Driving mBot

Overview

In this initial activity, students recreate the functionality of Mode A–Manual Drive of mBot's Default Program. Students receive a quick introduction to mBlock programing as they create programs and upload them to mBot. The programs written by students in this activity are reused and expanded upon in following activities, so we highly recommend beginning with this activity.

Objectives

- Learn about the three pre-programmed modes of operation in mBot's Default Program.
- Practice writing code, uploading and running a program on mBot, and troubleshooting code when problems arise.
- Learn about using the IR remote with mBot.
- Practice using variables, IF-THEN statements, and FOREVER loops in the mBlock software.
- Learn about using the LEDs on mBot.
- Write code to develop a basic manual-drive program for mBot.
- Learn about the importance of adding comments to your programs.

Computer Science Teachers Association (CSTA) Standards

2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.

2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.

2-AP-13 Decompose problems and subproblems into parts to facilitate design, implementation, and review of programs.

2-AP-19 Document programs in order to make them easier to follow, test, and debug.

Materials

- computer (PC/Mac) **or** Chromebook with mBlock software installed
- mBot
- mBot battery (battery pack with 4 AA batteries **or** Makeblock's 3.7 V rechargeable battery)
- mBot IR remote control (required CR2025 coin-cell battery not included)
- USB cable (included with mBot)

Tips

1. In the Electronic Resources you will find a PDF of the student pages as well as sample programs for this activity. The PDF allows you to print the activity for your students or distribute the activity electronically. You can use the sample programs as you prepare for teaching; we also include sample solutions to the Challenge Extensions. Sign in to your account at **vernier.com/account** to access the Electronic Resources.

2. Students will need room to test their mBot programs. The iterative process of testing mBot programs is most efficient if the robot-testing area is close to the students' computer or Chromebook.

3. Before beginning this activity, give your students time to play with mBot's Default Program. Once they have a clear understanding of mBot's default actions, they will have a better idea of what they will need to implement in their code.

4. If you have extra mBots, consider keeping one with the Default Program. In the first three activities, students write code to mirror how mBot behaves in the Default Program. Having a single mBot with the Default Program loaded saves students from switching back and forth between programs.

5. If you do not have an extra mBot, or you want to teach your students how to reload the Default Program to mBot, the procedure is fairly simple:
   a. Connect mBot to a computer or Chromebook with a USB cable (included with mBot).
   b. Open the mBlock software. To connect to mBot, choose Serial from the Connect menu and select the COM (USB) port to which mBot is connected.
   c. Once connected, choose Reset Default Program > mBot from the Connect menu.
   d. mBlock will reload the Default Program to mBot.

6. mBot's driving speed varies with the state of mBot's battery. When mBot is fully-charged, it will drive faster for a given, specified drive speed, than when the battery is low.

7. We recommend using rechargeable AA batteries in mBot or the rechargeable mBot battery.

Solutions to Challenge Extensions

Extension 1 Change Speed with Numeric Buttons

In this Challenge Extension solution, we use the 1, 2, and 3 buttons to set the speed variable.

Figure 1 Manual drive program with speed control (IR remote buttons 1, 2, and 3)

Of course, there are lots of other ways to control the speed. For example, you could have the 0 button decrease the value of the "speed" variable and the 1 button increase it.

Extension 2 Backup Warning

In this example solution, we use a high-pitched sound for a half beat (0.5 second) and then silence for 0.3 seconds. This will repeat as long as the backward button is pressed (and the car will also move backward).

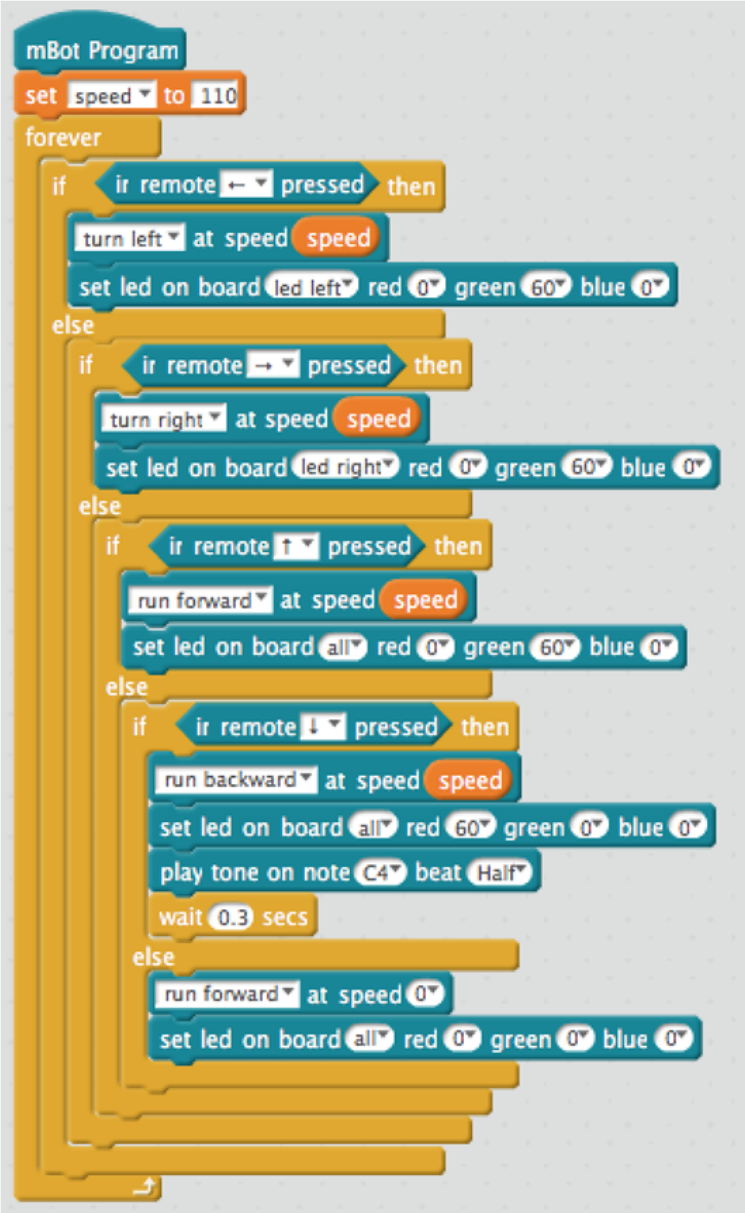

Figure 2 Manual drive program with “back up warning” using “play tone on note” block

Note: In the "Security and Emergency" activity, students learn more about incorporating sound into their programs.