STEM SCOUTS®

SCIENCE
TECHNOLOGY
ENGINEERING
MATH

Technology Lab: Leaders Guide – Building With mBot

Leaders Guide

Building With mBot

Leaders Guide

Building With mBot

Overview

This module uses the mBot robot to show how programming and robotics can be combined to create some amazing projects. Robots are quickly changing the way we work and live and will have an even larger impact on our lives in the future. Robotics is an exciting way to teach Scouts the different areas of STEM with activities that will challenge them to construct and program a robot. The mBot allows the Scouts to explore how motion, lights, sounds, and sensors are used to control the actions of their robot. They will use the Scratch Blockly programming environment to work collaboratively with their team to solve challenges during each meeting.

This module was developed for STEM Scouts by Eduporium. Eduporium (www.eduporium.com) is an all-encompassing partner of STEM educators and after-school programs, specializing in creating innovative education and computer technology solutions to better prepare young people for the 21st century. Their experts creatively combine technology tools into engaging solutions for use in small groups or large classrooms, and they design starter activities so young people can enjoy meaningful learning. Eduporium encourages early exposure to invention, problem-solving, coding, and collaboration to provide every youth with the chance to succeed.

This module takes six STEM Scout meetings of approximately 90 minutes each.

Meeting 1: Introduction to Robotics

Robots can be made in many ways, using all types of materials. Most robots share a great deal in common, and Scouts will discover these important features while building their mBot. Before assembling the mBot's different parts, the Scouts will learn about input and output devices, controllers, and power sources. Once the mBot is built, the Scouts will learn to control it using the preprogrammed features and the remote control.

Meeting 2: Introduction to Programming

Scouts will learn to control their mBot by programming with the mBlock software. The Scouts will go through an introduction to programming and the Scratch 2.0 software. They will develop programming skills while designing programs to make the robot move in a square and in a circle. Scouts will be able to explore and get a feel for the software as they begin their mBot programming journey.

Meeting 3: The Physics of Acceleration

Iteration and experimentation are important concepts in programming and engineering. This means developing something, trying it out, and developing it some more. In this meeting, Scouts will use the STEM Scouts Engineering Design Process to practice their programming and collaboration skills. Scouts will learn how to control the acceleration of their mBot and will be challenged to drive their robot up and down a ramp.

Meeting 4: Programming With Math, Light, and Sound

The mBot has several sensors that collect data. Using their math skills, the Scouts will work with this data to create a reaction from their mBot when the correct information is received. Scouts will discover the important role that math plays in programming. The mBot's light sensor and ultrasonic sensor will be activated to control the mBot's actions.

Building With mBot

Meeting 5: Programming With Music
The mBot has two RGB LED lights and a buzzer on the mCore processor. The Scouts will use these features to program their mBot to dance, play music, and change the colors of the lights. The Scouts will gain a better understanding of how RGB lights are used in pixels to create images that are able to be manipulated. Scouts will be challenged to reuse and remix a program to create their own unique project.

Meeting 6: Putting It All Together
Scouts will put together all of their programming knowledge and robotics skills developed over the course of the previous meetings as they are challenged to create a program that will enable the mBot to run through several different courses.

Pre-Module Preparation
The adult leaders should read through all of the meeting activities ahead of time. If this is your first time managing a Lab, please review the Lab Leaders Operations Guide, which contains general instructions, lab setup information, the Scout Oath and Scout Law, and more.

Review each meeting plan and identify what you want your Principal Investigator, Co-Principal Investigator, and Program Manager to be responsible for in that meeting. A few minutes before the meeting, sit down with those Scouts and review what they need to do for that meeting.

Special Advance Preparation

Meeting 1: Try building and programming one of the mBots so you are familiar with the materials and how the mBot works so you can help your Scouts if they have trouble putting it together.

Put the batteries in and test that each mBot works properly well in advance of the first meeting.

Make sure that you have internet access for the laptops. If you are using your sponsoring organization's computers, make sure you can access the links listed in the meeting plan.

Check to see whether the "**mBlock 3 for PC**" software is already installed on the computers you will be using. If not, download the "**mBlock 3 for PC**" software onto all of the computers prior to the meeting, http://www.mblock.cc/software/ and check that the mBot can load a program from the mBlock software. Just a reminder, please do **not** download "mBlock 5 for PC" for this module.

The mBot can connect to the computer using the USB cord.

Meeting 2: Make sure that you have internet access for the laptops. If you are using your sponsoring organization's computers, make sure you can access the links listed in the meeting plan.

Check to see whether the "**mBlock 3 for PC**" software is already installed on the computers you will be using. If not, download the "**mBlock 3 for PC**" software onto all of the computers prior to the meeting, http://www.mblock.cc/software/ and check that the mBot can load a program from the mBlock software. Just a reminder, please do **not** download "mBlock 5 for PC" for this module.

The mBot can connect to the computer using the USB cord.

Building With mBot

If you are using the laptops without plugging them in, make sure they are fully charged well ahead of the meeting.

Meeting 3: You will need to have enough open floor space for each team to build a ramp roughly a foot wide and 3 feet long with space around it for the mBot to go up and down the ramp.
Each team will need to use the mBot box as the center platform for their ramp.

If you are using the laptops without plugging them in, make sure they are fully charged well ahead of the meeting.

Meeting 4: The first part of this lab will measure the reaction of the light sensor on the mBot to the amount of light available. Look into whether you can vary the amount of light in your meeting space. The teams have a flashlight in their kits for very low light levels (assuming you can turn off the lights completely).

If you are using the laptops without plugging them in, make sure they are fully charged well ahead of the meeting.

Meeting 5: In this meeting, the Scouts will modify an existing program and use it. Before the meeting, download the following program:

https://github.com/Matt-Ma/Makeblock-Tutorials/raw/master/Singing%20and%20Dancing%20mBot.sb2.

Save it to the desktops of the teams' computers so it will be easy to access.

Meeting 6: You will need to set up a challenge course for the robots to run on that includes a ramp like the ones used in Meeting 3. All the teams will compete on the same challenge course. If you have a very large Lab, you may want to set up more than one course and run them in parallel. Have the course set up before the meeting.

Arrangements should be made to pick up computers and other council-supplied materials a few days before the lab and to return them as soon as possible after use. If software needs to be installed, be sure to allow adequate time for that in your preparation. If you are using your sponsoring organization's computers, make sure that you can access all links and arrange for any software to be installed ahead of time.

All materials needed for each weekly meeting are listed at the beginning of the meeting plan. You will want to check these well in advance and make sure nothing is missing.

Scouts will be divided into teams of four members for this module. Material quantities are defined for each team. The following are the council- and unit-supplied materials for this module. Unit-supplied materials can often be supplied by parents, if requested well in advance.

Meeting 1
- 1 laptop for each Scout team (unit- or, optionally, council-supplied)
- 1 laptop for Lab Manager (unit- or, optionally, council-supplied)
- 1 projector for Lab Manager (unit- or, optionally, council-supplied)
- Optional: 1 empty water bottle (unit-supplied)
- Optional: miscellaneous obstacles (books, blocks, shoes, kit boxes etc.) (unit-supplied)

Building With mBot

Meeting 2
- 1 laptop for each Scout team (unit- or, optionally, council-supplied)
- 1 laptop for Lab Manager (unit- or, optionally, council-supplied)
- 1 projector for Lab Manager (unit- or, optionally, council-supplied)

Meeting 3
- 1 laptop for each Scout team (unit- or, optionally, council-supplied)
- 1 laptop for Lab Manager (unit- or, optionally, council-supplied)
- 1 projector for Lab Manager (unit- or, optionally, council-supplied)

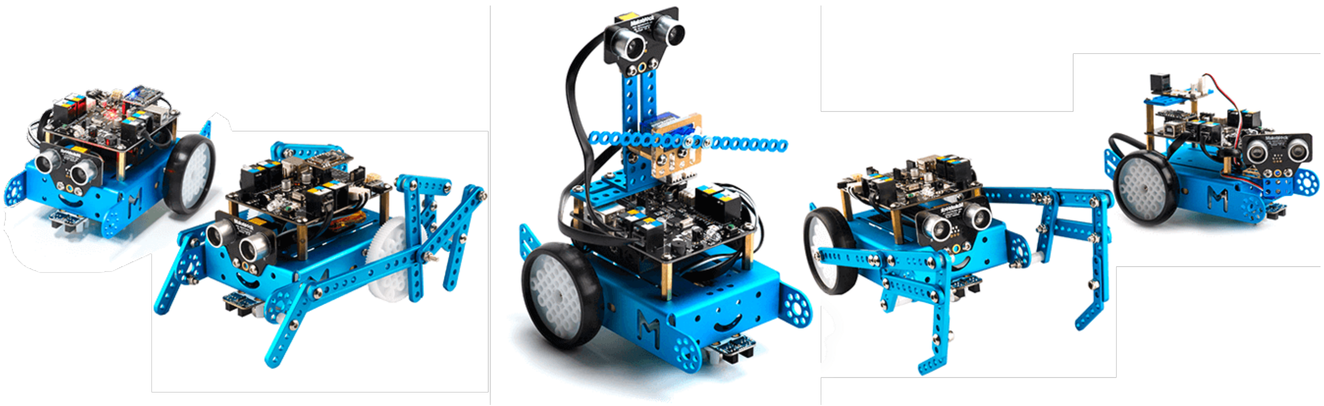
Meeting 4
- 1 laptop for each Scout team (unit- or, optionally, council-supplied)
- 1 laptop for Lab Manager (unit- or, optionally, council-supplied)
- 1 projector for Lab Manager (unit- or, optionally, council-supplied)
- 1 laptop for each Scout team (unit- or, optionally, council-supplied)
- 1 laptop for Lab Manager (unit- or, optionally, council-supplied)
- 1 projector for Lab Manager (unit- or, optionally, council-supplied)

Meeting 6
- 1 laptop for each Scout team (unit- or, optionally, council-supplied)
- 1 laptop for Lab Manager (unit- or, optionally, council-supplied)
- 1 projector for Lab Manager (unit- or, optionally, council-supplied)

Service Project Idea
Service projects are a good way for Scouts to give to their sponsoring organization and community. Don't try to do more than one service project per module. Check on the STEM Scouts Portal for a list of service project ideas.

Leaders Guide

Meeting 1: Introduction to Robotics



Meeting 1: Introduction to Robotics

Pre-Meeting Leader Preparation

Try building and programming one of the mBots so you are familiar with the materials and how the mBot works so you can help your Scouts if they have trouble putting it together.

After you have assembled a “test” mBot, you will want to use your assembled mBot to *test all of the other batteries* to make sure they work. Once tested, **disassemble** the mBot you put together so that all Scout teams have one to work with.

Make sure that you have internet access for the laptops. If you are using your sponsoring organization’s computers, make sure you can access the links listed in the meeting plan.

Check to see whether the mBlock software is already installed on the computers you will be using. If not, download the software onto all of the computers prior to the meeting (www.mblock.cc) and check that the mBot can load a program from the mBlock software.

The mBot can connect to the computer using the USB cord.

Materials and Tools Needed

- 1 laptop for Lab Manager (unit- or, optionally, council-supplied)
- 1 projector for Lab Manager (unit- or, optionally, council-supplied)

The following materials are shared between multiple teams.

- 1 pair of tweezers to tighten nuts if they become loose (shared between two teams)
- Optional: 1 empty water bottle (unit-supplied)
- Optional: miscellaneous obstacles (books, blocks, shoes, kit boxes, etc.) (unit-supplied)

Scouts will be divided into teams of four for this set of experiments. Material quantities are defined for each team.

- 1 mBot kit
- 1 small Phillips head screwdriver (shared between two teams)
- 4 AA batteries
- CR2025 button cell battery for remote control

Meeting 1: Introduction to Robotics

Opening

Have the Principal Investigator lead the group in reciting the Pledge of Allegiance and the Scout Oath and Scout Law.

Applying the Scout Law

Choose one of the 12 points of the Scout Law to discuss in application to today's lab. A suggested theme for this meeting is *kind*, as in *I will be kind to my teammates and help to make sure that everyone gets to participate in exploring the mBot*.

Ask the Scouts what being *kind* means to them and how they might apply that to today's lab.

Activity Overview

Scouts will assemble their mBot using the instructions in the mBot kit. They will learn about the parts of the robots and their purpose and importance. They will also learn to control the mBot using the preloaded programs on the remote control.

Background

Robots have evolved to have many uses, including movement and recognizing external objects. When combined with other forms of technology, robotics can be a valuable tool for learning the concepts of programming. People around the world work with robots that are small enough to fit in the palms of their hands or as large as people and program them to develop certain abilities and actions. These days, robots can accomplish just about anything, helping to show the importance of building problem-solving skills and developing creativity.

mBot is an easy-to-run robot kit that provides hands-on experience in graphical programming, electronics, and robotics. It is easy to assemble and comes with an Arduino open-source platform that allows the robot to be programmed using the mBlock software.

Although there are many different types and styles of robots, they all have similar core parts. The mCore main control board is the brain of the mBot. The wheels and chassis are the arms, legs, and body. Sensors in robots can often replicate the five senses.

All computers are directly programmed in what is commonly referred to as machine code. These are usually very low-level binary commands to individual parts of the computer. While machine code allows you to completely control every aspect of any computer, it is very difficult to understand and takes enormous amounts of time to create the code to perform even simple tasks.

Programming languages are important because they allow programmers to communicate with machines such as computers and robots at a higher level. A programming language is a set of instructions that tells a computer to perform specific tasks. Programming languages are like spoken languages. There are many different spoken languages, such as English, French, and Spanish. There are also many different programming languages, such as Python, Java, and C++, that have been developed. The vast majority of coding done today uses high-level programming languages. These are languages that look more like math or common human languages and translate our commands to machine code. An example is adding two numbers.

Meeting 1: Introduction to Robotics

A high-level language might let you type in:

A = B + C

Machine code commands might look like: 110100011011101

The same equation in Assembly code, which is a level above machine code, might look like the following:

Load r1, mem(5) (where mem(5) is the memory location where the value of B is stored)

Load r2, mem(6)

Add r1, r2, r3

Load mem(4), r3

Which do you think is easier to work with?

A software development environment (SDE) is the term that describes the system in which a person can use a programming language. SDEs often contain simulators of the code, the ability to download and test code, and visual elements that make programming simpler. The mBot is programmed using the mBlock SDE software, which uses Blockly as the programming language interface. Blockly is a visual block programming language that packages the underlying JavaScript into easily manageable blocks. Block programming makes programming for new or younger programmers easier and quicker to understand and use for projects.

Engaging Questions

Use some of the following questions to get the Scouts engaged. (*Typical answers are in italics.*)

Where are robotic devices used? *(Answers will vary: Manufacturing, medicine, construction, education, law enforcement, and military are some examples.)*

How are the parts of the mBot similar to the human body and how is a human programmed like an mBot? *(The mBot has an mCore control board like a brain, motors like a heart, wheels like arms and legs, and sensors like ears, eyes, and skin. When we learn new things, our brain is programming our body to do new things.)*

What is the purpose of programming languages? *(Programming languages provide a simpler and more highly organized way for people to instruct computers and computing devices to do different tasks.)*

Safety Moment

Tell the Scouts:

Keep track of your mBot parts when assembling and disassembling it. You need all these parts for the rest of the meetings.

Be careful not to force the parts together when assembling the mBot.

Make sure the mBot does not drive off tables or counters. If you are using it on the floor, make sure you don't step on it or trip over it.

Meeting 1: Introduction to Robotics

If you have long hair, tie it back following lab safety rules. Also, if you have any long, dangling necklaces or similar items, take them off or put them inside your clothing. You don't want them to get caught in the mBot's wheels.

Experiment

Get the Scouts into teams of four. They will stay in these teams for all six meetings in this module.

Ask them to follow the instructions in their Lab Notebook, reproduced below.

Materials List

- 1 pair of tweezers to tighten nuts if they become loose (shared between two teams)
- 1 mBot kit
- 1 small Phillips head screwdriver (shared between two teams)
- 4 AA batteries
- CR2025 button cell battery for remote control

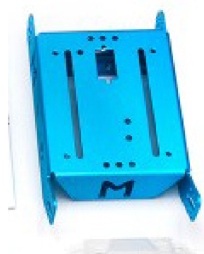
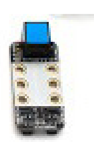
Step 1: Parts of Your Robot

As a team, discuss what you think each part does. Then write in the human equivalent for that part.

Meeting 1: Introduction to Robotics

What body part compares with the robotic part?

Lab Manager Note: Possible answers are in red; the Scouts will have that part of the table blank.

Image	*Robotic Name*	*Human Body Part*
	Chassis	The chassis is like the skeleton of the robot. All of the other parts of the robot are connected to it.
	Motors	The motors are like the heart of a robot. They move the robot's wheels.
	Wheels	Wheels are like the legs or feet of a robot. They allow the robot to move forward, backward, left, and right, depending on the motors.
	mCore Processor	The processor is like the brain of the robot. It receives commands and sends commands to the other parts that are connected.
 	Sensors Me Ultrasonic Sensor Me Line Follower	There are many types of sensors in robotics. The Me Ultrasonic Sensor is like sight and it can sense objects. The Line Follower is a light sensor that can sense different levels of light.

Meeting 1: Introduction to Robotics

Step 2: Assemble Your mBot (25 minutes)

Lab Manager Note: While the Scouts are working, check in with the teams to verify they are rotating roles.

You and your team will assemble your mBot following the instructions in the mBot kit. Each page in the assembly manual lists the parts needed for each step and explains how to assemble each part.

You and your teammates should take turns with the following roles when assembling the parts:

- **Inventory Manager:** collects the parts for assembly
- **Builder:** adds the parts to the mBot
- **Quality Manager:** checks that the parts are assembled correctly
- The fourth Scout can be the Inventory Manager for the next step and can begin finding parts for that step.

Inventory Manager: Find the parts needed on page 4 of the mBot assembly manual.

Builder: Using the parts from your Inventory Manager, assemble the parts as shown in the assembly manual.

Quality Manager: Verify that the Builder has the correct parts and has assembled the parts correctly.

You and your team should rotate roles after finishing each page.

Complete steps through page 7 (Wiring), with everyone taking different roles each time.

When the mBot is assembled, put any spare parts and tools back into the mBot kit.

Step 3: Power Up!

After you have assembled the mBot, turn it on using the switch on top of the mCore processor.

Lab Manager Note: Check each team's mBot and troubleshoot any connection or assembly issues.

Step 4: Power the Remote Control

Place the CR2025 button cell battery in the remote control.

Step 5: Remote Controls! (25 minutes)

The mBot can be controlled through programming or by using the remote control. The remote control has three preloaded programs that you can use. You and your team can use the gear button on the remote control to practice using these features. There is a good picture of the remote control on page 9 of the assembly manual you can refer to. Each of you should have a chance to try each of the three modes.

Mode 1 is the Remote Manual Control: Practice driving around the room, being careful not to drive into walls or other mBots.

Meeting 1: Introduction to Robotics

> **Lab Manager Optional Activity:** Have Scouts design a group race course around tables or chairs so the teams can compete to see whose mBot completes the course first using Mode 1. Ready, set, go! You can have it be a timed race or create a starting line and a finish line and have all teams compete at the same time.

Mode 2 is the Wall Avoidance Robot Mode: See where your robot takes you and what happens when it encounters obstacles using the Me Ultrasonic Sensor.

> **Lab Manager Optional Activity:** Create a 5x5-foot square with masking tape. Have one Scout on each side of the square with a water bottle or any obstacle (shoe, book, block). The object of the game is to have the Scouts anticipate where the mBot is going and place the bottle in front so that it will turn and not cross the edge of the square. Once it crosses a line, the Scout by that line is out and a new Scout takes their place for the next round.

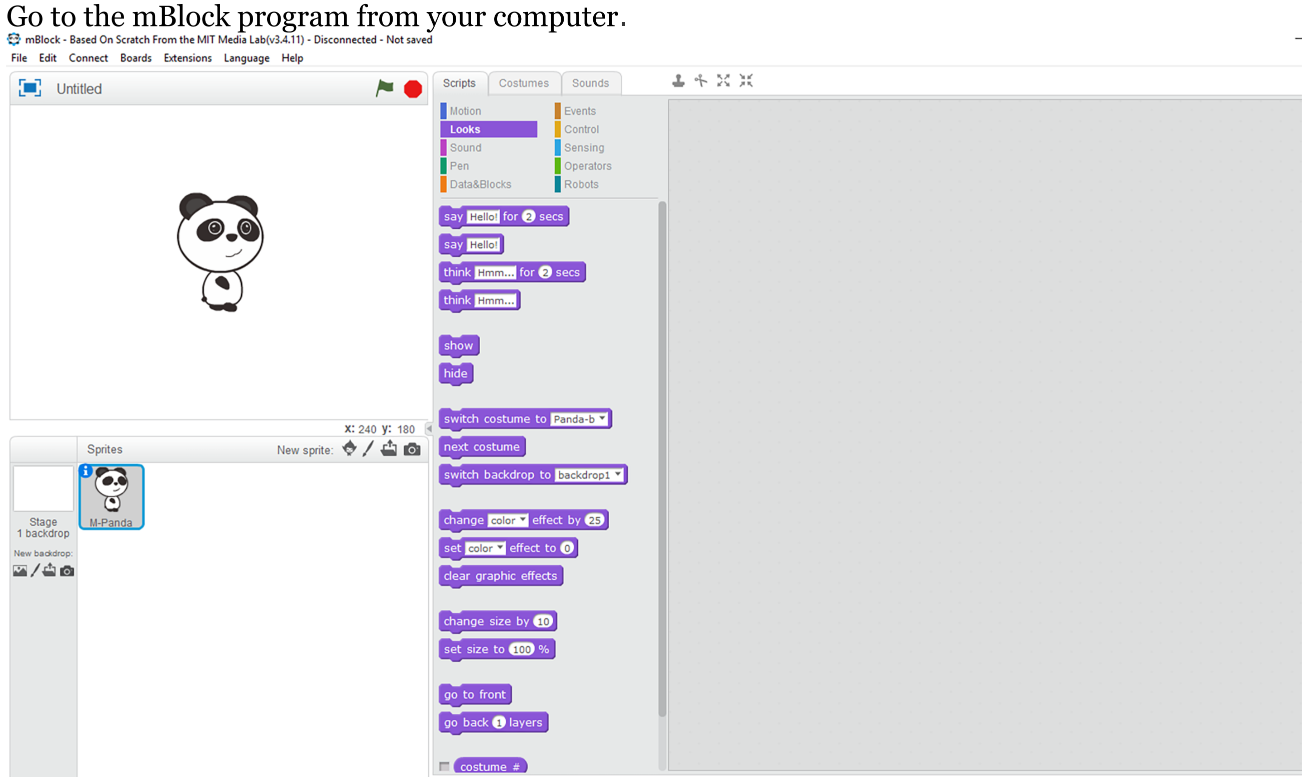
Mode 3 is the Line Follower Robot Mode: Using the Route Map that is included in the mBot kit, test and see how the mBot uses its Me Line Follower sensor.

Troubleshooting Problems:

If the mBots are not going in the correct direction, they are not wired correctly, so you will need to switch the M1 and M3 connections. (See the Wiring page in the manual.)

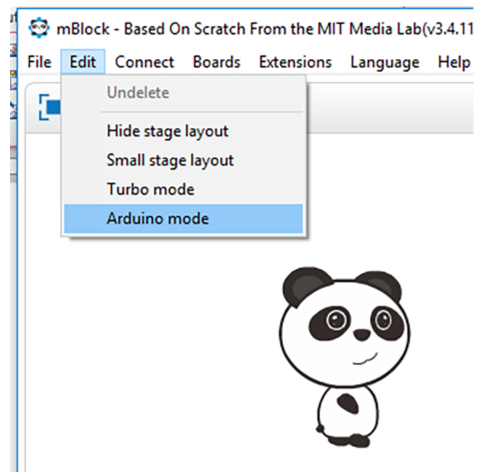
If the remote control does not connect, you may need to reset the default program.

1. Go to the mBlock program from your computer.



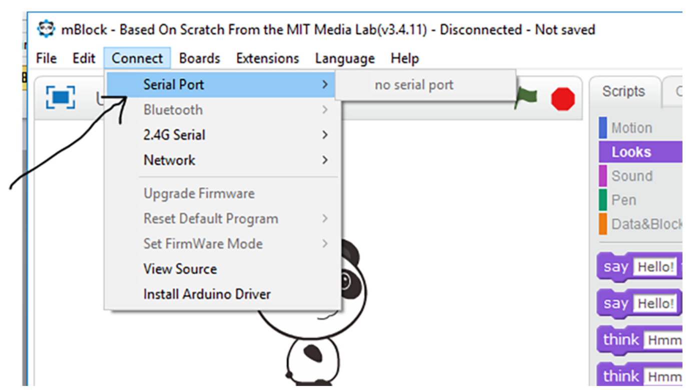
Meeting 1: Introduction to Robotics

2. Using the USB cable, connect the mBot to the computer and then turn on the mBot. (Do not use Bluetooth.)
3. Select **Edit > Arduino Mode**.



4. Select **Connect > Serial Port** (ports may vary; choose the one that connects).

   Serial ports will vary from computer to computer. To find the port for your mBot, see what ports are available with the mBot turned off. Then when the mBot is turned on, a new port should be visible. That is the port that should be selected.



5. Select **Connect > Reset Default Program.**

Check the nuts and bolts periodically to make sure they are not loosening. If they are loose, use the tweezers to tighten them.

Discussion

Lead a discussion on what the Scouts have done and their results. Ask open-ended questions to get them thinking about what they have done and learned, and what they might do differently next time.

Discussion Questions

What was a problem that you encountered and how did you solve it? *(Answers will vary, but it is important to note the strategies used to solve problems.)*

What was a big discovery or "wow" moment that you had? *(Answers will vary. The different answers may be new discoveries for the other groups.)*

Share one piece of advice for other Scouts who need to work in teams. *(Answers will vary, but look for the idea of collaboration.)*

Meeting 1: Introduction to Robotics

Cleanup

Tell the Scouts:

Turn off your mBot, remove the batteries, and put the mBot in the box with the spare parts, remote control, and USB cord. Be careful not to damage the box, as it will house the mBot and its parts for the rest of this module.

Clean your area, and be sure no trace is left behind.

STEM Innovator Moment

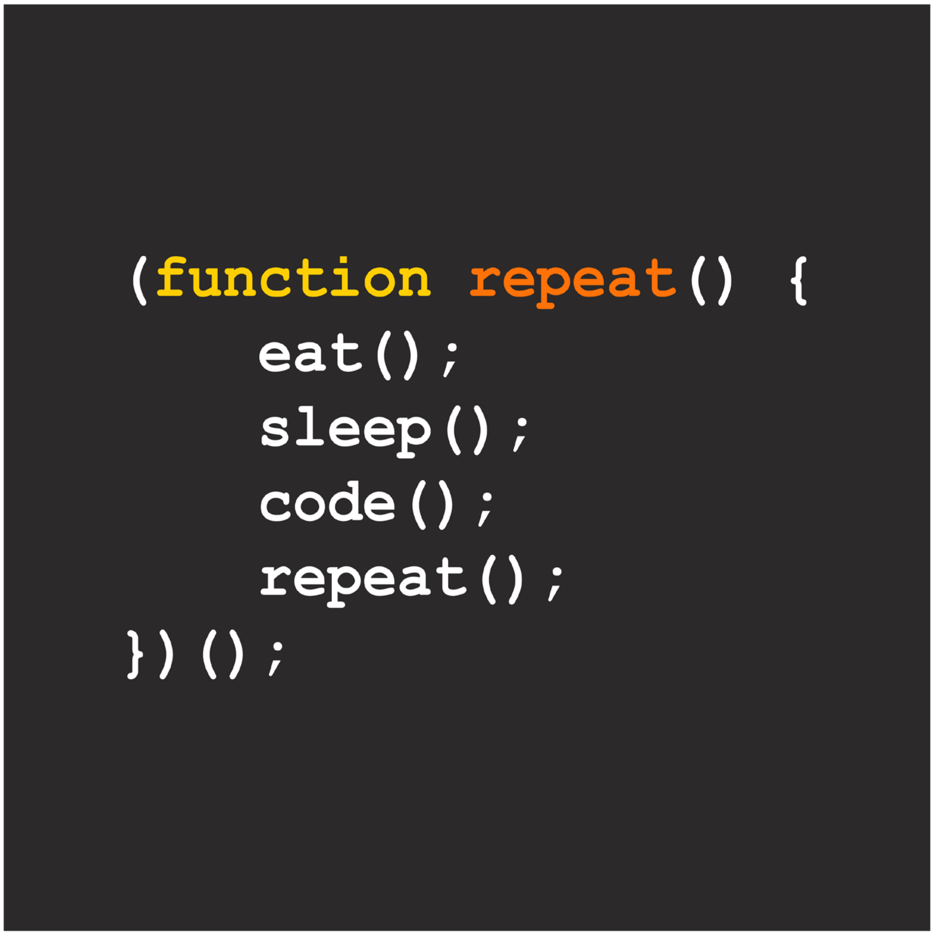
The Scouts should hear from or about a person who has expertise on the topic of the day's activities about how this topic applies to everyday life and careers. The person could visit the Lab or use teleconferencing software such as Skype. You can also use the innovator information below.

> *Scientists are finding many exciting ways to integrate robots into real-world activities. Learn more about the amazing ways robots are being used today and how they are being developed to do some of the jobs that people do. Discussion topic: Will this be a good thing? Play the following video:*
>
> Robotics Innovations are Shaping the Future of Our Society (Journeyman Pictures): www.youtube.com/watch?v=UxVYP0PCVFY&feature=youtu.be (6:26)

Closing

Discuss an ethical situation that occurred during the meeting as well as how the Scout Oath and Scout Law could be applied to deal with the issue. Remember to focus on things that occurred among the Scouts rather than on controversial issues in science today.

Meeting 2: Introduction to Programming



Shutterstock.com, courtesy–©Orange Vectors

Meeting 2: Introduction to Programming

Pre-Meeting Leader Preparation

Make sure that you have internet access for the laptops. If you are using your sponsoring organization's computers, make sure you can access the links listed in the meeting plan.

Check to see whether the mBlock software is already installed on the computers you will be using. If not, download the software onto all of the computers prior to the meeting (www.mblock.cc) and check that the mBot can load a program from the mBlock software.

The mBot can connect to the computer using the USB cord.

If you are using the laptops without plugging them in, make sure they are fully charged well ahead of the meeting.

Meeting 2: Introduction to Programming

Materials and Tools Needed

- 1 laptop for Lab Manager (unit- or, optionally, council-supplied)
- 1 projector for Lab Manager (unit- or, optionally, council-supplied)

The following materials are shared between multiple teams.

- 1 pair of tweezers to tighten nuts if they become loose (shared between two teams)

Scouts will be divided into teams of four for this set of experiments. Material quantities are defined for each team.

- 1 mBot kit (mBot already assembled from previous meeting)
- 1 small Phillips head screwdriver (shared between two teams)
- 4 AA batteries
- CR2025 button cell battery for remote control
- 1 laptop (unit- or, optionally, council-supplied)

Opening

Have the Principal Investigator lead the group in reciting the Pledge of Allegiance and the Scout Oath and Scout Law.

Applying the Scout Law

Choose one of the 12 points of the Scout Law to discuss in application to today's lab. A suggested theme for this meeting is *helpful*, as in *I will help my teammates as we all learn how to program the mBot together*.

Ask the Scouts what being *helpful* means to them and how they might apply that to today's lab.

Activity Overview

Scouts will learn how the parts of the mBot can be controlled through programming. They will be introduced to the mBlock software and learn how Blockly programming simplifies complex programming into easy-to-use blocks. Scouts will use mini-tutorials to learn about movement and rotation; controlling sensors, lights, and sounds; and loading and running their programs with the mBot.

Background

History of Programming: The middle of the 19th century saw much forward progress. Areas such as engineering, transportation, communication, architecture, science, and manufacturing were all changing rapidly. The people who worked in these fields relied on printed numerical tables for calculation. Human error was common when working with these numbers, however, and it was feared that these errors could be a disaster waiting to happen.

Teamwork: Early attempts at creating a computer were unsuccessful. The ENIAC, or Electronic Numerical Integrator and Computer, is considered the first successful programmable computer. Its success was attributed to the team of innovators that used past models and teamwork to build it.

In this lab, Scouts will be using mBlock to control the mBot. mBlock is a graphical programming language based on the Scratch software development environment. Scratch makes it easy to program the mBot and create other creative programs. Scratch was designed so that anyone can code. The blocks are made up of JavaScript commands. JavaScript can take time to master but when it's put into blocks, like in Scratch, users can start coding and creating in no time.

Meeting 2: Introduction to Programming

Lab Manager Note: Play the following video for the Lab.

What Makes a Computer, a Computer? (Code.org): www.youtube.com/watch?v=xfKn5OjHLqQ (5:09)

Engaging Questions

Use some of the following questions to get the Scouts engaged. (*Typical answers are in italics.*)

What are the possible dangerous consequences when an engineer has made a miscalculation? *(A miscalculation could cause a design to fall apart or not perform correctly. For example, a bridge or building could collapse.)*

The first computers were unsuccessful. Why are mistakes important in designing and creating? *(Mistakes allow the designers to learn what does not work and help to identify where the design needs to improve.)*

Safety Moment

Tell the Scouts:

Make sure the mBot does not drive off tables or counters. If you are using it on the floor, make sure you don't step on it or trip over it.

Connect and disconnect the cable carefully when testing programs so that you don't damage the cable or connector.

If you have long hair, tie it back following lab safety rules. Also, if you have any long, dangling necklaces or similar items, take them off or put them inside your clothing. You don't want them to get caught in the mBot's wheels.

Experiment

Get the Scouts into the same teams of four that they were in for the previous meeting.

Ask them to follow the instructions in their Lab Notebook, reproduced below.

Activity 1: Introduction to mBlock (20 minutes)

Materials List

- 1 mBot kit (mBot already assembled from previous meeting)
- 1 small Phillips head screwdriver (shared between two teams)
- 4 AA batteries
- CR2025 button cell battery for remote control
- 1 laptop (unit- or, optionally, council-supplied)

Lab Manager Note: Explain the following to the Scouts:

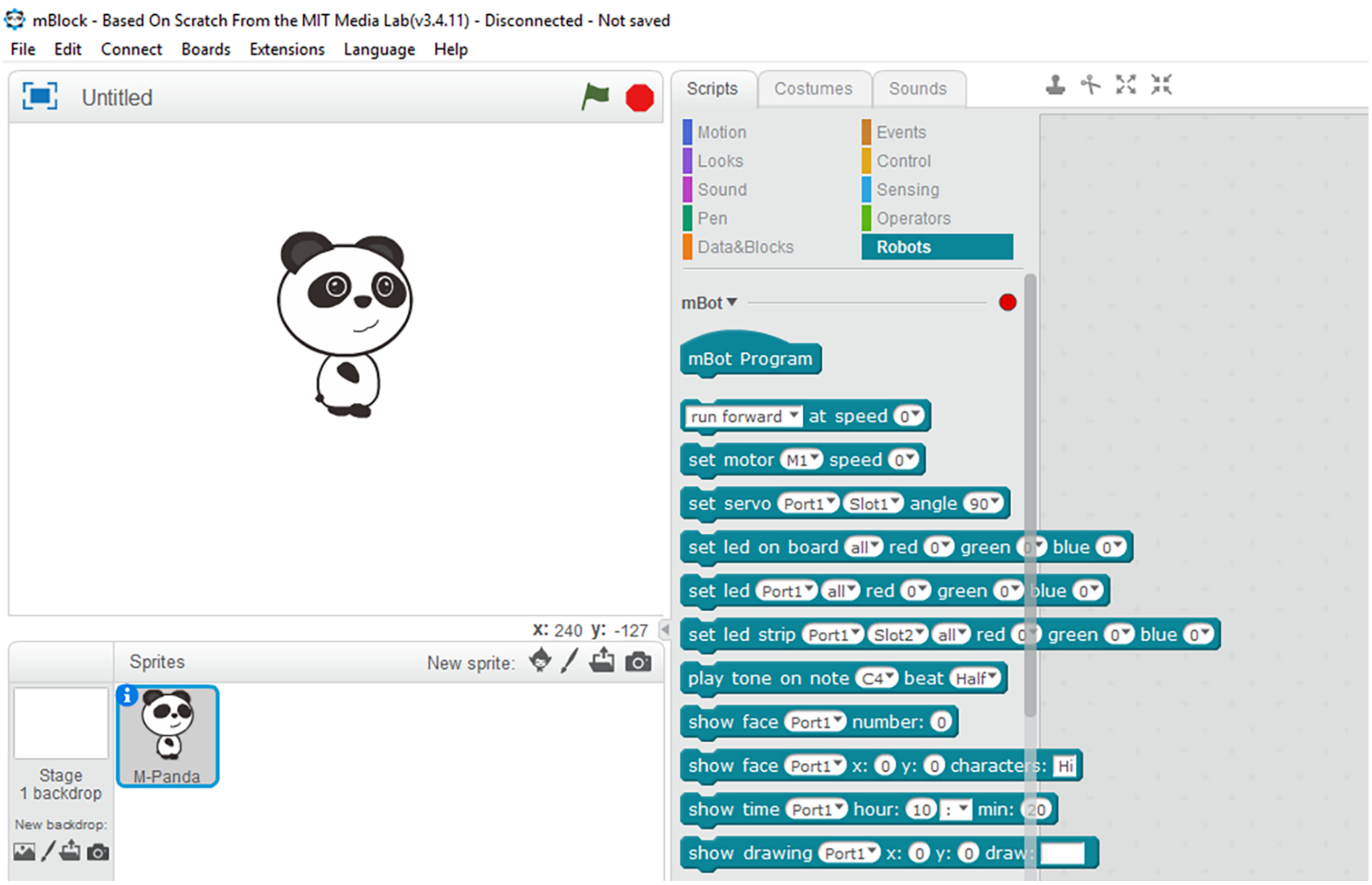
Programmers write programs to control computers using languages such as Java, C, or Python. Controlling a robot generally requires knowledge of Arduino C++ or a similar language. mBlock, however, packages more complex programming languages into blocks, making the writing of programs as easy as dragging and dropping blocks of code. Additionally, mBlock lets you design games and visual effects for your computer, as well as write programs for Arduino-based robots.

Put the batteries back into your mBot robot.

Meeting 2: Introduction to Programming

Step 1: You and your team should open the mBlock program that is loaded on your computer. Make sure that each member of your team has a chance to be the programmer.

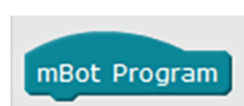
Lab Manager Note: Check to make sure each team can find and open mBlock. The screens should look like this.



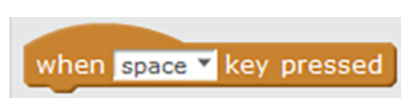
Step 2: Program Your Panda

Scouts will first practice programming their Panda before programming their mBot.

Sprites are the characters that you will program for this activity. You can change sprites; however, for this activity we will program a panda.



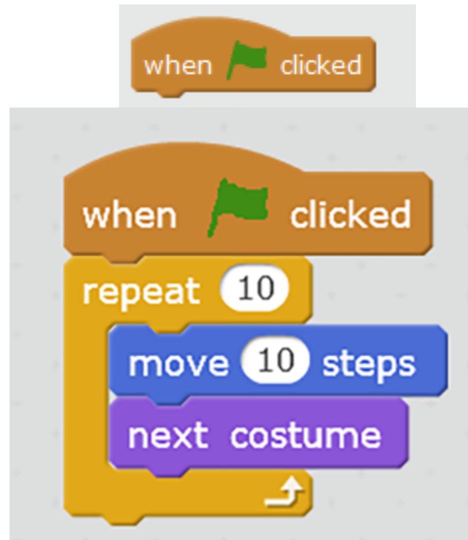
Event blocks or **event handlers** start action in your program. They are different from other blocks because they have a rounded top and can have nothing attached above them. NOTE: mBot program blocks must be connected to the rest of the blocks in order to successfully run the commands.



Some examples are:

Motion blocks: Control how your sprite will move around your screen.

Meeting 2: Introduction to Programming



Look blocks: Control the appearance of your sprites in the program.

Control blocks: Allow you to control your sprites.

Copy this program that uses Event, Control, Motion, and Look blocks.

What else can you make your panda do?

Each team member should take about 5 minutes to see what they can discover.

If your sprite goes off the screen, use a **go to x:0 y:0** command to get it back on-screen.

Lab Manager Note: Remind the teams to switch every 5 minutes.

Now it's time to program your mBot!

Your team will do two programming activities. Spend a few minutes discussing how your team will make sure each team member gets a chance to program.

Scouts who are not programming should be reviewing the program and managing the robot when it is running.

Activity 2: Go Around in Circles (25 minutes)

Lab Manager Note: Tell the Scouts that, while following these instructions, they should make sure that their robot is either propped up so the wheels aren't in contact with a surface or in a space where it can move freely. Make sure the robot does not fall off of any desks or tables.

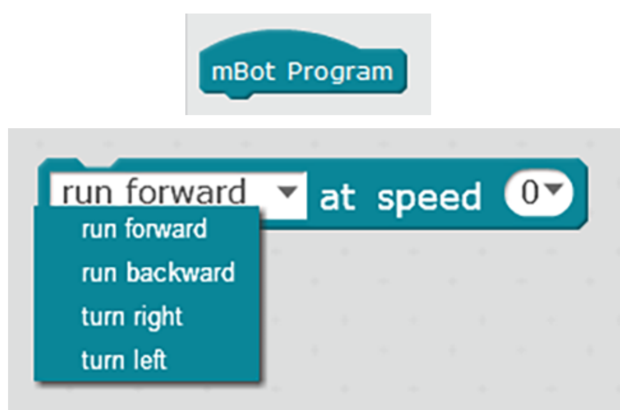
Remember to switch programmers so everyone has a chance to code.

Remove your mBot and your connection cord from the box.

When programming the mBot, use **Scripts> Robots.**

To start your program, use the mBot Program block. Then connect the other blocks below this block.

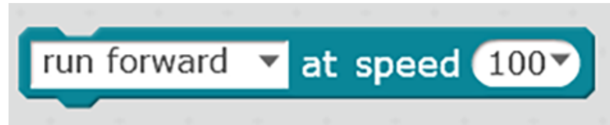
Robots are made to move, and in this activity you are going to learn how. The block that controls robot movement has two drop-down menus. The first menu controls the direction in which the robot moves:



The second menu controls power. The top power is 255, 0 stops the motors, and negative numbers reverse direction. (NOTE: Very low power levels may not be strong enough to get the robot moving.)

Meeting 2: Introduction to Programming

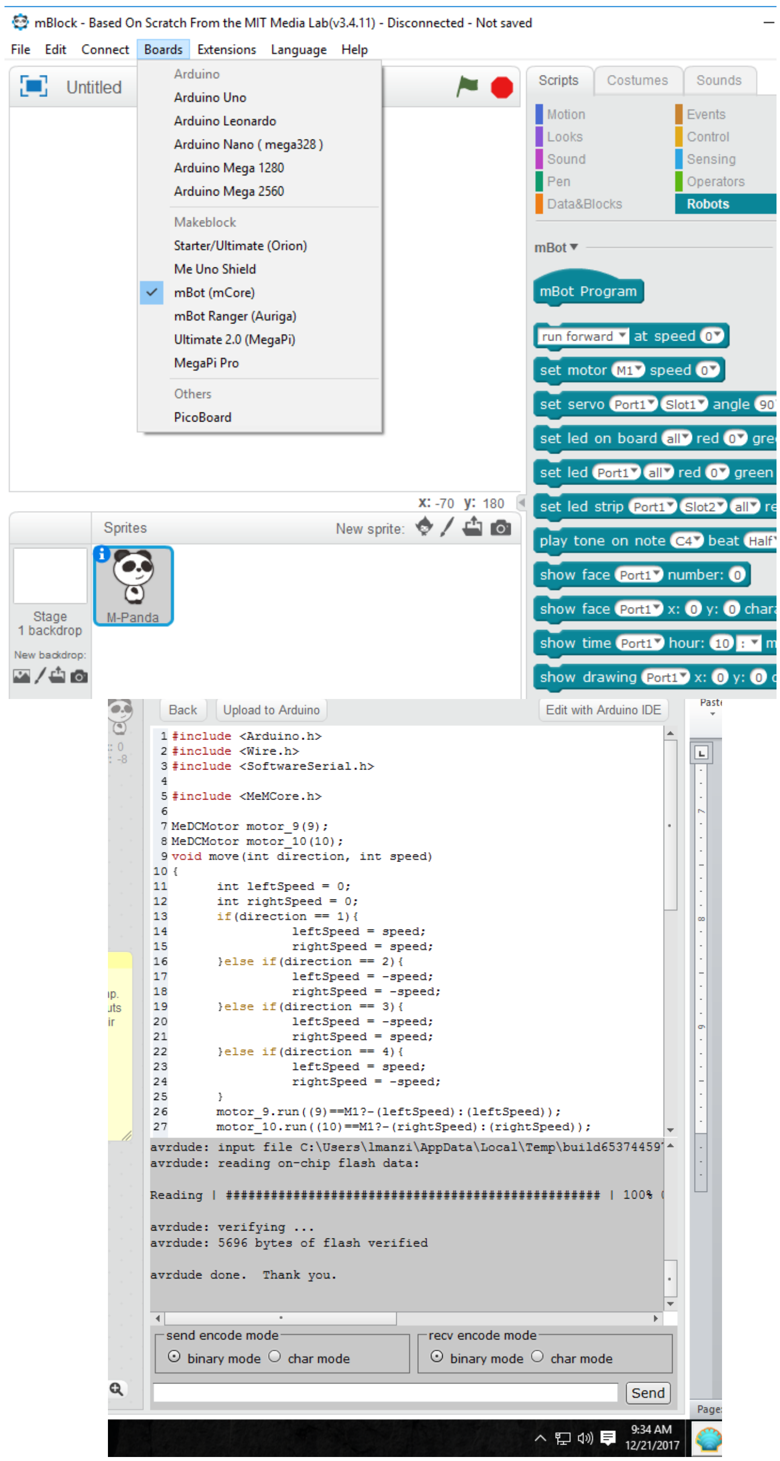
The **Programmer** should select the following to get the robot moving forward at a speed of 100:



Connecting your mBot to run your programs:

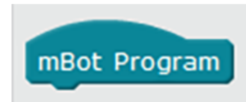
- Plug your connection cord into a USB port on your computer.
- Plug the other end into your mBot.
- Hold your mBot and be careful to keep your hands away from the wheels.
- Turn on your mBot. (Note that the last program that was entered may start running.)
- On the mBlock program, open the **Boards menu** and select **mBot (mCore).**
- Next open the **Connect menu** and select **Serial Port.**

Serial ports will vary from computer to computer. To find the port for your mBot, see what ports are available with the mBot turned off. Then when the mBot is turned on, a new port should be visible. That is the port that should be selected.



Meeting 2: Introduction to Programming

- Click on the mBot Program block to open Arduino mode.
- Click the Upload to Arduino button to upload your program to your mBot.



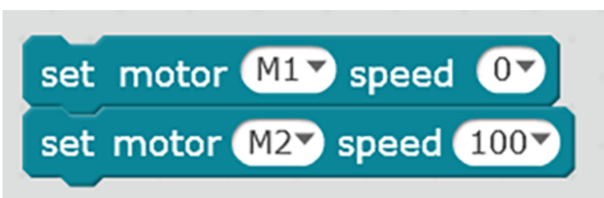
Sometimes you don't want to move in a straight line or perform a point turn (as the turn left/turn right commands do). In this case, you can use the set motor block:

You can choose a motor (M1 is the left; M2 the right) and a power level (negative values make the mBot move in reverse). In this way, you can perform swing turns, in which one motor is turned off and the other is on):

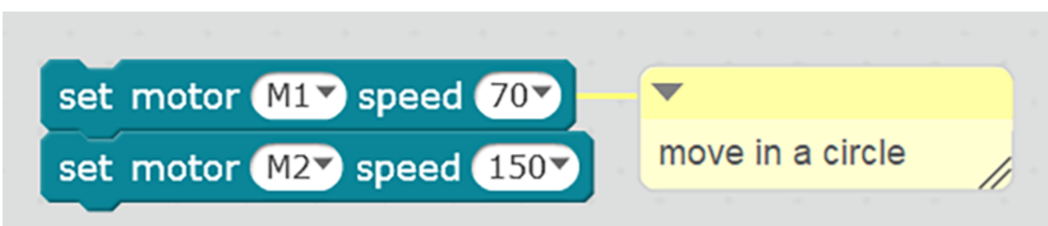
Spend some time discussing different ways your robot can turn in a circle.

Lab Manager Note: Check in with the teams after about 5 minutes of discussion. Teams may come up with the following ideas.

Turn off one motor while the other is running.



Make one motor slower than the other.



Lab Manager Note: The program starts as soon as the mBot is turned on. Some Scouts may not be consistent where they set the mBot and flip the switch. A simple solution is to add a wait block at the start of the program to give them a chance to turn it on and set it in place.

Challenge:

Write a program so your robot moves in a figure eight.

Write a program so your robot moves to spell your initials.

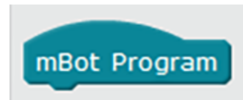
Meeting 2: Introduction to Programming

Activity 3: Move in a Square (25 minutes)

Remember to switch programmers so everyone has a chance to code.

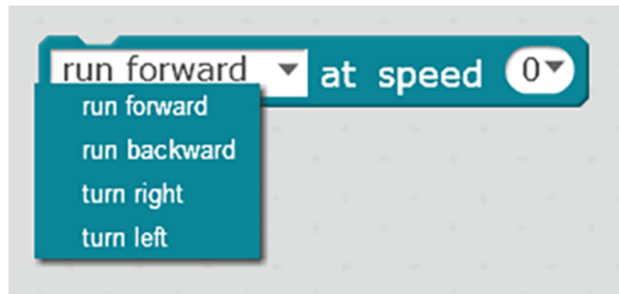
Step 1:

Remove your mBot and your connection cord from the box.



When programming the mBot, use **Scripts> Robots.**

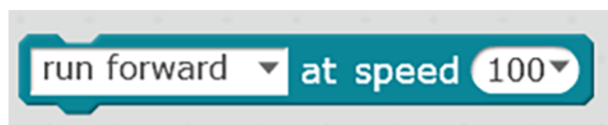
To start your program, use the mBot Program block. Then connect the other blocks below this block.



Robots are made to move, and in this activity you are going to learn how. The block that controls robot movement has two drop-down menus. The first menu controls the direction in which the robot moves:

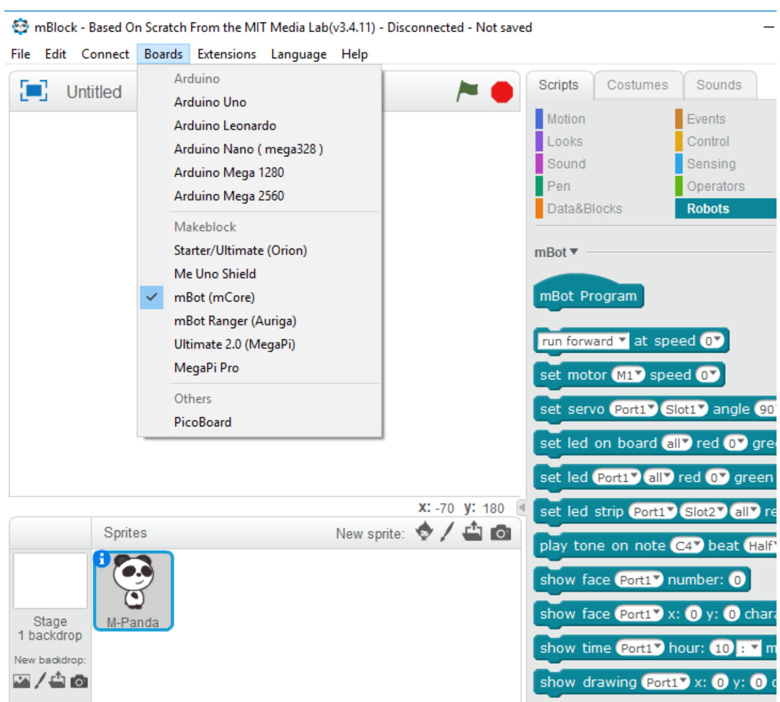
The second menu controls power. The top power is 255, 0 stops the motors, and negative numbers reverse direction. (NOTE: Very low power levels may not be strong enough to get the robot moving.)

The **Programmer** should select the following to get the robot moving forward at a speed of 100:



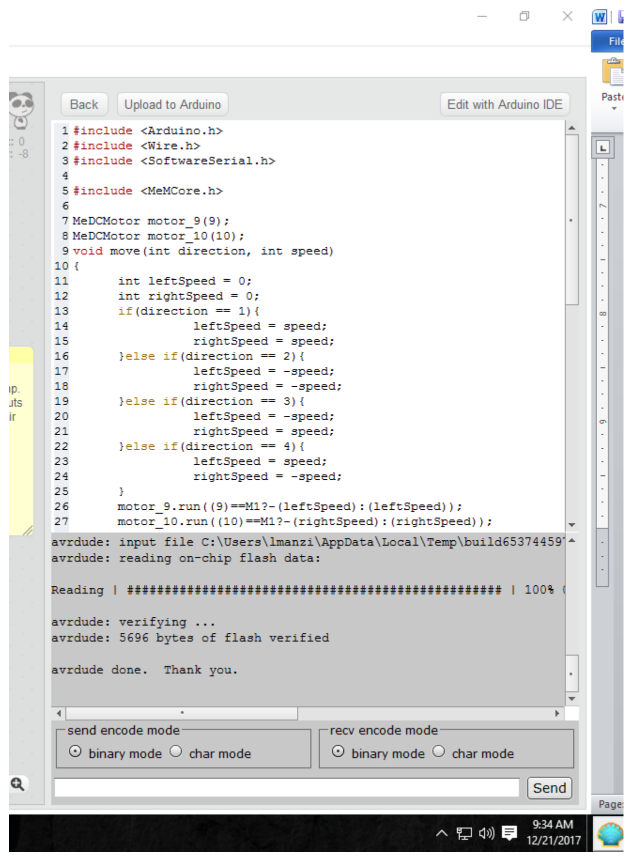
Connecting your mBot to run your programs:

- Plug your connection cord into a USB port on your computer.
- Plug the other end into your mBot.
- Hold your mBot and be careful to keep your hands away from the wheels.
- Turn on your mBot. (Note that the last program that was entered may start running.)
- On the mBlock program, open the **Boards menu** and select **mBot (mCore).**
- Next open the **Connect menu** and select **Serial Port.**



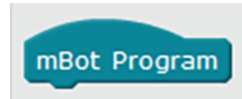
Meeting 2: Introduction to Programming

Serial ports will vary from computer to computer. To find the port for your mBot, see what ports are available with the mBot turned off. Then when the mBot is turned on, a new port should be visible. That is the port that should be selected.

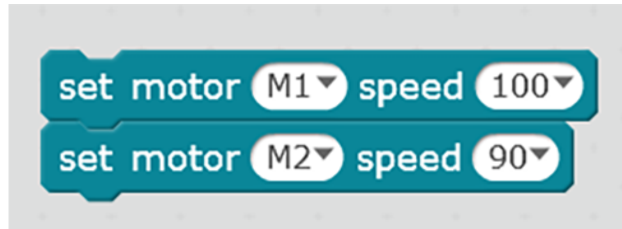


Meeting 2: Introduction to Programming

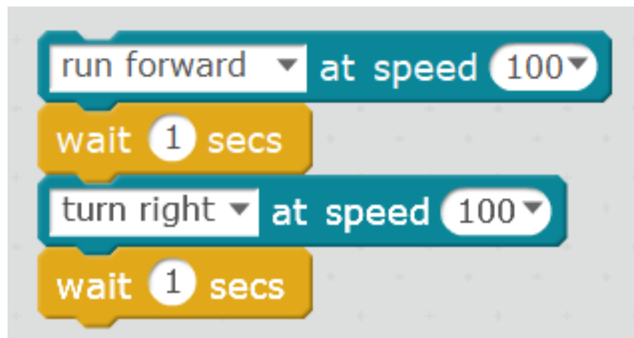
- Click on the mBot Program block to open Arduino mode.
- Click the Upload to Arduino button to upload your program to your mBot.



It is possible that the robot does not run perfectly straight. This could be for a few reasons, such as one of the wheels not being perfectly aligned or one of the wheels being more tightly attached to the robot, causing more friction. If this is the case, you could fix this by replacing the run forward block with two blocks setting the motor power levels separately:



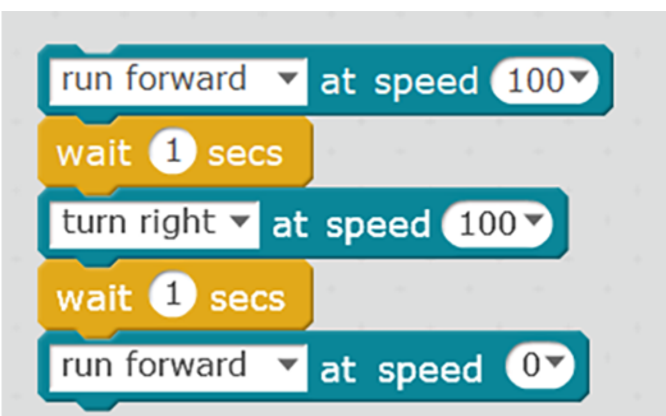
Programmer: If your robot is going straight, continue using the run forward block. You want to write a program that makes the mBot move in a square. So go forward for one second, then turn right. Try running the following program:



Does this program run as you wanted? Discuss with your team why it does not work and how you can correct it before moving forward.

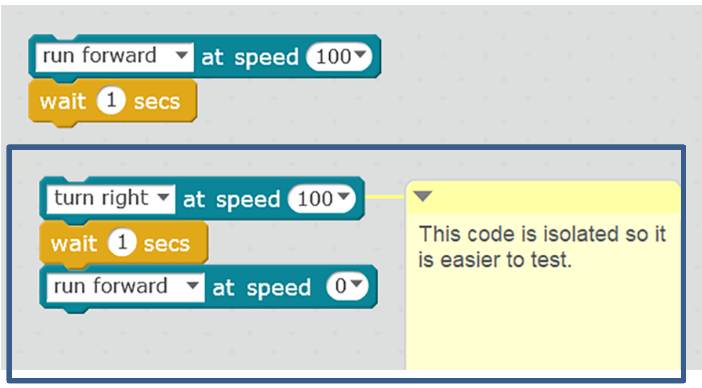
Lab Manager Note: The problems are that the program does not stop and the mBot is most likely not turning at a 90-degree angle. Scout teams should note that the robot runs the first command of the program and then finishes. The second command is to turn right. Then the robot waits for one second. Then the program ends. *At no point in the program are the motors turned off*. So the motors keep running and the mBot never stops. The Scouts need to add a line to stop the motors.

Programmer: You and your team should copy the following program:



Meeting 2: Introduction to Programming

Now your robot moved a little too far. One way that programmers test their programs is by isolating parts and running them separately. So if your team wants to test how far to turn, and that is unrelated to the part of the code that moves forward, you can isolate the turning part of the code. This will make testing easier and faster. See the image below:

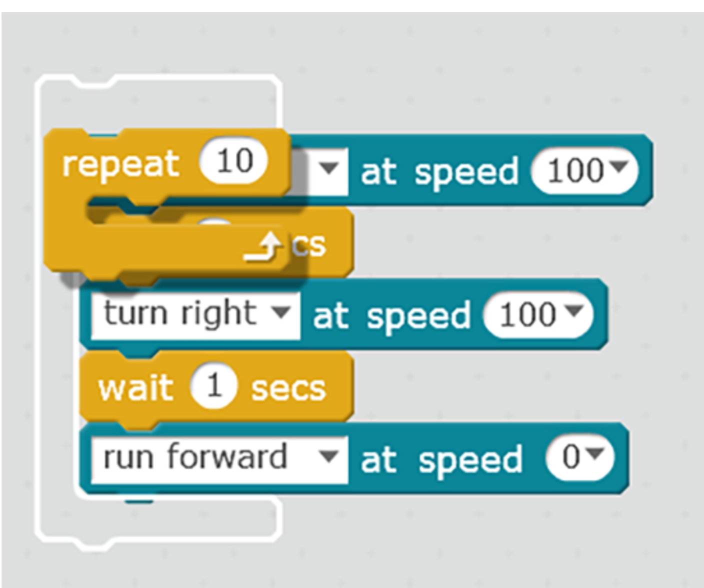


Discuss with your team what each team member thinks the timing should be for a 90-degree angle. Test your team's ideas on your robot.

Does your time to turn 90 degrees match any other team's time?

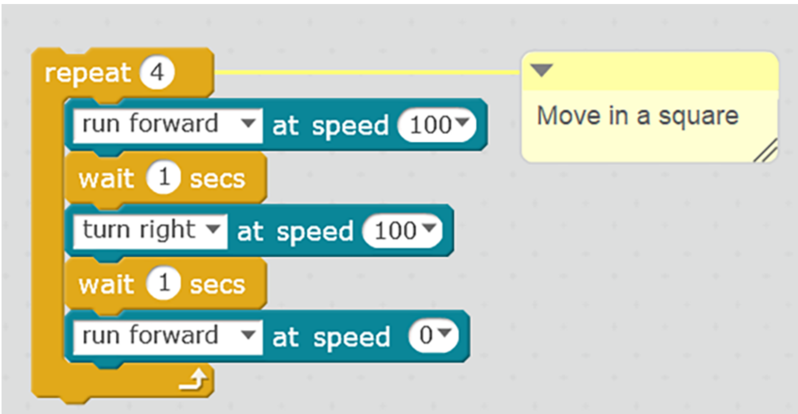
The time mBot needs to turn 90 degrees depends on many things. For example, the kind of battery you are using, how charged the battery is, how much friction there is between the motor and the wheel, and the speed setting of the turn will all affect the time needed to turn 90 degrees. So, when you do your tests, it is more than likely you will get a different time than other teams.

You should use loops to make your square. Loops are commonly used in programming languages to repeat sets of commands. Your program will need to go forward and turn four times to make four sides. There is a repeat block you can use to make a loop. Put all the things you want to be repeated inside the repeat block.



Meeting 2: Introduction to Programming

And change the number of repeats to 4:



When completed, you should have successfully made a square with your mBot! Congratulations!

If you finish early, try these challenges:

> Do you need the last movement command in the repeat loop? Can you make the code more streamlined so it still does what you want but with fewer commands issued?
>
> Can you write a program that moves the robot in a rectangle?
>
> Can you extend your program so that it gives a warning sound before starting and has lights on while moving?

Discussion

Lead a discussion on what the Scouts have done and their results. Ask open-ended questions to get them thinking about what they have done and learned, and what they might do differently next time.

Discussion Questions

What was your team's most difficult task?

What was your team's greatest success?

What was one way you solved a difficult problem?

Cleanup

Tell the Scouts:

Turn off your mBot, remove the batteries, and put the mBot in the box with the spare parts, remote control, and USB cord. Be careful not to damage the box, as it will house the mBot and its parts for the rest of this module.

Exit all applications on the laptop, fully power it down, and return it to the Program Manager.

Clean your area, and be sure no trace is left behind.

Meeting 2: Introduction to Programming

STEM Innovator Moment
The Scouts should hear from or about a person who has expertise on the topic of the day's activities about how this topic applies to everyday life and careers. The person could visit the Lab or use teleconferencing software such as Skype. You can also use the innovator information below.

> *Who was first to invent the computer? That is a question that is debated because there were several people who developed computer-type devices for different purposes. These machines all led to the development of modern-day computers.*
>
> Who Invented the Computer? (Computer History Museum): https://youtu.be/d1pvc9Zh7Tg (4:55)

Closing
Discuss an ethical situation that occurred during the meeting as well as how the Scout Oath and Scout Law could be applied to deal with the issue. Remember to focus on things that occurred among the Scouts rather than on controversial issues in science today.

Meeting 3: The Physics of Acceleration

Shutterstock.com, courtesy–©Vasilyev Alexandr

Meeting 3: The Physics of Acceleration

Pre-Meeting Leader Preparation

You will need to have enough open floor space for each team to build a ramp roughly a foot wide and 3 feet long with space around it for the mBot to go up and down the ramp.

Each team will need to use the mBot box as the center platform for their ramp.

If you are using the laptops without plugging them in, make sure they are fully charged well ahead of the meeting.

Materials and Tools Needed

- 1 laptop for Lab Manager (unit- or, optionally, council-supplied)
- 1 projector for Lab Manager (unit- or, optionally, council-supplied)

The following materials are shared between multiple teams.

- 1 pair of tweezers to tighten nuts if they become loose (shared between two teams)
- 1 roll of electrical tape for the teams to share (for ramp)

Meeting 3: The Physics of Acceleration

Scouts will be divided into teams of four for this set of experiments. Material quantities are defined for each team.

- 1 mBot kit (mBot already assembled from Meeting 1 PLUS mBot box)
- 1 small Phillips head screwdriver
- 4 AA batteries
- CR2025 button cell battery for remote control
- 1 laptop (unit- or, optionally, council-supplied)
- 4 7x7-inch pieces of cardboard (for ramp)
- 5-10 craft sticks to use as support for the ramp as needed

Opening

Have the Principal Investigator lead the group in reciting the Pledge of Allegiance and the Scout Oath and Scout Law.

Applying the Scout Law

Choose one of the 12 points of the Scout Law to discuss in application to today's lab. A suggested theme for this meeting is *brave*, as in *I will be brave and learn about how to control the mBot, even if I am not a big fan of physics*.

Ask the Scouts what being *brave* means to them and how they might apply that to today's lab.

Activity Overview

In this activity, Scouts will create a ramp and use acceleration and timing in programming to have their mBot go up the ramp, stop, and then go down the ramp with control. They will learn about the importance of iteration in the STEM Scouts Engineering Design Process.

Background

Physics and Acceleration of Ramps (Inclined Planes)

An inclined plane is one of the simple machines that was used in ancient history. An inclined plane—often called a ramp—has no moving parts. It allows us to move objects up and down an even sloping surface without having to move them directly upward or downward. Ramps are used to solve all sorts of problems.

Ancient Egyptians built pyramids of heavy stone. How did they get those huge blocks to the tops of pyramids? There has been much speculation, but the most likely scenario involves intricate ramps.

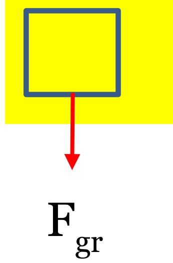
Lab Manager Note: Play this video for the Lab:

How Did the Egyptians Raise Huge Stoneworks? (Science Channel): www.youtube.com/watch?v=ZV-ULfy1XQo (1:28)

Meeting 3: The Physics of Acceleration

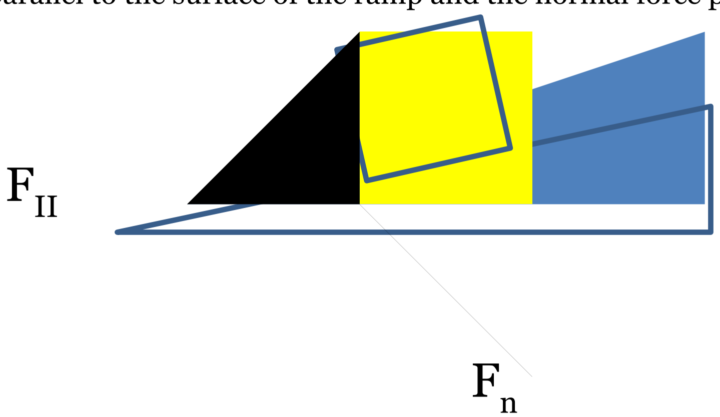
How does the physics of acce t this? Any object is subject to the acceleration of gravity, and that force is:

F_{object} = Mass * $F_{gravity}$



A 10,000-pound block of stone has a force on it of 320,000 foot-pounds/second. To lift it, you have to exert more than that force to get it off the ground.

When you put that same block on a ramp, that gravitational force generates two components—the force parallel to the surface of the ramp and the normal force pulling the block against the ramp.



The acceleration along the surface of the plane is: F_{II} = mass * g * sine θ/mass, where θ is the angle of the ramp. If we increase the slope of the ramp, the acceleration down the ramp is greater, and so the force needed to push the block up the ramp is greater.

If the slope of the ramp is gentle, the object will be pushed or pulled over a longer distance but with less effort. If the slope is steep, the object has to be pushed or pulled over a shorter distance but with more effort.

When the Scouts build their ramps, have them think about how the slope of the ramp and the speed of the mBot's motors will affect the ramp design.

The STEM Scouts Engineering Design Process is similar to the process that engineers use when designing a product, system, or environment to solve a problem. Designers do not always move step by step through the process. They often jump back and forth between steps as they move toward their final solution. This method consists of the following steps:

Meeting 3: The Physics of Acceleration

Step 1: Define the Problem (What is it that you're trying to accomplish?)

Step 2: Conduct Background Research (How have others before me accomplished this? What do I already know that can help?)

Step 3: Specify Requirements (What do I need to do in order to solve my problem from Step 1?)

Step 4: Create Alternative Solutions (Is there any other way to solve this problem?)

Step 5: Build a Prototype (Build something that meets your requirements from Step 3.)

Step 6: Test and Redesign as Necessary (Verify that your prototype does what you wanted it to, and adjust it if it doesn't. Be sure to write down any changes you make!)

Step 7: Communicate Results (Tell everyone else about the design you made to solve your problem. Diagrams are a bonus!)

Engaging Questions

Use some of the following questions to get the Scouts engaged. (*Typical answers are in italics*.)

Where are inclined planes or ramps used in your world and what purpose do they have? *(Answers will vary; some examples are wheelchair ramps, roofs, and delivery truck ramps.)*

What are the advantages of brainstorming a solution? *(Answers will vary; some examples are they provide different perspectives or different ideas.)*

Mistakes are often made in the design process. What should you do after you discover a mistake? *(Go back to brainstorm a solution to that problem.)*

Safety Moment

Tell the Scouts:

Make sure the mBot does not fall off the ramp. Make sure you don't step on the mBot or the ramp or trip over them.

Connect and disconnect the cable carefully when testing programs so that you don't damage the cable or connector.

If you have long hair, tie it back following lab safety rules. Also, if you have any long, dangling necklaces or similar items, take them off or put them inside your clothing. You don't want them to get caught in the mBot's wheels.

Meeting 3: The Physics of Acceleration

Experiment
Get the Scouts into the same teams of four that they were in for previous meetings.

Ask them to follow the instructions in their Lab Notebook, reproduced below.

Activity 1: Build a Ramp (20 minutes)

Materials List
- 1 pair of tweezers to tighten nuts if they become loose (shared between two teams)
- 1 mBot kit (mBot already assembled from Meeting 1 PLUS mBot box)
- 1 small Phillips head screwdriver
- 4 AA batteries
- CR2025 button cell battery for remote control
- 4 7x7-inch pieces of cardboard (for ramp)
- 1 roll of electrical tape for the teams to share (for ramp)
- 5-10 craft sticks to use as support for the ramp as needed

Step 1: Define the Problem

Engineers use their science, technology, engineering, and math skills to solve problems. Machines often have to drive over difficult terrain, such as rivers, or around or over obstacles that are impossible to move. You and your team will use your skills to solve this problem: **How do we get our mBot to drive up and over an obstacle?**

In today's activity, you will design a ramp that will allow your team to drive an mBot up and over an imaginary obstacle. Working with your team, you will go through the engineering design process to design a ramp and a program that will take your mBot to the other side.

There is a challenge in this: You want the mBot to be able to go over the highest obstacle possible without falling off the ramp.

Lab Manager Note: Ensure that the teams understand that the problem is: How do we get our mBot to go up and over an obstacle?

Step 2: Conduct Background Research

With your team, discuss the following issues:
- How steep should the ramp be? (Various angles; if it's too steep, the robot will tip.)
- At what speed should the robot move? (Programming speeds are 50, 100, and 255.)

How will you figure this out?

Lab Manager Note: The angle of the ramp can be steeper if the speed is higher. However, at some point the robot will tip backward if the ramp is too steep. Look for the Scouts to come up with some tests using the remote control to see how well the mBot goes up ramps of different angles at different speeds as a part of their solutions and testing. If they don't think of this, hold off saying anything until they truly get stuck.

Meeting 3: The Physics of Acceleration

Step 3: Specify Requirements

Your mBot must go up the ramp, stop at the top, then proceed down the other side without falling off the side or tipping over.

Discuss what your solution criteria might look like. You will probably have some different ideas.

What resources are available? *(4 7x7-inch pieces of cardboard, the mBot box, electrical tape, craft sticks)*

What will a successful outcome look like? *(When a ramp is built that allows the robot to go up, stop, and go down and when a program is written to control the robot.*)

Lab Manager Note: The simplest ramp is to use the mBot box as the center platform with two pieces of cardboard, taped together with an overlap, as the ramps, one from the ground to the top of the box and the other from the top of the box to the ground on the other side. The craft sticks are available for support if the cardboard sags where pieces are taped together.

Step 4: Create Alternative Solutions

Spend a few minutes brainstorming ideas for building your ramp. When you and your team think you have a good design, write it down as your primary solution to be built first as your prototype.

What are some other suggestions for ramps and different programming options? Do you need to do any experiments to optimize your design?

Step 5: Build a Prototype

Build your primary solution.

Activity 2: Program Your mBot (40 minutes)

Remember to switch programmers so everyone has a chance to code.

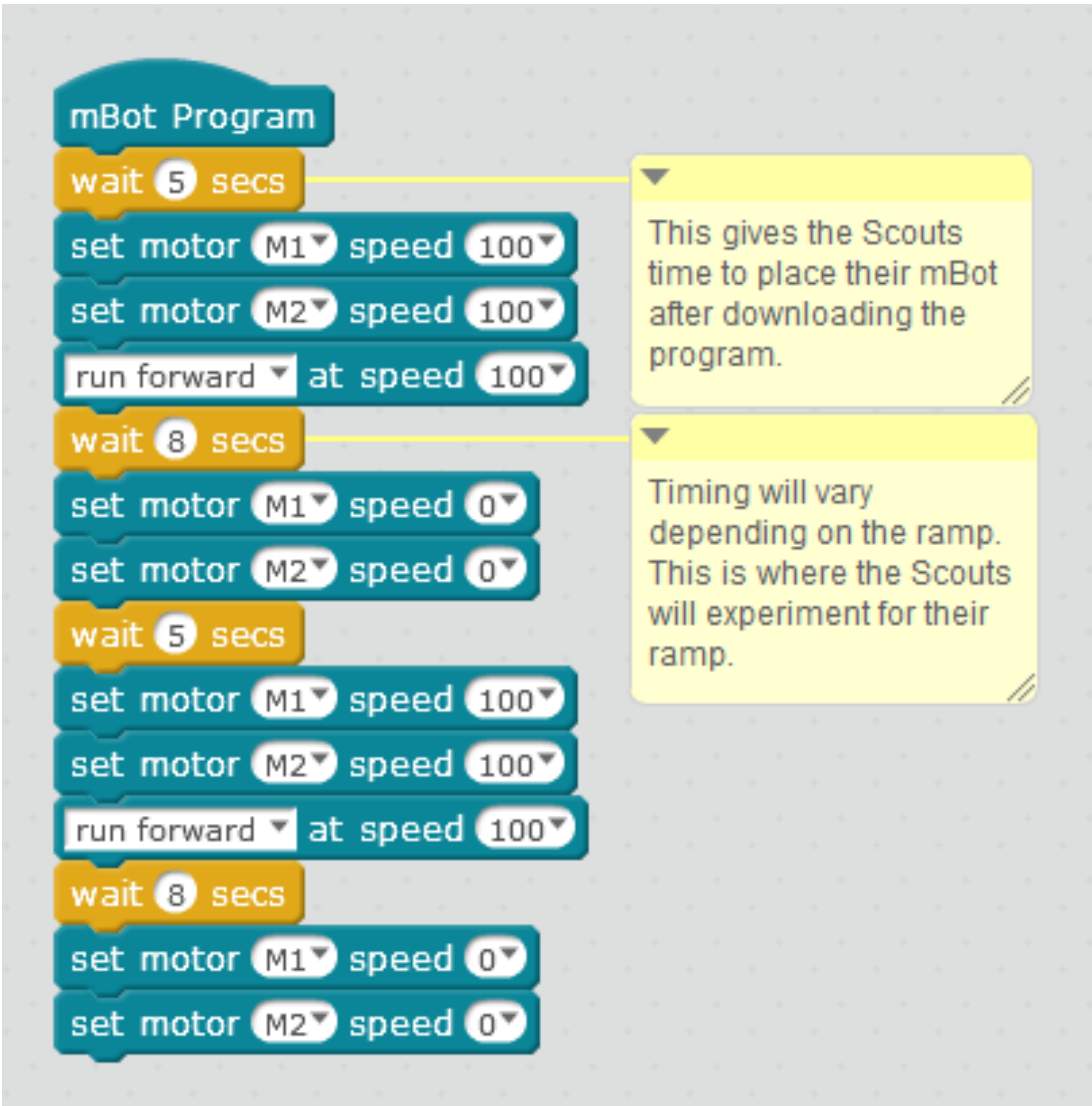
Materials List

- 1 pair of tweezers to tighten nuts if they become loose (shared between two teams)
- 1 mBot kit (mBot already assembled from Meeting 1 PLUS mBot box)
- 1 small Phillips head screwdriver
- 4 AA batteries
- CR2025 button cell battery for remote control
- 4 7x7-inch pieces of cardboard (for ramp)
- 1 roll of electrical tape for the teams to share (for ramp)
- 5-10 craft sticks to use as support for the ramp as needed
- 1 laptop

Now is the time to test out your ramp and your programming skills. Your job is to design a program so the mBot moves forward up the ramp and then stops at the top of the ramp to enjoy the view. After a few seconds, the mBot should go down the ramp.

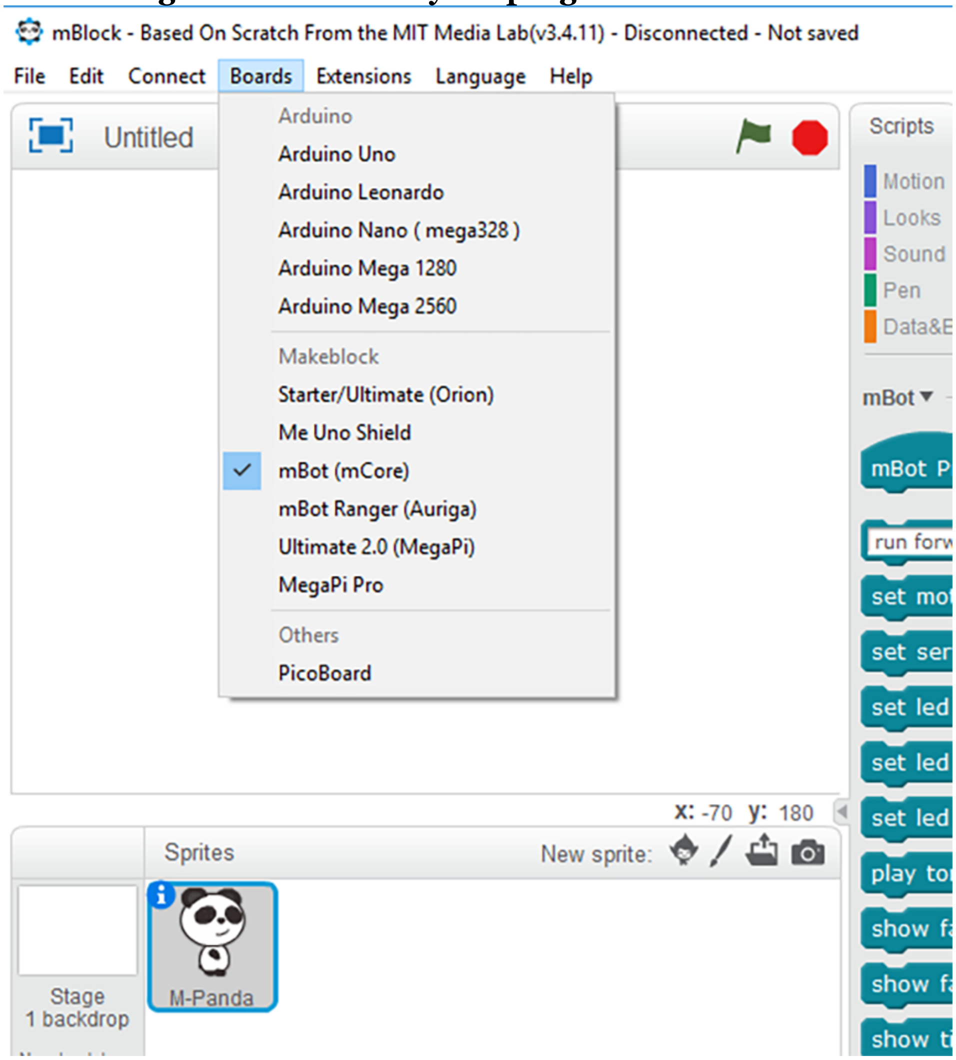
Meeting 3: The Physics of Acceleration

Lab Manager Note: Some sample programming ideas are provided for your information below.



Meeting 3: The Physics of Acceleration

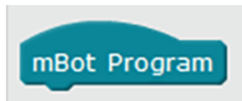
Connecting the mBot to run your programs:



- Plug your connection cord into a USB port on your computer.
- Plug the other end into your mBot.
- Hold your mBot and be careful to keep your hands away from the wheels.
- Turn on your mBot. (The last program that was entered may start running.)
- On the mBlock program, open the **Boards menu** and select **mBot (mCore).**
- Next open the **Connect menu** and select **Serial Port.**

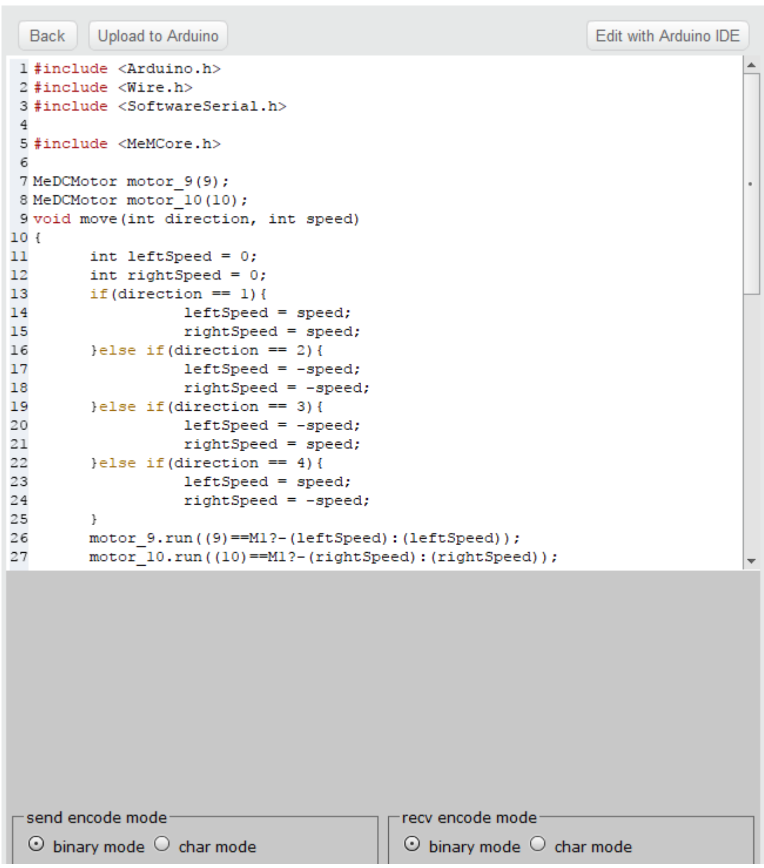
Meeting 3: The Physics of Acceleration

Lab Manager Note: Serial ports will vary from computer to computer. To find the port for your mBot, see what ports are available with the mBot off. Then when the mBot is on, a new port should be visible. That is the port that should be selected.

- Click on the mBot Program block 

 to open Arduino mode.

- Click the **Upload to Arduino** button to upload your program to your mBot.



Step 6: Test and Redesign as Necessary

Review your results and modify your ramp design and/or program to optimize your solution.

Meeting 3: The Physics of Acceleration

Step 7: Communicate Results

Be prepared to present to the rest of the teams how you structured your experimental procedure and then present your results and conclusions.

Extensions:

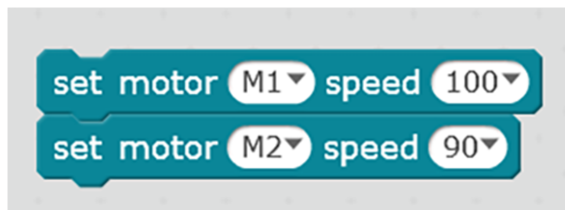
- How would you program your mBot to go up a second ramp?
- What happens when you decrease or increase the speed of your motors?

Troubleshooting Problems:

When uploading the program, make sure that your robot is either propped up so the wheels aren't in contact with a surface or in a space where it can move freely.

Please make sure it doesn't fall off any desks or tables.

It is possible that your robot does not run perfectly straight. This could be for a few reasons, such as one of the wheels not being perfectly aligned or one of the wheels being more tightly attached to the robot, causing more friction. If this is the case, you could fix this by replacing the run forward block with two blocks setting the motor power levels separately:



Discussion

Lead a discussion on what the Scouts have done and their results. Ask open-ended questions to get them thinking about what they have done and learned, and what they might do differently next time.

Discussion Questions

Why is timing important when programming?

Many programs do not work the first time they are developed. If your program did not work as expected the first time you ran it, what did your team do?

Cleanup

Turn off your mBot, remove the batteries, and put the mBot in the box with the spare parts, remote control, and USB cord. Be careful not to damage the box, as it will house the mBot and its parts for the rest of this module.

Exit all applications on the laptop, fully power it down, and return it to the Program Manager.

Take the ramps apart carefully and save the materials; you will need them for Meeting 6.

Clean your area, and be sure no trace is left behind.

Meeting 3: The Physics of Acceleration

STEM Innovator Moment
The Scouts should hear from or about a person who has expertise on the topic of the day's activities about how this topic applies to everyday life and careers. The person could visit the Lab or use teleconferencing software such as Skype. You can also use the innovator information below.

Robotics is being used in many ways to help people who have lost physical abilities. David Gow, the inventor of the i-limb, shows how robotics is being used in the field of prosthetics.

Advanced bionic hand is changing lives (touchbionics):
www.youtube.com/watch?v=0Su7LTkAmCQ&feature=youtu.be (2:29)

Closing
Discuss an ethical situation that occurred during the meeting as well as how the Scout Oath and Scout Law could be applied to deal with the issue. Remember to focus on things that occurred among the Scouts rather than on controversial issues in science today.

Meeting 4: Programming With Math, Light, and Sound

Shutterstock.com, courtesy–©Shaiith

Meeting 4: Programming With Math, Light, and Sound

Pre-Meeting Leader Preparation

The first part of this lab will measure the reaction of the light sensor on the mBot to the amount of light available. Look into whether you can vary the amount of light in your meeting space. The teams have a flashlight in their kits for very low light levels (assuming you can turn off the lights completely).

If you are using the laptops without plugging them in, make sure they are fully charged well ahead of the meeting.

Materials and Tools Needed

- 1 laptop for Lab Manager (unit- or, optionally, council-supplied)
- 1 projector for Lab Manager (unit- or, optionally, council-supplied)

The following materials are shared between multiple teams.

- 1 pair of tweezers to tighten nuts if they become loose (shared between two teams)

Meeting 4: Programming With Math, Light, and Sound

Scouts will be divided into teams of four for this set of experiments. Material quantities are defined for each team.

- 1 mBot kit (mBot already assembled from Meeting 1)
- 1 small Phillips head screwdriver
- 4 AA batteries
- CR2025 button cell battery for remote control
- 1 small flashlight
- 1 laptop (unit- or, optionally, council-supplied)

Opening
Have the Principal Investigator lead the group in reciting the Pledge of Allegiance and the Scout Oath and Scout Law.

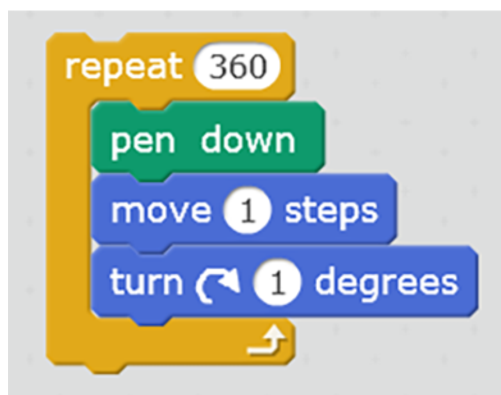
Applying the Scout Law
Choose one of the 12 points of the Scout Law to discuss in application to today's lab. A suggested theme for this meeting is *courteous*, as in *I will be courteous to everyone and not disrupt their measurements by shining my flashlight at their robot or disturbing their measurements in any way*.

Ask the Scouts what being *courteous* means to them and how they might apply that to today's lab.

Activity Overview
The Scouts will use the mBot light sensor and the ultrasonic sensor to collect data. They will use this data to program the mBot to have different reactions. Using the data they collect, they will be able to create an interactive robot and will be challenged to show how their mBot can respond.

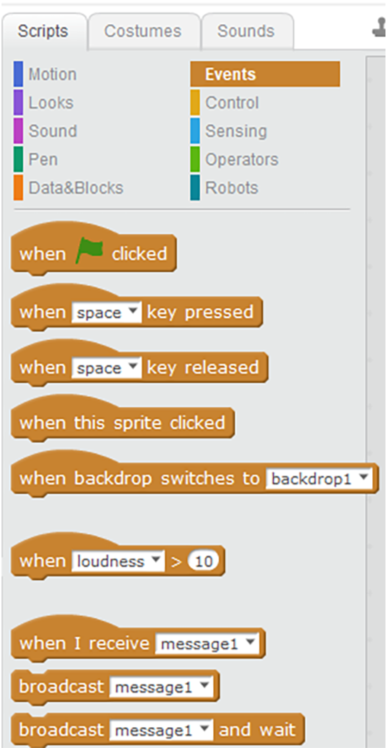
Background
Math is one of the most important elements in computer science. In addition to basic logic, math, and algebra, computer scientists use other computational concepts that are common to many programming languages. The Scratch programs identify seven concepts, which are highly useful in a wide range of Scratch projects and which transfer to other programming (and nonprogramming) contexts. They are:

- Sequence: identifying a series of steps for a task
- Loops: running the same sequence multiple times (For example, this program will repeat to make a circle.)
- Parallelism: making things happen at the same time



Meeting 4: Programming With Math, Light, and Sound

- Events: one thing causing another thing to happen



- Conditionals: making decisions based on conditions

Meeting 4: Programming With Math, Light, and Sound

- Operators: support for mathematical and logical expressions

- Data: storing, retrieving, and updating values

Data, or input, is needed in order to use math and logic in your programs. One way to collect data is by using sensors. This data can be used in a program to generate responses. Scouts will collect data from the mBot's sensors to create a program that uses math and logic to generate a response from their mBot.

Engaging Questions

Use some of the following questions to get the Scouts engaged. (*Typical answers are in italics.*)

How is math used in a video game that changes levels based on the score? *(Once the score is greater than the set value, it will switch to the next level.)*

"**IF** the light is on **THEN** drive forward" is an example of what concept? *(Conditional)*

What are some examples of how these concepts are used outside of programming? *(Answers will vary.)*

Safety Moment

Tell the Scouts:

Be aware of other mBots when operating your mBot.

When using the light sensor, there may be times when the area needs to be darker. Be aware of your surroundings and other mBots.

Connect and disconnect the cable carefully when testing programs so that you don't damage the cable or connector.

If you have long hair, tie it back following lab safety rules. Also, if you have any long, dangling necklaces or similar items, take them off or put them inside your clothing. You don't want them to get caught in the mBot's wheels.

Meeting 4: Programming With Math, Light, and Sound

Experiment

Get the Scouts into the same teams of four that they were in for previous meetings.

Ask them to follow the instructions in their Lab Notebook, reproduced below.

Activity 1: Reading Sensors (10 minutes)

Remember to switch programmers so everyone has a chance to code.

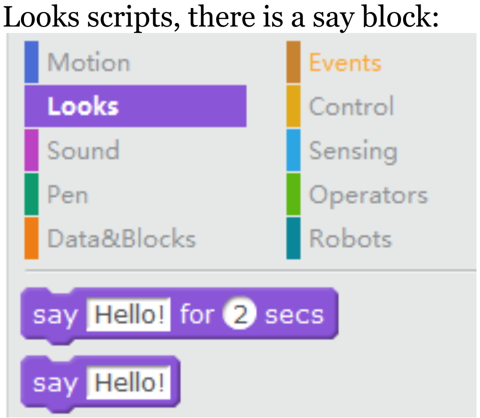
Materials List

- 1 mBot kit (mBot already assembled from Meeting 1)
- 1 small Phillips head screwdriver
- 4 AA batteries
- CR2025 button cell battery for remote control
- 1 small flashlight
- 1 laptop

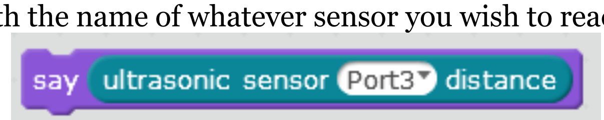
Put the batteries back into the mBot.

In this activity, you will use your mBot and the mBlock software to see how the sensors collect data. Your program/sprite will tell you what values the sensors are reading.

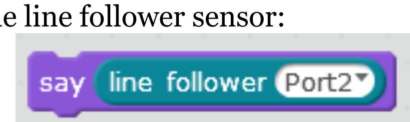
The easiest way to find out what value a sensor is giving is to have a sprite (panda by default) "say" it. In the Looks scripts, there is a say block:



If you ask the panda to say "Hello!", the panda is going to say "Hello!" Fortunately, you can replace this text with the name of whatever sensor you wish to read. This is for the ultrasonic sensor:

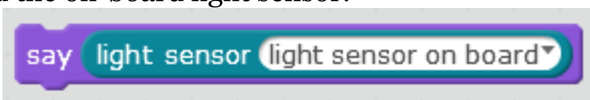


The line follower sensor:



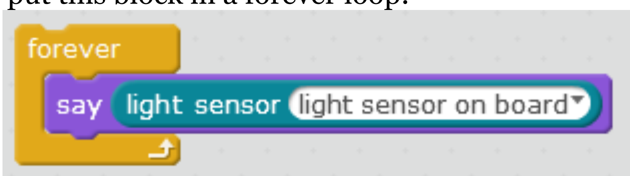
Meeting 4: Programming With Math, Light, and Sound

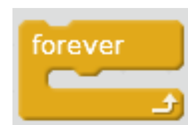
And the on-board light sensor:



For the plug-in sensors, make sure the port selected is the same port you have plugged your sensor into.

Now, you can click this block and it will tell you the value of the sensor. To make it continually update, you can put this block in a forever loop:





As you can see, the shape of the forever block is different from the shape of the other blocks you have seen so far. That is because this is a control block that allows the program to enter a loop. In programming, a loop is a feature that allows a part of the code to be repeated.

Using the above code, add a “when green flag clicked” event to the top. Connect to your serial port.
***TIP:** If program does not load, RESET THE DEFAULT PROGRAM.

Light Sensor Collection Activity		
Challenge	***Value***	***How did you get your result?***
Lights on		Turned on all the lights
Lights off		Turned off all the lights
Brightest result		
Darkest result		

Meeting 4: Programming With Math, Light, and Sound

Once you have completed working with the light sensor, do the same for the line follower and ultrasonic sensors to map out the limits of all three sensors and record them. As a team, figure out how to test these other sensors and make up your own challenges.

Line Follower Collection Activity		
Challenge	***Value***	***How did you get your result?***

Ultrasonic Collection Activity		
Challenge	***Value***	***How did you get your result?***

Meeting 4: Programming With Math, Light, and Sound

Activity 2: Light Sensor (30 minutes)

Remember to switch programmers so everyone has a chance to code.

Materials List

- 1 mBot kit (mBot already assembled from Meeting 1)
- 1 small Phillips head screwdriver
- 4 AA batteries
- CR2025 button cell battery for remote control
- 1 small flashlight
- 1 laptop

The light sensor on the mCore board measures light—the brighter it is, the higher the value; the darker it is, the lower the value. The range of the light sensor signal is 0 to 1023.

You are going to program the mBot to play one note if it senses there is a lot of light and play a different note if it senses there is not a lot of light. For this, you will need the if block. There are three parts to an if block:

1. The "if" itself
2. The condition
3. The code that is run if the condition is met

Step 1: Conditions for Sound

You need to write something like this:

If (condition)
---- Do this code

The first thing you want to do in this activity is to play a particular sound if it is dark. The condition is "it is dark," and the code is "play sound":

If (it is dark)
---- Play sound.

Meeting 4: Programming With Math, Light, and Sound

Remember: The light sensor returns a value from 0 to 1023, where high numbers represent a light environment. As the programmer, you have to choose what value represents “dark.” For this example, a light sensor reading of less than 500 will represent a dark environment. You can then write the code as:

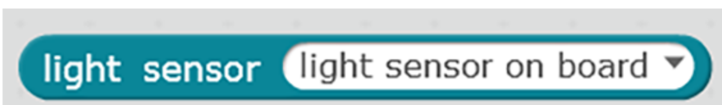
If (the value on the light sensor IS LESS THAN 500)
---- Play sound.

Writing this in mBlock code, you need:

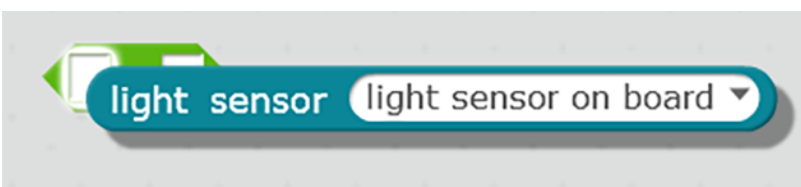
1. An “if”

2. A condition. The condition in this example is made up of two parts. You need the green less than block in the Operators script:

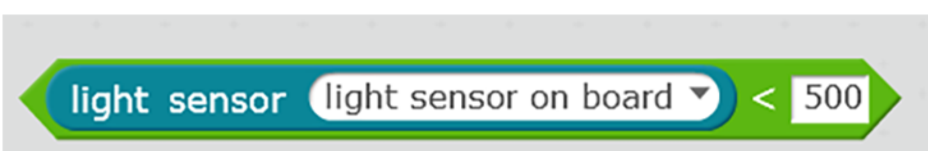
And you need the light sensor block from the Robots script:



The condition is “the light sensor is less than 500,” so you can put the light sensor block in the left side of the less than block and enter “500” in the right side. When putting the blue block inside the green block, make sure the left side of the blue block is aligned with the appropriate space of the green block. When the space in the green block has a white outline, it is ready to receive the blue block:

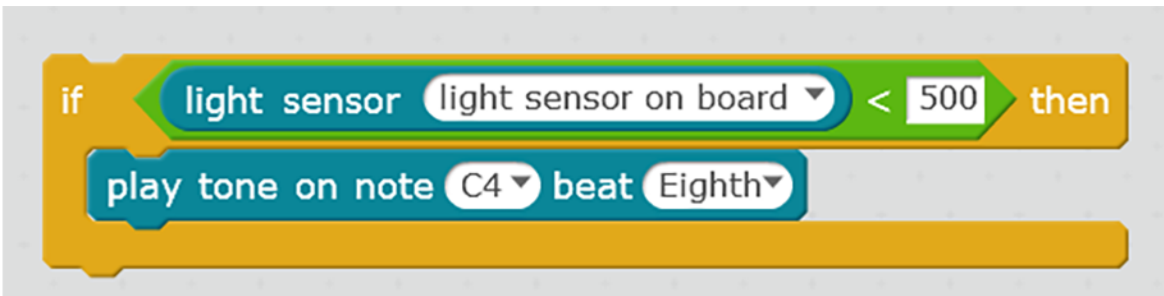


When dropped in, it looks like this:



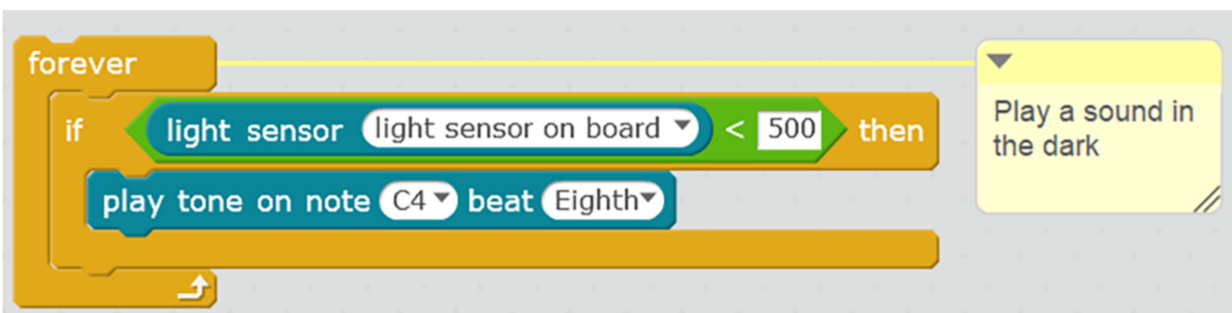
3. Some code that will be run if the condition is met:

When this code is put together, it looks like this:



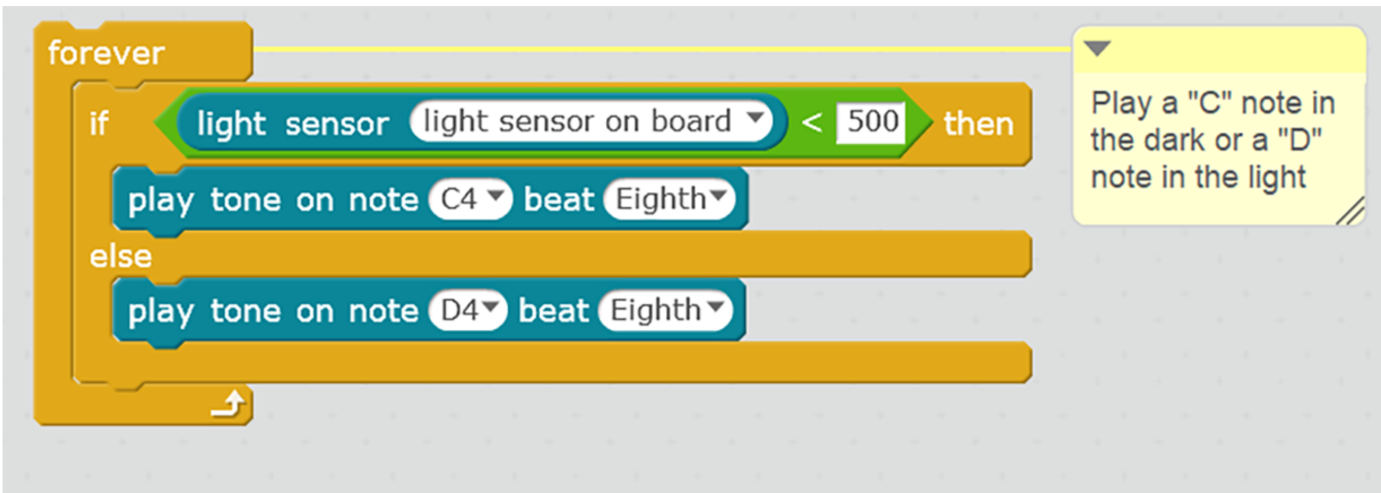
When this code runs, it will test the condition once and then stop running. If you want your mBot to continue to play music while it's in a dark environment, you can add a forever loop. It is always a good practice to include a descriptive comment. Comments like this help you to remember why you put something in your code, making it easier to debug and maintain the code later on.

To add a comment, right-click on the code and select **Comment.**



Meeting 4: Programming With Math, Light, and Sound

What if you wanted to play note C4 in the dark but note D4 in the light? For this, you could use an if/else control block:



Lab Manager Note: This activity has fewer step-by-step instructions. It is designed to have the Scouts try different programs and options. They should be reminded that mistakes are a great way to learn and are common in programming.

Using the table provided in the Lab Notebook, write what light sensor data values will make your mBot respond under the following challenges.

Challenges:

- Develop a program where the robot runs forward in the light and stops in the dark.
- Develop a program where the robot dances in the light (be creative with your dance moves) and rests in the dark.
- Develop a program where the robot goes at different speeds depending on the light—the lighter it is, the faster it goes.
- Develop a program that plays lots of different notes—the more light sensed, the higher the note.

Light Sensor Data Values		
Challenge	***Value***	***How did you get your result?***
Runs forward in light, stops in dark		
Dances in light, rests in dark		
Goes faster with more light		
Plays higher note with more light		

Meeting 4: Programming With Math, Light, and Sound

Discussion for Light

Lead a discussion on what the Scouts have done and their results. Ask open-ended questions to get them thinking about what they have done and learned, and what they might do differently next time.

Discussion Questions

What did you find out about how much light it takes to make the light sensor react?

Would any of you like to demonstrate one of the programs you developed using the light sensor?

Meeting 4: Programming With Math, Light, and Sound

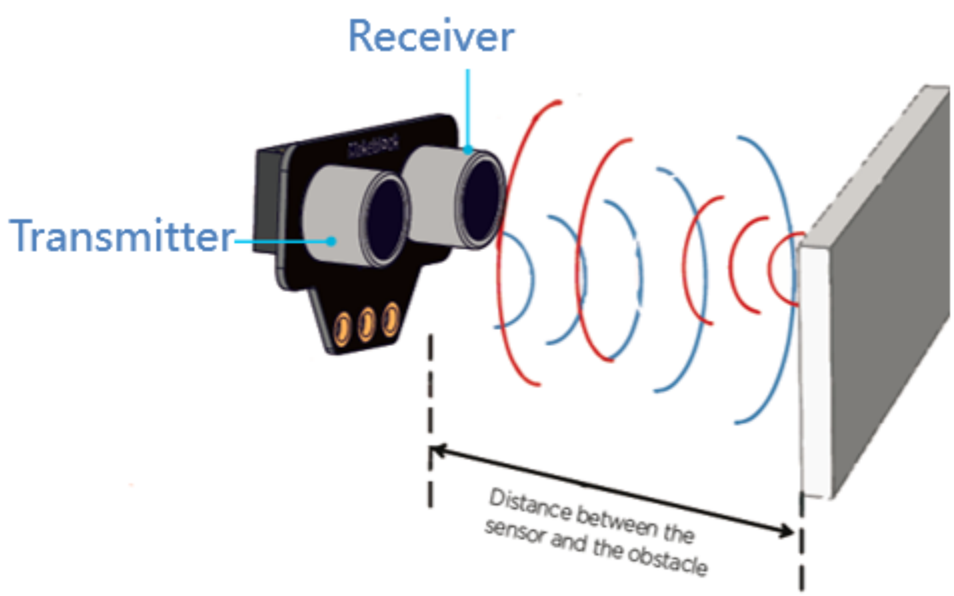
Activity 3: Ultrasonic Sensor (30 minutes)

Remember to switch programmers so everyone has a chance to code.

Materials List

- 1 mBot kit (mBot already assembled from Meeting 1)
- 1 small Phillips head screwdriver
- 4 AA batteries
- CR2025 button cell battery for remote control
- 1 small flashlight
- 1 laptop

The ultrasonic sensor measures distance. One of the "eyes" transmits a sound, and the other waits for the echo of the sound to return. In the time this process takes, the distance of the object from the sensor can be calculated. The ultrasonic sensor has a range of 3 to 400 centimeters. If an object is outside this range, the sensor will return a value of 400.

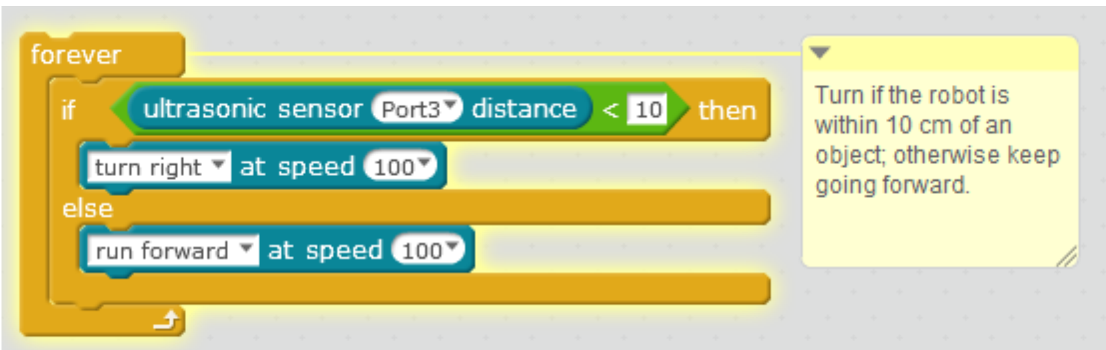


Meeting 4: Programming With Math, Light, and Sound

Have the robot go forward until it is close to an object, then turn away from the object and go off in a new direction:

Forever:
---- If an object is detected
---- ---- Turn to a new direction
---- Else
---- ---- Go forward

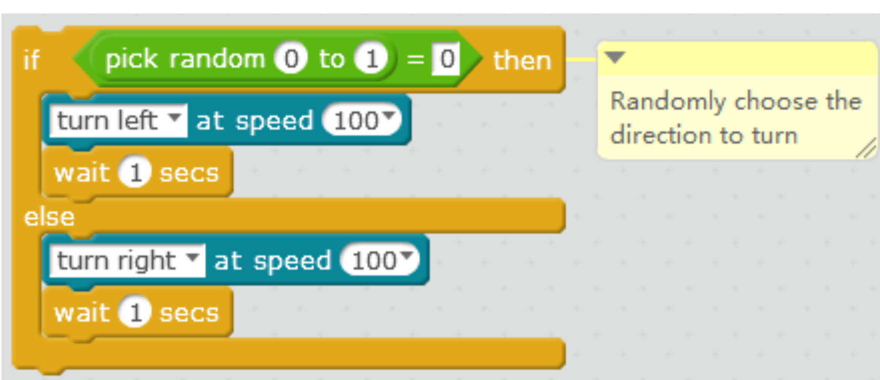
In code, that looks like this:



On testing this, you will find that the robot isn't turning fast enough when it encounters an object, so you can add a wait command after the turn.

Now, let's say you want it to turn in a random direction—sometimes left and sometimes right. Fortunately, there is an operator block that you can use to do just that:

The robot can turn only left or right, so choose a random number from 0 to 1. If the random number is 0, the robot will turn left. If the random number is 1, the robot will turn right. So then the turning behavior code will look like this:



Meeting 4: Programming With Math, Light, and Sound

Challenges:

- Can you put the code above together to make the complete program?
- Can you write a program that follows an object? If the object is too close, the robot goes backward; if the object is far away, it goes forward; and if the object is not near or far, the robot stops.
- Can you change your program from the previous challenge to make the robot move at different speeds?

Discussion for Ultrasound

Lead a discussion on what the Scouts have done and their results. Ask open-ended questions to get them thinking about what they have done and learned, and what they might do differently next time.

Discussion Questions

Why do you use a range or a less than or greater than operator when working with sensors? Why not use an exact number?

Would any of you like to demonstrate one of the programs you developed using the ultrasound sensor?

Cleanup

Turn off your mBot, remove the batteries, and put the mBot in the box with the spare parts, remote control, and USB cord. Be careful not to damage the box, as it will house the mBot and its parts for the rest of this module.

Exit all applications on the laptop, fully power it down, and return it to the Program Manager.

Clean your area, and be sure no trace is left behind.

STEM Innovator Moment

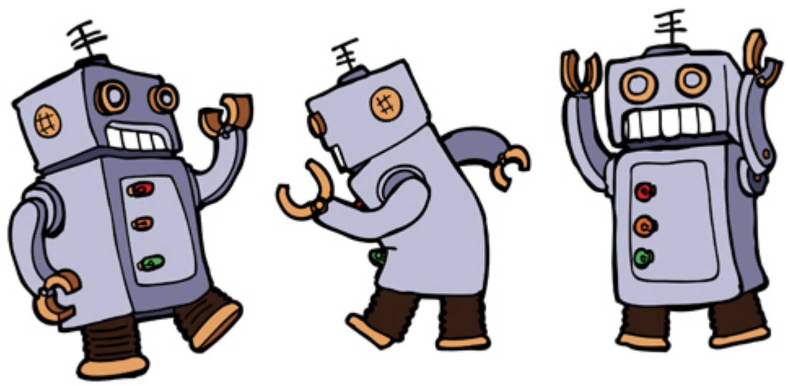
The Scouts should hear from or about a person who has expertise on the topic of the day's activities about how this topic applies to everyday life and careers. The person could visit the Lab or use teleconferencing software such as Skype. You can also use the innovator information below.

> *Austin Russell is the CEO of Luminar Technologies, a company he founded when he was 17 years old. He has developed an advanced lidar sensor that helps self-driving vehicles see and hear the world around them. Lidar is similar to radar, only it uses lasers instead of radio waves.*
>
> Meet Luminar (Luminar): https://vimeo.com/228269192 (1:47)

Closing

Discuss an ethical situation that occurred during the meeting as well as how the Scout Oath and Scout Law could be applied to deal with the issue. Remember to focus on things that occurred among the Scouts rather than on controversial issues in science today.

Meeting 5: Programming With Music



Shutterstock.com, courtesy–©lineartestpilot

Meeting 5: Programming With Music

Pre-Meeting Leader Preparation

In this meeting, the Scouts will modify an existing program and use it. Before the meeting, download the following program:

https://github.com/Matt-Ma/Makeblock-Tutorials/raw/master/Singing%20and%20Dancing%20mBot.sb2.

Save it to the desktops of the teams' computers so it will be easy to access.

Materials and Tools Needed

- 1 laptop for Lab Manager (unit- or, optionally, council-supplied)
- 1 projector for Lab Manager (unit- or, optionally, council-supplied)

The following materials are shared between multiple teams.

- 1 pair of tweezers to tighten nuts if they become loose (shared between two teams)

Meeting 5: Programming With Music

Scouts will be divided into teams of four for this set of experiments. Material quantities are defined for each team.

- 1 mBot kit (mBot already assembled from Meeting 1)
- 1 small Phillips head screwdriver
- 4 AA batteries
- CR2025 button cell battery for remote control
- 1 laptop
- 1 small flashlight

Opening
Have the Principal Investigator lead the group in reciting the Pledge of Allegiance and the Scout Oath and Scout Law.

Applying the Scout Law
Choose one of the 12 points of the Scout Law to discuss in application to today's lab. A suggested theme for this meeting is *thrifty*, as in *I will be thrifty in my coding and try to make my programs simple and easy to understand.*

Ask the Scouts what being *thrifty* means to them and how they might apply that to today's lab.

Activity Overview
In this activity, Scouts will learn to program LED lights to make lights change colors and to create musical sounds with their mBot. Each team will develop a unique song-and-dance routine for their mBot.

Background
Art and technology are used in many ways in our world today and can be found in many different industries. Everywhere we look we are surrounded by sights and sounds that were created by artists and then enhanced by computer programming. Programming allows us to modify and enhance an image or sound.

Lab Manager Note: Show the following videos:

Digital Compression explained by Aloe Blacc (Code.org): www.youtube.com/watch?v=By30SCp-Tsw (3:23)

Images, Pixels, and RGB (Code.org): www.youtube.com/watch?v=15aqFQQVBWU (5:49)

Meeting 5: Programming With Music

Engaging Questions
Use the following questions, and any others of your choosing, to get the Scouts engaged. (*Typical answers are in italics*.)

When you stretch out an image on your screen, why does it get distorted? *(The pixels are the same size but are spread out, making the image blurry.)*

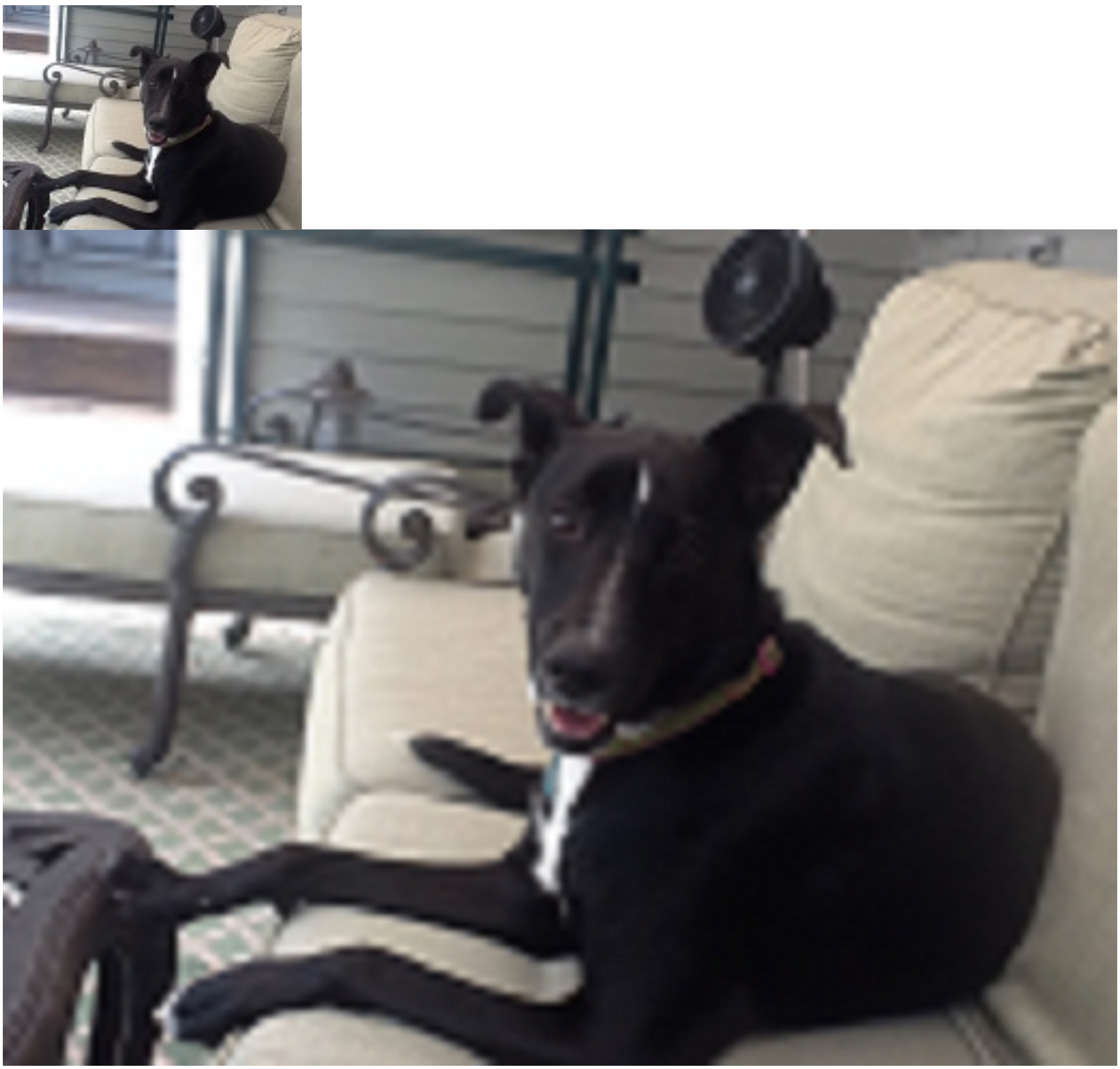


Image by Lisa Manzi

Meeting 5: Programming With Music

What are some ways sounds are programmed to keep us safe? *(Possible answers: fire alarm, crosswalk beeps, clocks, and timers)*

How are music and sounds used in video games and why are they important? (*Possible answers: graphics, sound effects.)*

Safety Moment
Tell the Scouts:

Be aware of other mBots when operating your mBot.

Connect and disconnect the cable carefully when testing programs so that you don't damage the cable or connector.

If you have long hair, tie it back following lab safety rules. Also, if you have any long, dangling necklaces or similar items, take them off or put them inside your clothing. You don't want them to get caught in the mBot's wheels.

Experiment
Get the Scouts into the same teams of four that they were in for previous meetings.

Ask them to follow the instructions in their Lab Notebook, reproduced below.

Activity 1: The LED Display (20 minutes)

Remember to switch programmers so everyone has a chance to code.

Materials List

- 1 mBot kit (mBot already assembled from Meeting 1)
- 1 small Phillips head screwdriver
- 4 AA batteries
- CR2025 button cell battery for remote control
- 1 laptop

There are two RGB LEDs on the mCore. RGB stands for red-green-blue and LED stands for light-emitting diode. Each LED can be assigned a level of red light, a level of green light, and a level of blue light from 0 (light turned off) to 255 (light turned on full power). By combining these lights, you can make a wide range of different colors:

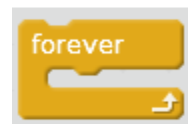
To start playing with the on-board LEDs, you need to drag and drop this block:

Meeting 5: Programming With Music

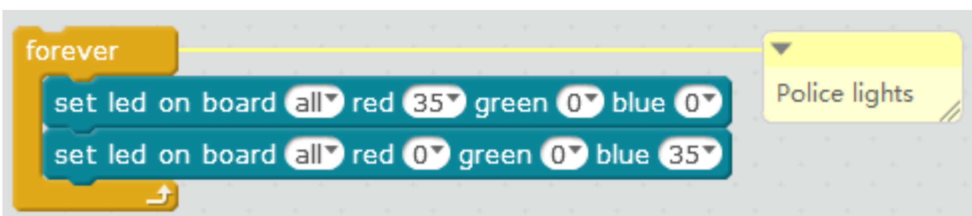
The first drop-down menu gives you a choice of which on-board LED you want to control. On the mBot, you have three choices:

You want to control all the LEDs so you are going to select **all**. The other three drop-downs allow you to control how bright the LED displays the relevant color. It gives you options of 0, 20, 60, 150, and 255, but you can also type in the value you want. Set your LED's red value to 35 and click the block to see the LEDs light up.

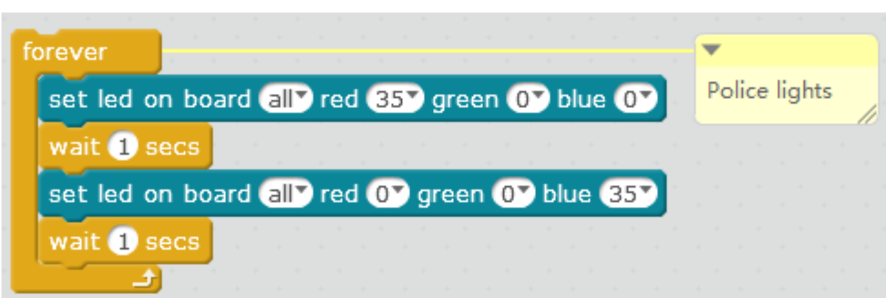
It might be fun to have the robot flashing lights like a police car, going from red to blue and back again. If you would like this to go on forever, there is a forever block in the Control scripts:



In this example, the loop is going to repeat forever, repeating all the blocks that are inside the forever loop:



The LEDs will be set to red, then blue. Then the program will go back to the top of the loop, and the LEDs will be set to red again, then blue, then loop forever. That is good, but the lights change color far too quickly. How would you slow that down? Hint: You used this command in the last meeting.



Adding two wait blocks—one after the red LED is turned on and one after the blue LED is turned on—allows you to see each color clearly before it changes.

Meeting 5: Programming With Music

Challenges:

- Play with different settings of the LEDs. What different colors can you make? What settings of the RGB LED create these colors?
- Can you write a program that sets the left LED and the right LED at different times to your favorite color so that if the right LED is on, the left LED is off, and vice versa?
- Can you write a program that gives a light show accompanied by music? Remember to add comments.

Suggested notes for songs.

Happy Birthday	Mary Had a Little Lamb	Twinkle, Twinkle, Little Star	Little Mermaid
GGAGCB	EDCDEEE	CCGGAAGFFEEDDC	BCDECDEECDEG
GGAGDC	DDDEGG	GGFFEEDGGFFEED	FFEGFFEFECG
GGGECBA	EDCDEEE	CCGGAAGFFEEDDC	FECGGBC
FFECDC	EDDEDC		

Meeting 5: Programming With Music

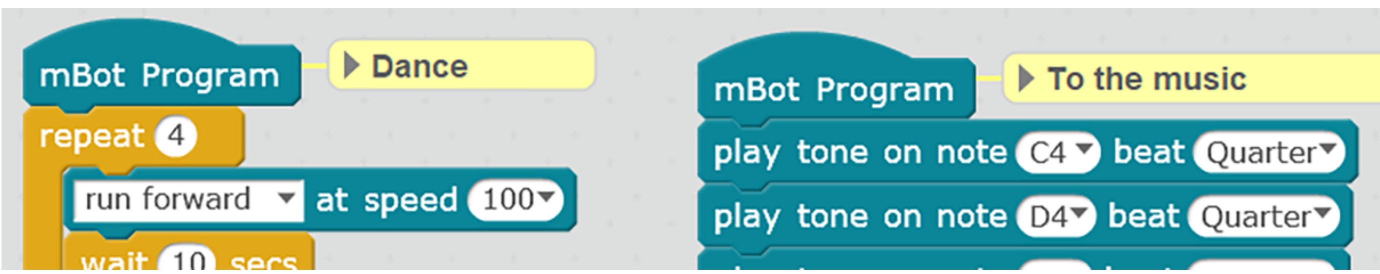
Activity 2: Singing and Dancing mBot (40 minutes)

Remember to switch programmers so everyone has a chance to code.

Materials List

- 1 mBot kit (mBot already assembled from Meeting 1)
- 1 small Phillips head screwdriver
- 4 AA batteries
- CR2025 button cell battery for remote control
- 1 small flashlight
- 1 laptop

The options are endless for this activity. Use any song you like or write your own song, and then get the mBot grooving to the beat. Write some code that plays a song and gets the mBot dancing.



Reusing and **remixing** are the computational practices of making something by building on existing projects or ideas.

In this activity, you and your team will take an existing program and remix it to create your own program.

- Open the mBlock program.
- Click on the **File** menu and choose **Load Project.**
- From the **File Manager,** locate and open the **Desktop.**
- Locate the file named **Singing and Dancing mBot.sb2** and **Open** it.
- This will load a program that will allow your mBot to first move, and then play a song.

Challenge: Use your skills from the LED activity to add some lights to your program.

Discussion

Lead a discussion on what the Scouts have done and their results. Ask open-ended questions to get them thinking about what they have done and learned, and what they might do differently next time.

Discussion Questions

Does anyone want to show off their dancing mBot?

What new ideas did you learn in doing this lab?

Meeting 5: Programming With Music

Cleanup
Turn off your mBot, remove the batteries, and put the mBot in the box with the spare parts, remote control, and USB cord. Be careful not to damage the box, as it will house the mBot and its parts for the rest of this module.

Exit all applications on the laptop, fully power it down, and return it to the Program Manager.

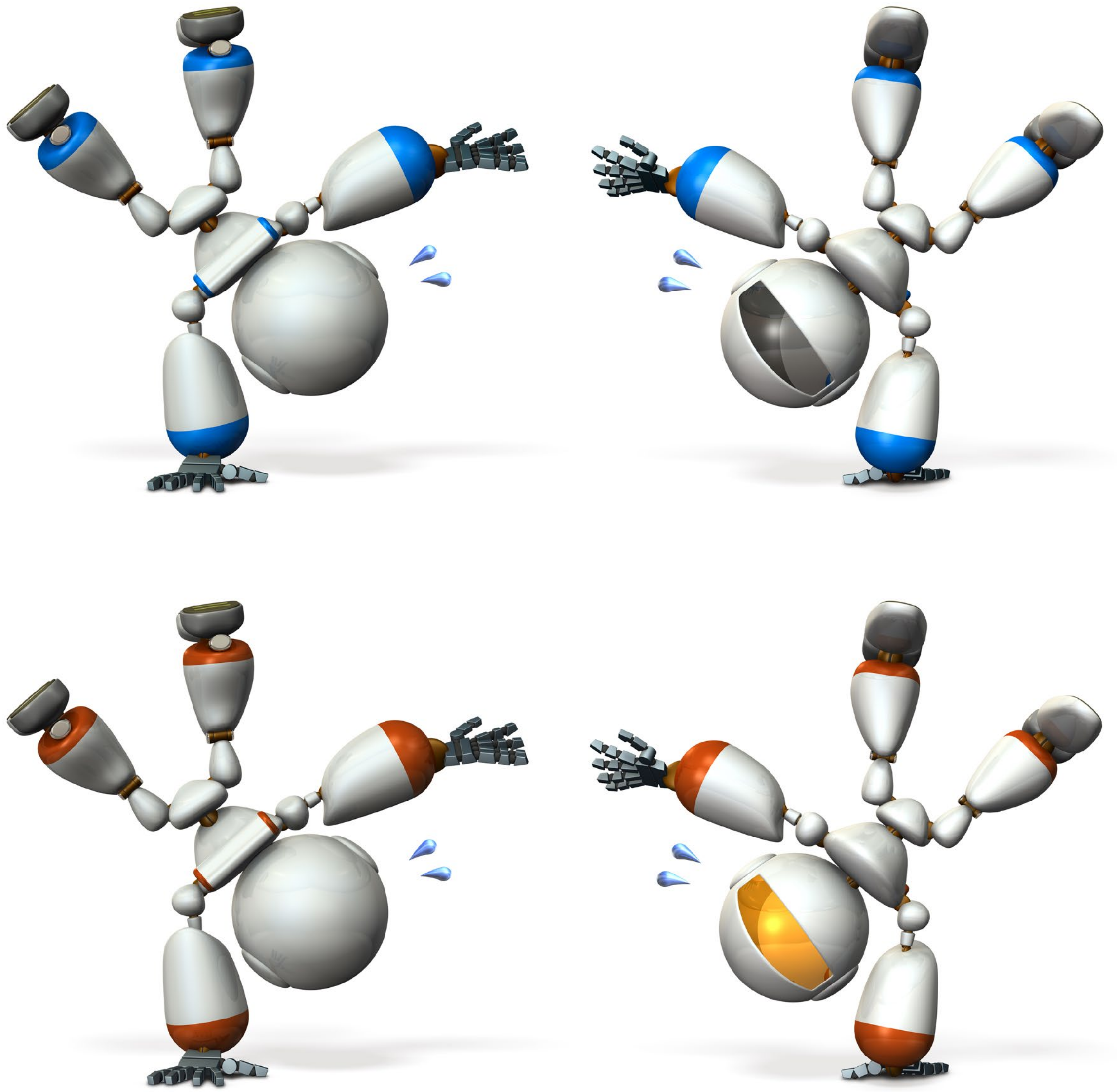
Clean your area, and be sure no trace is left behind.

STEM Innovator Moment
The Scouts should hear from or about a person who has expertise on the topic of the day's activities about how this topic applies to everyday life and careers. The person could visit the Lab or use teleconferencing software such as Skype. You can also use the innovator information below.

> *Brina Lee never thought she'd like computer science, but she went on to become Instagram's first female software engineer.*
>
> Faces of Computer Science: Brina From Instagram (Code.org):
> www.youtube.com/watch?v=elaR3WhguVk&feature=youtu.be (1:35)

Closing
Discuss an ethical situation that occurred during the meeting as well as how the Scout Oath and Scout Law could be applied to deal with the issue. Remember to focus on things that occurred among the Scouts rather than on controversial issues in science today.

Meeting 6: Putting It All Together



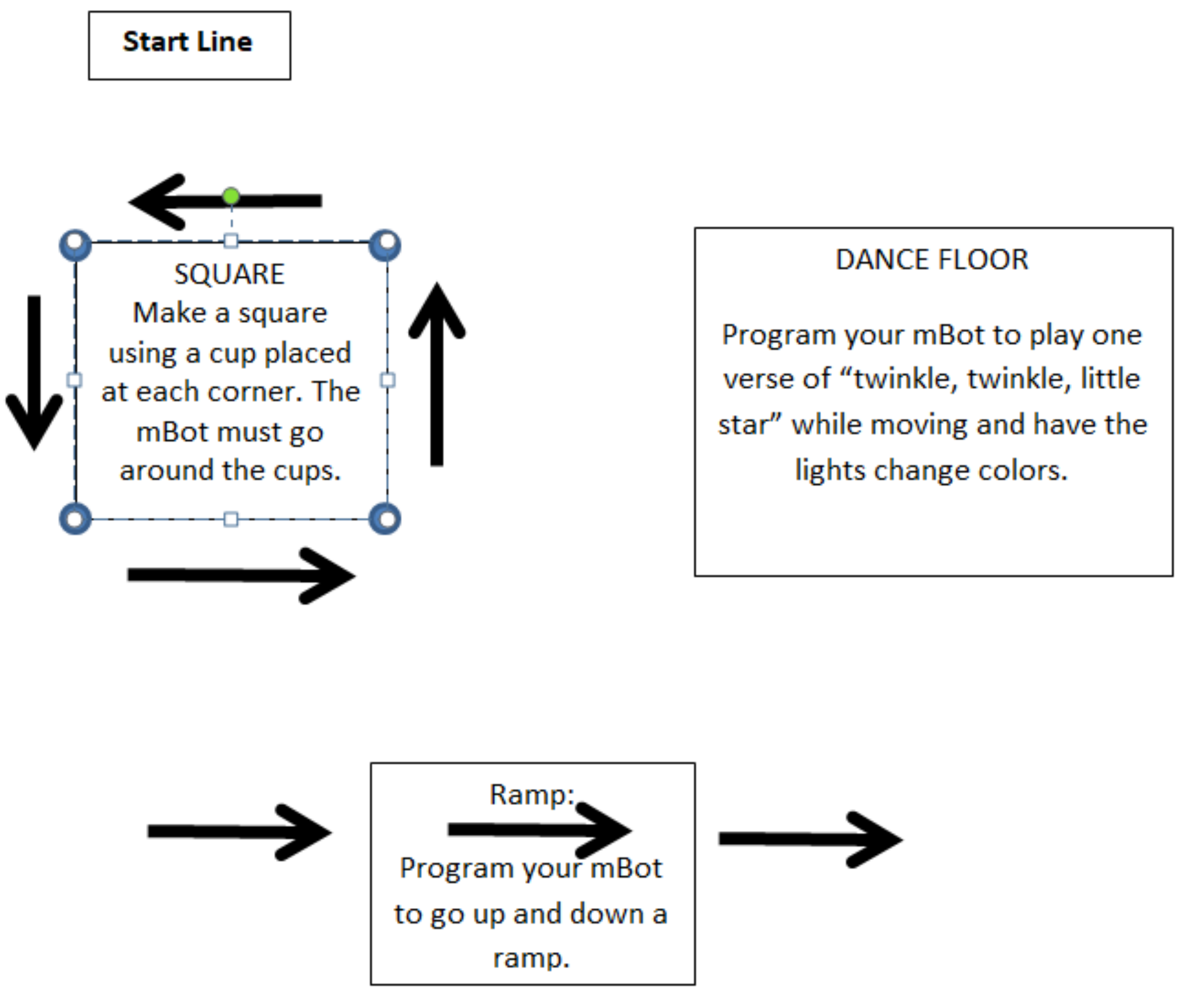
Shutterstock.com, courtesy–©CYCLONEPROJECT

Meeting 6: Putting It All Together

Pre-Meeting Leader Preparation

You will need to set up a challenge course for the robots to run on that includes a ramp like the ones used in Meeting 3. All the teams will compete on the same challenge course. If you have a very large Lab, you may want to set up more than one course and run them in parallel. Have the course set up before the meeting. A rough diagram of the challenge course is shown below. It can be adjusted based on your meeting space.

Meeting 6: Putting It All Together



Meeting 6: Putting It All Together

Materials and Tools Needed

- 1 laptop for Lab Manager (unit- or, optionally, council-supplied)
- 1 projector for Lab Manager (unit- or, optionally, council-supplied)

The following materials are shared between multiple teams.

- 1 pair of tweezers to tighten nuts if they become loose (shared between two teams)
- 4 7x7-inch pieces of cardboard (for ramp)
- 1 roll of electrical tape (for ramp)
- Craft sticks to use as support for the ramp as needed
- 4 plastic cups for challenge course

Scouts will be divided into teams of four for this set of experiments. Material quantities are defined for each team.

- 1 mBot kit (mBot already assembled from Meeting 1)
- 1 small Phillips head screwdriver
- 4 AA batteries
- CR2025 button cell battery for remote control
- 1 laptop
- 1 small flashlight

Opening

Have the Principal Investigator lead the group in reciting the Pledge of Allegiance and the Scout Oath and Scout Law.

Applying the Scout Law

Choose one of the 12 points of the Scout Law to discuss in application to today's lab. A suggested theme for this meeting is *friendly*, as in *I will remain friendly to all Scouts as we compete against each other*.

Ask the Scouts what being *friendly* means to them and how they might apply that to today's lab.

Activity Overview

This is the final activity of this module where the Scouts will combine all of the skills they have learned from each of the past five meetings. The Scouts will use their skills from Meeting 2 to complete the square obstacle course. They will then re-create their ramp program from Meeting 3 to go up and over the ramp. Finally, they will use skills learned in Meetings 4 and 5 to make a music and light show with their mBot.

Background

Building a complete program is like building a house. There are many different skills needed to complete the project. In building a house, you may need experts in foundations, carpentry, electrical, plumbing, painting, etc. In programming and robotics, different programs have to be mastered in order to run the whole project. In today's meeting, you and your team are going to combine all of your programming skills from the past meetings to program a robot to go through different challenges.

Meeting 6: Putting It All Together

Engaging Questions

Use the following question, and any others of your choosing, to get the Scouts engaged. (*Typical answers are in italics.*)

What programming skills did you learn from past meetings? *(Controlling the motors, making lights turn on and off, using the light sensor and ultrasonic sensor, and making sounds.)*

Safety Moment

Tell the Scouts:

Be aware of other mBots when operating your mBot.

Connect and disconnect the cable carefully when testing programs so that you don't damage the cable or connector.

If you have long hair, tie it back following lab safety rules. Also, if you have any long, dangling necklaces or similar items, take them off or put them inside your clothing. You don't want them to get caught in the mBot's wheels.

Experiment

Get the Scouts into the same teams of four that they were in for previous meetings.

Ask them to follow the instructions in their Lab Notebook, reproduced below.

Remember to switch programmers so everyone has a chance to code.

Materials List

- 1 mBot kit (mBot already assembled from Meeting 1)
- 1 small Phillips head screwdriver
- 4 AA batteries
- CR2025 button cell battery for remote control
- 1 small flashlight
- 1 laptop

Lab Manager Note: Assign each team an area on the challenge course to start programming. Do not have them all start at the beginning because it will make the area very crowded when testing. For example, team 1 may be assigned to start with the square, team 2 to start with the ramp, and team 3 to start with the dance floor.

Step 1: You and your team will be assigned an area to start with. Work with your team to develop a program for that area. When complete, check in with a Lab Leader to move on to the next area, moving counterclockwise on the challenge course. Keep track of time, as you have three areas to conquer on the challenge course. You and your team should look over the entire course and decide how to work together as a team to get it all done.

Lab Manager Note: Some teams may want to split up into two teams of two to work on the different areas of the challenge course. That is OK, but one sub-team will have to design their program on paper, since they have only one laptop to share between the four team members. Instruct the teams to make comments in their programs so they can tell which blocks of code go to which part of the challenge course.

Meeting 6: Putting It All Together

Step 2: Program your next area by adding a new section to the program for your first area. Keep the code separate but in the same program. You will connect the code when all three sections are complete. Continue until you have programmed all three areas.

Step 3: Put it all together by combining the three program segments in the right order.

What do you need to add between the obstacles to get from one area to the next?

Step 4: Present your final project.

Discussion

What was most rewarding about your challenges?

How could you continue practicing and expanding your robotics or programming skills in the future?

Discussion Questions

How did your final designs work? What might you do differently next time?

Cleanup

Turn off your mBot, remove the batteries, and put the mBot in the box with the spare parts, remote control, and USB cord.

Exit all applications on the laptop and fully power it down.

Return all materials to the Program Manager.

Clean your area, and be sure no trace is left behind.

STEM Innovator Moment

The Scouts should hear from or about a person who has expertise on the topic of the day's activities about how this topic applies to everyday life and careers. The person could visit the Lab or use teleconferencing software such as Skype. You can also use the innovator information below.

> *Technology allows us to collect vast amounts of data, or information. Computer programs are being used in medicine to help find solutions so people can get better quicker.*
>
> Data in Medicine: http://www.youtube.com/watch?v=bMrDHtGHFR4&feature=youtu.be (6:07)

Closing

Discuss an ethical situation that occurred during the meeting as well as how the Scout Oath and Scout Law could be applied to deal with the issue. Remember to focus on things that occurred among the Scouts rather than on controversial issues in science today.