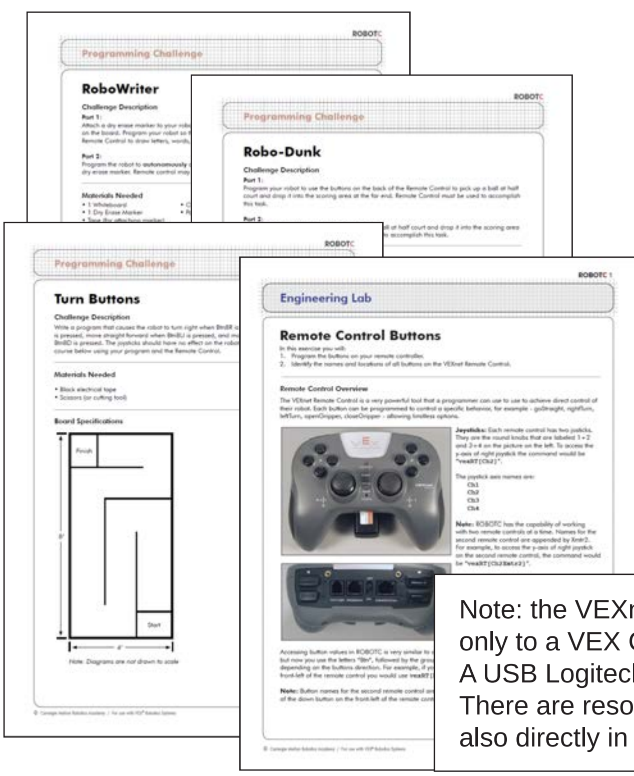
Additional Remote Control labs - The Remote Control Engineering Lab provides students with code and a set of activities designed to take them step-by-step through a remote control lab. Robo-Dunk, RoboWriter, and Turn Buttons are additional labs designed to provide a set of hands on activities to check a student's understanding of programming buttons.

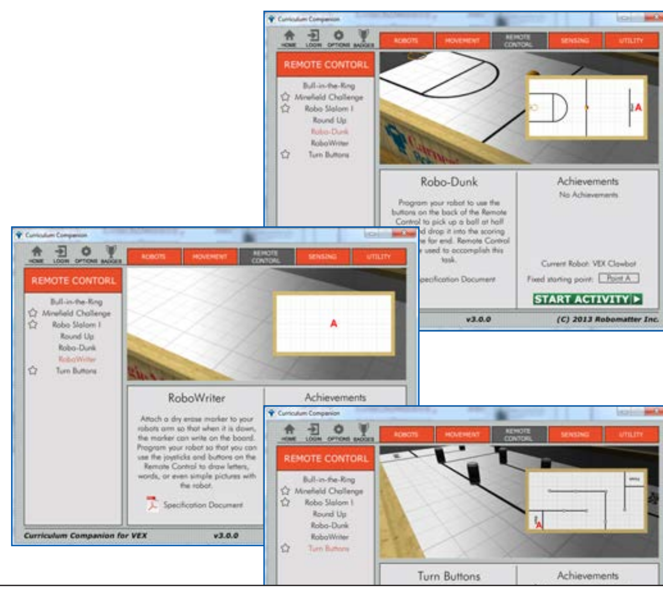
Remote Control/Buttons (continued)

Introduction to Remote Control Buttons

Controlling the Arm - Broken into 3 videos due to the complexity and length of the topic.

Button Labs - Available in PDF format and as a Robot Virtual World programming challenge.





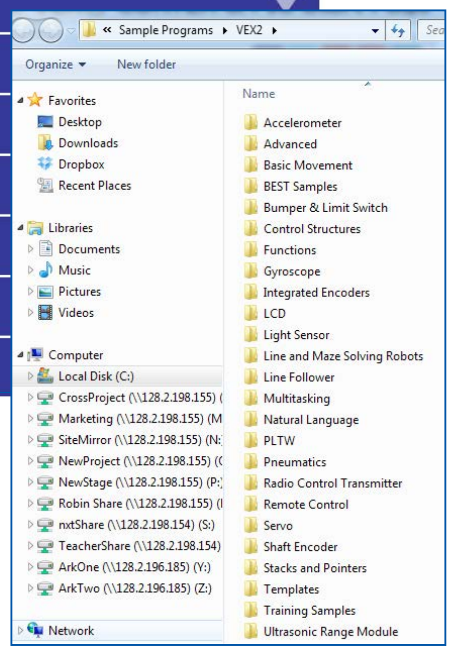
Note: the VEXnet Joystick does not communicate its values to the PC, only to a VEX Cortex, so it cannot be used to control the Virtual Robot. A USB Logitech Joystick is recommended for use with the virtual worlds. There are resources at the ROBOTC Wiki for using any USB joystick and also directly in ROBOTC's Help Documentation.

Sensing

Robots separate themselves from simpler machines through their ability to detect and respond to their surroundings. In this section, you will learn how to take advantage of the multiple sensors available to the VEX Cortex.

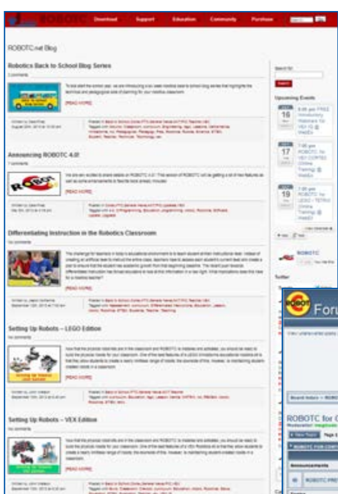
EXPAND ALL

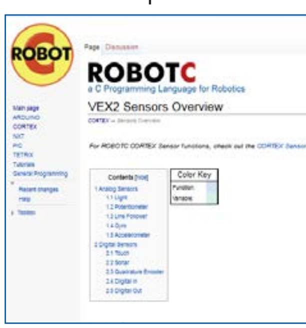
1. Challenge
2. Limiting the Arm
3. Behaviors and Functions
4. Forward until Near
5. Line Tracking
6. Turn for Angle
7. Using the LCD

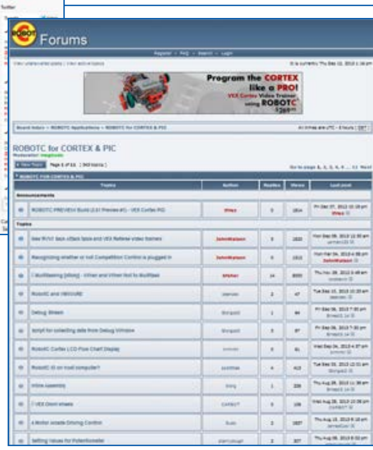


Sensing/Additional Resources

Since the release of the VEX Cortex robot VEX Robotics has added the Gyro, Accelerometer, and Integrated Motor Encoders. There are four places that you can find help with new sensors or ROBOTC features.







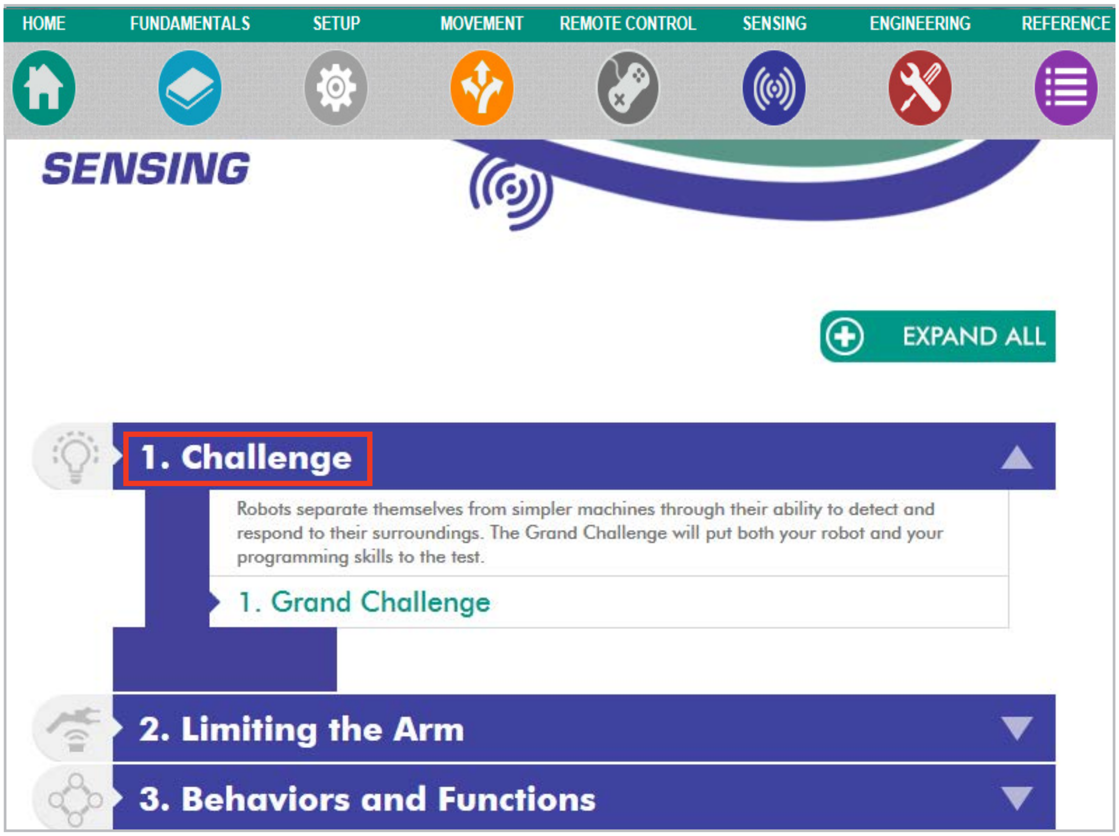
1. ***Sample programs*** - whenever ROBOTC adds new sensor functionality you will find sample programs that demonstrate how the sensor works.

2. ***ROBOTC Help Documentation***- Developers also document new features at the wiki. http://www.robotc.net/wiki/VEX2_Sensors_Overview

3. ***The ROBOTC Blog*** - Developers like to show the community what is new and blog about the new features. http://www.robotc.net/blog/

4. ***The ROBOTC Forum*** - ROBOTC has an active community that helps others with their code. http://www.robotc.net/forums

Sensing/The Grand Challenge



The Grand Challenge

This challenge is a teacher designed course that reinforces behavior-based programming and the engineering process. The challenge course is teacher-designed course and not revealed to students until the day of the competition.
Before the competition, students are provided with a list of conditions and situations to prepare for (width of the robot's path, 30 degree ramps, line following, obstacle detection, etc). On the day of the competition, the student's programming knowledge and preparation are put to the test as they work to traverse the course in a limited amount of time. The robot that makes the most progress without stalling out or deviating from the course wins!

Example Robot Behaviors/Functions - The list below are example behaviors and functions that students might use to solve the Grand Challenge.

- Moving Straight using Encoders
- Accurately Move a Specific Distance
- Move Forward Until the Light Sensor Sees a Dark Line
- Program using Remote Control
- Precise Turning using Encoder Feedback
- Precise Turning using Gyro Sensor Feedback
- Programming Remote Controller Buttons to Precise Turning using Encoder or Gyro Sensor Feedback
- Obstacle Detection using Feedback from Sonar or Touch Sensors (autonomous)
- Track a Line
- Ability to Push Object
- Ability to Detect and Count Lines
- Ability to Display Characters to the LCD Remote Screen

Grand Challenge resources are found on the next page.

Grand Challenge Problem Resources

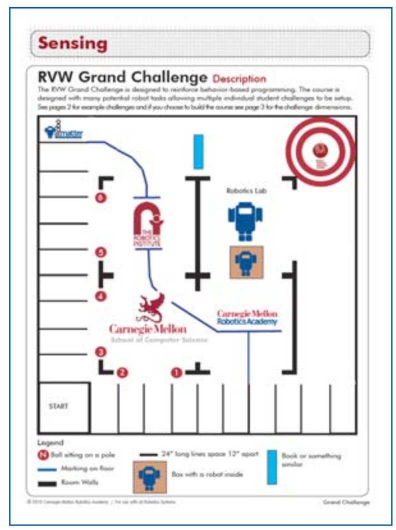
Physical Robot Resources

The Grand Challenge Design Specification - A PDF that describes the challenge.

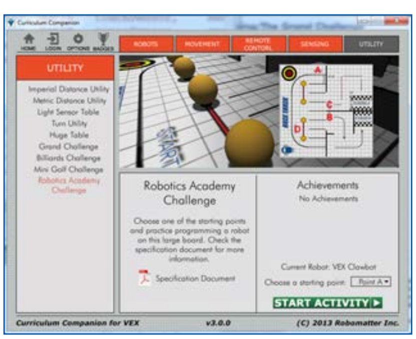
Grand Challenge Example Solution Video - This solution video is an example of a course. The course that you use will be designed by you and your students.

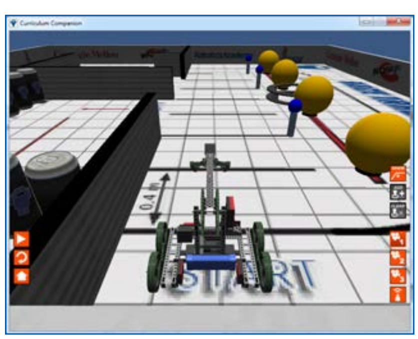
Robot Virtual World Resources

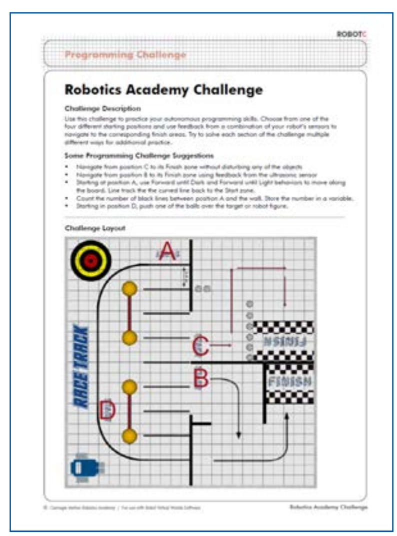
The Robot Virtual World Grand Challenge Design Specification - This world is designed so that it can be solved many ways allowing the teacher to assign student challenges based on ability. Below, at the left is a picture of the home screen for the RVW Grand Challenge, it is found in the Utility Tables section, in the center is a screen shot of the RVW, and at the right is the RVW Grand Challenge programming description PDF.



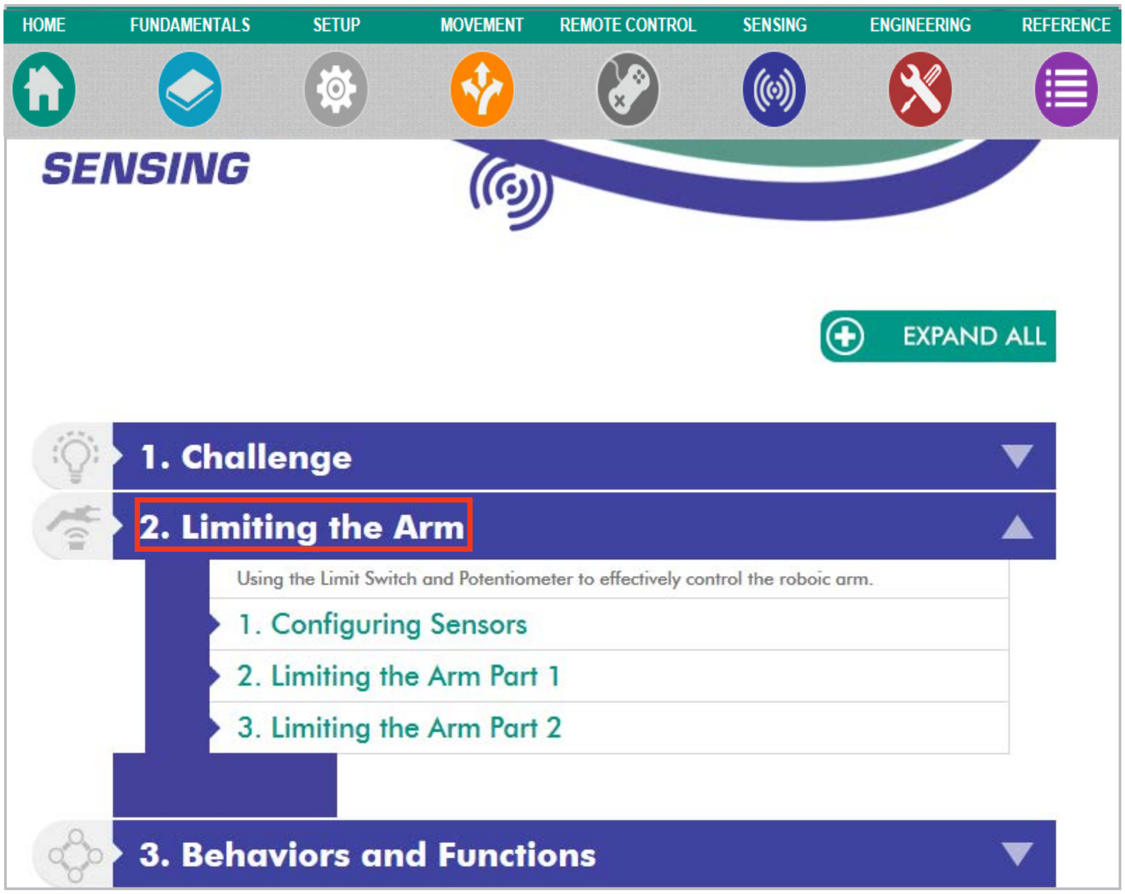
The Robot Virtual World Robotics Academy Grand Challenge Design Specification - This world is also designed so that it can be solved many ways allowing the teacher to assign student challenges based on ability. This world is a little easier than the RVW Grand Challenge above. Below, at the left is a picture of the home screen for the RVW Grand Challenge, it is found in the Utility Tables section, in the center is a screen shot of the RVW, and at the right is the Robotics Academy RVW Grand Challenge programming description PDF.







Sensing/Limiting The Arm



Limiting the Arm Resources

The Limiting the Arm lesson set teaches students how to configure and integrate sensors into the lift-arm mechanism on the robot. The Lesson Set has the following resources:

The Configuring Sensors Video - The video explains how to configure sensors using ROBOTC's Motors and Sensors Setup wizard and reviews the rules for naming sensors.

The Sensors Reference Guide PDF - A 12 page reference guide that describes the Analog and Digital ports found on VEX Controllers and describes how touch sensors work.

Servo Motors Reference Guide PDF - A 2 page reference guide that shows students how to control servo motors.

Limiting the Arm Part 1 Video - This video explains how the touch sensor works as a limit switch to control how far the motor can move. Students will also learn how to apply the AND Logical Operator in their conditional statement enabling them to check two conditions at the same time.

if-else Statement Reference Guide PDF - Provides students with a review of how the if-else Statement works.

Switch Cases Reference Guide PDF - Provides students with working code that shows how a Switch Case works.

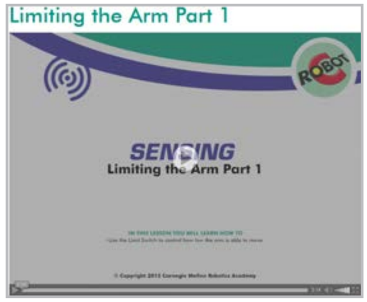
Limiting the Arm Part 2 Video - This video teaches about potentiometers and how they work. Students will learn how to integrate the potentiometer into the mechanical arm mechanism using ROBOTC's debug window.

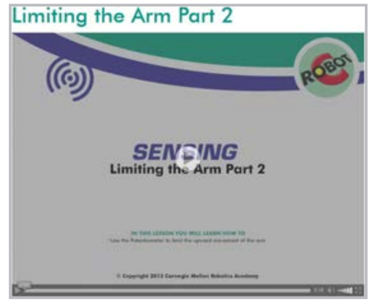
Potentiometer Reference Guides - The reference guides are designed to be used as study guides or teaching tools and explain how the sensor works.

Lifting the Arm resources are continued on the next page.

Sensing/Limiting the Arm Resources (continued)

Instructional Videos





PDF Reference Guides

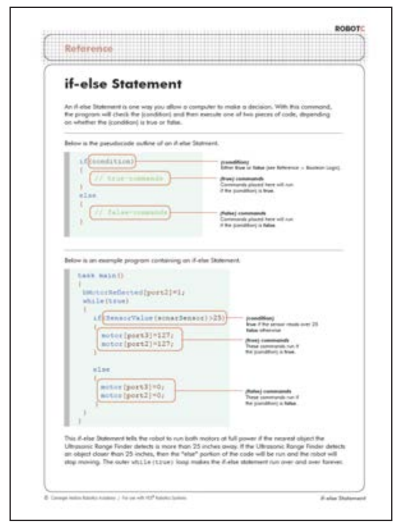

Sensor

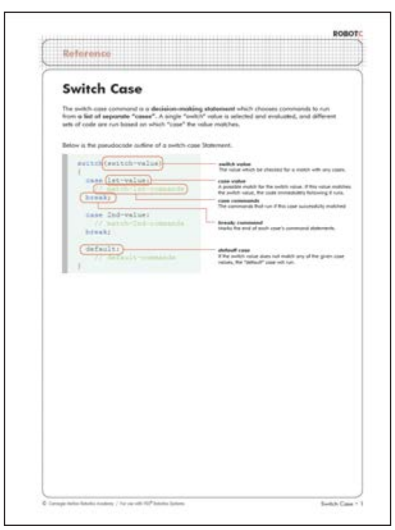
Sensor

Table of Contents:

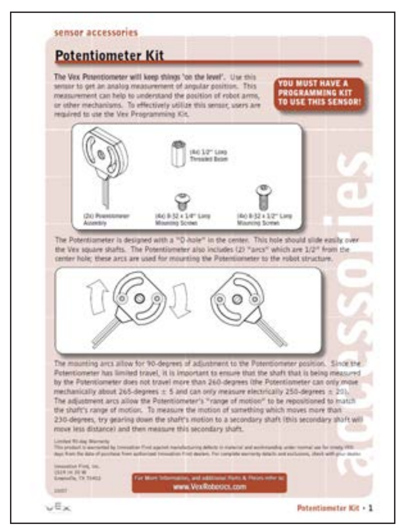
Inventor's Guide 5 • 1


Servo Module Overview


if-else Statement


Switch Case


Potentiometers Overview

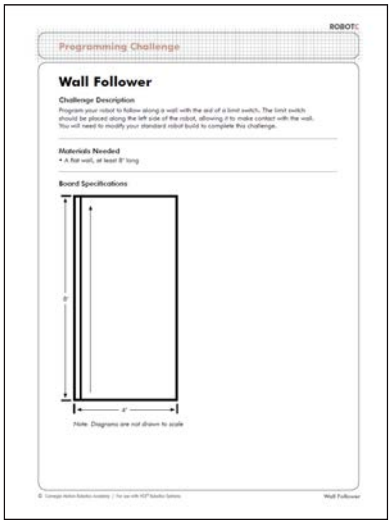

sensor accessories

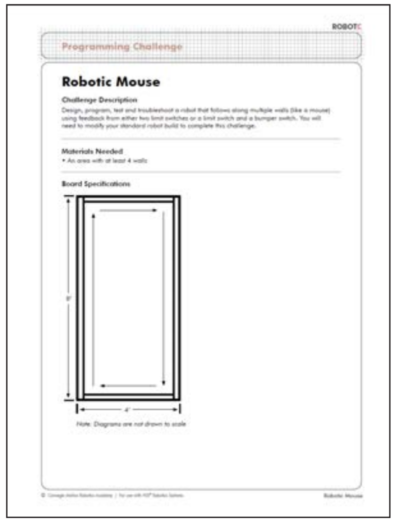
Potentiometer Kit

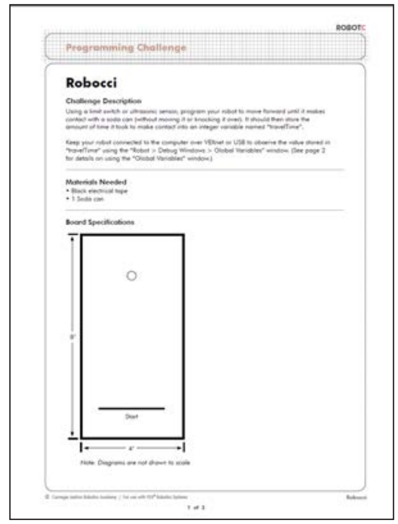
YOU MUST HAVE A PROGRAMMING KIT TO USE THIS SENSOR!

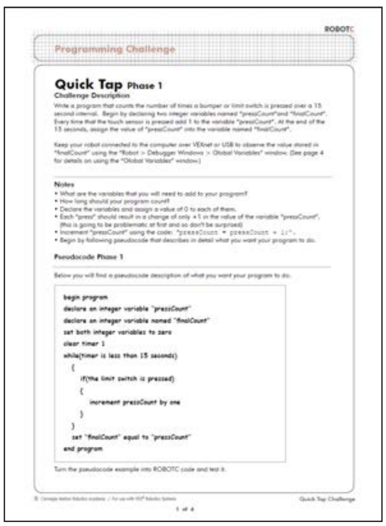
Sensing/Limiting the Arm Resources (continued)

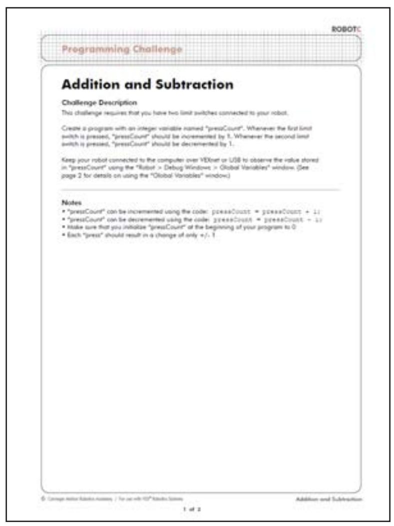
Physical Robot Programming Challenges



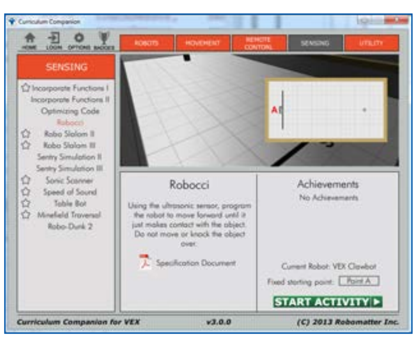


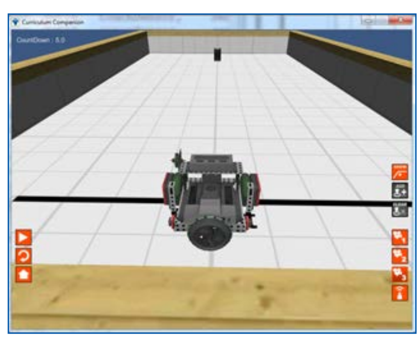




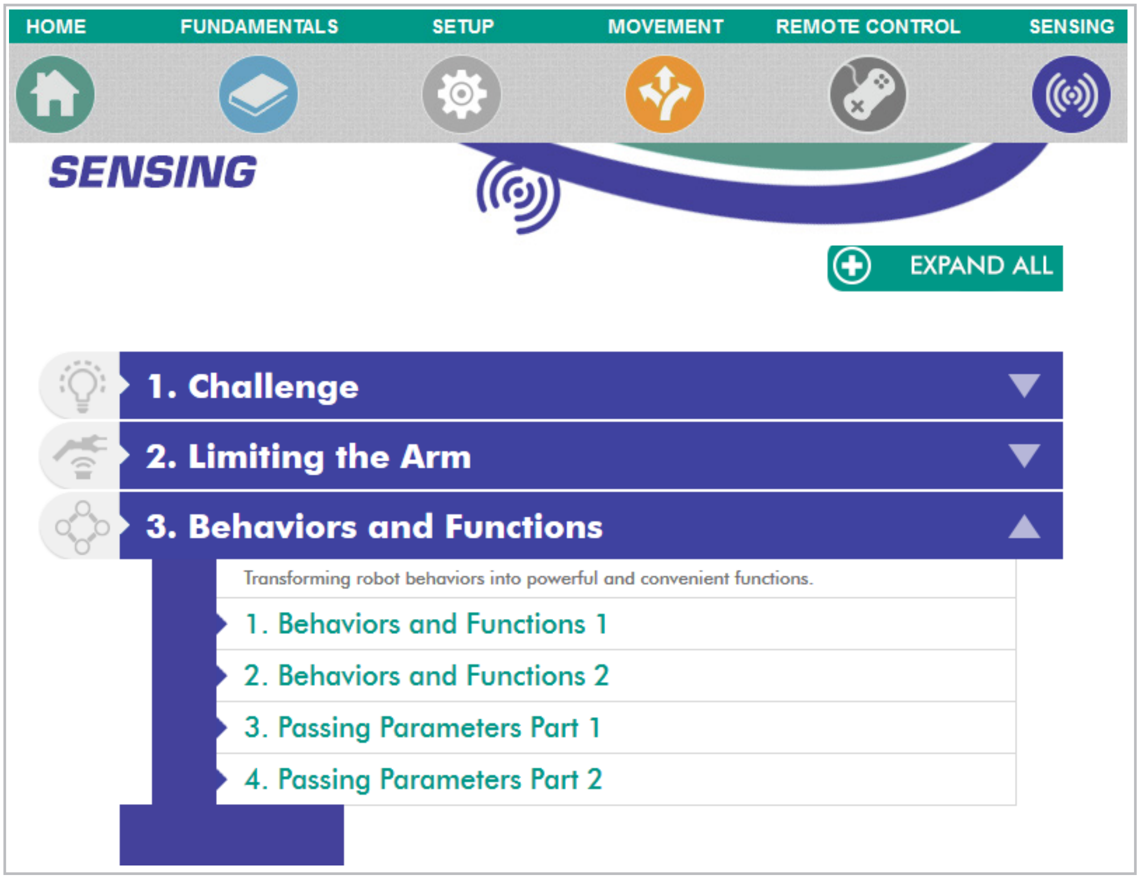


Robot Virtual World Programming Challenges





Sensing/Behaviors and Functions



This lesson set will teach students to build their own behaviors by creating functions. They will also learn to pass parameters that make their functions more portable.

The Behaviors and Functions Part 1 Video - This video introduces students to the power of functions enabling them to make their programs shorter and easier to read.

The Behaviors and Functions Part 2 Video - In this lesson students will learn to transform their code into functions. They will learn how to declare a function and what a parameter is.

The Shaft Encoders Reference Guide - This PDF explains how Shaft Encoders work.

VEX Integrated Encoders Module - Video lessons, challenges, and helper pages.

Optimizing Code and Incorporating Functions Challenges - Two challenges that give students the chance to revisit old code that they wrote and make the code easier to read by incorporating functions into their code.

Passing Parameters Part 1 Video - In this video students learn how to pass parameters allowing them to write more powerful functions, they will also learn the rules to naming and specifying parameters.

Passing Parameters Part 2 Video - In this lesson students learn how to implement the parameter including the type of data the parameter will pass and the name of the parameter.

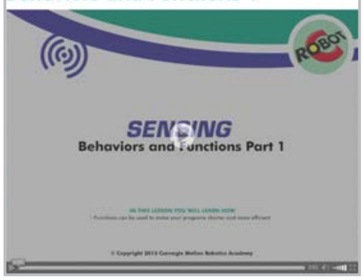
Function Challenges - The Real World Values, RoboDunk2, Seeing the Difference, and Robot Acceleration Programming Challenge enable students to revisit code that they've previously written and optimize it by incorporating functions and parameters.

Additional resources for the Sensing/Behaviors and Functions lesson set can be found on the next page.

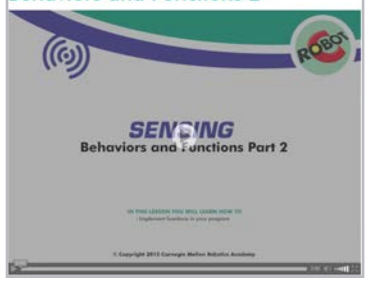
Sensing/Behaviors and Functions Resources (Continued)

Instructional Videos

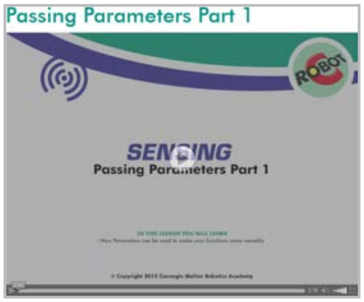
Behaviors and Functions 1



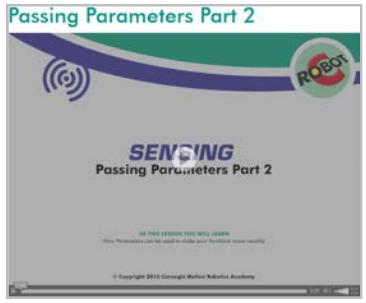
Behaviors and Functions 2



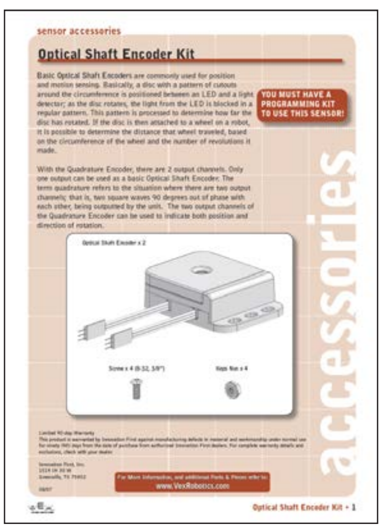
Passing Parameters Part 1

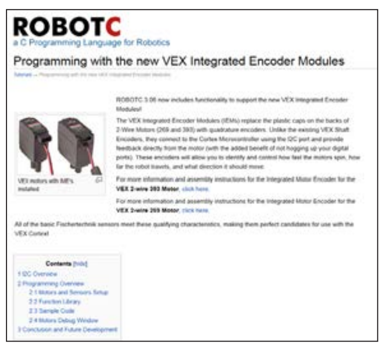


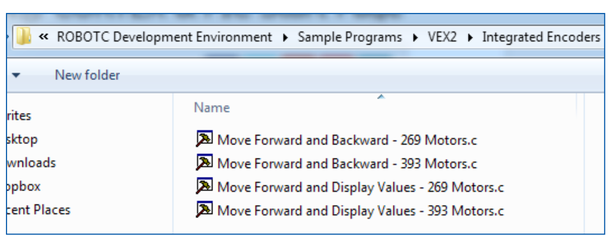
Passing Parameters Part 2



PDF Reference Guides





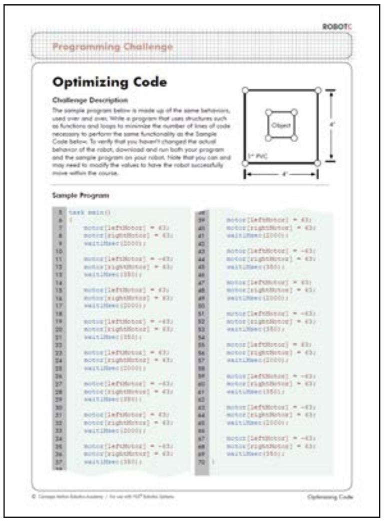


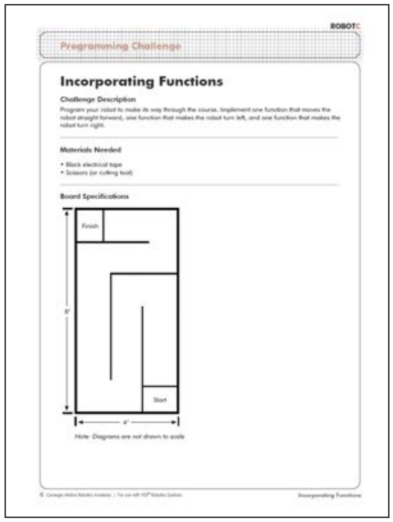
Commented Integrated Encoder sample programs are found in the current ROBOTC Software build.

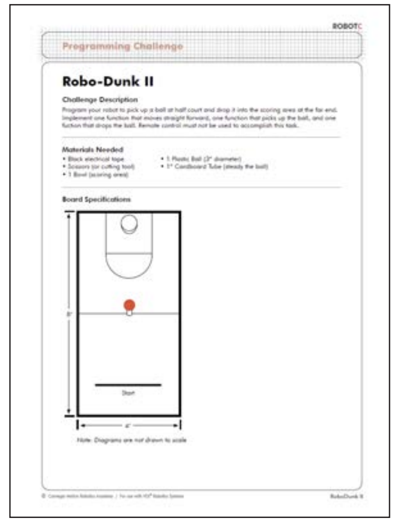
Additional resources are available on the next page.

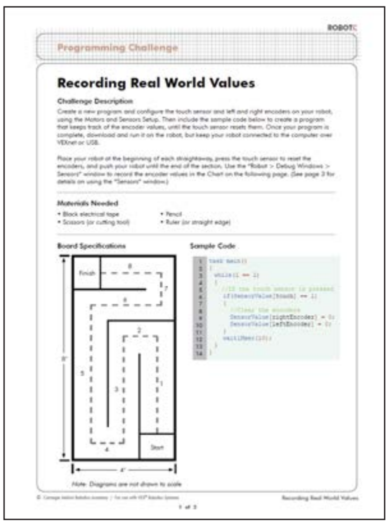
Sensing/Behaviors and Functions Resources (Continued)

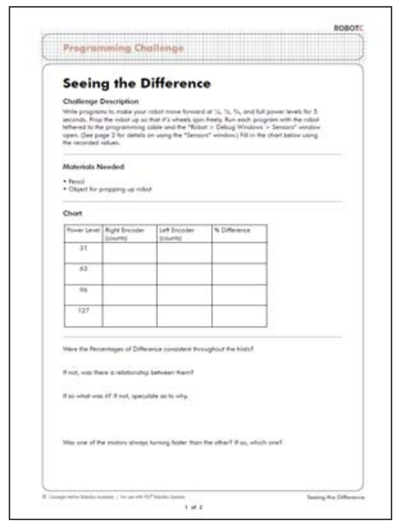
Physical Robot Programming Challenges





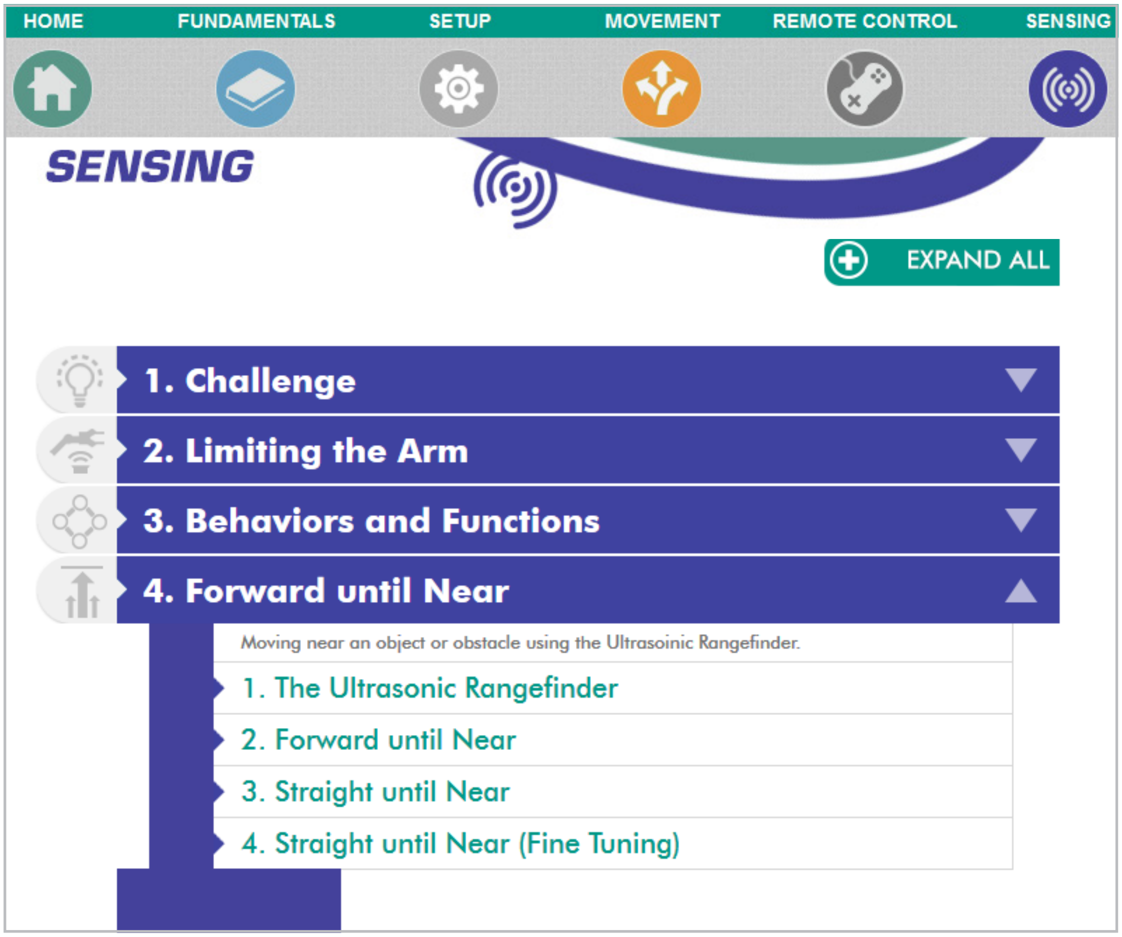






Robot Virtual World Programming Challenges

Sensing/Forward Until Near



This lesson set will teach students how to program the VEX Ultrasonic Rangefinder Sensor

The Ultrasonic Rangefinder Video - This video teaches students about the Ultrasonic Rangefinder.

The Ultrasonic Rangefinder Reference Guide - This guide explains how the Ultrasonic Rangefinder works.

The Forward Until Near Video - Shows students how the code for the Forward Until Near behavior works with the ultrasonic rangefinder including: using the Motors and Sensors wizard, the while loop, and the related code.

The Thresholds Reference Guide - A reference guide that explains what a threshold value is.

The Straight Until Near Video - This lesson combines the Forward Until Near behavior with the Automated Straightening behavior. They will also review how encoders work and why you need to clear them before use.

The Straight Until Near Fine Tuning Video - In this lesson the student will optimize their program using the debugging window so that it moves to the object and stops when it hits the object.

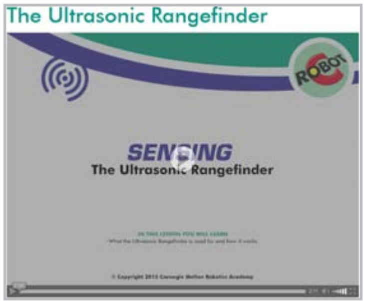
Forward Until Near Programming Challenges - This lesson set contains four increasingly difficult programming challenges that use the Ultrasonic Rangefinder: Sentry Simulation Two, The Speed of Sound, Sentry Simulation Level Three, and the Sonic Scanner programming challenges.

Robot Virtual World Programming Challenges - The RVW simulation software has the Sentry Simulation Level Two, Sentry Simulation Level Three, Speed of Sound, and the Sonic Scanner programming challenges in simulation.

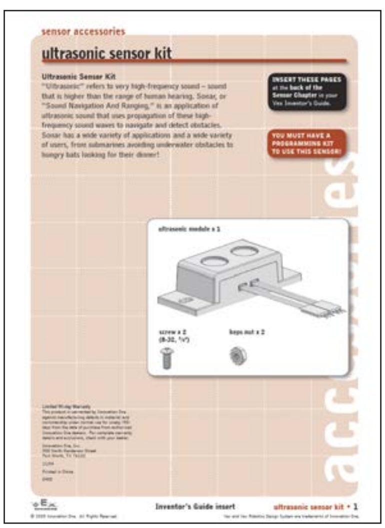
Additional resources for the Sensing/Behaviors and Functions lesson set can be found on the next page.

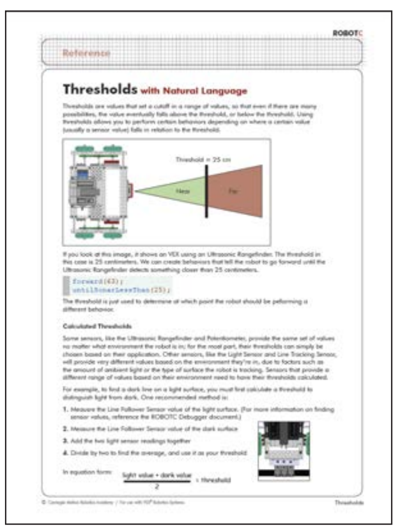
Sensing/Forward Until Near (continued)

Instructional Videos



PDF Reference Guides

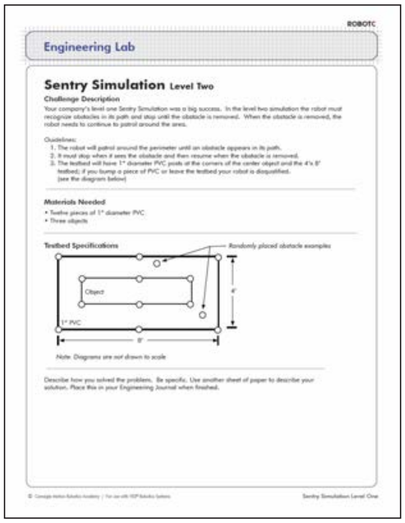


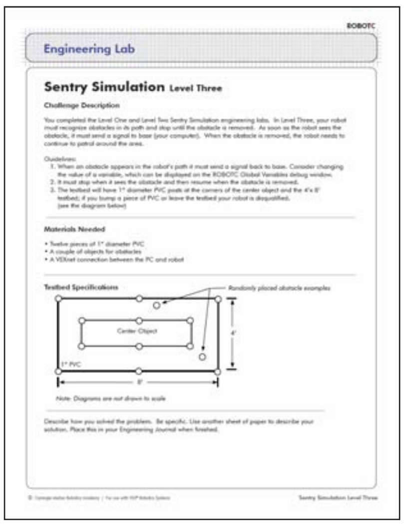


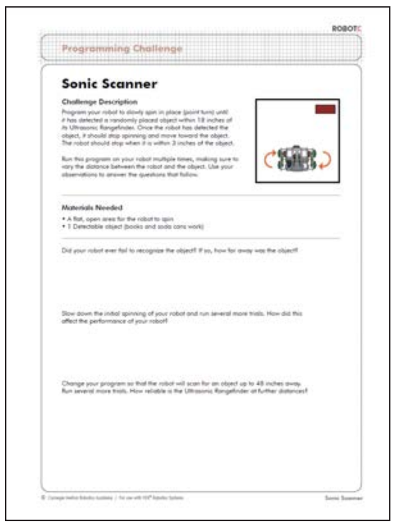
Additional resources are available on the next page.

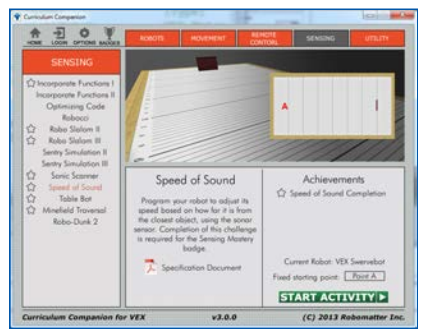
Sensing/Forward Until Near (continued)

Physical Robot Programming Challenges

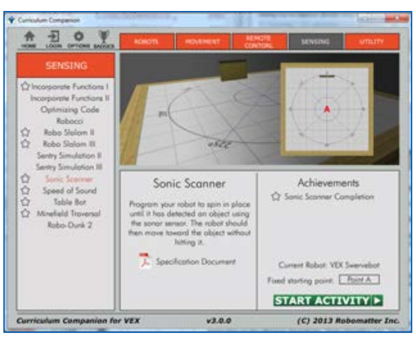


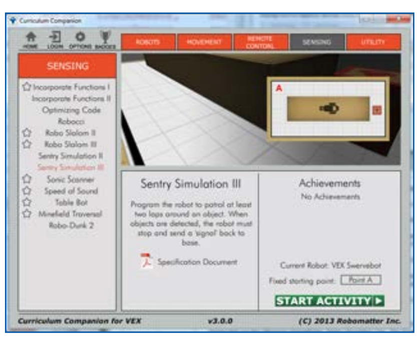




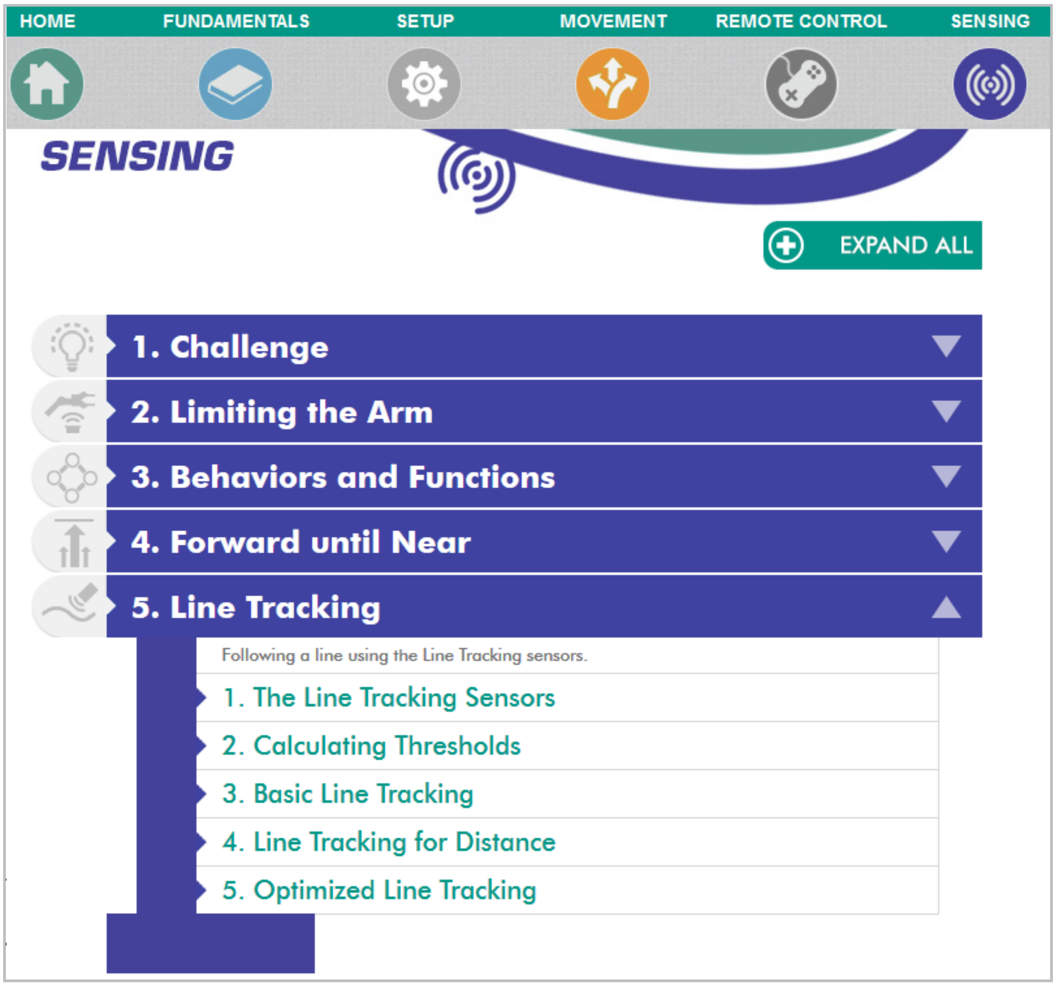


Robot Virtual World Programming Challenges





Sensing/Line Tracking



The SwerveBot Building Instructions - The Swervebot tracks lines well and can be used for this challenge.

The Line Follower Kit Reference Guide- This PDF shows students how the light sensor kit works including the values the light sensor sees and the ports that it connects to.

The Line Tracking Calculating Thresholds Video - This video teaches students how to use ROBOTC's debugger to calculate threshold values for the light sensor.

The Thresholds Reference Guide - This reference guide explains what a threshold value is and how to calculate it.

The Variables Reference Guide - This reference guide explains what variables are and how to use them.

The Basic Line Tracking Video - This video teaches students how to build a line tracking behavior.

Line Tracking for Distance Video - This video teaches students how to use a combination of encoder values and the Line Track kit to track a line for a specific distance.

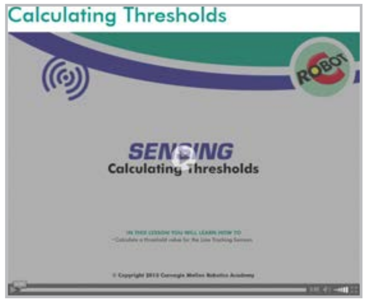
The Optimized Line Tracking Video - Line tracking can be slow, this video shows how you can modify the motor powers to significantly increase the speed your robot tracks a line.

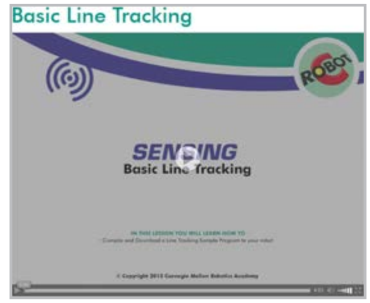
Programming Challenges - This unit has the Forward Until Dark, TableBot, Robo500, Robo Slalom Level 2, Minefield Traversal and Robo Slalom Level three programming challenges to give students practice programming.

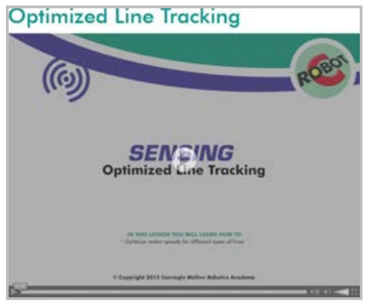
Additional resources for the Line Tracking lesson set can be found on the next page.

Sensing/Line Tracking (continued)

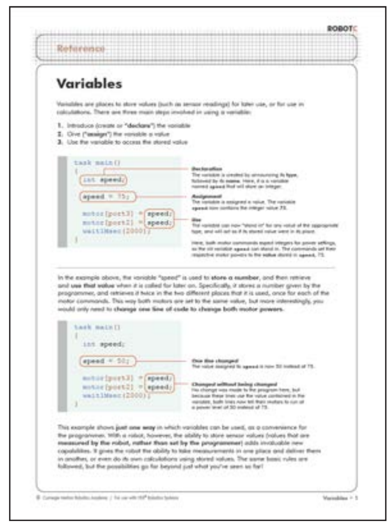
Instructional Videos

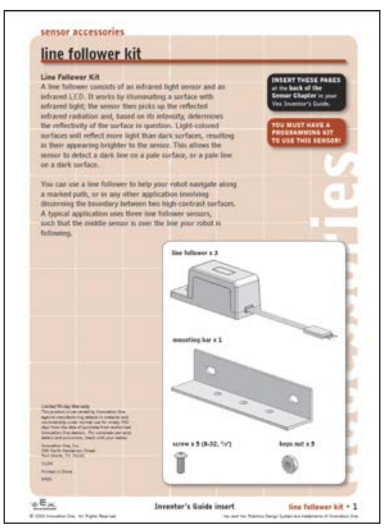


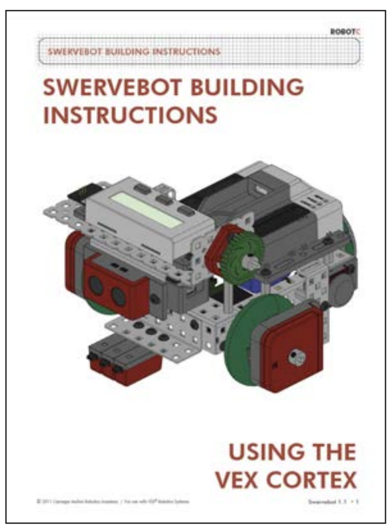


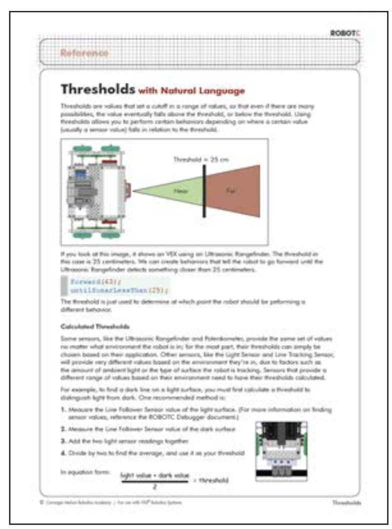


PDF Reference Guides



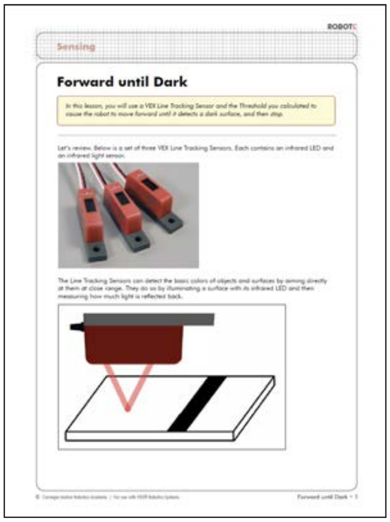


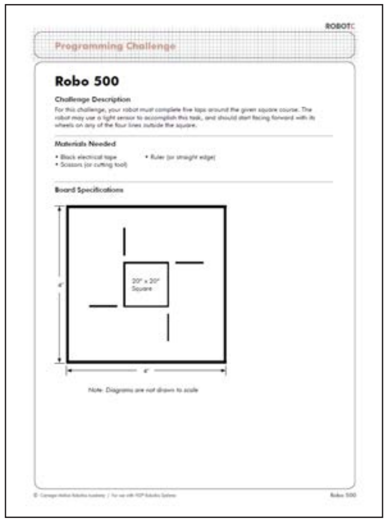


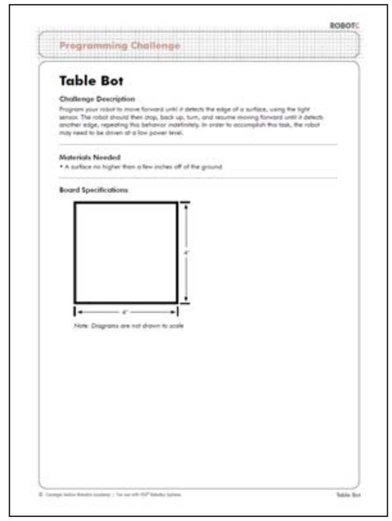


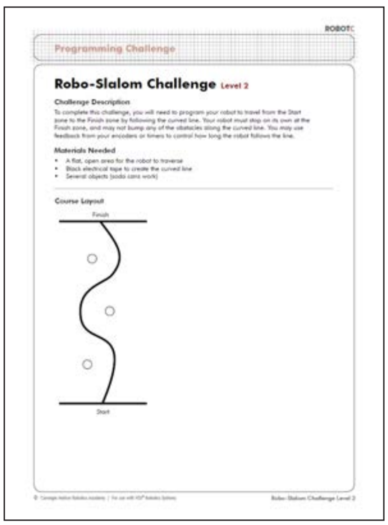
Sensing/Line Tracking (continued)

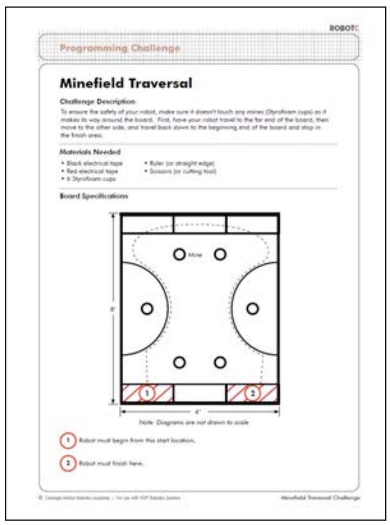
Physical Robot Programming Challenges

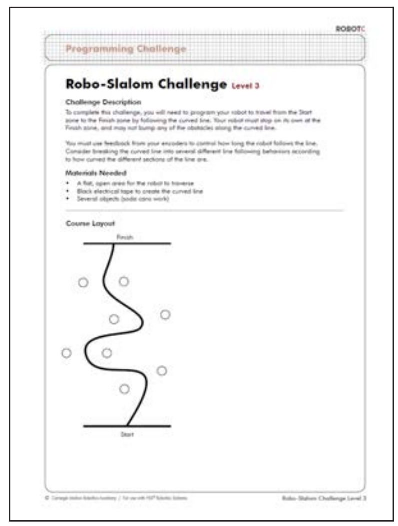






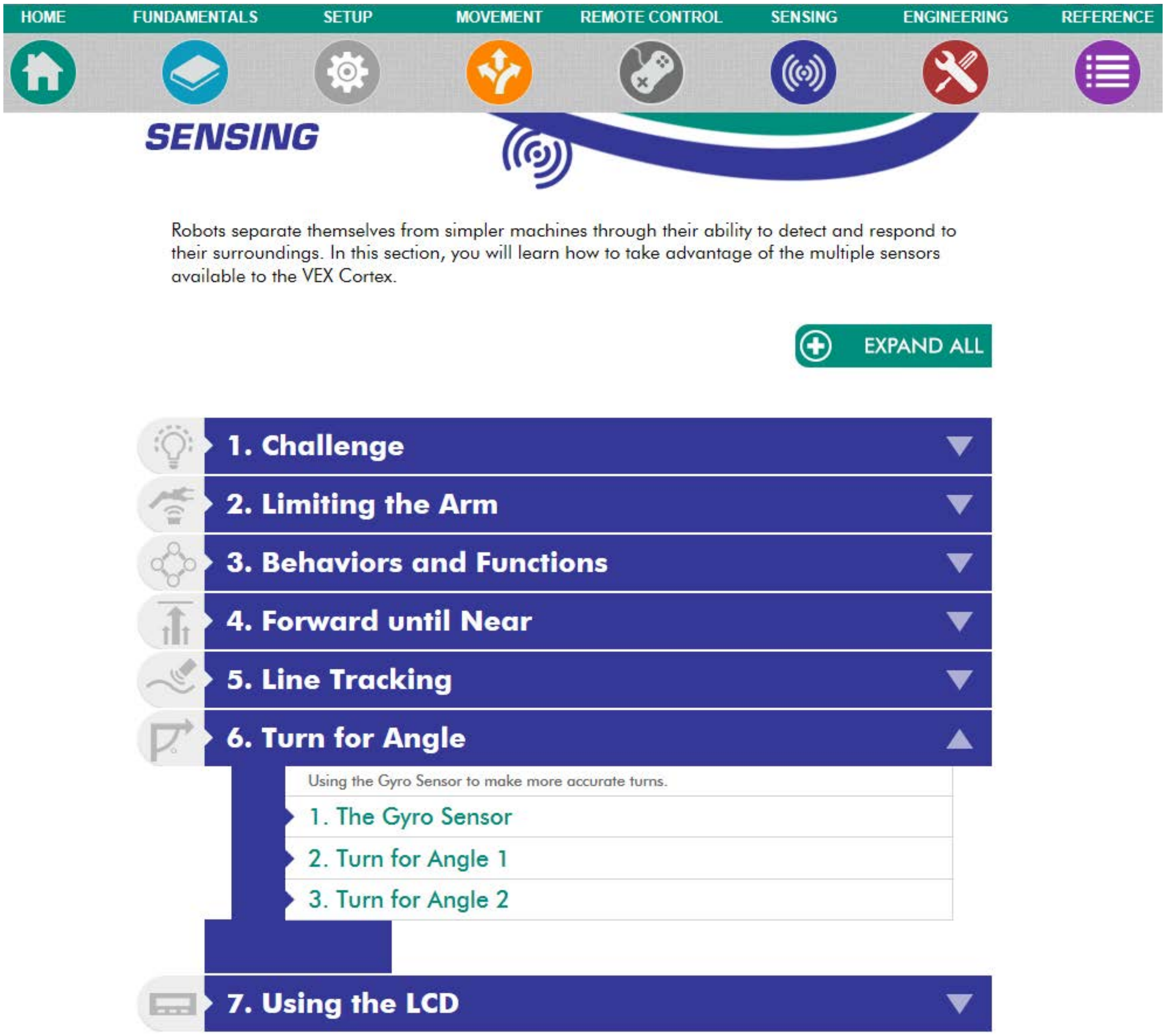






Robot Virtual World Programming Challenges

Sensing/Turn for Angle



This lesson set will teach students how to program the VEX Gyro Sensor

The Gyro Sensor Video - This video introduces the student to how the Gyro Sensor works.

Sensing Turn for Angle Part 1 - Part one of a video set that teaches how to reset and program the gyro sensor to work with your VEX robot.

Sensing Turn for Angle Part 2 - Part two of a video set that teaches how to reset and program the gyro sensor.

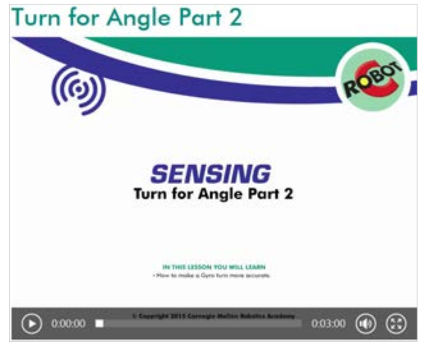
The ROBOT 500 Challenge PDF and RVW - This challenge comes in both a physical and virtual version.

The Minefield Challenge PDF and RVW - This challenge comes in both a physical and virtual version.

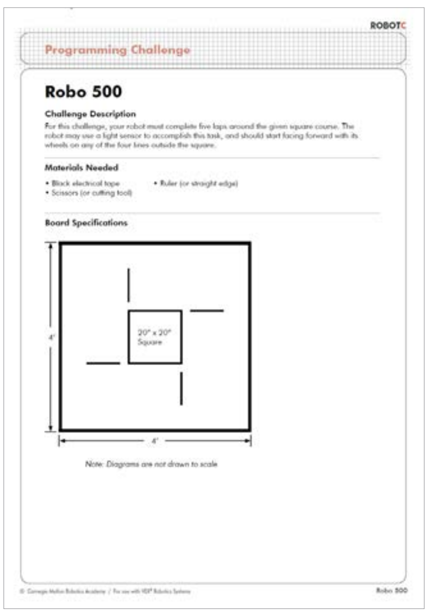
Additional resources for the Sensing Turn for Angle lesson set can be found on the next page.

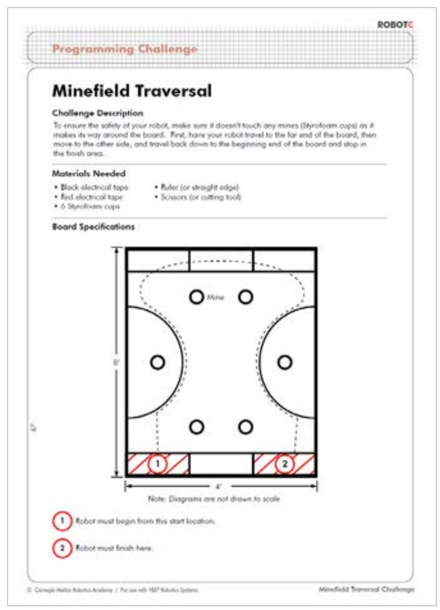
Sensing/Turn for Angle (continued)

Instructional Videos



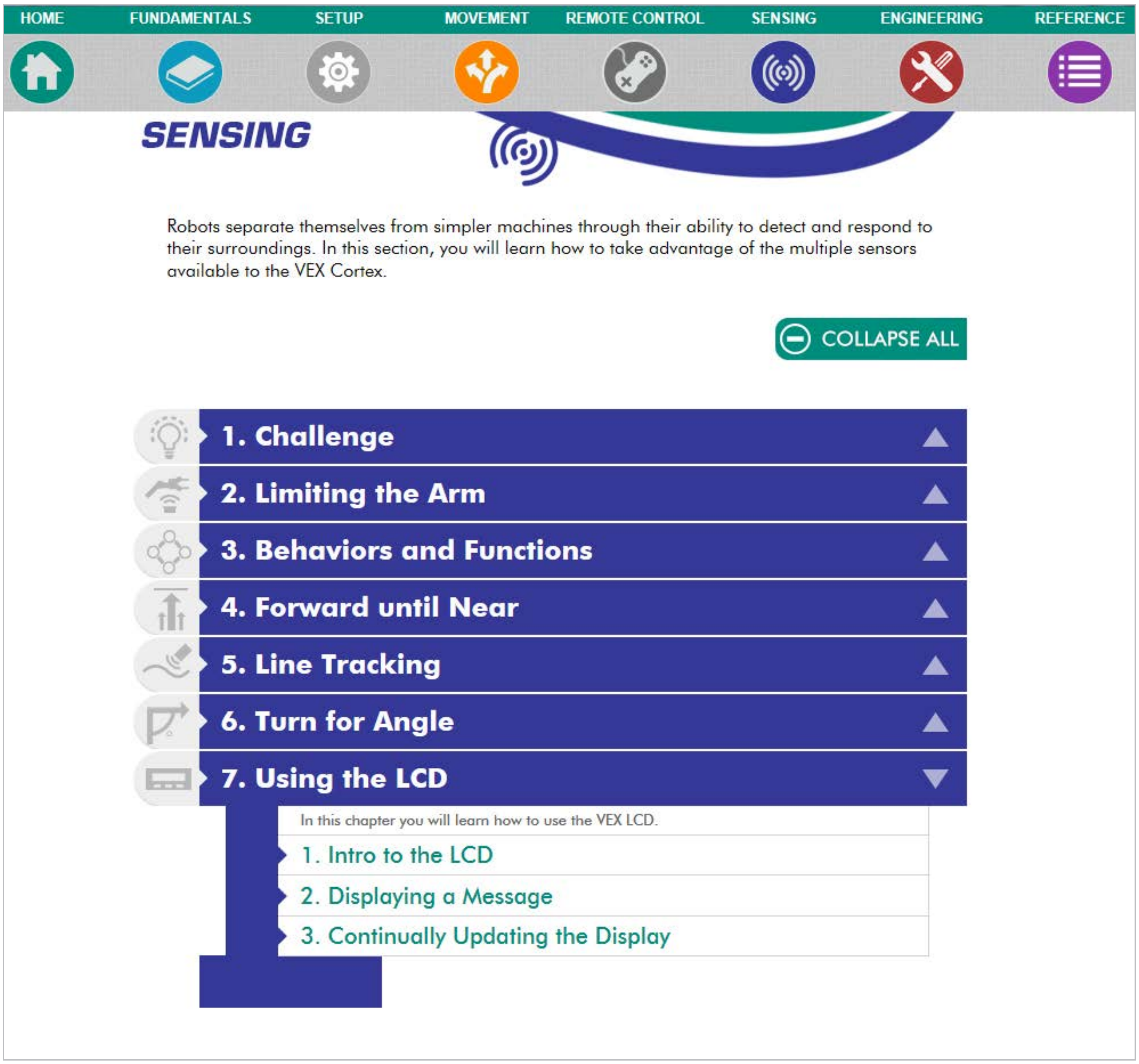
Physical Robot Programming Challenges





Virtual Robot Programming Challenges

Sensing/Using the LCD



This lesson set will teach students how to program the VEX LED Screen

The VEX LCD Video - This video introduces the many features of the LCD screen.

Displaying Sensor Values on the VEX LCD Video - This video teachers the student how to display continually changing values to the LCD.

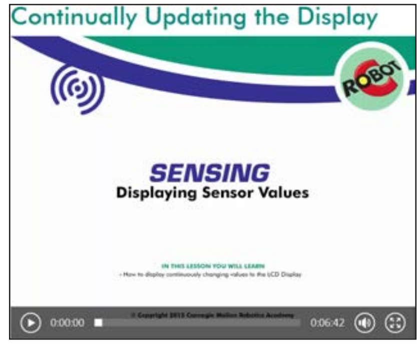
Continually Updating the VEX LCD Video - This video teachers the student how to display continually changing values to the LCD.

The LCD Display Handout PDF - This handout was produced by Innovation First and explains how to connect the LCD to the VEX controller

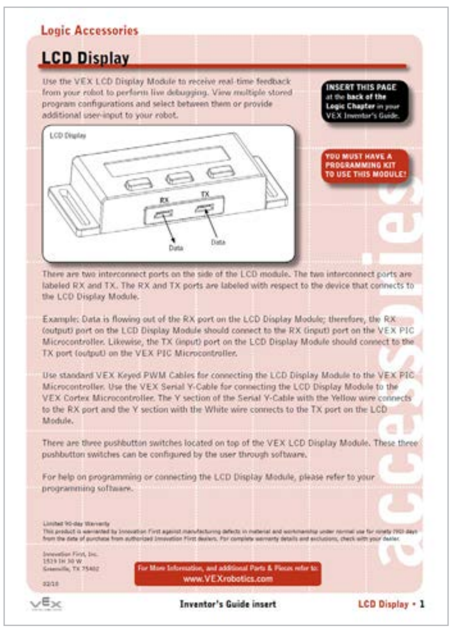
Additional resources for the Using the LCD lesson set can be found on the next page.

Sensing/The LCD Display (continued)

Instructional Videos



LCD Handout



Logic Accessories

LCD Display

Use the VEX LCD Display Module to receive real-time feedback from your robot to perform live debugging. View multiple stored program configurations and select between them or provide additional user-input to your robot.

INSERT THIS PAGE at the back of the Logic Chapter in your VEX Inventor's Guide.

YOU MUST HAVE A PROGRAMMING KIT TO USE THIS MODULE!

LCD Display

RX

TX

Data

Data

There are two interconnect ports on the side of the LCD module. The two interconnect ports are labeled RX and TX. The RX and TX ports are labeled with respect to the device that connects to the LCD Display Module.

Example: Data is flowing out of the RX port on the LCD Display Module; therefore, the RX (output) port on the LCD Display Module should connect to the RX (input) port on the VEX PIC Microcontroller. Likewise, the TX (input) port on the LCD Display Module should connect to the TX port (output) on the VEX PIC Microcontroller.

Use standard VEX Keyed PWM Cables for connecting the LCD Display Module to the VEX PIC Microcontroller. Use the VEX Serial Y-Cable for connecting the LCD Display Module to the VEX Cortex Microcontroller. The Y section of the Serial Y-Cable with the Yellow wire connects to the RX port and the Y section with the White wire connects to the TX port on the LCD Module.

There are three pushbutton switches located on top of the VEX LCD Display Module. These three pushbutton switches can be configured by the user through software.

For help on programming or connecting the LCD Display Module, please refer to your programming software.

Limited 90-day Warranty
This product is warranted by Innovation First against manufacturing defects in material and workmanship under normal use for ninety (90) days from the date of purchase from authorized Innovation First dealers. For complete warranty details and exclusions, check with your dealer.

Innovation First, Inc.
1519 IH 30 W
Greenville, TX 75402

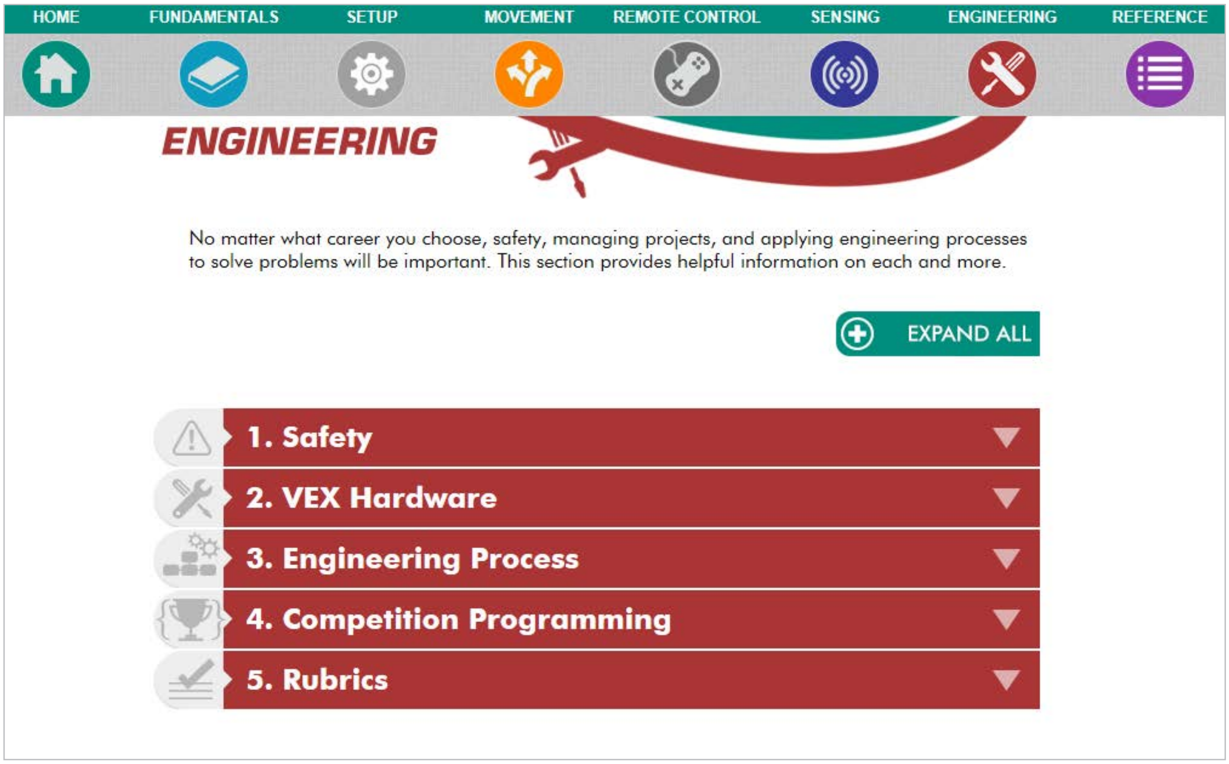
For More Information, and additional Parts & Pieces refer to:
www.VEXrobotics.com

accessories

Inventor's Guide insert

LCD Display • 1

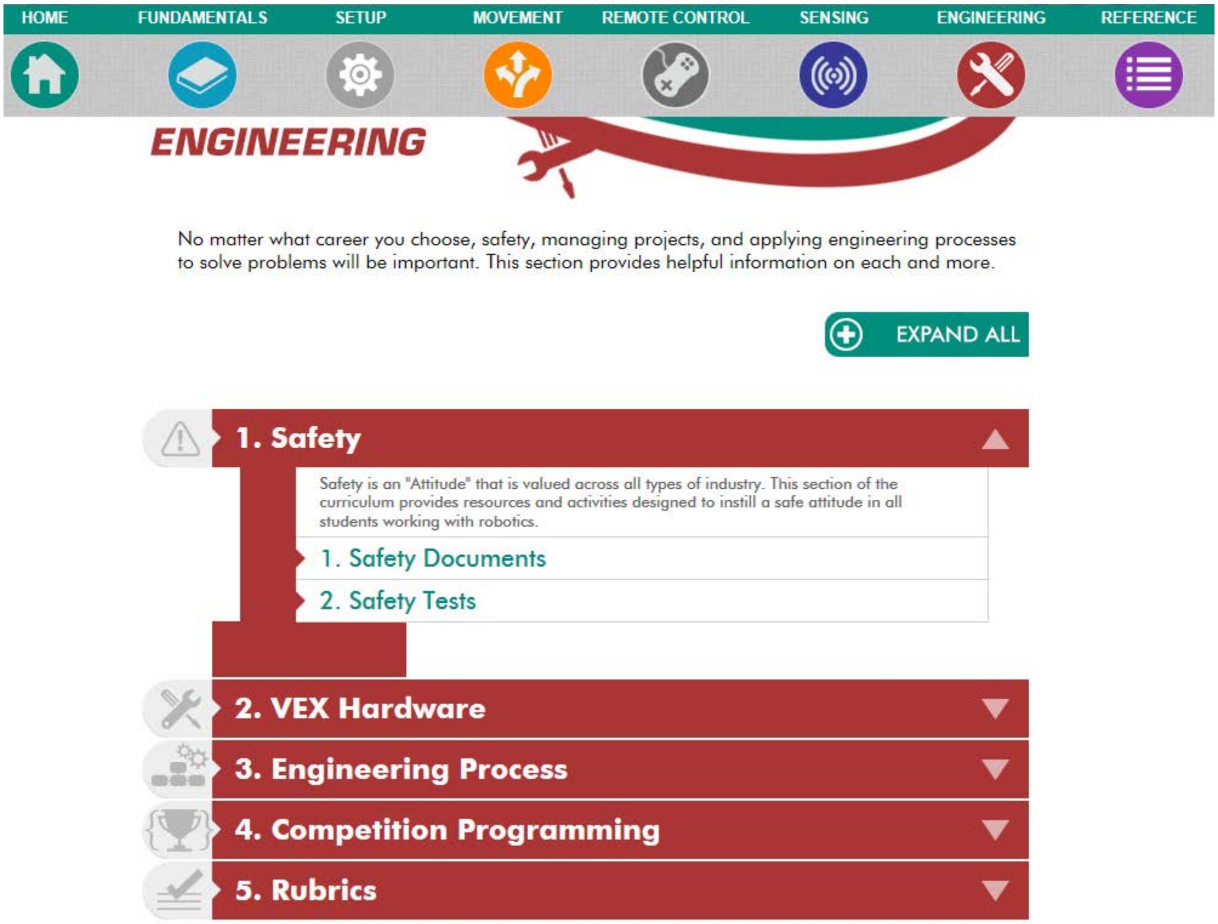
Engineering



Engineering

The Engineering section is broken into 5 sections and includes lots of handouts, instructional videos, and rubrics for assessment.

Engineering/Safety



Safety

Any course that involves moving parts, handling and processing materials and students requires safety training. Safety begins with the development of a safe attitude. Most accidents can be avoided if a student develops a safe and conscientious attitude. The safety lesson begins by challenging a student's general beliefs about safety and concludes with a safety inspection of the robotics lab.

Safety is an Attitude - A one page handout that defines what safety is and what safety is not, and concludes with statements that support the fact that most accidents are preventable with the development of a safe attitude.

General Lab Safety - A four page handout that spells out general safety rules, describes features of a safe classroom, safe storage, material handling, disposal of materials, tools and equipment, and ends with a list of definitions of terms that students may not know.

Safety Checklist - A three page handout that contains a safety checklist, rules to consider when you are moving things around the lab, and a one page safety poster.

Electrical Safety - A two page handout that describes safety rules when working with electricity and common causes of electrical accidents, including defective equipment, unsafe practices, and lack of electrical knowledge.

Power Tool Safety - A one page handout that sets rules and expectations for when students use power tools in the robotics lab.


Safety

Safety is an Attitude

Safety in General

A. Safety is defined as freedom from danger or hazards.
B. Safety is a personal responsibility. Most accidents are caused by one person and happen to one person.
C. Poor safety attitudes are the biggest hazards in most cases.

Safety is NOT...

A. Cynicism- "Safety is kid's stuff"
B. Fatalism- "I can't do anything to stop accidents."
C. Laziness- "It's just too much trouble."
D. Temper- "I'll do it my way or else!"
E. Forgetfulness-"I meant to but forgot."
F. Carelessness-"Oh, why bother."
G. Overconfidence-"I'll never get hurt." "It will never happen to me!"
H. Recklessness-"Danger is the spice of life!"
I. Ignorance-"I don't need to wear specific glasses for this."
J. Preoccupation-"I was thinking of something else instead of paying attention."

Accidents are Preventable

A. Accidents are a result of attitudes and of actions which you can eliminate. Safety is a state of mind.
B. People could prevent over 90% of all accidents if they wanted to because over 90% of all accidents are caused by people--not mechanical failures or freaks of nature.
C. Be your own safety expert by knowing how to prevent accidents.
D. Come to work relaxed--good health is a factor in accidents. Fatigue is a frequent factor in accidents.
E. Know and follow the safety rules of your job--be familiar with the operation of all machines and equipment that you use.
F. Know yourself--your abilities, skills, and limitations.
G. Watch out for the practical joker; "clowning" is a major cause of accidents.
H. Know what to do in an emergency; know first aid and emergency procedures.
I. Know whether your area responds to the 911 telephone call number for emergencies. Know what information you should give.
J. Have positive attitudes such as self-control, sound judgment, and the desire to be safe.

VEX® Cortex® Video Trainer using ROBOTC®

Safety continued next page.

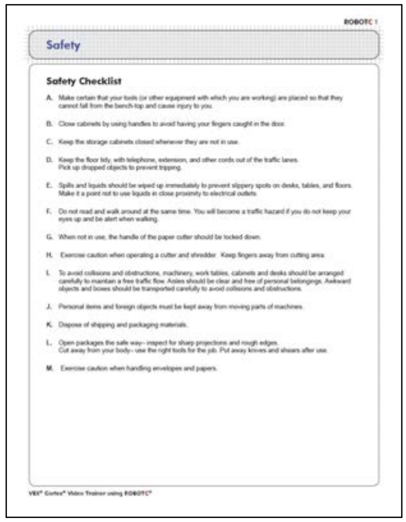
Engineering/Safety continued

Handouts

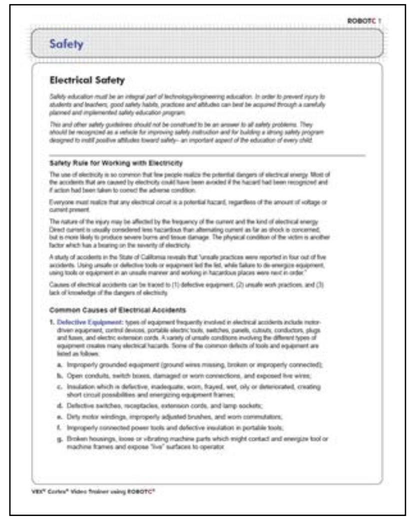

Safety

General Safety Considerations

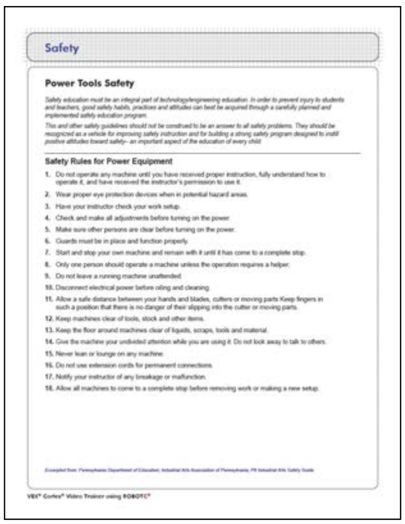
FACILITY DESIGN


Safety

Safety Checklist

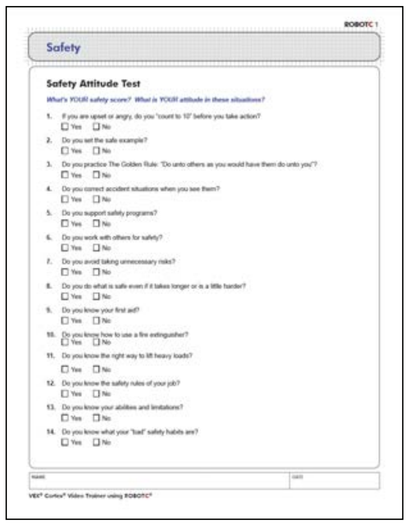

Safety

Electrical Safety

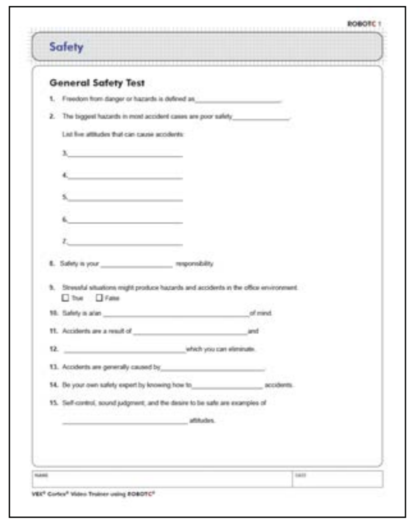

Safety

Power Tools Safety

Safety Rules for Power Equipment


Safety

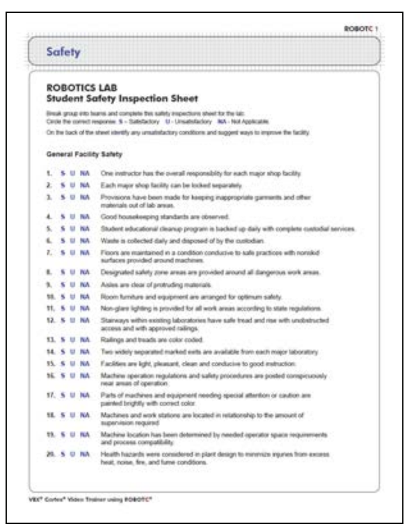
Safety Attitude Test


Safety

General Safety Test

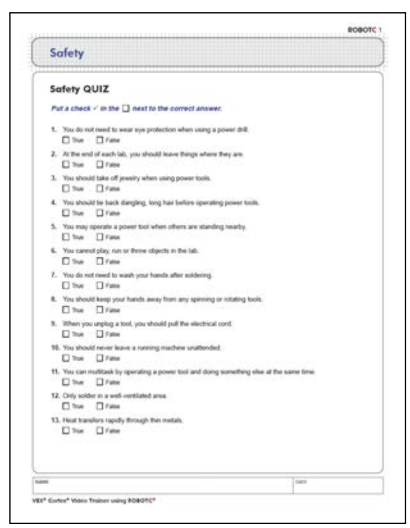
Safety Tests and Answers - Three different safety quizzes designed to check students' understanding of the importance of safety.

Robotics Lab Safety Inspection Sheet - Helps students to understand that they need to monitor the robotics classroom for safety.


Safety

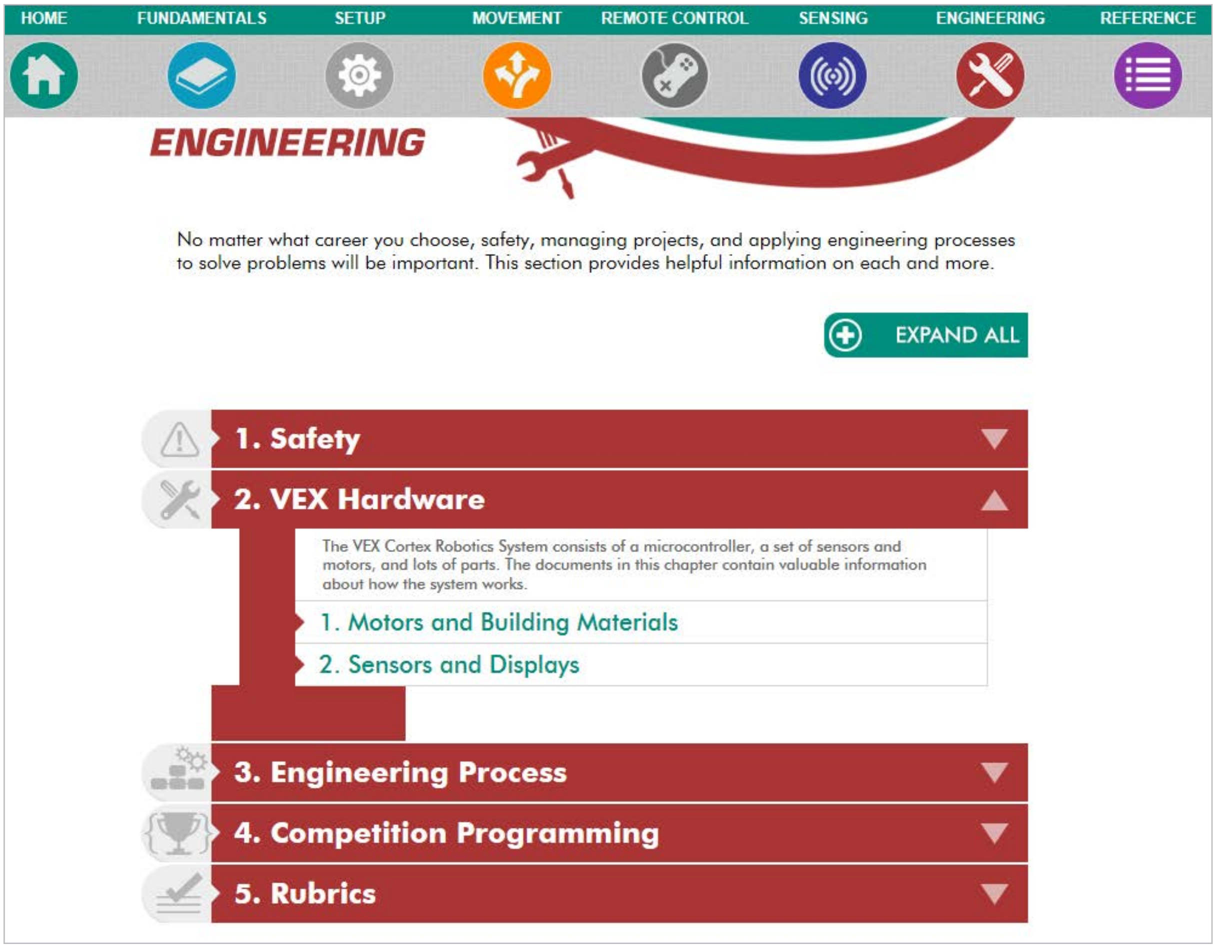
ROBOTICS LAB
Student Safety Inspection Sheet

General Facility Safety


Safety

Safety QUIZ

Engineering/VEX Hardware



VEX Hardware

The VEX Cortex Robotics System consists of a micro controller, a set of sensors and motors, and lots of parts. The documents on the next couple of pages contains valuable information about how the system works. These documents are available online for free student access and can be used as reference documents or assigned as homework study guides.

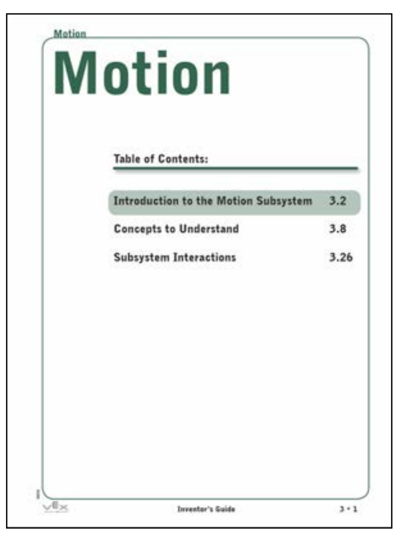
Structure

Structure

Table of Contents:

Introduction to Structure Subsystem	2.2
Concepts to Understand	2.7
Subsystem Interactions	2.17

Inventor's Guide 2 • 1



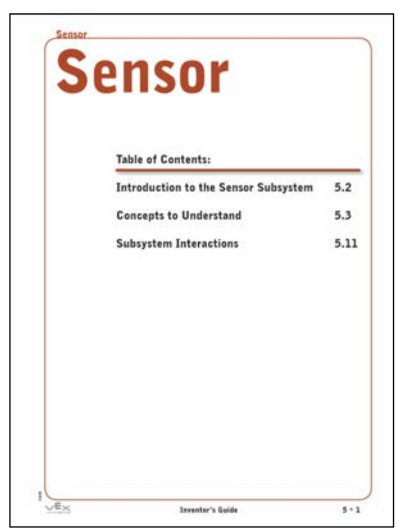
Motion

Motion

Table of Contents:

Introduction to the Motion Subsystem	3.2
Concepts to Understand	3.8
Subsystem Interactions	3.26

Inventor's Guide 3 • 1



Sensor

Sensor

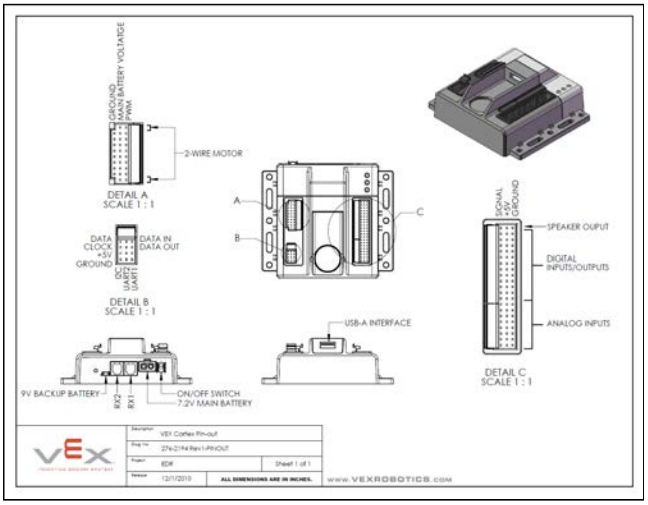
Table of Contents:

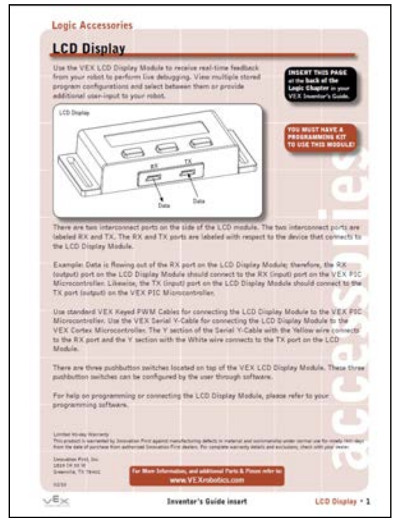
Introduction to the Sensor Subsystem	5.2
Concepts to Understand	5.3
Subsystem Interactions	5.11

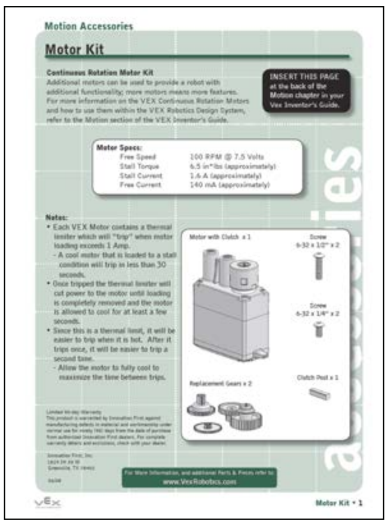
Inventor's Guide 5 • 1

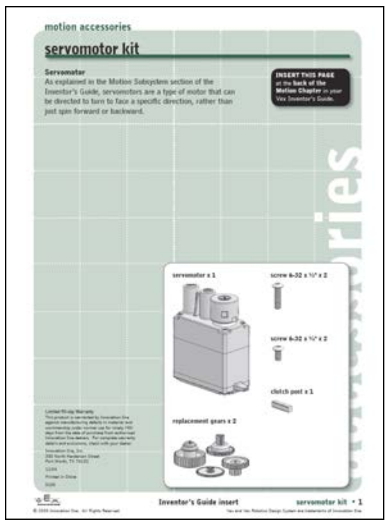
Fundamentals/VEX Hardware (continued)

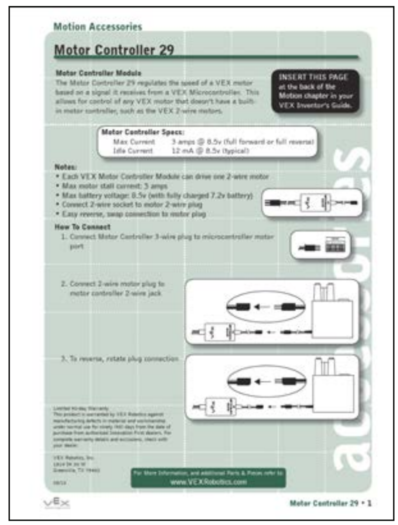
http://www.education.rec.ri.cmu.edu/products/teaching_robotc_cortex/index.html

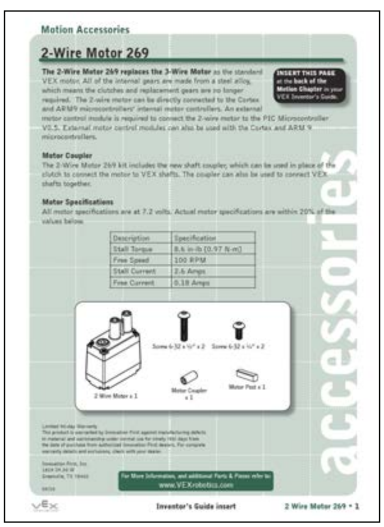


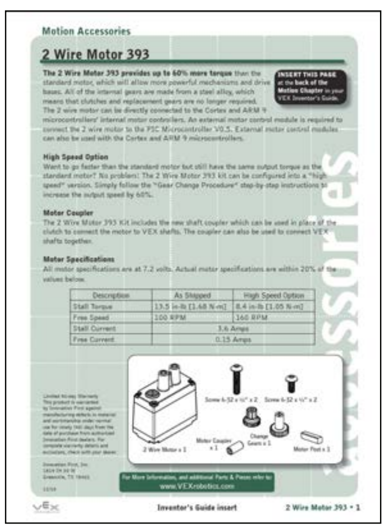


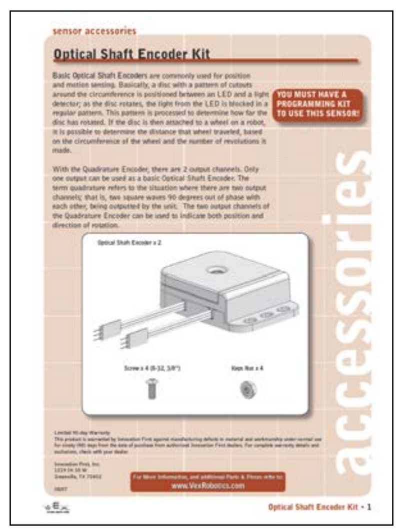






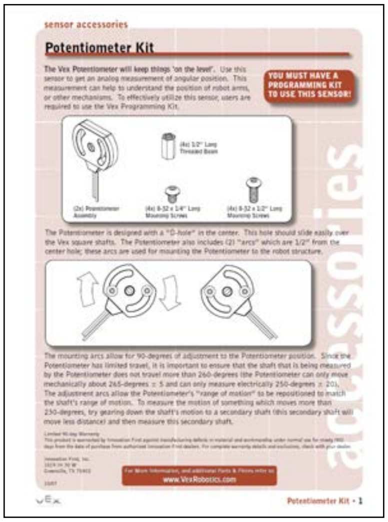


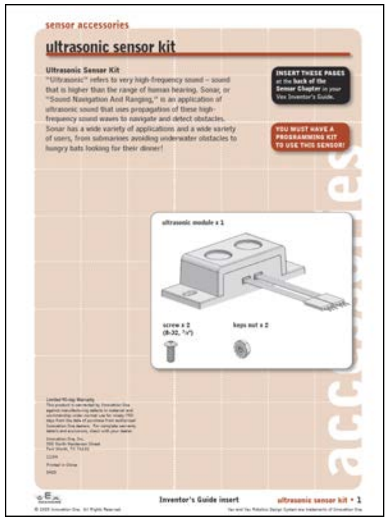


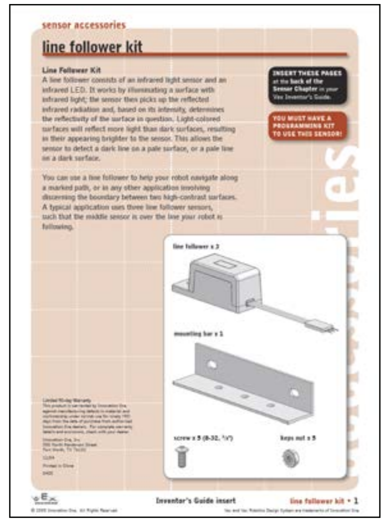


Engineering/VEX Hardware (continued)

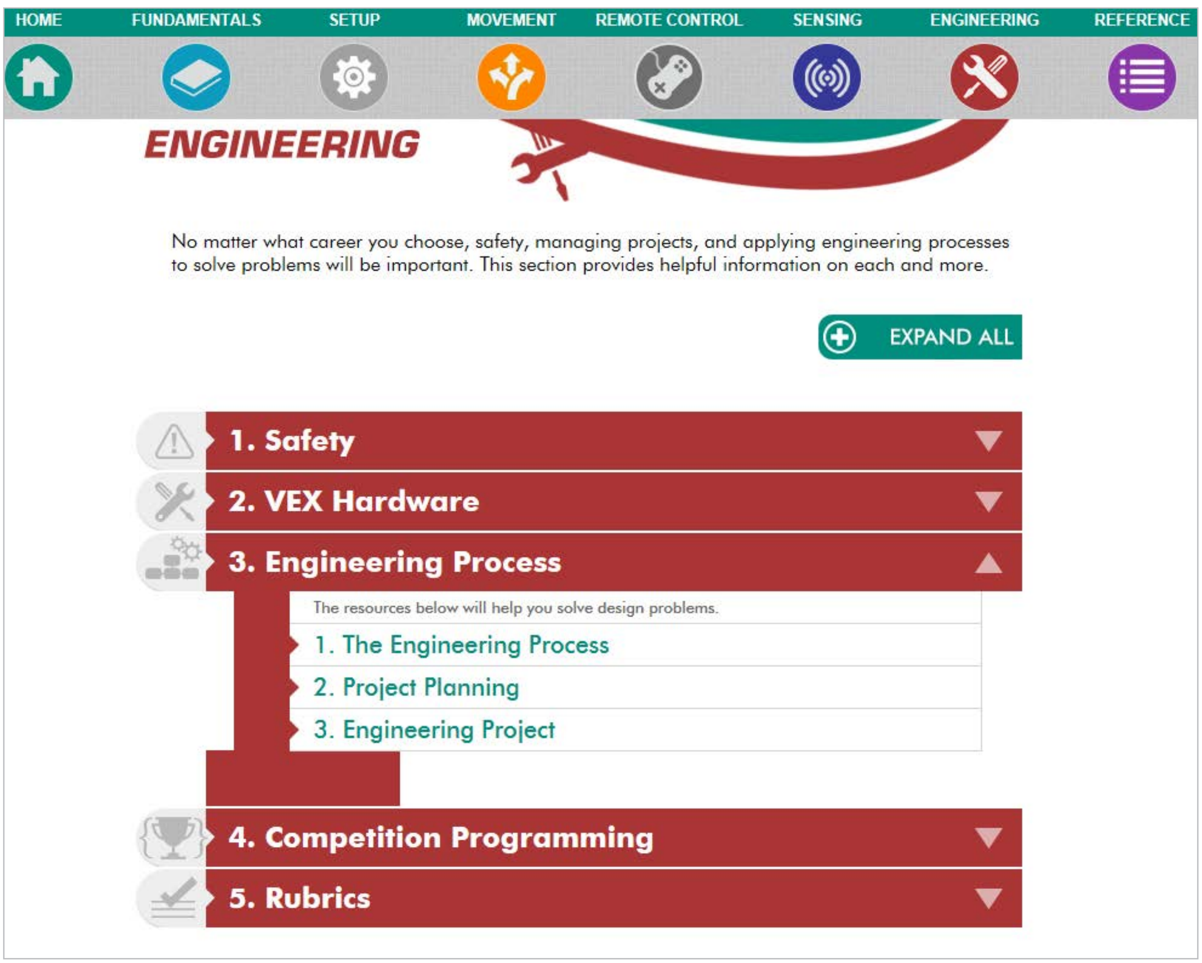
Handouts







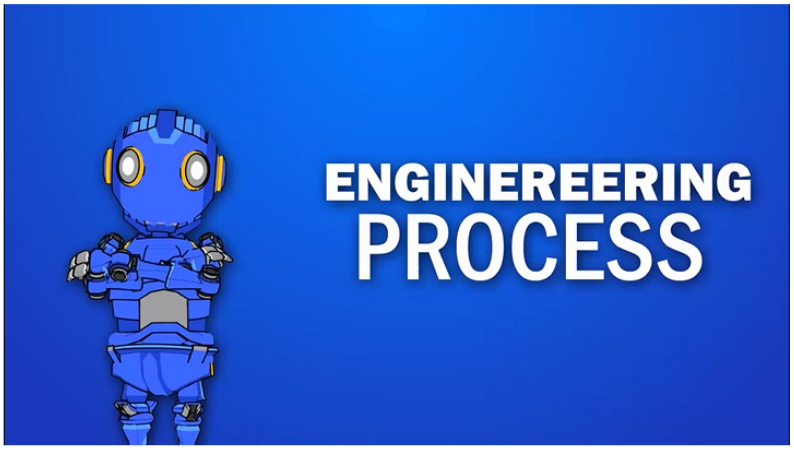
Engineering/Engineering Process



Many schools compete in robotic competitions; other schools are using the VEX Cortex and IQ systems to teach engineering. The resources in the engineering section provide students with materials that teach how to manage and solve engineering design problems.

Engineering/Engineering Process *continued*

The Engineering Process



Engineering Process Reference

Keeping an Engineering Journal

Design Reviews

Understanding the Problem

Brainstorming

Engineering Definitions

Engineering Process Resources

Engineering Process Video - Engineering Process Video - The five minute Engineering Process Video highlights the importance of research, planning, developing prototypes, and iterative testing when solving engineering design problems.

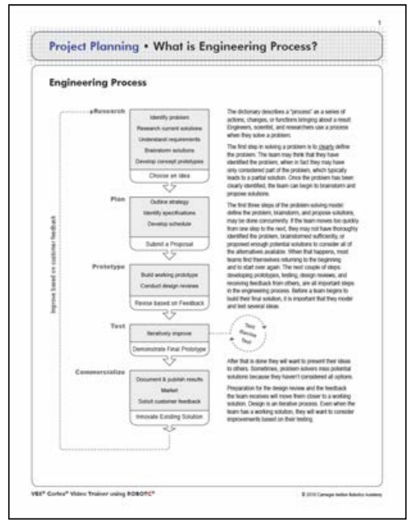
Engineering Process Reference - This three page set of handouts describes steps that engineers use to solve problems, provides a set of definitions for the word “engineering”, and describes the iterative nature of design.

Understanding the Problem PDF - One of the keys to solving any problem is “Understanding the Problem”. This three page set of handouts consists of: Defining the Problem, Technical Research, and Creating a Design Specification.

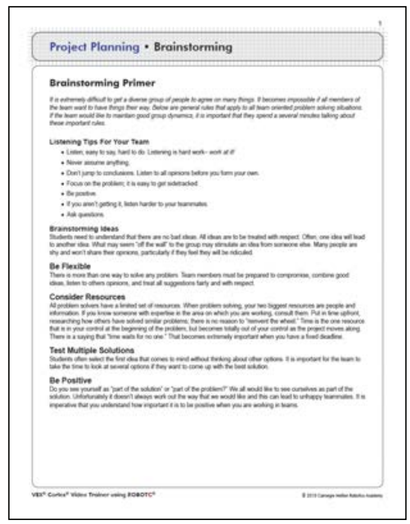
Brainstorming PDF - Another key problem-solving step involves meetings where people brainstorm together to develop potential solutions. This three page set of handouts is broken into: a Brainstorming Primer, Things to Think About, and Brainstorming Tips. Each handout can be used individually or as part of a set.

Design Reviews PDF - Engineers conduct design reviews on a regular basis. This two page handout describes how to conduct weekly team design reviews, as well as preliminary and detail design reviews.

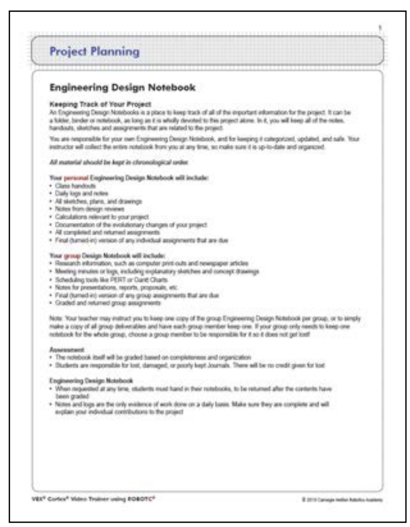
Engineering Design Notebook - Each student is required to keep an engineering design notebook. This two page handout describes the what is kept in the notebook and what a daily log is.


Project Planning • What is Engineering Process?

Engineering Process


Project Planning • Brainstorming

Brainstorming Primer

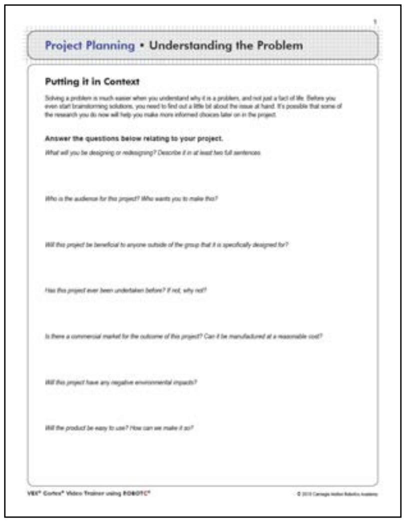

Project Planning

Engineering Design Notebook

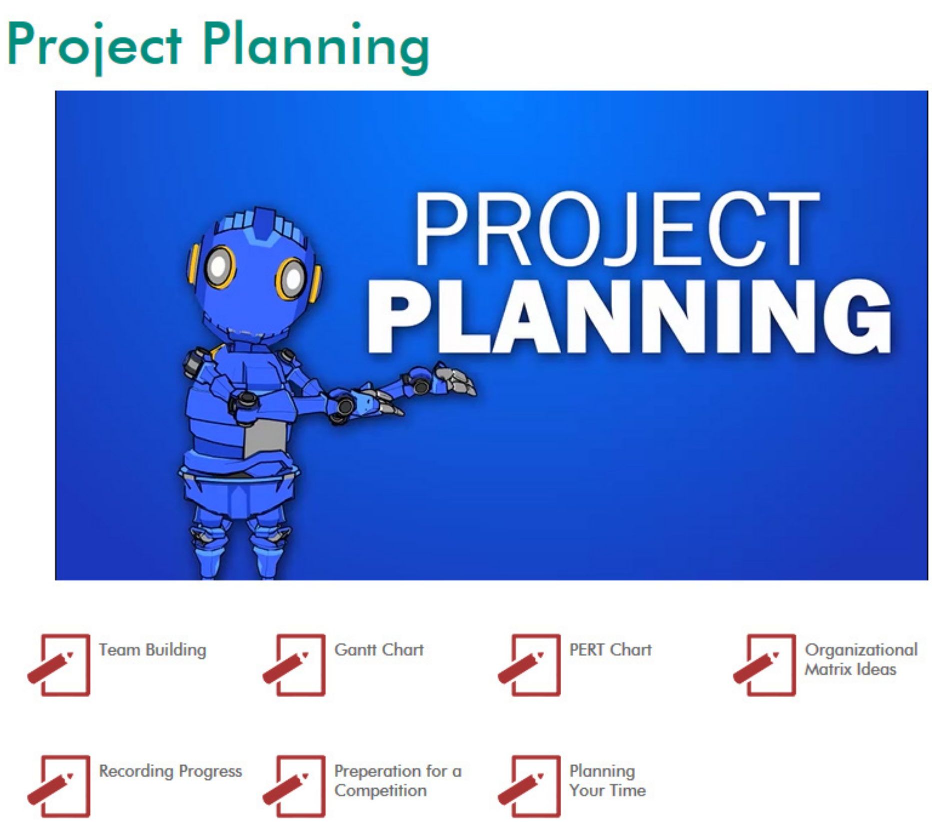
Engineering process resources continued next page.

Engineering/Engineering Process *continued*

Handouts



Engineering/Project Planning



Proper planning is very important to solving any design problem - This section provides many resources designed to help students learn to manage their time.

Project Planning resources can be found on the next page.

Engineering/Project Planning *continued*

Project Planning Handouts and Resources

Project Planning Video - The five minute Project Planning Video uses a combination of humor and examples to describe what a well planned project looks like.

Organizational Matrix Ideas PDF - This three page handout graphically shows three methods of organizing projects.

Recording Progress PDF - The recording progress tools offer the project manager three solutions that can be used to help team members to document the team's progress toward the project goals.

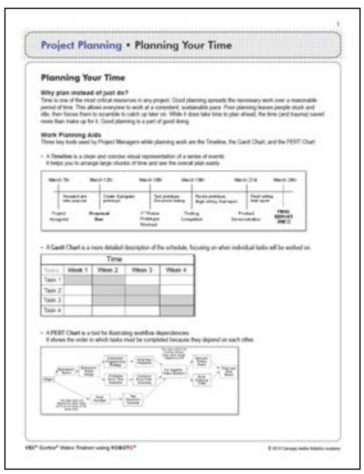
Gantt Chart PDF - A Gantt chart provides a graphical illustration of a schedule to help plan, coordinate, and track specific tasks in a project. This one page handout is designed to teach students how Gantt charts work.

PERT Chart PDF - A PERT chart is a tool that graphically illustrates when parts of the project become due. The advantage of the PERT chart is that it shows which things must be completed in sequence and which things need to be completed simultaneously. This one page handout is designed to teach students how PERT charts work.

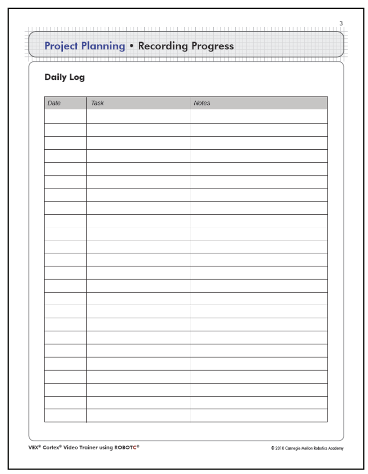
Preparing for a Competition PDF- Robotic competitions offer unique opportunities to teach students about time management, resource allocation, teamwork, and problem solving, all within a context that they find challenging but fun. The Preparing for a Competition handout is designed to support robotics teams as they plan for the competition.

Planning Your Time PDF - Time management is a crucial skill to develop. This four page handout uses a simple activity, planning a birthday party, to describe a critical skill set that everyone should learn: how to manage time.

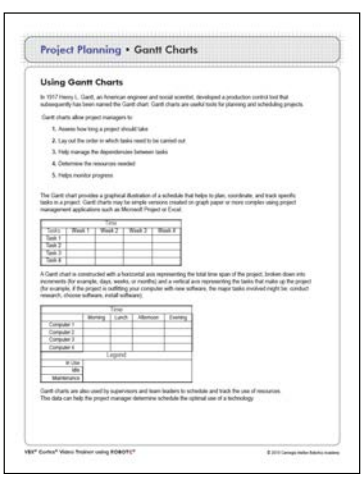

Project Planning • Definitions of Engineering


Project Planning • Planning Your Time

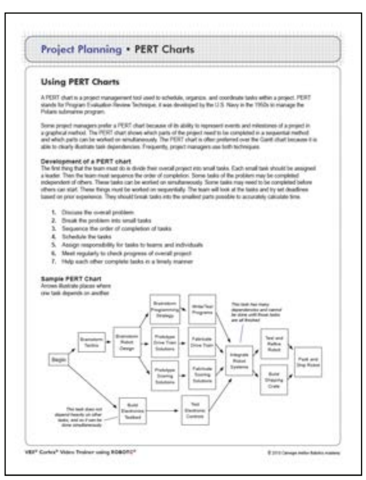
Planning Your Time


Project Planning • Recording Progress

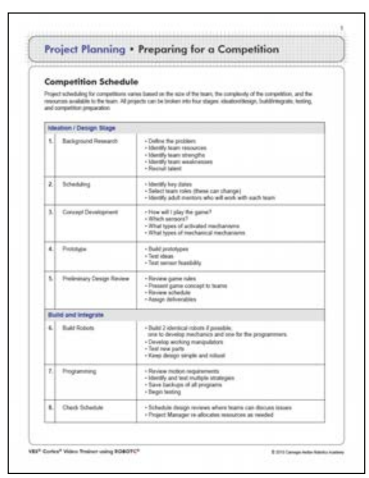
Daily Log


Project Planning • Gantt Charts

Using Gantt Charts

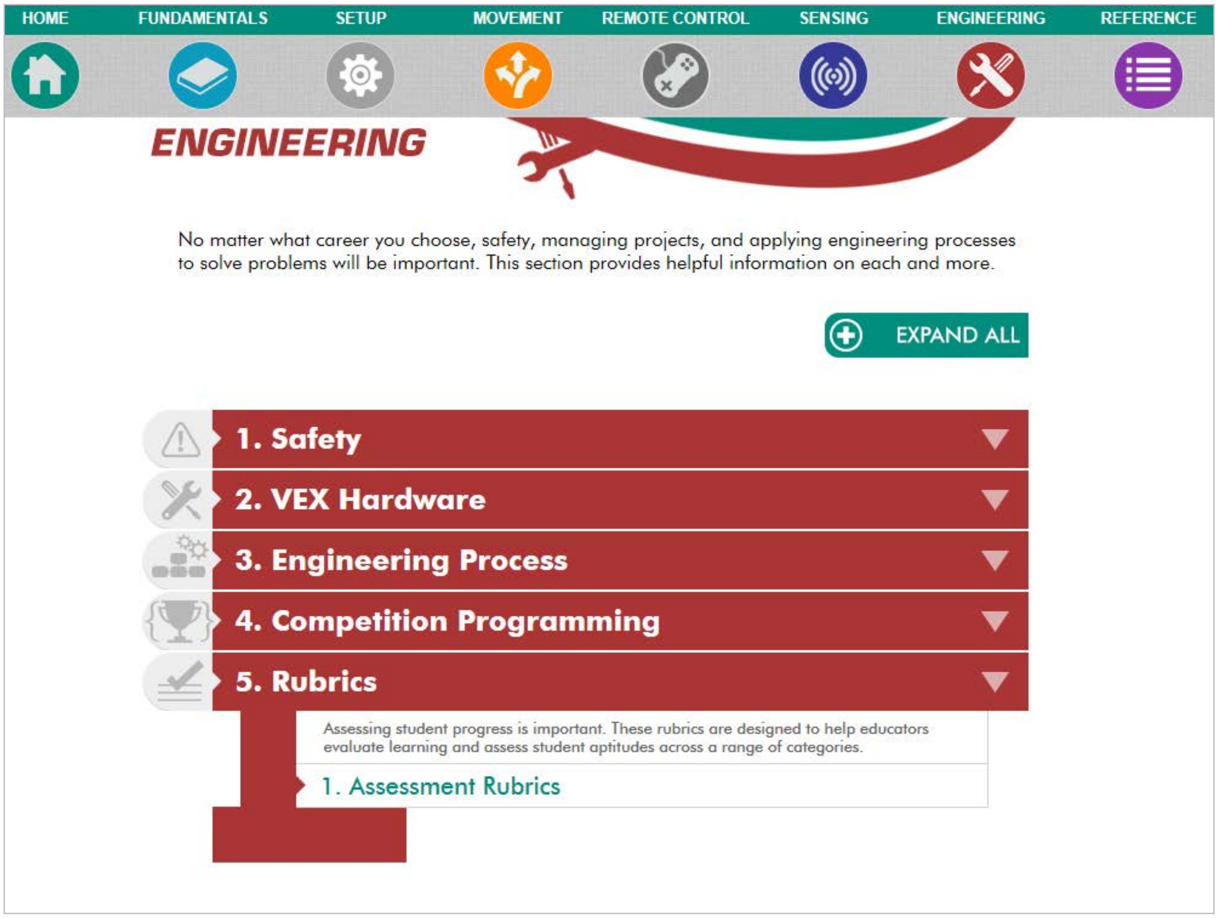

Project Planning • PERT Charts

Using PERT Charts


Project Planning • Preparing for a Competition

Competition Schedule

Engineering/Assessment Rubrics



Assessment Rubrics

Timely assessment is paramount in today's educational environment. A clear expectation of what is being assessed is a key to training students. Traditional assessments are provided in the curriculum; ie quizzes. The assessments found in this section are assessment rubrics for project-based learning. There are many other tools that a teacher may use, but this section provides some examples. Rubrics allow all stakeholders to see what is being measured.

Writing Criteria Rubric - Writing is a process and good writing requires several steps: brainstorming, outlining, pre-writing, and editing. This is a simple rubric that check for those steps.

Engineering Journal Rubric - Explains to students what is expected in their engineering journals.

Presentation Rubric - Helps students determine what a good presentation should include.

Request for Proposal Rubric - Helps students to determine what is being evaluated in their RFP submission.

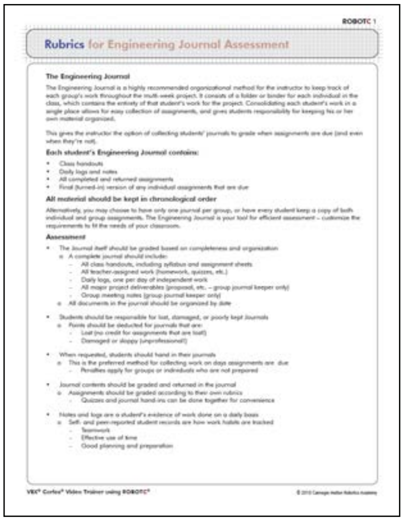
Work Habit Evaluation - This is a great tool for students to use to develop strong work habits.

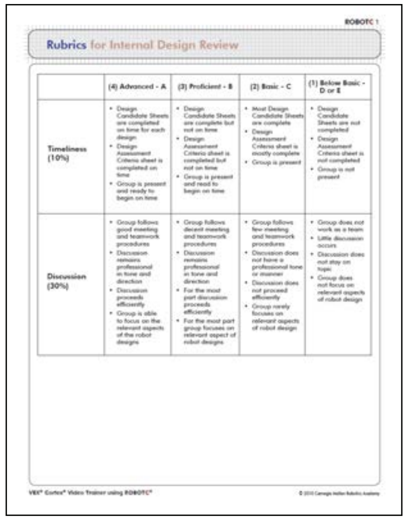
Workplace Competencies Rubric - This rubric helps students to develop the skills that are valued by industry.

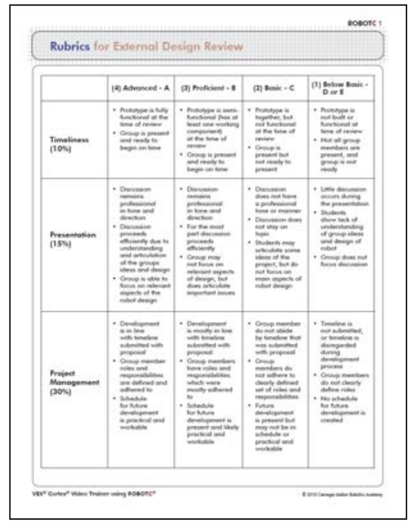
Internal Design Rubric - This evaluation tool helps students understand the expectations and preparation needed for an internal design review.

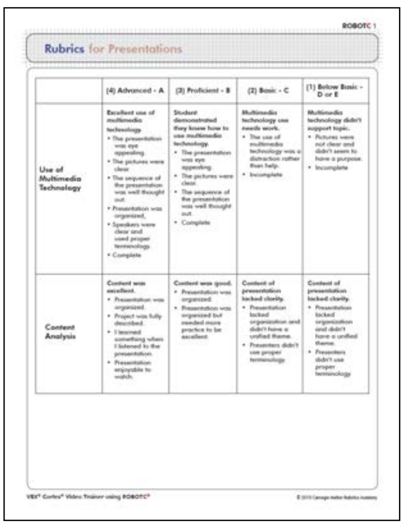
External Design Review - This evaluation tool helps students understand the expectations and preparation needed for an external design review.

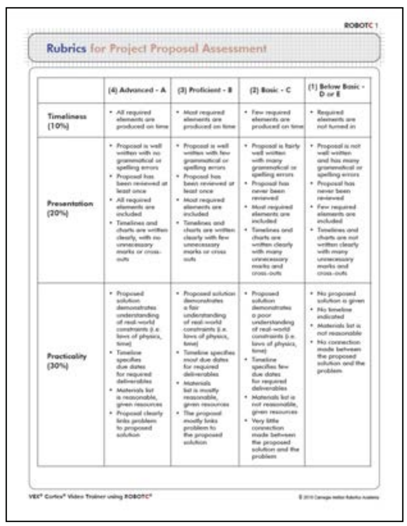
Engineering/Assessment Rubrics continued

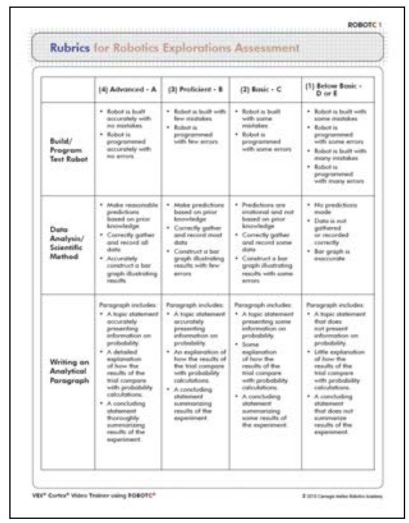

Rubrics for Engineering Journal Assessment

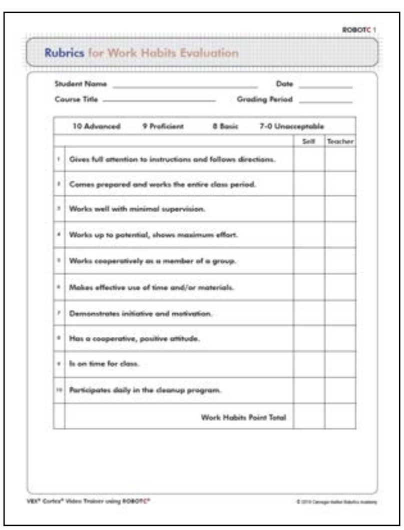

Rubrics for Internal Design Review


Rubrics for External Design Review


Rubrics for Presentations


Rubrics for Project Proposal Assessment


Rubrics for Robotics Explorations Assessment

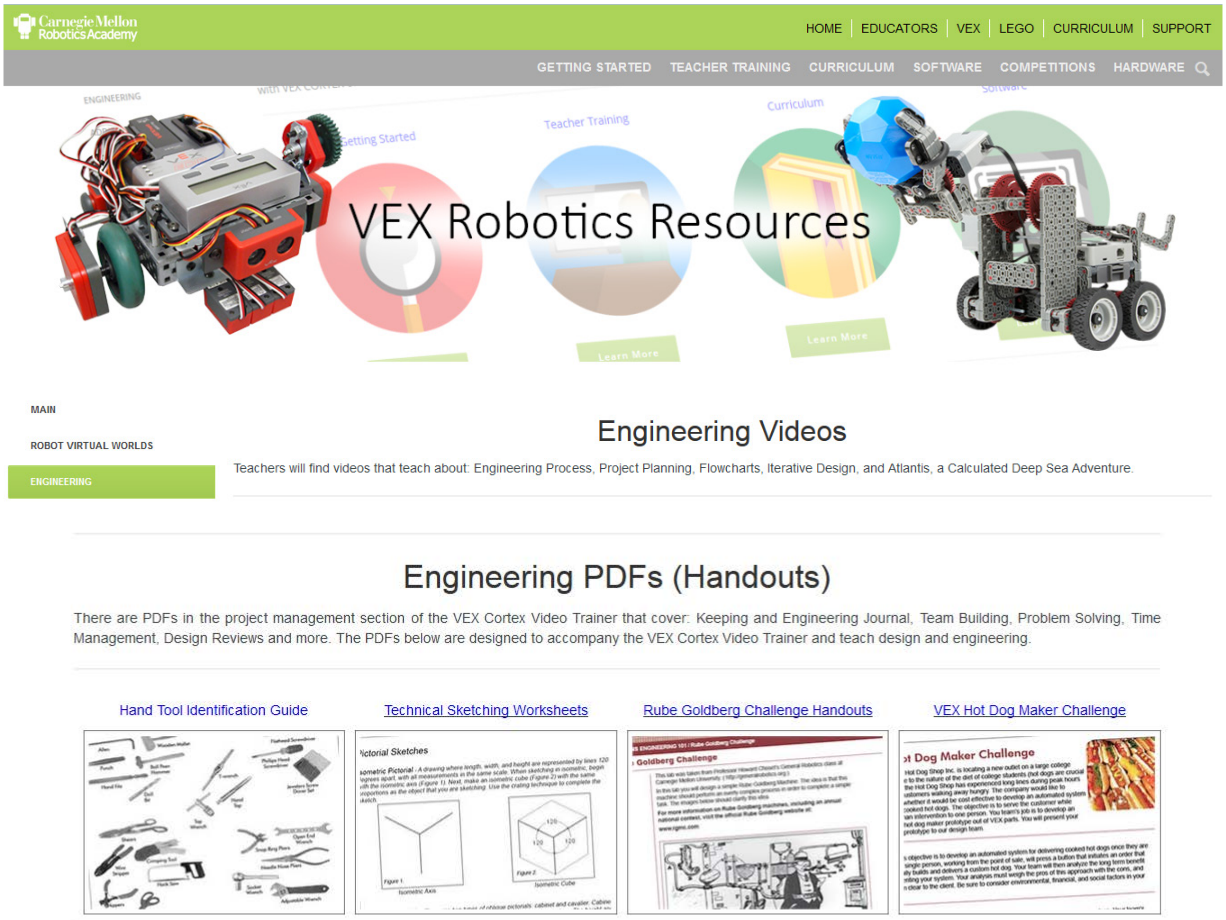

Rubrics for Work Habits Evaluation


Rubrics for Workplace Competencies Evaluation


Rubrics for Robotics Class Writing

Reference Index

Additional Resources at vexteacher.com



Teaching The Engineering Design Process

"The Engineering Design Process is the formulation of a plan to help an engineer build a product with a specified performance goal."

This process involves a number of steps, and parts of the process may need to be repeated many times before production of a final product can begin. Engineering design teaches problem solving, brainstorming, time and resource management, cooperation and collaboration, and the soft skills that today's workforce demands. Robotics and robotics competitions provide a high energy organizer to teach the Engineering Design Process.

VEX Robotics Competitions - Each year the Robotics Education and Competition (REC) Foundation hosts an International VEX competition, in 2014 there were over 10,000 teams. The Competition is incredibly well run and can provide a life-changing experience for students. The Robotics Academy is a strong supporter of the formalized robotics competitions because they are well thought out, provide real deadlines, and provide students with a high energy 21st century learning experience.

School-based Competitions - Many teachers use school based competitions. They setup classroom design challenges; some of them are competitive and others are cooperative in nature. There are several examples of possible school based cooperative design problems like The Rube Goldberg Challenge, and the Hot Dog Maker and Automated Workcell examples included in this curriculum.

The next couple of pages introduce resources that are available to teach the Engineering Design Process.

What is Engineering and Teaching the Engineering Design Process

Teaching engineering is an iterative process, just like engineering. Students are not going to learn engineering process by attempting to solve one engineering problem. They need to engineer solutions for many problems before they become good at it. There are multiple ways to introduce engineering and 5th grade teachers are going to approach the topic much differently than high school teachers. Start with small engineering projects that are a couple days in duration and introduce new organization concepts like brainstorming and ideation, cooperation and collaboration, Gantt and PERT Charts, and developing prototypes and Design Reviews as each project requires them. Always have students keep an Engineering Design Journal for every project. Below are some suggestions that others have used to introduce students to Engineering and Engineering Design Process.

1. Ask the class to discuss the difference between Engineering and Engineering Design Process. Help them to develop their own internal definition of what engineering is. Introduce students to engineering by showing the Engineering Process Video found on page 10 and provide them with the Definitions of Engineering PDF from page 12. Have them write a one-page paper describing “what is engineering”.

2. Introduce students to what the engineering process looks like by providing them with the Engineering Process PDF found on page 11.

3. Teach students to keep and Engineering Design Journal and require them to document their project by keeping all notes, sketches, and code snippets. Have them keep all of their handouts in their Engineering Design Notebook. There is a PDF describing the notebook on page 10.

4. Require students to work in teams. This is a very important skill for students to develop and it takes practice. Begin by having them review and discuss the “First Team Meeting” handout found on page 11.

5. Teach students how to brainstorm without offending each other. There is a handout on page 10 called “Brainstorming”. Give students ample opportunities to solve new problems regularly.

6. Require evidence of project planning and “Time Management” by having them develop PERT and GANTT Charts for various projects. Pass the Gantt and PERT chart handouts found on page 12.

7. Discuss how to break a project into manageable parts, assigning deliverables, and self assigning due dates. Have them watch the Project Planning Video on page 10.

8. Build prototypes and conduct design reviews. Use the design review handout on page 11.

9. Invite other adults to sit in on design reviews to review student solutions.

10. Require project documentation.

11. Iteratively test solutions and have students brainstorm how to improve them.

12. Give students the opportunity to present their solution

13. Debrief, talk about what worked, what didn’t work, and how to improve the process.

14. Give students other problems to solve - The students will improve each time they participate in the engineering design process. The first project may be a disaster for some teams. The debrief session is the most important part of the above process for new learners. Students need to recognize that this process will be with them in some form for the rest of their lives and so it is important that they become good at it.

Planning Your Project: Technical Sketching

Sketching provides a quick way to share ideas; this module introduces students to technical sketching, a fundamental tool of engineers. This section is not meant to replace a formal Engineering Design CAD course, but to teach kids basic skills that enable them to share ideas. The resources are divided into the following organizers and it will be up to the teacher to determine what is appropriate for their students:

1. Introduction to Technical Sketching
2. Drawing Conventions/Dimensioning
3. Pattern Developments
4. Pictorial Sketching
5. Precision Measurement
6. Sketching Fasteners

The Lesson section consists of a variety of hands-on technical sketching exercises of a variety of parts. It will depend on the teacher's point of emphasis, but this section of the course could take anywhere from two days to a week. Introduce students to sketching early, challenge them with a design problem that involves building a working model. Have them plan their project before they build it. This process involves sketches, dimensioning, and design reviews. Team members share ideas, and then make improvements based on testing and feedback. Require students to keep all sketches in their engineering design journal.

Introduction to Technical Sketching resources consists of a series of printable handouts that demonstrate sketching technique, a set of handouts that explains sketching technique, and multiple sketching exercises.

Drawing Conventions/Dimensioning consists of two printable handouts illustrating technical sketching line conventions and general dimensioning rules and several parts to dimension.

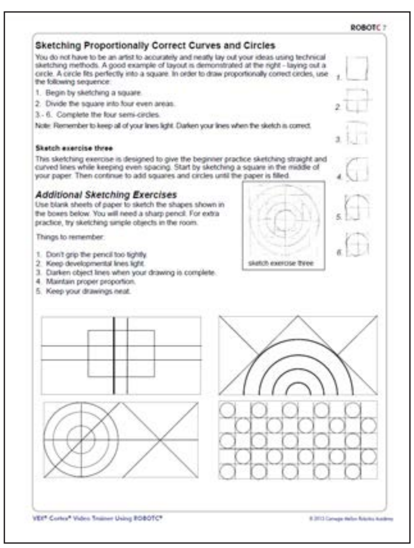
Pattern Developments is a handout that introduces the topic of pattern development and provides several examples of student-designed manipulators that use pattern developments.

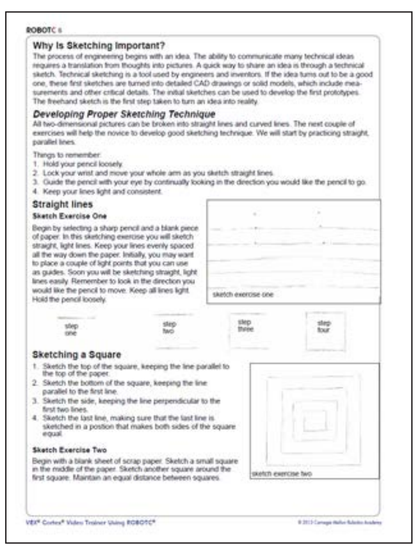
Pictorial Sketching consists of a series of printable handouts that reinforces "crating technique" when sketching pictorial drawings and a pictorial sketching handout.

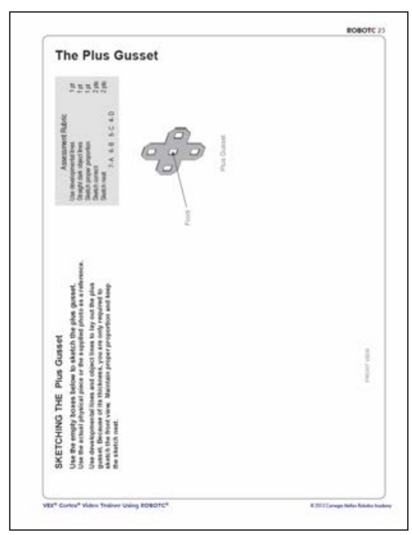
Precision Measurement consists of a couple of VEX robot parts that need to be accurately measured. The handouts are designed to give students practice measuring accurately, and a handout on units of measurement.

Sketching Fasteners - fasteners are used on many robotics engineering projects. This set of exercises give students the opportunity to both perfect their sketching technic and also learn the names of the various fasteners.

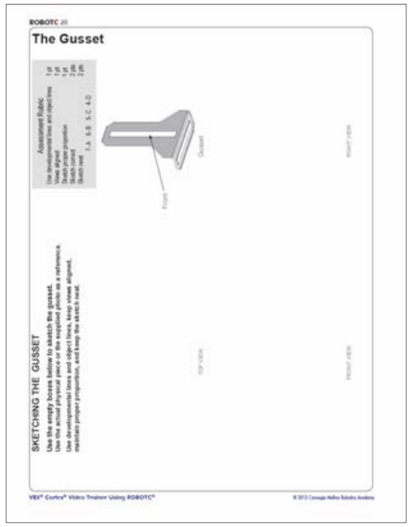
Introduction to Technical Sketching

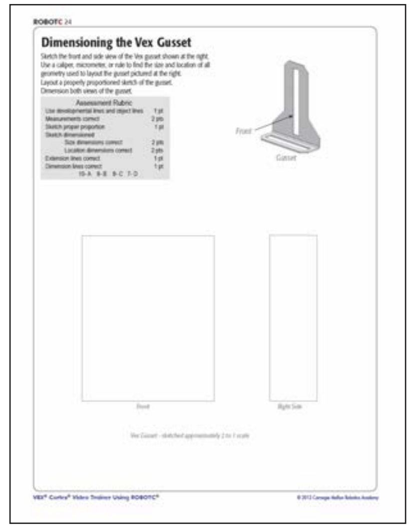




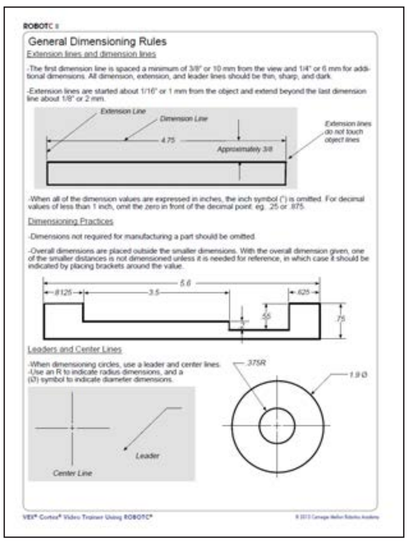


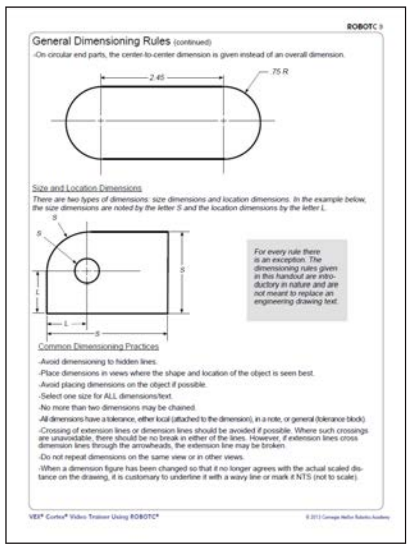
Introduction to Technical Sketching Continued

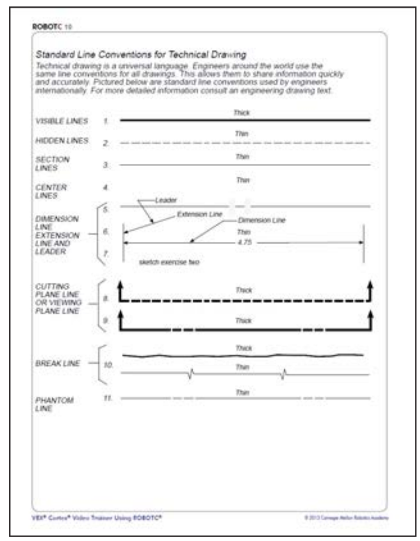


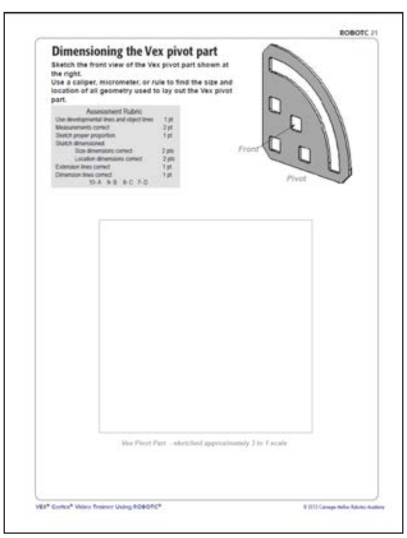


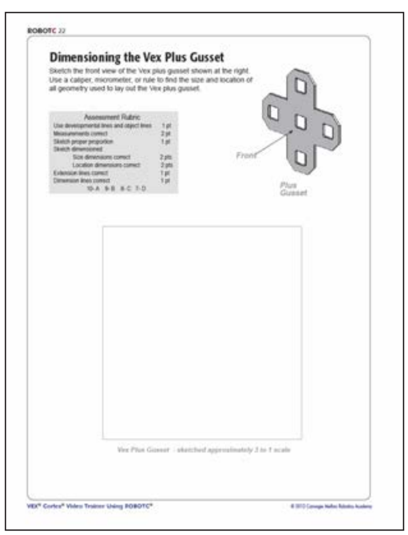
Drawing Conventions/Dimensioning

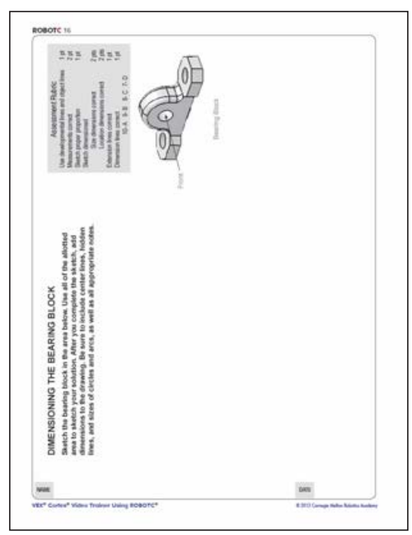




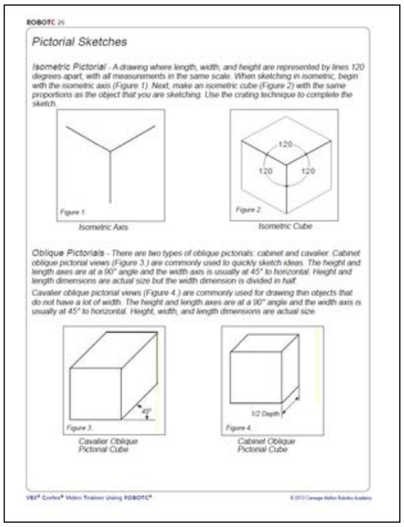








Pictorial Sketching


ROBOTC 25

Pictorial Sketches

Isometric Pictorial - A drawing where length, width, and height are represented by lines 120 degrees apart, with all measurements in the same scale. When sketching in isometric, begin with the isometric axis (Figure 1). Next, make an isometric cube (Figure 2) with the same proportions as the object that you are sketching. Use the crating technique to complete the sketch.

Figure 1. Isometric Axis

Figure 2. Isometric Cube

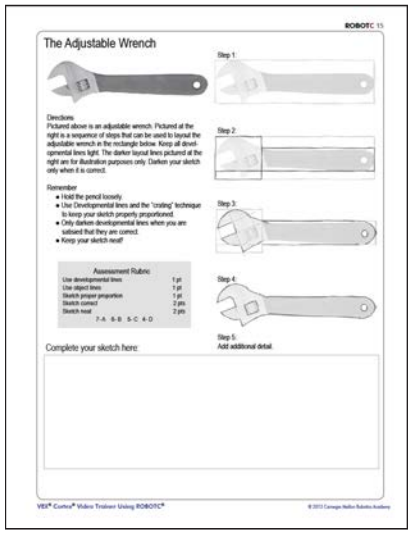
Oblique Pictorials - There are two types of oblique pictorials: cabinet and cavalier. Cabinet oblique pictorial views (Figure 3.) are commonly used to sketch ideas. The height and length axes are at a 90° angle and the width axis is usually at 45° to horizontal. Height and length dimensions are actual size but the width dimension is divided in half.

Cavalier oblique pictorial views (Figure 4.) are commonly used for drawing thin objects that do not have a lot of width. The height and length axes are at a 90° angle and the width axis is usually at 45° to horizontal. Height, width, and length dimensions are actual size.

Figure 3. Cavalier Oblique Pictorial Cube

Figure 4. Cabinet Oblique Pictorial Cube

VEX® Cortex® Video Trainer Using ROBOTC®


ROBOTC 15

The Adjustable Wrench

Step 1, Step 2, Step 3, Step 4, Step 5: Add additional detail

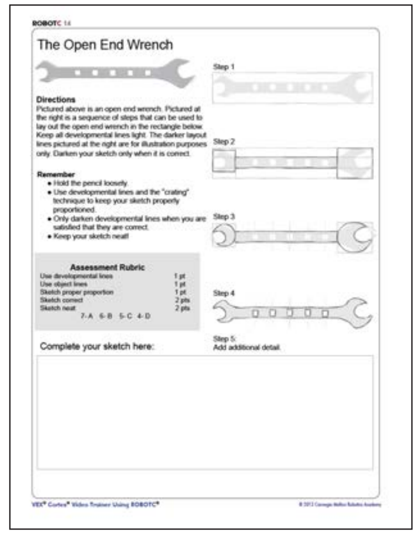
Directions

Pictured above is an adjustable wrench. Pictured at the right is a sequence of steps that can be used to layout the adjustable wrench in the rectangle below. Keep all developmental lines light. The darker layout lines pictured at the right are for illustration purposes only. Darken your sketch only when it is correct.

Remember
- Hold the pencil loosely.
- Use Developmental lines and the "crating" technique to keep your sketch properly proportioned.
- Only darken developmental lines when you are satisfied that they are correct.
- Keep your sketch neat!

Assessment Rubric	
Use developmental lines	1 pt
Use object lines	1 pt
Sketch proper proportion	1 pt
Sketch correct	2 pts
Sketch neat	2 pts
7-A 6-B 5-C 4-D	

Complete your sketch here:


ROBOTC 14

The Open End Wrench

Step 1, Step 2, Step 3, Step 4, Step 5: Add additional detail

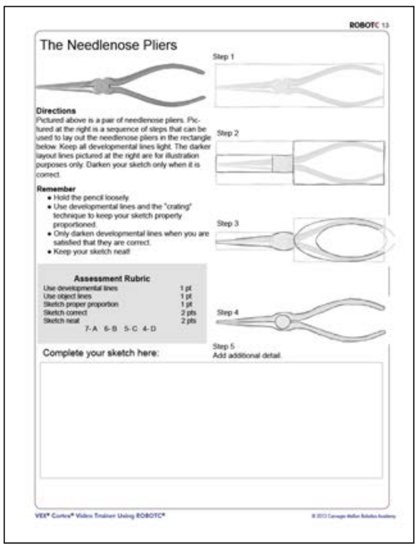
Directions

Pictured above is an open end wrench. Pictured at the right is a sequence of steps that can be used to lay out the open end wrench in the rectangle below. Keep all developmental lines light. The darker layout lines pictured at the right are for illustration purposes only. Darken your sketch only when it is correct.

Remember
- Hold the pencil loosely.
- Use developmental lines and the "crating" technique to keep your sketch properly proportioned.
- Only darken developmental lines when you are satisfied that they are correct.
- Keep your sketch neat!

Assessment Rubric	
Use developmental lines	1 pt
Use object lines	1 pt
Sketch proper proportion	1 pt
Sketch correct	2 pts
Sketch neat	2 pts
7-A 6-B 5-C 4-D	

Complete your sketch here:


ROBOTC 13

The Needlenose Pliers

Step 1, Step 2, Step 3, Step 4, Step 5: Add additional detail

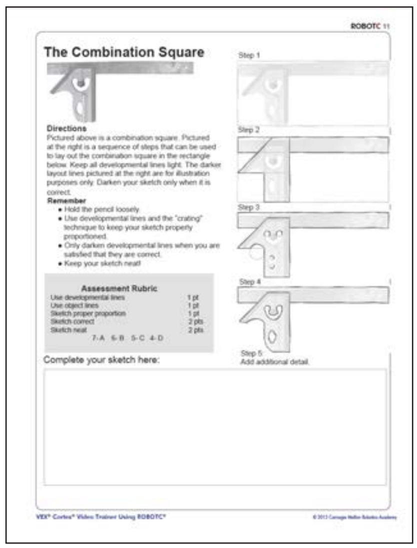
Directions

Pictured above is a pair of needlenose pliers. Pictured at the right is a sequence of steps that can be used to quickly sketch the needlenose pliers in the rectangle below. Keep all developmental lines light. The darker layout lines pictured at the right are for illustration purposes only. Darken your sketch only when it is correct.

Remember
- Hold the pencil loosely.
- Use developmental lines and the "crating" technique to keep your sketch properly proportioned.
- Only darken developmental lines when you are satisfied that they are correct.
- Keep your sketch neat!

Assessment Rubric	
Use developmental lines	1 pt
Use object lines	1 pt
Sketch proper proportion	1 pt
Sketch correct	2 pts
Sketch neat	2 pts
7-A 6-B 5-C 4-D	

Complete your sketch here:


ROBOTC 11

The Combination Square

Step 1, Step 2, Step 3, Step 4, Step 5: Add additional detail

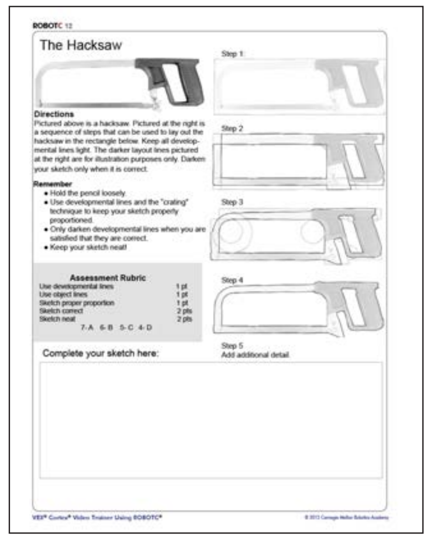
Directions

Pictured above is a combination square. Pictured at the right is a sequence of steps that can be used to lay out the combination square in the rectangle below. Keep all developmental lines light. The darker layout lines pictured at the right are for illustration purposes only. Darken your sketch only when it is correct.

Remember
- Hold the pencil loosely.
- Use developmental lines and the "crating" technique to keep your sketch properly proportioned.
- Only darken developmental lines when you are satisfied that they are correct.
- Keep your sketch neat!

Assessment Rubric	
Use developmental lines	1 pt
Use object lines	1 pt
Sketch proper proportion	1 pt
Sketch correct	2 pts
Sketch neat	2 pts
7-A 6-B 5-C 4-D	

Complete your sketch here:


ROBOTC 12

The Hacksaw

Step 1, Step 2, Step 3, Step 4, Step 5: Add additional detail

Directions

Pictured above is a hacksaw. Pictured at the right is a sequence of steps that can be used to lay out the hacksaw in the rectangle below. Keep all developmental lines light. The darker layout lines pictured at the right are for illustration purposes only. Darken your sketch only when it is correct.

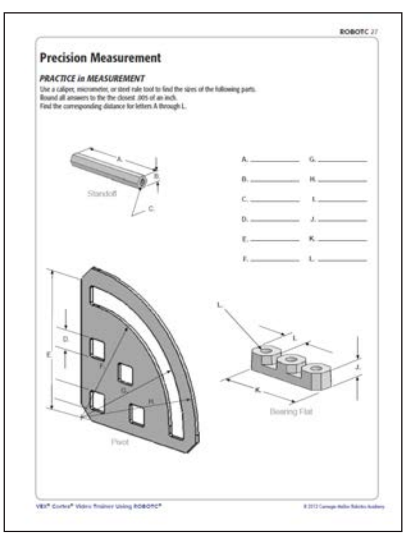
Remember
- Hold the pencil loosely.
- Use developmental lines and the "crating" technique to keep your sketch properly proportioned.
- Only darken developmental lines when you are satisfied that they are correct.
- Keep your sketch neat!

Assessment Rubric	
Use developmental lines	1 pt
Use object lines	1 pt
Sketch proper proportion	1 pt
Sketch correct	2 pts
Sketch neat	2 pts
7-A 6-B 5-C 4-D	

Complete your sketch here:

Precision Measurement

Sketching Fasteners


ROBOTC 27

Precision Measurement

PRACTICE in MEASUREMENT

Use a caliper, micrometer, or steel rule tool to find the sizes of the following parts. Round all answers to the the closest .005 of an inch. Find the corresponding distance for letters A through L.

A. ____ G. ____
B. ____ H. ____
C. ____ I. ____
D. ____ J. ____
E. ____ K. ____
F. ____ L. ____

Standoff

Pivot

Bearing Flat


ROBOTC 19

Screw Head Types

Directions: Sketch and label the appropriate views of the nuts and bolts below.

Nuts/Fasteners

Socket and Screw Head Drives

Assessment Rubric	
Use developmental lines	1 pt
Use object lines	1 pt
Sketch proper proportion	1 pt
Sketch correct	2 pts
Sketch neat	2 pts
7-A 6-B 5-C 4-D	


ROBOTC 18

Fasteners

What is the difference between a bolt and a screw?

What are the three properties that identify bolts?
1.
2.
3.

Sketch the following head types of bolts: (Front view)

When specifying a bolt or screw, what three bits of information must be included?

What does 1/4-20 mean?

Sketch the following drive head shapes. (Top view)

Describe the differences between a nylon insert locknut and a keps nut.

Describe the properties of a standoff.

NAME

DATE

VEX Robotics Competitions

VEX Robotics Competitions - Type Robotics Competitions into a search engine and you will have millions of hits. They are incredibly popular with over 30,000 US teams. The major advantages of competing in an annual robotics competition are:

- The competition game is new each year so your students won't get bored
- The rules are generally well thought out and developed by a team of experts
- There are hard deadlines that students need to meet.
- It is incredibly exciting for students to compete against other students from across the region and the world.
- There are multiple events that your team can compete in.
- Taking your school to these types of events ads prestige to your program.

The REC Foundation - Each year the Robotics Education and Competition (REC) Foundation hosts an International VEX competition, in 2012 there were over 7,000 teams. The Competition is incredibly well run and can provide a life-changing experience for students. There are competitions run in every state and in most countries in the world. To learn more go here: http://www.roboticseducation.org/

BEST Robotics http://www.bestinc.org/ BEST stands for Boosting Engineering Science and Technology. It hosts and annual competition each year and has about 1,000 teams. To learn more go here: http://www.bestinc.org/documents/Competition%20Overview.pdf

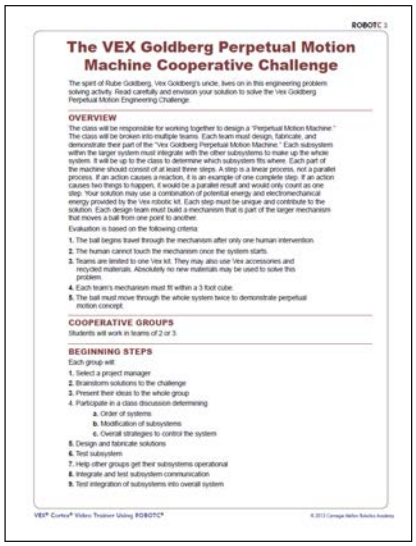
VEX In-School Design Problems

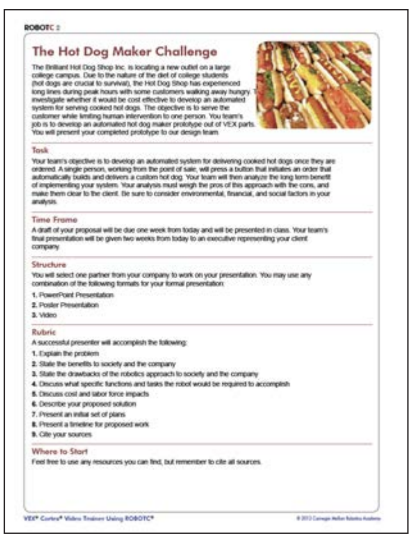
On the next page you will find handouts that you can use to introduce these three challenges. The challenges are described below and are designed to challenge students to develop innovative solutions using the engineering design process.

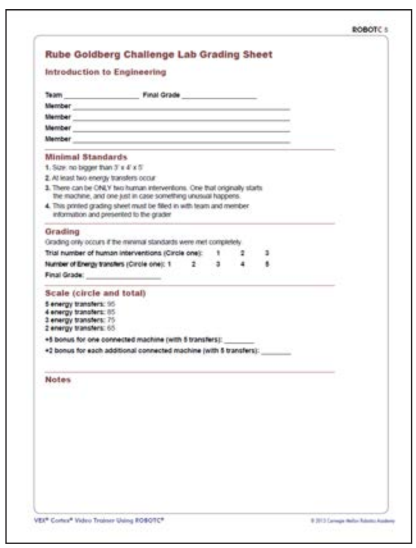
Project	Time Required	Experience Level	Other Requirements	Focus/Coverage
Rube Goldberg	1 - week	Introductory		**Focus**: Engineering process, teamwork, communication, systems concepts (input, output, subsystems, state), design **Coverage**: Non-Vex building of stationary "cells" in a system that can only perform their functions once, and do not have electronic control
Hot Dog Maker Challenge	3-5 weeks	**Recommended:** Students must know how to program with sensors	**Required:** Programming kit, various additional sensors, hardware and tools	**Focus**: Engineering process, teamwork, communication, collaboration, systems concepts, reliability **Coverage**: Inter-team collaborative design and Vex building of subsystems that must work together reliably to accomplish a multi-step process with dissimilar functions at each step. Project management on several levels is key to successfully completing this activity.
Automated Work Cell	4-5 weeks	**Required:** Students must know how to program with sensors	**Required:** Programming kit, various additional sensors, hardware and tools	**Focus**: Engineering process, teamwork, communication, collaboration, systems concepts, reliability **Coverage**: Inter-team collaborative design and Vex building of subsystems that must work together reliably to accomplish a multi-step process with dissimilar functions at each step. Project management on several levels is key to successfully completing this activity.

Notes on Projects:

1. Rube Goldberg Challenge – This challenge can be accomplished with or without VEX parts. Our intention with this design problem is to get the kids engineering early and have them develop an outside the box solution. We believe that the project can be done without VEX parts using all recycled parts. The project can be done at home as a homework assignment over the weekend and then brought to class. There are many examples of Rube Goldberg Machines on the Internet. There is also a handout below which uses VEX parts.
2. The Hot Dog Maker & Automated Work Cell – The nature and themes of these projects can be modified however the teacher sees fit (hot dog maker and work cell are just two ideas), and should be designed to fit available resources. The idea behind the projects is to challenge the students to develop an automated system that requires every group to be doing something different, yet integral, to the final solution. The solution to the system is up to the students.







Breaking Programs into Behaviors

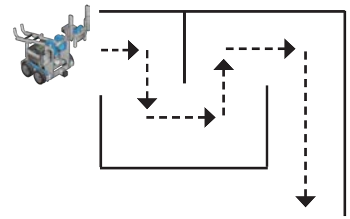
What Are Behaviors?

A behavior is really anything your robot does: turning on a single motor is a behavior, moving forward is a behavior, tracking a line is a behavior, navigating a maze is a behavior. There are three main types of behaviors that we are concerned with: complex behaviors, simple behaviors, and basic behaviors.

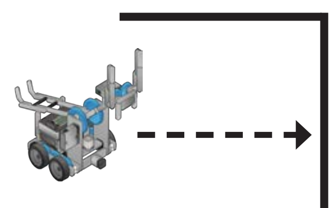
Complex Behaviors

These are behaviors at the highest levels, such as navigating an entire maze. Though they may seem complicated, one nice property of complex behaviors is that they are always composed of smaller behaviors. This means that if you observe a complex behavior, you can always break it down into smaller and smaller behaviors until you eventually reach something you recognize.

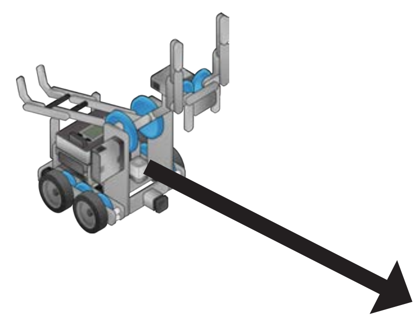
Complex Behavior:
Maze Navigation



Contains the Simple Behavior:
Move Forward until Touch Sensor is Pushed



Contains the Basic Behavior:
Turn on Left Motor in Forward Direction



Simple Behaviors

Simple behaviors are small, bite-size behaviors that allow your robot to perform a simple, yet significant task, like moving forward for a certain amount of time. These are perhaps the most useful behaviors to think about, because they are big enough that you can describe useful actions with them, but small enough that you can program them easily several lines of code.

```
setMotorSpeed(leftMotor, 50);
setMotorSpeed(rightMotor, 50);
sleep(2000);
```

The simple behavior directly above enables a robot to move forward for two seconds.

Basic Behaviors

At the most basic level, everything in a program must be broken down into tiny behaviors that your robot can understand and perform directly. These are behaviors the size of single lines of code, like turning on a single motor, or checking a single sensor port.

```
setMotorSpeed(leftMotor, 50);
```

Basic behavior - turn on one motor.

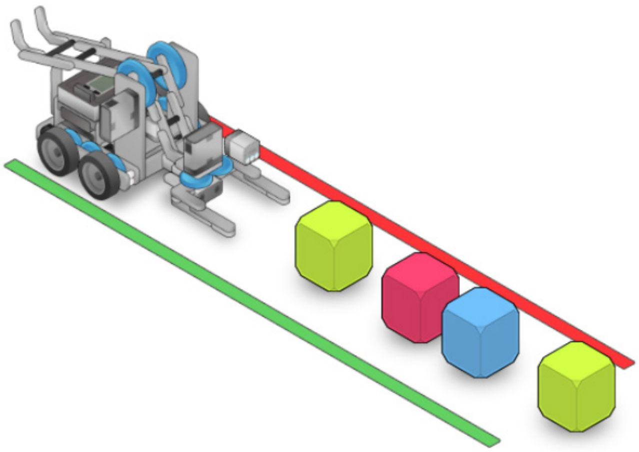
Exercises

1. What level of behaviors can your robot perform directly?
2. Why is it useful to think about a robot's actions in terms of behaviors?

Sense Plan Act

How does a robot think?

One easy to understand how a robot thinks is **Sense-Plan-Act**. A robot must be able to Sense its environment, Plan a course of action based on that data, and Act on that plan.

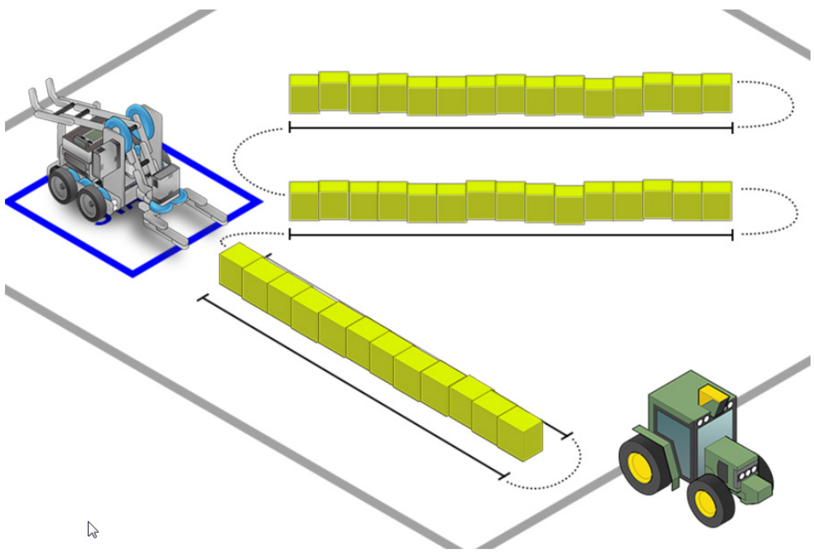


Sense

Using a variety of available sensors, the robot gathers data from its surroundings. Sensors include anything that provides the robot with information on its environment, such as the color sensor mounted on the robot in the picture, which will provide feedback about the color of the blocks in front of it.

Plan

The robot will process the information gathered in the Sense phase, and formulate an appropriate plan of action to react to what it saw. This step is most often performed by software (like your ROBOTC software) that has been loaded onto the robot in advance. The program illustrated here tells the robot to go forward until it sees a color.



Act

The robot acts in the world through the use of actuators– any component which allows the robot to create a change in its surroundings, such as motors, which move the robot through the environment. The robot in the picture will drive through the maze.

Answer the following questions

1. Define what a robot does.
2. Describe how your robot senses, plans, and acts to solve the challenge that you are working on.

Teaching Pseudocode and Flowcharts

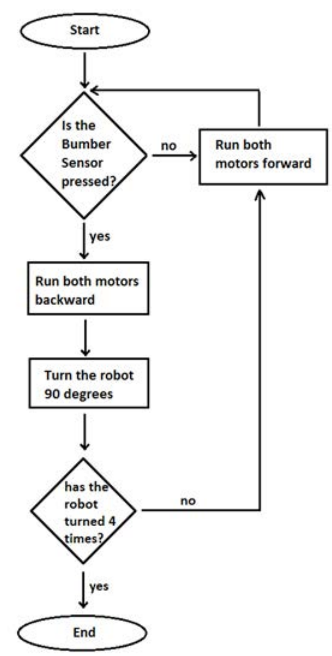
Introduction

Pseudocode - is a native language description of what the robot is required to do. With practice pseudocode eventually resembles ROBOTC code.
Flowchart - is a graphical representation of program flow.

It is a very good practice to have students begin each programming task by breaking the task into its smallest parts and develop pseudocode that describes the robot's behaviors. You will find much more detail on this practice in the *Introduction to Pseudocode lesson*.

Example pseudocode

Example flowchart



The practice of developing pseudocode is informally introduced on the first day of class when students are asked to write a program to control a humanoid robot to make a sandwich. See *Introduction to Robot Programming* for a full description of that activity. In that activity, students will not yet know a robot programming language, but they can begin to write pseudocode to describe the behaviors the robot will need to complete to make the sandwich. (E.g., lift the arm, open the hand, reach for the bread, grab the bread, etc.).

In this teacher's guide, we use the term *flowchart* in a very general way to convey that students are constructing visual representations of the problem and its solution. The advantage to flowcharts is that they help a student identify robot decision making.

This curriculum doesn't formally introduce flowcharts until the Remote Control Chapter because the first chapter, Movement, only require simple behaviors (movingForward, turning, moveUntil an encoder count). During these activities, students will learn the basic lexicon of the programming language allowing them to be begin writing robot behaviors using pseudocode. Teachers should encourage students to use pseudocode - even if it initially starts as common descriptions and slowly improves to reflect a more accurate programming language. Pseudocode will eventually be used to describe individual parts within a flowchart since each pseudocode phrase describes a unique robot behavior.

In the Remote Control Chapter, students will begin programming their robots to solve problems that contain multiple steps with decisions, that is when this curriculum formally introduce flowcharts. With practice, students will begin to see that flowcharts provide a visual representation of the robot's decision making process and then begin using flowcharts to solve their programming challenges.

Introduction to Pseudocode

Overview

In order to plan a program and write efficient code students need to be able to write clear instructions for the robot. This lesson introduces students to the idea of writing clear instructions and then introduces them to pseudocode.

Objectives

Students will be able to:

- Listen carefully and follow instructions
- Communicate clear instructions
- Break tasks down into smaller pieces
- Write pseudocode for a simple maze
- Understand the necessity of planning clear steps

Materials

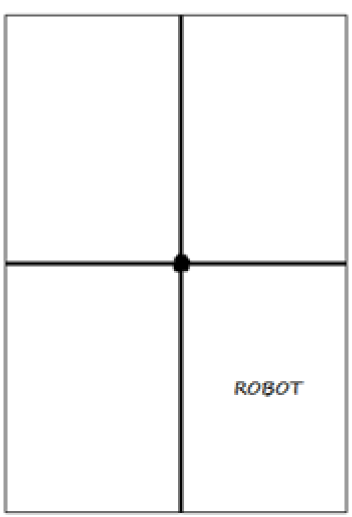
- Blank paper for each student
- Pencil and a ruler for each student
- A large table top or floor surface to setup a simple maze
- Tape to mark out maze boundaries
- Small box or object to represent a robot

Clear Communications Lesson

Procedure

1. Introduce the following "Clear Communications" drawing activity. Tell students:

Robots will follow the program that they are given, even if that program doesn't make sense. The robot needs to be given a program that it can understand to produces the desired outcome. Let's begin with a drawing activity to help us think about this idea of giving clear instructions.



Have all students begin with a blanks sheet of paper and then give them the following verbal instructions:

1. *Draw a dot in the center of your page*
2. *Draw a vertical line from the top of your page to the bottom of your page, passing through the center dot*
3. *Draw a horizontal line from the top of your page to the bottom of your page, passing through the center dot*
4. *Write the word ROBOT center of the square created at the bottom right of your page*

Have students hold up their drawings when everyone is finished. Check to see that everyone's is the same. Discuss any differences if they arise.

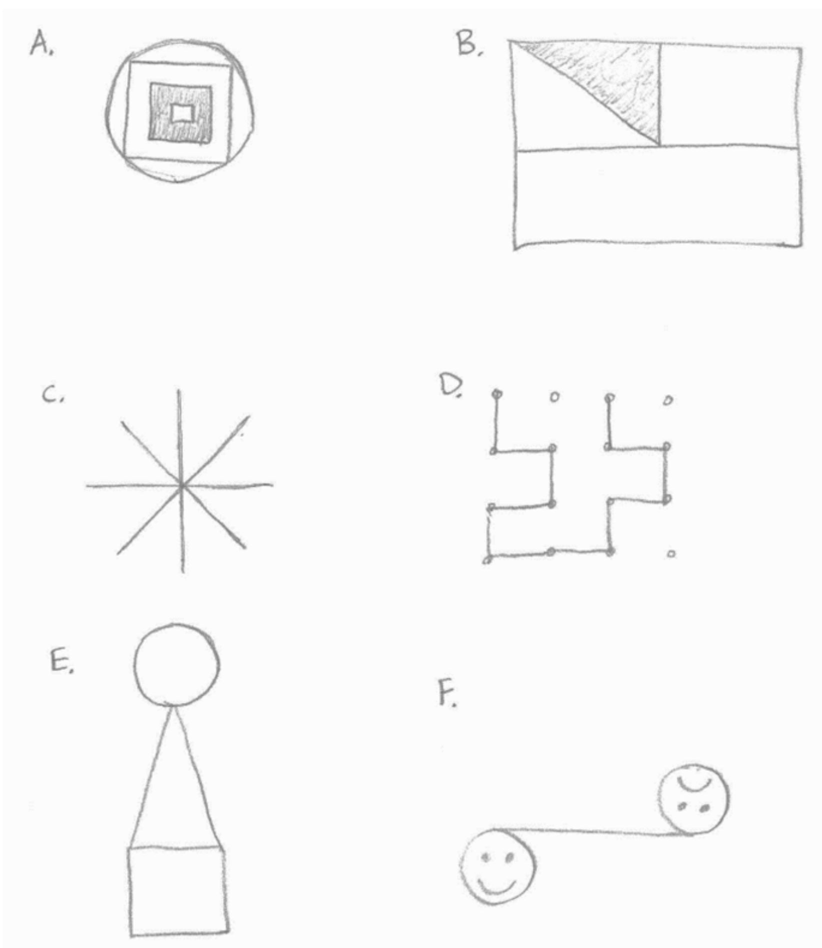
Introduction to Pseudocode

Introduction to Clear Communications

Choose a student to be the communicator and give them a simple drawing (see examples at the bottom of this page) making sure no one else sees the picture that you've given the student communicator. Have the communicator describe the drawing for the others to reproduce on their paper. The other students may ask questions for clarification and the communicator may adjust their instructions if they see mistakes being made. Encourage students to see how quickly and accurately the picture can be reproduced – each drawing should not take more than 5 minutes.

Repeat the exercise with a new picture and a new communicator student. This time the other students are not allowed to talk or ask any questions, but the communicator may still adjust their instructions if they see mistakes being made.

Repeat the exercise one last time with a new picture. Now the communicator must sit behind a screen or with their back turned and no questions may be asked. The communicator must clearly give their instructions one time for the others to reproduce the drawing.



Discuss this last exercise with the class.

Ask the students who had to be the communicator:

How did you decide the order in which to give instructions?

How confident were you that your instructions would work?

Ask the students who were reproducing the drawing:

Did you find the instructions easy to follow?

Was there anything you drew differently because the instructions were unclear?

What makes a good set of instructions?

[Emphasize this question, and encourage students to conclude that the instructions need to be broken down into simple steps because instructions that are too big can be confusing and lead to error.]

Introduction to Pseudocode

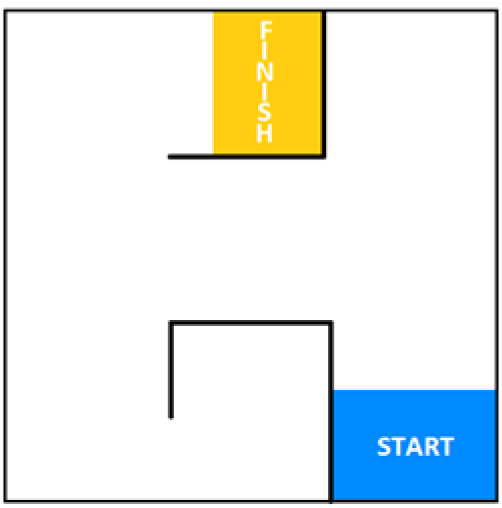
Introduction to Robot Programming

Tell students: *This last exercise is most like programming robots – the programmer (or communicator) gives a set of instructions which will be followed exactly. The instructions need to be broken down into the simplest possible pieces so that they can be performed accurately and without confusion.*

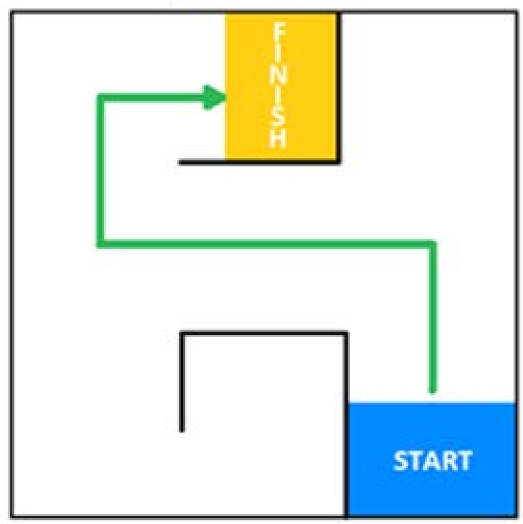
Tell students: ***Behaviors** are a very convenient way to talk about what a robot is doing and what it must do. Every action of a robot can be described as a behavior: move forward, turn on a motor, look for an obstacle, stop, and solve a maze are all examples of robot behaviors.*

Direct students to the simple maze shown below on the left. The goal is for the robot to solve the maze by following the path below on the right. Tell students: *To do this, we will think about the robot's actions in terms of behaviors.*

Simple Maze

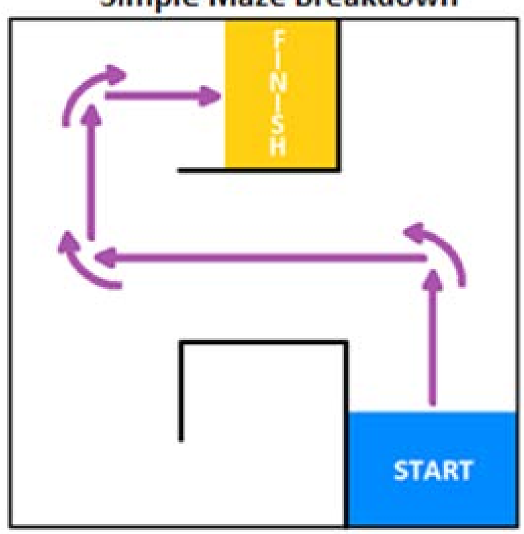


Simple Maze Solution



Give a student a robot. Have a student volunteer move the object along the path that would solve the maze. Point out that we can easily see what behavior the robot needs in order to solve the maze.

Simple Maze Breakdown



Tell students:
Some behaviors are too big to give to the robot as instructions, so we need to think about breaking them down into smaller behaviors. "Solve the maze" is actually a very complex behavior because it involves many steps and the robot wouldn't know what to do.

Introduction to Pseudocode

Instruct students to:

Write down the behavior “Solve the maze” on the top of your paper.

Now, tell them to:

Break down that behavior into smaller behaviors and write them below in order.

If students are unsure, ask:

What does the robot need to do to follow the solution path?

(Move forward, turn left, move forward, turn right, etc.)

Ask students:

Is your list of behaviors is clear enough to instruct the robot through the maze.

They should realize that we are close, but the robot doesn’t know how far to move forward each time or how much to turn.

Finally, instruct students to:

Use your rulers to figure out the distance the robot needs to move for each behavior.

(Exact distances will depend on your maze setup).

Write this new specific behavior next to their last list of behaviors.

We now have a list of behaviors specific enough to give as instructions to the robot.

Tell students:

By starting with a very large solution behavior and breaking it down into smaller and smaller sub-behaviors, you have a logical way to figure out what a robot needs to do to accomplish its task.

Talking about and writing the code in English is the first step in good pseudocode practice, which allows us to plan robot behaviors before we translate them to code.

Example Pseudocode

Solve the Maze

Move forward ---> move forward 60 cm
turn left ---> turn left 90 degrees
move forward ---> move forward 80 cm
turn right ---> turn right 90 degrees
move forward ---> move forward 60 cm
turn right ---> turn right 90 degrees
move forward ---> move forward 50 cm

Pseudocode Exercise

What is Pseudocode?

Robots need very detailed and organized instructions in order to perform their tasks. Before a programmer can begin programming they need to break a robot's behaviors down into simple behaviors and figure out when each behavior should run. Some programmers like to use pseudocode to begin constructing the programming problem.

pseudo

adj : not genuine but having the appearance of;

Source: WordNet ® 1.6, © 1997 Princeton University

Pseudocode is a hybrid language, halfway between English and code. It is not real code yet, but captures the details that will be important in translating your ideas to code, while still allowing you to think and explain things in plain language. Good pseudocode will make it very straightforward to write real code afterwards, because all the behaviors and logic will already be contained in the pseudocode.

Pseudocode example

If you wanted to program a robot to stop when it saw and object and move forward when it didn't see and object your pseudocode might look like:

pseudocode

1. Move forward
2. If (sonar sensor detects and object)
   stop
3. When the sonar sensor no longer
   sees and object move forward.
4. Do this forever

Exercise

1. Convert these instructions to pseudocode and into a flowchart:
   a. "If it's raining, bring an umbrella."
   b. "Keep looking until you find it."
   c. "Take twenty paces, then turn and shoot."
   d. "Go forward until the touch sensor (on port 1) is pressed in."
   e. "Turn on oven. Cook the turkey for 4 hours or until meat thermometer reaches 180 degrees."
   f. "Crossing the street" Hint, make sure that you look both ways!

2. Compare the advantages and disadvantages of flowcharts and pseudocode.
   Explain in your own word why you believe one is better than the other.
   Is one of them always better than the other, or are both good in different situations?
   Can you use both to help solve the same problem? Should you?

Introduction to Flowcharts

Overview

A flowchart is a visual representation of program flow and is used by programmers to break down and model robot behaviors. This lesson provides teachers with a guide to introduce flowcharts in the *Remote Control Chapter*, and then this lesson should be applied to all of the subsequent Units. Students have already been introduced to pseudocode and program flow in the Introduction to Robot Programming activity; this lesson focuses on how to graphically describe robot decision making. In this lesson the teacher will model how integrate pseudocode and decision making into a flowchart.

Objectives

Students will be able to:

- Break down a problem into simple components
- Organize the components into a proper sequence
- Represent a sequence of behaviors in a flowchart
- Use a flowchart to analyze robot behaviors

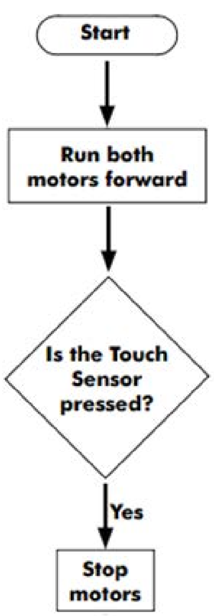
For this lesson, you will need to have programmed a robot that moves forward until the touch sensor is pressed and have it ready to demonstrate to students.

Procedure:

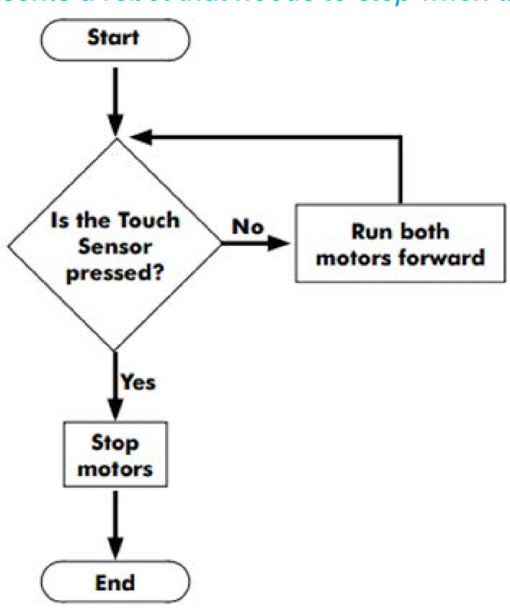
In the example below students will see a program that requires the robot to make a decision and includes multiple steps. Note: Show all students the picture of the flowchart 1 and flow-chart 2 below.

Tell students: *Now that the robot is using a sensor to make decisions, we can use a flowchart to understand how our robot's behavior is broken down into steps within a program. So far, our flowcharts have just been single steps of pseudocode, but now we will need to add decision blocks which ask a "yes or no" question.*

Show both of the following flowcharts to the class and ask:
Which flowchart best represents a robot that needs to stop when the touch sensor is pressed?



Flowchart 1



Flowchart 2

Draw the example flowcharts on the left on the board or project them onto the screen for students to see and discuss.

Introduction to Flowcharts Page 2 of 3

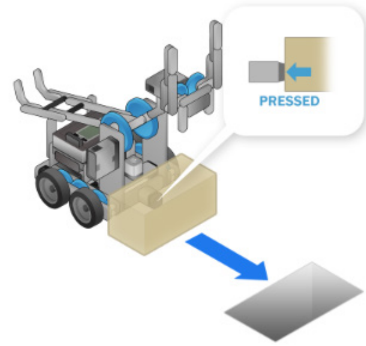
If students struggle or disagree, ask:

When does the robot stop moving forward? (When the touch sensor is pressed in.)

When does the robot check the touch sensor to make the decision to move forward or stop?
(The robot continually decides to move forward when the touch sensor is not pushed, and decides to stop when the touch sensor is pushed)

Which flowchart represents a robot continually making decisions based on the bumper sensor?
(The flowchart on the right requires the robot to keep asking if the touch sensor is pressed)

Read through Flowchart 2 (on the previous page) to the class to show how each robot action is represented in the flowchart. Teach students the difference between start/stop, action, and decision blocks. Tell students:
The robot must continually check the value of the touch sensor to decide what to do. If the touch sensor is not pressed, the robot continues running both motors forward. If the touch sensor is pressed, the robot stops all motors. The flow of these decisions is given in the flowchart 2 on the right, but missing from flowchart 2 on the left.



Imagine a scenario where the robot's touch sensor is pushed in while it is pushing a box off of the table. (Pictured at the right) When the box is pushed of the table, the robot's touch sensor is no longer pressed.

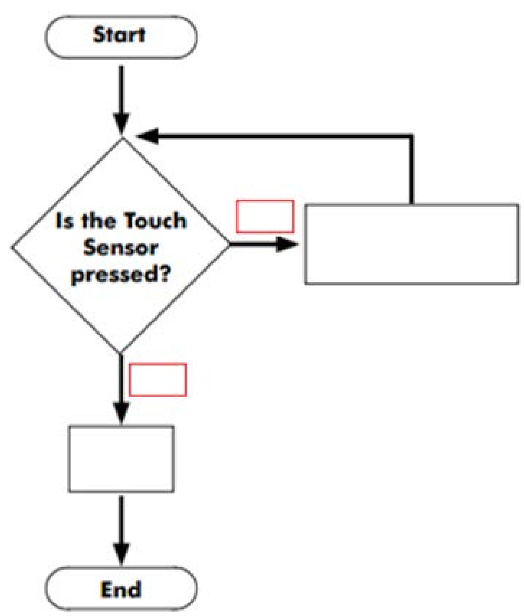
Refer to flowchart 2 as the class thinks about the following question:
What happens if you start the program with the touch sensor pressed in?"

Give them a moment to think, and then ask students to explain their answer to the class using a flowchart. Tell the students: *Use a flowchart to break down each step of the robot's behavior so that we can see what actions the robot will perform in any situation.*

See the flowchart 3 at the right. Have students copy it onto their papers and fill in the blanks to properly represent program flow while a robot is pushing a box until its touch sensor is released.
If students struggle, ask:
Now when will the robot stop moving forward?
(When the touch sensor is released)

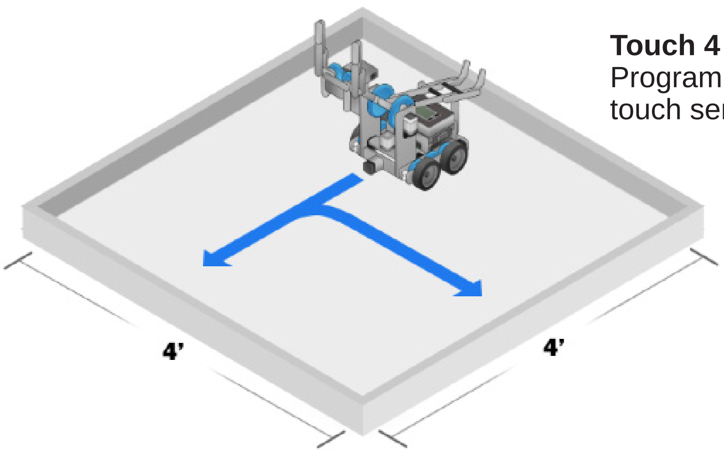
When students have successfully filled out their flowcharts, ask the class:
What would happen if the box was not all the way against the touch sensor when the program started?

(The robot would not move)
Use a flowchart to show this pathway.

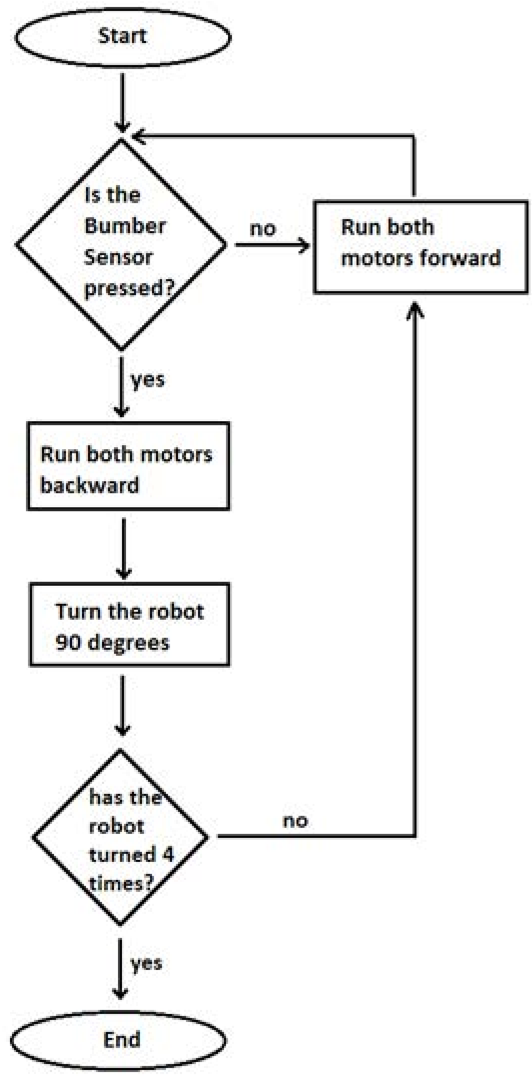


Flowchart 3

Introduction to Flowcharts



Touch 4 Walls Challenge
Program the robot to touch all four walls using the touch sensor to know when it has reached a wall



Pictured above is the solution for the Touch 4 Walls Challenge

See the challenge pictured above. On a new piece of paper, have students design a flowchart to represent the robot's behavior in this challenge. The robot will need to perform more actions after the bumper sensor is read each time.

Give the students several minutes to work on their flowchart. Have them work in small groups if necessary.

If students struggle, ask:

What does the robot need to do after the bumper sensor is pressed? (Backup and turn towards another wall)

Does the robot ever need to repeat its behaviors?
(Yes, it will have the same behavior for all 4 walls)

How will the robot know when it has contacted all the walls? (When it has repeated its behavior 4 times)

After students have finished, ask a group to draw their solution for the class. Follow the flow and check the logic for any missing steps or problem areas.

Ask if other students created a different flowchart they would like to share.

Discuss aspects of each solution until the class comes to an agreement on a correct flowchart. It's possible to have multiple correct solutions.

Tell students:

Flowcharts help us understand the decision making process of a robot. The answer to a "yes or no" decision will determine what action the robot does. By creating a properly organized flowchart, we can see the plan a robot needs to follow to be successful.

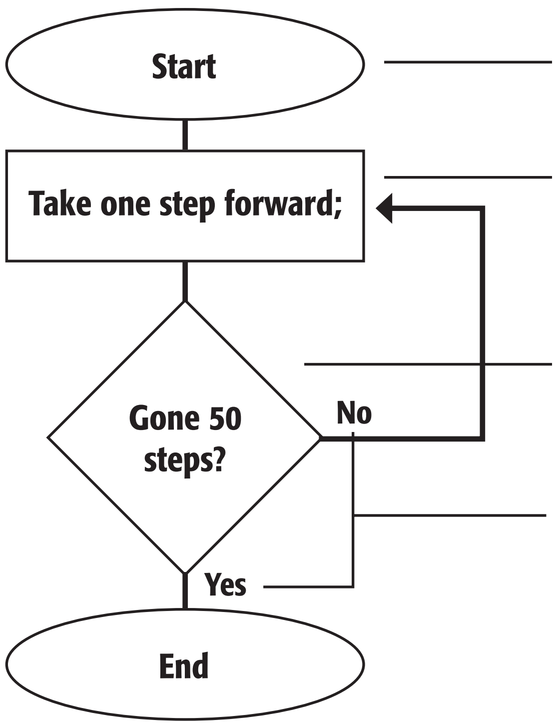
Flowchart Exercise

What are Flowcharts?

Robots need very detailed and organized instructions in order to perform their tasks. The programmer must break things down into simple behaviors and figure out when each behavior should run. A flowchart is a tool that can be used by programmers to determine program flow.

A flowchart provides a way of visually representing and organizing individual behaviors and decisions within a program -- it provides a diagram of the "flow" of the program. Programmers use flowcharts to lay out the steps that will be needed in their final program, and to help determine how the robot's behaviors should be broken down.

Parts of a Flowchart



Start of Program - Marks the beginning of the program, begin here. Follow the line to get to the next block.

Statement Block - A statement to execute, or a behavior to perform.

Decision Block - A decision point in your program. Ask a simple question, and do different things depending on the answer.

Yes/No - Answers to the question posed in the decision block. Follow the line labeled with the appropriate answer.

End of Program - Marks the end of the program. If you reach this point, the program is done!

Exercise

1. Make a flowchart organizing the "flow" of getting ready to go to school in the morning. Be sure to include the following steps in your chart, but don't be afraid to add other things if you need them!

Select something to wear	Look for your shoes	Put your shoes on
Take a shower	Brush your teeth	Hit snooze button
Eat breakfast	Put toast in the toaster	Get dressed
Walk or get a ride to school	Check your alarm clock	Comb your hair
Get out of bed	Turn on shower	Check the time

Introduction to Robot Programming

Programming is Precise

One of the big ideas of the Movement Chapter is that "programming is precise". It will be important for students to change the way that they think about providing directions. The following activity will highlight the attention to detail required for students to become successful programmers.

The Humanoid Sandwich Programming Challenge

This activity requires students to program a human robot to make a two ingredient sandwich. This activity provides students with an opportunity to practice breaking down what appears to be a very simple task into its smallest parts. Students will quickly learn that they need to use very precise commands when programming robots. Making a two-ingredient sandwich (plus bread) is presumably familiar to most students, and that experience will allow them to think critically about the behaviors needed to program a humanoid robot. Initially, the majority of students will not be prepared to break this task down to a level of detail necessary for robot programming.

Procedure

The teacher says:

Your task is to program a humanoid robot to make a two ingredient sandwich. Begin by making a list of behaviors your robot needs to perform make a sandwich, then turn that list into pseudocode. Pseudocode is a simple set of instructions that you want the robot to execute. We will test your "code" by programming a human to execute your sandwich making program.

Students can begin writing their program individually or in small groups.

Identify one student to assume the role of the robot, they will stand in a place where everyone can see her/him. The props for making the sandwich can be actual bread, spreads/meats, and a utensil - or they can be stand-ins (paper for bread and/or meats, a pencil for a knife, cups for jars of spreads). The "robot" (student) will do *exactly* what fellow students say to do.

The teacher can ask the class for the first behavior to be programmed, followed by the second, third, etc. Most times, it quickly becomes apparent that students have not fully considered the level of detail required for programming. For example, the command "pick up a slice of bread" is inadequate. The robot needs to know in which direction to move to get the bread, how to detect the bread (use of sensory data), how to pick it up, etc. Those are each individual lines of code. It is not important to continue this exercise once students recognize why greater detail is required - even for a task as simple as making a sandwich.

Additional Lessons

This teacher's guide provides multiple additional lessons that the teacher can use to introduce students to pseudocode and flowcharts. It is up to the discretion of the teacher to use or not use these lessons. For additional lessons go to the Index and find:

- Pseudocode and Flowcharts
- Introduction to Pseudocode
- Introduction to Flowcharts

Teaching How to Troubleshoot Programs

Stop, Trace, Analyze, Revise (STAR)

All teachers want students to learn how to troubleshoot their robot problems. There are times when a teacher will quickly give a student the answer, but this doesn't teach them how to troubleshoot their own programming problems, it teaches dependency. If a teacher immediately answers all student questions, the student quickly learns that all they have to do is ask the teacher and their problem will be solved. We recommend the STAR approach to teaching troubleshooting, this approach is designed to teach students stop and trace the program flow, analyze where there is a difference between what should happen and what happened, and then revise and test the program.

Teacher's Role in STAR Troubleshooting

Stop and Determine Student Intent The teacher needs to determine what does the student think is happening in the program. The teacher needs to ask the students to:

"*Please describe step-by-step what is happening in your program*".

Trace Together, the student and teacher, need to trace through the program and identify where the robot's behavior diverges from the student's intent.

Analyze The teacher needs to help the student to analyze what they misunderstood.

Revise Help the student to correct their misunderstanding and to fix their program. Often, correcting a student's misunderstanding means re-framing the problem or highlighting some discrepancy in how the student sees the problem.

STAR - Student's Role in Troubleshooting Programs

Stop and Reflect What do I think should be happening in each step of my program?

Trace I need to trace through the program step-by-step and identify what my program is telling the robot to do. If it is a complex program, then I should develop a flowchart that allows me to see my program flow.

Analyze If the robot is misbehaving, I need to analyze what parts of the program works and identify the point where the robot stopped doing what I wanted it to do. Then I need to figure out what I need to change to fix the problem

Revise I need to scientifically correct the program one step at a time and test each part of the program and fix the problem.

Note: Teachers might choose to post the student version of STAR somewhere in their classrooms and direct students' attention to it as they begin asking for assistance.