MICROBOT WARS / AFTERNOON LAB

Lesson Description: This lesson aims at teaching students different types of attacks like Denial of Service and replay attacks. These concepts are demonstrated through micro:bots (robots built combining micro:bits with moto:bits).

Prerequisite Knowledge: Students should have knowledge of radio communications over micro:bits and be able to write programs that send and receive messages over radio channels.

Length of Completion: 1 to 1:30 hours.

Level of Instruction: This lesson is intended for middle school and high school students. This lesson is intended for all three levels as it has activities that start off with novice level and difficulty builds off of that level.

Applicable First Principles &/or Concepts: Please select the Principles or Concepts covered in this lesson.

GenCyber First Principles

Domain Separation	Abstraction
Process Isolation	Data Hiding
Resource Encapsulation	Layering
Modularity	Simplicity
Least Privilege	Minimization

GenCyber Cybersecurity Concepts

CYBERSECURITY EDUCATION AND RESEARCH CENTER
@ The University of Wyoming
www.uwyo.edu/CEDAR
www.cowpokes.camp
www.uwcedar.io/community/cowpokes/wikis

Defense in Depth	Availability
Confidentiality	Think Like an Adversary
Integrity	Keep it Simple

Resources that are Needed: The resources needed to complete the lesson are two micro:bits, a moto:bit, a robot, computer and a USB cable. The software needed to start this lesson are JavaScript files robot-ctrl.js and robot-johno.js. A white board might also be needed for explanation.

Accommodations Needed: Written directions (hardcopy or electronic) are provided to all students.

LEARNING OUTCOMES

LESSON LEARNING OUTCOMES

- Students will be able to control robots built with moto:bits by sending out commands for navigation on radio channels of the micro:bits.
- Students will be able to identify different attacks such as a Denial of Service attack and a replay attack.
- Students will be able to program their robots to be more secure against hacking.

LESSON DETAILS

Interconnection: Beacons, Cyberhand Unplugged, Micro:bits 101, Ethics, CIA Triad.

Assessment: Formative assessment - Students are asked questions on how their micro:bits can be made more secure starting with the base code provided. They are asked to discuss at each activity level, the cons of having the existing code and are asked to modify the code to prevent their robots from getting hacked.

Extension Activities: "Garage Door" Lab, "Micro-Pay" Lab.

Differentiated Learning Opportunities: The baseline code targets novice learners where they use the code to establish a communication between controlling micro:bit and the robot and navigate the robot. As difficulty increases, they will be asked to change the channel of communication and change the message that is sent to the robot micro:bit.

LESSON

Warm Up: To begin the lesson, students are shown a micro:bot and all its parts (moto:bit, micro:bit, battery pack).
To link this lesson with previous lessons they are shown how the controlling micro:bit communicates with the micro:bit on top of the robot and are shown a demo of the robot following the directions of the controller micro:bit.

Lesson: The lesson is a guided inquiry lab.

Students are provided with micro:bots and the code to flash onto their robot micro:bit and the controller micro:bit. They begin by controlling their robots and navigating them. They start by realizing that one controller micro:bit is capable of controlling all the robots. The instructors ask the students to find a solution to this problem.

The second step is to choose a secret channel of communication between each controller micro:bit and robot micro:bit pair. In this situation, if two robots are programmed to the same channel, one controller is still be able to control them both.

Then, in the third step, the instructors ask the students to change the messages sent by the controller micro:bit to the robot micro:bit. This way, their robots get more secure against hacking.

In the fourth step, the instructors demonstrate a replay attack by loading a 'Deathinator' code on a micro:bit which listens to messages on each channel and just sends them back. The instructors introduce them to replay attacks and discuss advanced topics to prevent replay attacks or Denial of Service attacks.

There can be an extension to this lesson where students are asked to program a rolling key message transfer between the two micro:bits based on the skills of the students.

Lab Procedure

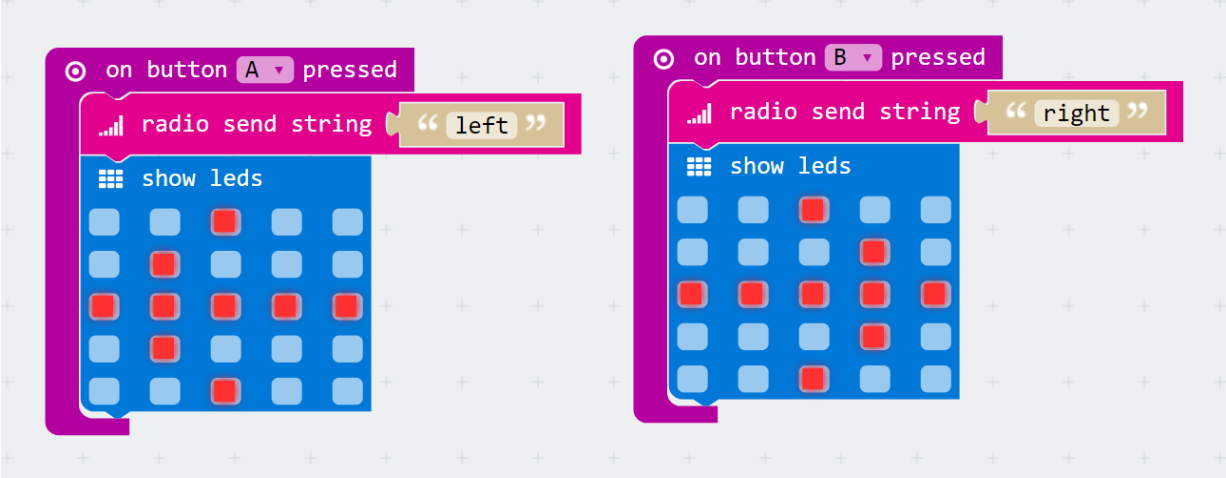
Step 1: Gather Materials

- ❑ Micro::Bit (1 per person / 2 per team)
- ❑ Micro-USB Cable
- ❑ Battery Pack (Equivalent to two AA batteries)
- ❑ Microbot

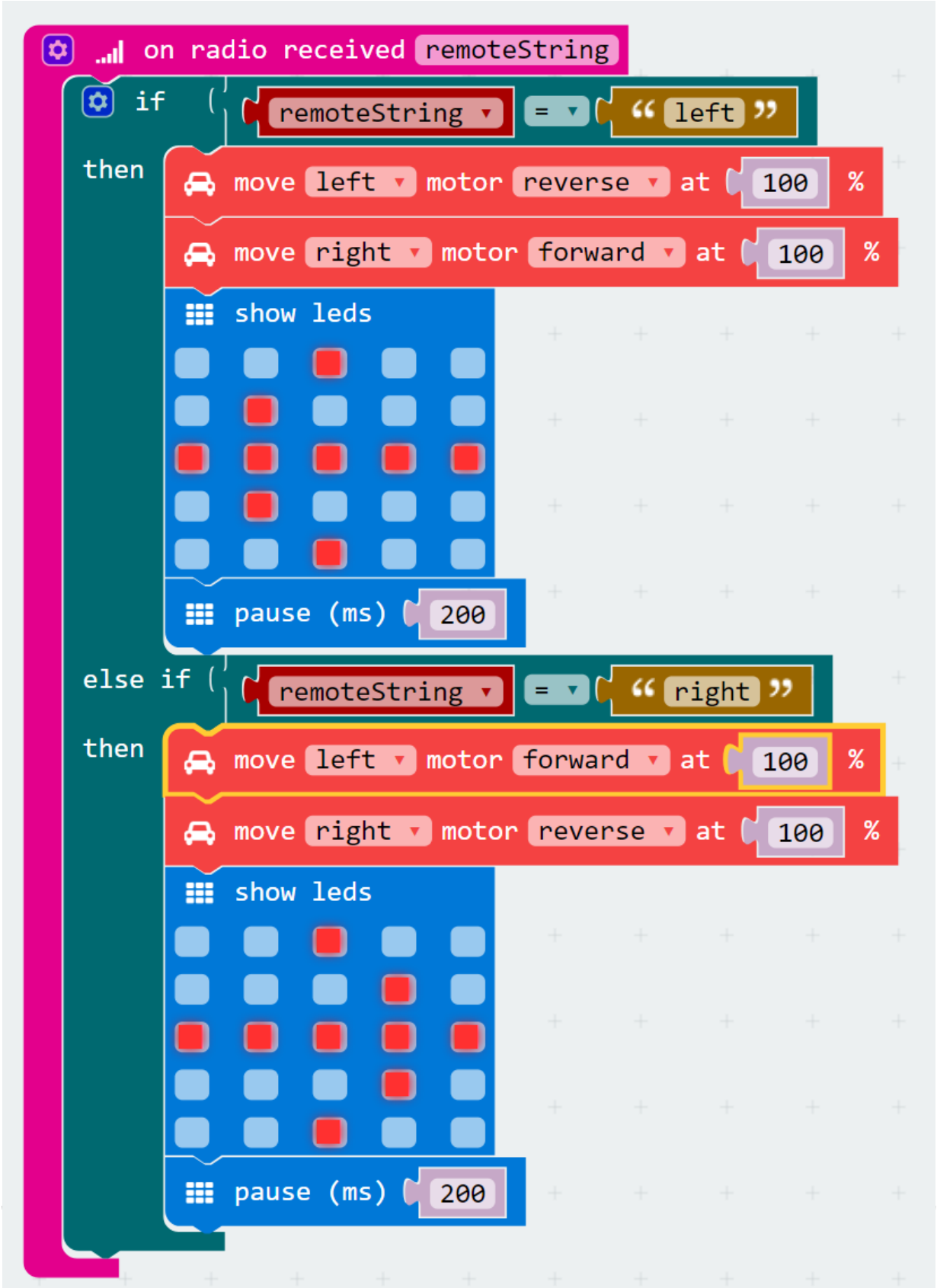
Step 2: Get Familiar With Microbots

Radios work by sending and receiving messages over different channels. This allows you to use on Micro:Bit to send messages and another to receive them. In this lab, you will use a Micro:bit as a controller to get user input and then send messages based on those inputs. Then, you will use a second Micro:bit to receive messages from the controller and then control the robot based on those received messages.

Start this lab by going over this basic programming example.



CYBERSECURITY EDUCATION AND RESEARCH CENTER
@ The University of Wyoming
www.uwyo.edu/CEDAR
www.cowpokes.camp
www.uwcedar.io/community/cowpokes/wikis



The first set of code is for the Micro:Bit controller. This controller waits for the user to press the A button or the B button. When the user presses the A button, the controller sends the message "left" to the robot. When the user presses the B button, the controller sends the message "right" to the robot.

The second set of code is for the Micro:Bot robot. This Micro:Bit waits to receive a message. If they receive a "left" message, it tells the robot to go left. If they receive a "right" message, it tells the robot to go right.

Step 3: Download The Files

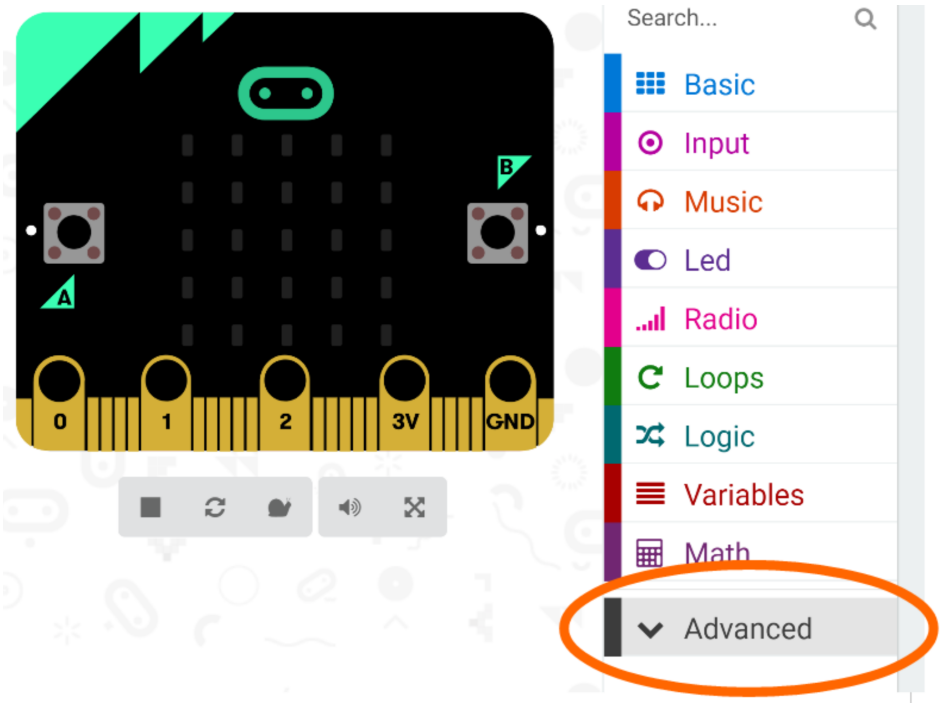
- Download (click on) the following file to get started: MicroBotLab.zip
- Unzip MicroBotLab.zip to your desired folder. For instructions on how to unzip click here
- NOTE: There is an extra folder for this lab, Attackers
- Attackers: This directory holds code to simulate an attack that jams the radio channel and makes the robots behave chaotically.

Step 4: Open your editor

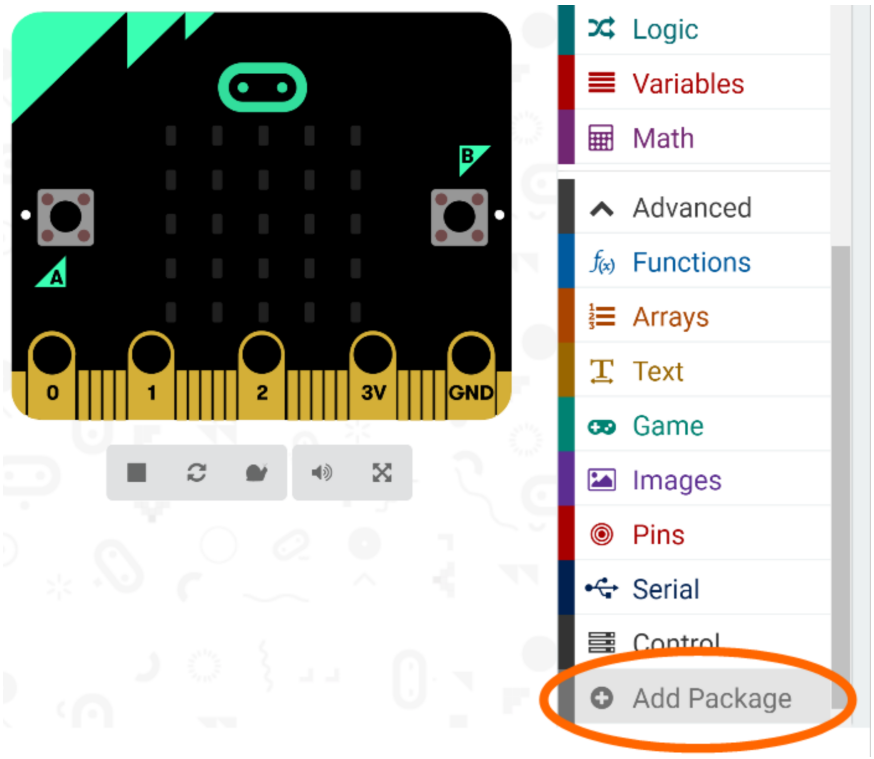
This base code is also available in the directory that you downloaded as well as this repository. If you don't remember how to open code in your editor, see the instructions here.

Step 5: Get Moto-Bit library

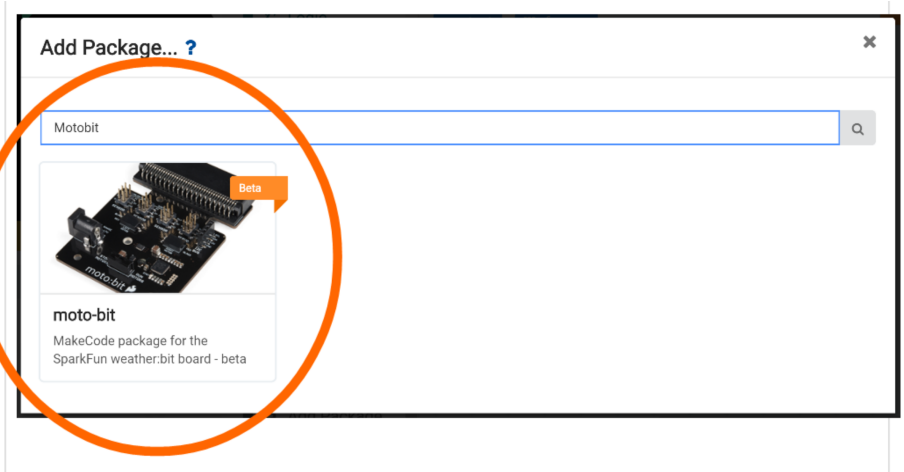
For JavaScript/Blocks, start by selecting the "Advanced" drop down box in the libraries toolbar.



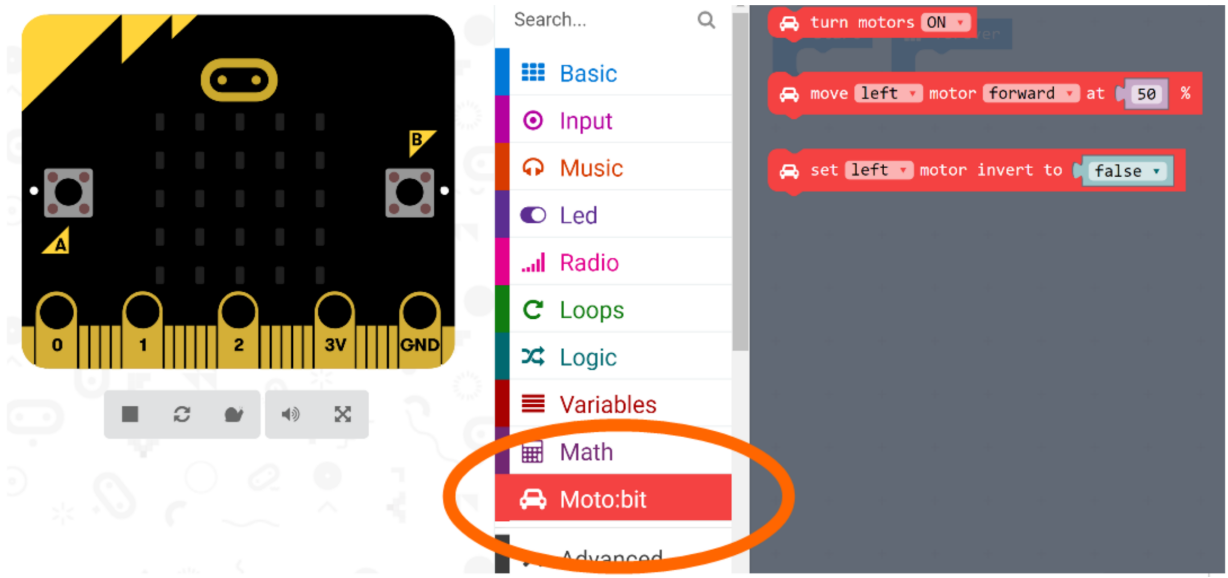
Then, scroll down to the bottom of the advanced packages. Select the "Add Package" button.



In the packages pop-up, search for "Motobit". Select the moto-bit library in the results section.



You can check to make sure that you have the library by searching for the library in your libraries toolbar.



Step 6: Programming!

6.A Complete the Robot Code

Complete the robot code. You will add channel choosing functionality and give the motobit commands after receiving a radio transmission from the controller:

- ❑ When button a and button b are pressed, set the radio group
- ❑ When button a is pressed, if the radio group has not been set, decrement the radio group by 1
- ❑ When button b is pressed, if the radio group has not been set, increment the radio group by 1

- ❏ When the command forward is received over the radio, drive the motobit Forward
- ❏ When the command stop is received over the radio, stop the motobit
- ❏ When the command left is received over the radio, drive the motobit left
- ❏ When the command right is received over the radio, drive the motobit right
- ❏ Generate your robot .hex file and flash your robot Micro::Bit.

6.B Complete the Controller code

Complete the controller base code. Students will add channel choosing functionality and send radio commands depending on the buttons pressed:

- ❏ When button a and button b are pressed, if the radio channel has not already been set, set it.
- ❏ If the radio channel has already been set, send a forward command over the radio
- ❏ When button a is pressed, if the radio group has not been set, decrement the radio group by 1.
- ❏ If the radio group has been set, send a left command over the radio
- ❏ When button b is pressed, if the radio group has not been set, increment the radio group by 1.
- ❏ If the radio group has been set, send a right command over the radio