

Introduction

Why should I do robotics with my middle school students?

One of the most frequent and widely supported calls for change in contemporary education is for newly envisioned curriculum that integrates math, science, and technology content and skills. Although robotics may at first seem exotic, on closer examination it provides a natural fit with good pedagogy that emphasizes meaningful problem-based learning, integration and application of knowledge, hands-on learning in cooperative groups, and demonstrable mastery of new learning. It attracts students to content domains such as mechanics, electrical circuits, and applied mathematical reasoning that all too often are stumbling blocks for middle school students. Robotics engages students in complex, strategic problem-solving and higher-order thinking—a set of skills that is a high priority for 21st century education. What is more, this kind of problem-solving can be introduced in a gradual, self-motivated way, so beginner students can experience satisfying achievements right away and can quickly move on to new challenges in a continuous progression toward greater levels of sophistication.

Perhaps the most compelling reason for a robotics curriculum is that it introduces students to knowledge, concepts, and skills that are needed for understanding the intelligent information-based technology of the future: technology that is highly interactive, multi-modal, adaptive and autonomous. Technology is no longer the private domain of a self-selected group of “nerds,” and sophisticated technological systems are no longer reserved for academic research labs or the military or government. Indeed, you and your students are likely to have computers sitting on your desks that far exceed the capacity of the most sophisticated systems in the world just a generation ago. It is hardly controversial to predict that technology systems will be more and more ubiquitous—not just in school and work settings, but in home and recreational lives as well, for people from all walks of life. We can also predict that technology will be increasingly complex and intelligent. Both the capacity of hardware and the capabilities of software expand at a fierce pace. Artificial intelligence is no longer the stuff of science fiction, but everyday reality.

Further, the outward appearance and ways a user interacts with new technology systems are more and more variable. Even today, information technology doesn't just look like a desktop computer that you sit in front of and type on. To be productive in employment and higher education, today's students will have to be able to learn new technology systems on their own, on demand, and embedded in different kinds of situations. Surface-level familiarity with current applications (which is often the form that technology education takes in schools) won't be sufficient. Students need a deeper foundation of conceptual understanding of information technology systems and productive working strategies for self-learning and trouble-shooting.

Most middle school students today are enthusiastic users and consumers of information technology. They hardly remember a time when there was no Internet, e-mail, cell phones, electronic toys, or digital media. Indeed, while the adults in their lives may struggle to adapt to the influx of increasingly sophisticated information technologies, students take it for granted. Our objective in developing a curriculum that emphasizes active problem-solving with intelligent technology is to put students in the role of creating and designing with information technology—not just using and consuming it.

This curriculum was also motivated by a particular concern for addressing the “digital divide” in American education. While fields such as business, law, medicine, and the life and social sciences have made dramatic gains in recent decades in the participation of women, minorities, and students from families lacking a history of higher education, the fields of computer and information science, engineering, and the physical sciences continue to lag behind. Of particular concern is evidence that many students develop attitudes and preferences and begin to make educational choices during middle school that have later consequences for their ability to explore and be successful in educational and career paths in these areas. That is, before they really have a chance to explore these exciting new interdisciplinary domains—many of which have very promising career paths—they have already withdrawn from the science and math courses that would keep them open as options. In many cases, this withdrawal is accompanied by an unfounded belief (unfortunately shared and reinforced by many adults) that they lack ability and that only geniuses can be successful in these domains. They move forward with an unnecessary handicap—and a dangerous one, because even if they do not choose to be science, engineering, or technology professionals, as productive citizens and workers, they will nevertheless need a reasonable degree of fluency with intelligent technology systems to be effective decision-makers, leaders, and problem-solvers.

With support from the National Science Foundation’s Directorate for Education and Human Resources (specifically the Program for Gender Equity), this curriculum has been designed and evaluated to make sure that it provides a friendly, motivating introduction to intelligent technology with which ALL middle school students can engage and be successful. It has been field-tested with girls and boys, urban and rural students, gifted and talented students, and students in academic distress. We’ve been inspired and gratified by their enthusiasm for robotics and their creative and skillful accomplishments. We hope that you will be, too.

How do I introduce robotics to my students?

There are several different models you might consider for introducing robotics to your students. We have tested this curriculum with all of them, and this guide includes suggestions for modifications to suit your circumstances. If you want to do robotics as part of your in-school curriculum, you might choose to do it as a unit in a science or technology class. The curriculum includes activities for approximately 15-20 class sessions, depending on how long your periods are, the size and level of your class, and which variations you choose. Many of our teachers prefer to do robotics for two consecutive periods (totaling about 1.5 hours) when they have the flexibility to do so. Longer working sessions are especially convenient when students start to build more complex robots or to solve problems that require multiple rounds of programming and testing. Some teachers may choose to coordinate classes and co-teach robotics. For example, students may build robots during a science class period and then program them when they are scheduled in a computer lab for technology classes. Another successful model is running a robotics elective course for one or two quarters. Finally, either as a way to try things out for the first time or as an ongoing supplement to the daily science, math, and technology curriculum, you may consider organizing an after-school robotics club and using the curriculum in a more informal way. This may also be a good option if you have difficulty gaining access to a sufficient number of computers during the regular school day. As you consider your options, keep in mind group size, the layout of your physical space (e.g., sufficient table or desk space for building and an area with a smooth floor for running robots), storage space for kits and work in progress, access to computers (PCs running Windows 98 or ME), and the availability of assistants (see more on this below). Most of the

activities assume that the students have ready access to computers running the Mindstorms™ software.

What is in the Agents for Change curriculum package?

This unit has six complementary components:

- Teacher's Guide
- Student Activity Sheets
- Building Instructions
- Teaching Transparencies
- Quick Start
- Technical Guide for Mindstorms™ Robotics Invention System 2.0
- Program Library

Together, these comprise a complete field-tested curriculum package specifically designed to introduce fundamental information technology and engineering concepts, skills, and strategies to middle-school students.

This book, the Teacher's Guide, contains an integrated set of lesson plans, background information, sample programs, and materials lists—everything you need to plan focused, effective lessons for middle school students. This is your primary planning tool, and the other elements of the curriculum provide supplementary support. The activities presented in this Teacher's Guide are designed to be flexible, and often you have a choice of whether to introduce an activity to the whole group as a teacher-led demonstration or group lesson or to have the students work more independently in small groups, with you taking the role of coach or facilitator. In many situations, it makes sense to alternate between these modes. For example, you may want to gather the class together to introduce a new programming concept using overhead transparencies before dividing into small groups, or you may move to a whole-class format to recap certain key content ideas or to compare different solutions to the same problem after groups have worked on their own for awhile. You are the one who best understands your students, your classroom situation, and constraints such as schedules, computer availability, and so forth.

The activities have a sequenced thread running through them, in that students begin by building and programming simpler robots, then become familiar with sensors, and gradually master more complicated programming concepts. However, it is not necessary to follow a strictly linear order through the curriculum. For example, you could introduce touch and light sensors in either order. Similarly, you might want deliberately to structure an activity around introducing all of your students to "Repeat" commands or you might suggest it more casually to particular groups as it seems appropriate to the problems they are working on. The Table of Contents gives an overview of the activities in the curriculum in a flow chart format that you can use to plan a sensible sequence for your students. As students develop a larger repertoire of knowledge and experience, they will naturally want to work with more complex designs (for example, adding various attachments to their robots). As their programming skills increase, you can present them with appropriately challenging problems and projects. Some students will also ask to initiate their own projects, as their imaginations are primed by their increasing understanding of how their robots work and how they can control them.

In the activities that follow, each one starts with an introductory section titled *Before You Begin*, which states the explicit objectives of the lesson, outlines the new content

introduced in the activity and the general “flow” of the lesson, lists the materials that will be used, and describes anything that must be prepared ahead. Because this curriculum emphasizes hands-on investigations and problem-based learning, there is no highly scripted lesson. Different teachers will make different choices about how to match the activity to their circumstances. However, the Teacher’s Guide provides rich discussions and focused examples of the content to be conveyed in each activity, issues to anticipate in students’ learning, and specific strategies for addressing them.

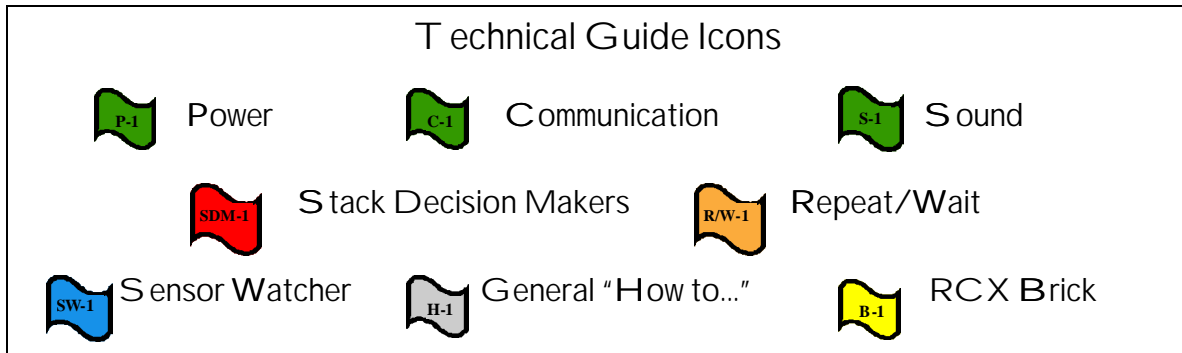
The Discussion Points section offers detailed advice on various ways to implement the activity (e.g., in a teacher-led lesson or in small groups), along with clear explanations of how to work through important demonstrations or teaching examples with your class. This is followed by a Programming Highlights section, which uses a two-column format. The left-hand side highlights and explains the main programming concepts that should be explored by the students in each activity (e.g., new commands that are introduced, which are also flagged by icons for easy identification and as pointers to the Technical Guide). The right-hand side addresses issues related to students’ learning, pointing out common confusions or places where you might want to do some particular facilitation with students. Each point raised in this column as something you should “be on the lookout” for is accompanied by a suggestion for what you can do to help students through the learning. In many cases, you will find a reference to the teaching transparencies that accompany this curriculum. These transparencies are designed to help you present new programming concepts, to show examples of programming solutions to various problems, and to draw out some of the more subtle, non-intuitive, or powerful features of the Mindstorms™ programming environment. The Teacher’s Guide reproduces miniatures of the actual transparencies and includes additional notes about how to use them. The color transparencies are available in a separate packet.

Activities in the Teacher’s Guide generally conclude with a Going Independent section, which includes ideas and sample solutions for problems students can solve using the new concepts and strategies introduced in the main activity. Although these independent problems are optional and different students may choose to work on different problems, this general type of independent problem-solving is an important part of your students’ learning process. The independent problems provide an opportunity for students to extend and consolidate their learning, to practice teamwork skills, and to apply their understanding in increasingly complex situations. Solving the independent problems (some of which are fairly challenging) can also be a source of satisfaction and motivation and an inspiration for creativity for your students. The suggested problems can generally be solved in more than one way, and it can be a rewarding classroom experience to have students present, explain, and analyze their unique solutions. You should also be open to letting students propose their own problems to work on, especially as they gain more control of the programming environment. If you have small groups that are moving at different rates, you can use the independent problems to give faster or more advanced groups additional challenges to work on while other groups complete the main activity.

Many of the activities call for a Student Activity Sheet, which is what students working in small groups will use to plan and carry out their work. Activities in the Teacher’s Guide and in the set of Student Activity Sheets are identified by the same name, to make it easy to match them to each other. The Student Activity Sheets are much less detailed than the Teacher’s Guide and are appropriate for a typical middle-school reading level. One of the activities, M. Pathfinder, uses a custom set of Building Instructions. Other building instructions can be found in the Constructopedia that is included in each

Mindstorms™ kit. Activities include the page numbers for easy reference to building instructions in the Constructopedia.

The spiral bound Technical Guide is an easy-to-use quick reference. It explains how to do routine tasks in the Mindstorms programming environment; how to understand and use the RCX brick (the on-board computer); how the most important and frequently used Commands work; and how to use Stack Controllers and Sensor Watchers in your programs. Both the Teacher's Guide and the Student Activity Sheets have icons flagging the relevant sections of the Technical Guide, marking the section and page number.



What do I need to get started?

All of the major national standards for science and technology education (e.g., the *National Science Education Standards*, *Benchmarks for Science Literacy*, and *Standards for Technological Literacy*) and most state and local standards now strongly advocate active hands-on curricula in the sciences, math, and technology. This kind of pedagogy requires materials and logistics that go beyond a text, paper and pencil, and blackboard. The Agents for Change curriculum is a hands-on, minds-on curriculum that involves students directly in building their own robots, writing their own programs, and pursuing their own investigations and design problems. We are fortunate to be able to recommend affordable, readily available, user-friendly robotics kits called the Mindstorms™ Robotics Invention System, developed by Lego. Although Mindstorms™ was designed as an informal educational product for children as young as 12, it is actually a very flexible and powerful system, and some colleges and universities use it in their introductory robotics courses. Although younger (and older) students generally enjoy playing around with the kits, in our experience, most of them do not discover or understand the more powerful features and potential of the hardware and software in the kit on their own. When used with an appropriate curriculum and the support of an informed teacher, students can have powerful learning experiences in the Mindstorms™ environment that meet many of the technology, engineering (design), and physical science objectives of the most recent generation of standards.

With the exception of the first activity (the Wall Hugging Mouse), the activities in this curriculum and the supporting technical resources have been designed for use with the Lego Mindstorms™ Robotics Invention System, version 2.0, which can be ordered directly from Lego or purchased from many computer centers and some toy stores for about \$200 per kit. (Please note that there is a family of products in the Mindstorms™ line; this curriculum is specific to just one of these products—the Robotics Invention System™ version 2.0.) The kits contain motors and connector wires; touch and light sensors; an RCX unit which is a computer that can be built right into a robot and used to store and run

programs; an infrared transmitter that is used to download programs from a PC to the RCX; the Mindstorms™ programming software on a CD-ROM; building instructions; and hundreds of specially designed Lego building pieces. All of the pieces are designed to snap together, and everything can be recombined and reused indefinitely. Illustrated building directions for a variety of robot bases, along with variations that can be added to them, are shown in the Constructopedia building guide. More experienced and adventuresome builders can invent their own designs. There are also several books related to Mindstorms™ and many active websites with a variety of interesting projects and ideas and advice on using the kits.

It is important to have enough kits for students to work in small groups. We have found groups of 3 or 4 to be an optimal size in school settings. (While obtaining enough kits for a large class may at first seem very expensive, it actually compares favorably to the prices for current standards-aligned science modules now on the market for middle school. The good news is that more and more schools are starting to recognize the need to commit significant funds to field good curriculum in science and technology.) Everything in the Mindstorms™ kits is fully reusable, and many replacement parts can be purchased from Lego if pieces get lost or damaged. The programming software is provided on a CD that will run on PCs with Windows as the operating system (except for Windows NT). Unfortunately, there is no version available that runs on Macintosh computers. If you will be using a serial IR tower (purchased separately) you will need a 9 volt battery for each tower. Additional suggested materials include AA batteries, floppy disks for students to store their programs, small flashlights, and bins or large shoeboxes to store completed and partially built robots safely.

Once you have acquired a set of Mindstorms™ kits, make sure to allow some time to set them up prior to using them with your students. The kits come with instructions for installing software onto the computers you will be using, setting up the infrared transmitters, installing batteries, and loading “firmware” into the RCXs. Eventually, you may want to train some of your students to be able to do some of these set-up chores. For example, after changing the batteries in an RCX, it’s often necessary to download firmware again, which takes about 5 minutes. Rather than taking time out of class, you could have some of your students perform routine maintenance. When you first get your kits, we also recommend removing the small Lego pieces from the sealed cellophane packages they come in and organizing them by type (e.g., axels, gears, wide plates, bricks, connectors, etc.) in resealable clear plastic bags such as Ziplocs™. This makes it much easier for the students to keep their kits organized and to prevent lost pieces. It also saves time when students are looking for building parts. Inside the back cover of the Constructopedia is a complete list of all of the parts in a kit. We suggest that you or your student helpers periodically sort through kits to keep them organized and monitor lost or misplaced parts.

The first activity starts with a simpler kit that provides an excellent introduction to mechanical and electrical design. It is called the Wall Hugging Mouse and is available from Tamiya Inc. for about \$20. (We’ve seen prices ranging from \$18 to \$28, so it is worth shopping around for a better price.)

Can I really teach this?

Robotics is new to everyone. No one expects teachers to have a specialized background in fields related to robotics, and it is perfectly possible to follow this curriculum without such a background. Because it is new, it provides a special opportunity for teachers to model for students the very important sets of skills and

attitudes that will help them become life-long learners. Your approach can help students share in the enthusiasm for learning new things (and can also help them manage the frustrations that sometimes crop up in the process). You may want to make a point of modeling and encouraging some of the following strategies:

- Breaking a larger problem into manageable, step-by-step pieces
- Testing your understanding and your solutions as you go
- Working systematically
- Persisting rather than giving up when something doesn't work the first time
- Using all available resources (e.g., the accompanying Technical Guide and resources on the Internet) to find answers to questions, look for new building ideas, and improve your expertise
- Relying on, cooperating with, and learning from teammates and classmates

A number of more specific strategies and problem-solving techniques for activities such as debugging a program are covered later in this Guide. Just like your students, you will find that the more you do this, the easier it gets. We have never seen a learner (adult or child) so "stuck" that there wasn't a fairly immediate way to simplify the problem or break it down into smaller pieces and get the learning moving again. More often, students are struck by how obvious and systematic their improvement is, even over just a few sessions.

Consider developing your own personal support team as you undertake robotics. You may team up with a colleague or recruit some community volunteers. We suggest making contacts with college students in science and engineering fields, parents who enjoy mechanical and electrical hobbies, mentors from local industry (e.g., engineers), and retired professionals. The Mindstorms™ kits are popular with adults as well as children, and many adults would welcome the chance to mentor students using the kits. (It is particularly helpful to have someone who is computer savvy help you install and test the software and the downloading processes before you get started to make sure that everything works in your environment.) You can also recruit middle school students who can be trained as team leaders, equipment managers, and classroom assistants. Train group leaders in an after-school club and assign them to different groups when you are ready to teach it to the entire class.

References and Resources

American Association for the Advancement of Science, Project 2061. *Benchmarks On-Line*. www.project2061.org/tools/benchol/bolframe.htm.

International Technology Education Association. *Standards for Technological Literacy*. www.iteawww.org

National Research Council. *National Science Education Standards*. National Academy Press. www.nap.edu/readingroom/books/nses/

Lego Mindstorms™. <http://www.mindstorms.lego.com>