

Personal Weather Assistant

Introduction

Electronic personal assistants are everywhere. From smartphones to Alexa, people are able to interact with electronic devices in a more "human" way to access information and accomplish tasks. In this activity, we will turn mBot into a personal assistant that reports something that is of interest to us all—the weather.

Understanding Inputs and Outputs

Inputs and outputs are key components when developing code. Consider the school bell that rings when class is over. Chances are there is a computer running a simple program that continually checks the time (input). When the time is equal to a programmed value (the time for class to end), the program causes the bell to ring (output).

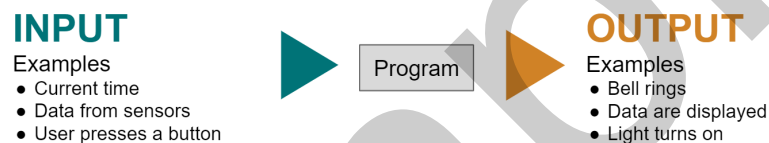


FIGURE 1

Of course you could monitor the time without the help of a computer. But, if you were to get distracted, you might ring the bell a few minutes late or forget to ring it altogether! Computers never get bored or distracted, making them ideal for these types of repetitive activities.

The STEM Classroom Kit includes the Me Temperature and Humidity Sensor, which combines a temperature sensor with a relative humidity sensor. The data from these two sensors will be the two inputs for your weather program. Your output will be a display of the weather data.

Equipment

For this project, you need the following equipment from the STEM Classroom Kit:

- mBot/mCore
- Me LED Matrix
- Me 7-Segment Display
- Me Temperature and Humidity Sensor

Make the following connections on mCore:

- Port 1: LED Matrix
- Port 2: 7-Segment Display
- Port 3: Temperature and Humidity Sensor

CONSTRUCTION TIP: Notice that the sensors and displays are color coded, as are the connection ports on mBot. Always connect a sensor or display to a port with the matching color.

Weather Station Program

PART I Launch mBlock and activate Upload mode

Start by writing a simple program that will display the temperature and humidity values on the 7-Segment Display.

1. Launch mBlock 5 on your computer or Chromebook.
2. By default, mBlock assumes that you are programming a Codey Rocky robot. To add mBot to the list of Devices, click the + button in the Devices pane. Select mBot, and then click OK.
3. To be able to upload code to mBot, activate Upload mode by clicking the Upload mode toggle switch (see Figure 2).

With mBot selected and Upload mode activated, mBlock 5 is set up for coding (see Figure 3).

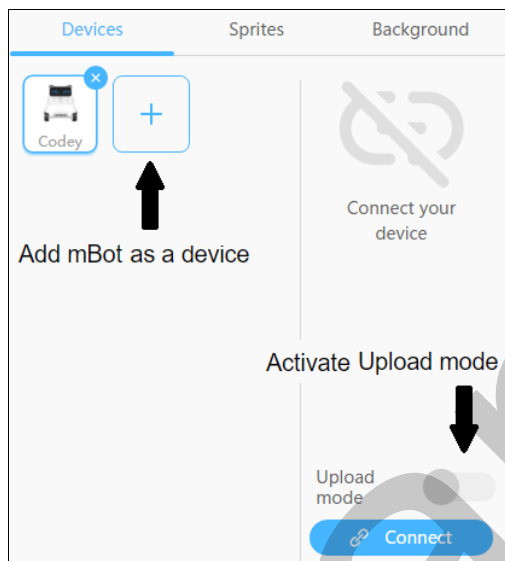


FIGURE 2 Add mBot and activate Upload mode

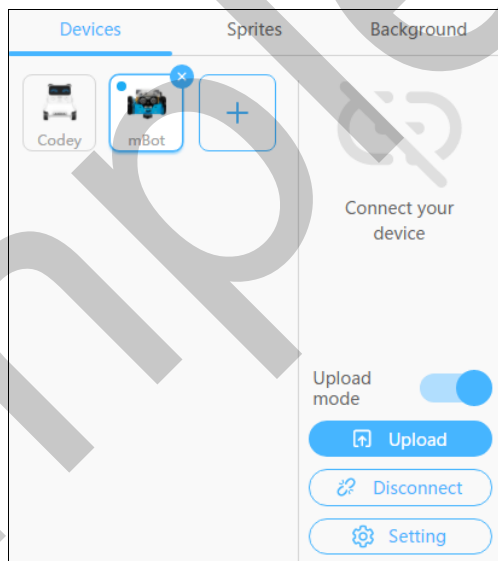











FIGURE 3 Connected to mBot, Upload mode active

- Take a minute and look through the palettes, which are located in the middle of the mBlock screen. Palettes contain blocks—the raw material from which you will build your programs. Blocks can be dragged to the *programming area*, the right-side pane, which is where you build your program.

Table 1	
Palette	Description and use
 Looks	Use the Looks palette to configure the display on the LED Matrix.
 Show	Program the onboard sound generator and LEDs with the Show palette.
 Action	The Action palette controls the motion of the mBot. The blocks assume you have the motors connected according to the mBot construction instructions.
 Sensing	Blocks from the Sensing palette read onboard sensors (e.g., Ultrasonic Sensor). This palette also has blocks for the IR receiver, the IR transmitter, and a timer.
 Events	The Events palette contains a block that is required to start programs. The "when mBot(mcore) starts up" hat block is the only block that we will use from this palette.
 Control	Create loops and other logic statements with Control palette blocks.
 Operators	Operators allow for mathematical and comparison functions (e.g., round to an integer, "greater than" or "less than"). The blocks can also be used to evaluate and manipulate strings (computer-speak for words), as opposed to just numbers.
 Variables	Use this palette to create variables. Once created, additional blocks that use the variable are automatically generated by mBlock.
 My Blocks	Use the My Blocks palette to create your own blocks.

PART II Display temperature and humidity

- To start coding, select the Events palette and drag the "when mBot(mcore) starts up" block onto the programming area. This type of block is called a hat block. The "when mBot(mcore) starts up" hat block is required at the start of every program you want to run on mBot.



FIGURE 4 "when mBot(mcore) starts up" hat block

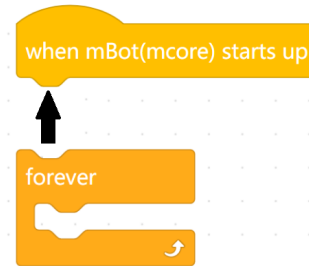


FIGURE 5 Blocks snap together when moved close together

- Select the Control palette and drag the "forever" block to the programming area. Drag the "forever" block close to the bottom side of the "when mBot(mcore) starts up" block (see Figure 4). Notice that blocks snap together when they get close to each other.

- mBlock 5 refers to the Temperature and Humidity Sensor as the "humiture sensor." You might expect to find this sensor block in the Sensing palette. However, the Sensing palette only contains blocks for sensors that are built into mBot (e.g., the Light Sensor).

To program the Temperature and Humidity Sensor, you first need to add a palette to mBlock 5. To do this, click the "+" button under the list of palettes and add the Maker's Platform palette. This palette contains blocks for all of the input and output devices that are included in the STEM Classroom Kit.

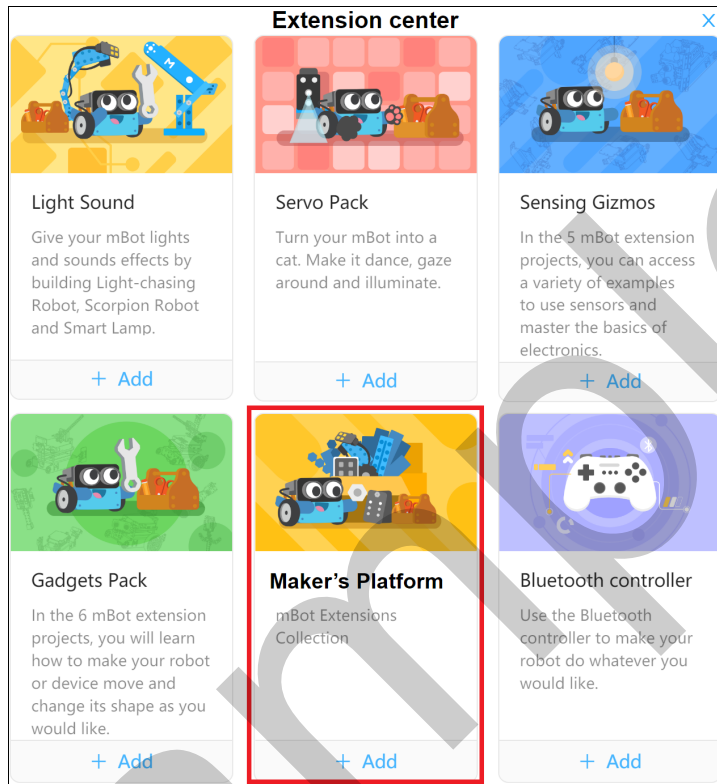


FIGURE 6

- Once you have added the Maker's Platform palette, drag the "seven-segments" block and the "humiture sensor" block onto the programming area.
- Repeat Step 8 so you have a total of four new blocks in the programming area (two "seven-segments" blocks and two "humiture sensor" blocks).
- It takes a little practice to get the hang of how to move the blocks around and connect them. Combine the "seven-segments" block with the "humiture sensor" block as shown in Figure 7.

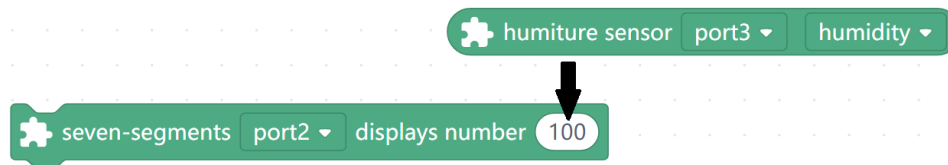


FIGURE 7 To display the sensor reading on the 7-Segment Display, drag the "humiture sensor" block into the 100 field of the "seven-segments" block.

- Drag two "wait" blocks from the Control palette onto the programming area. Connect the "seven-segments" blocks with the "wait" blocks (see Figure 8).

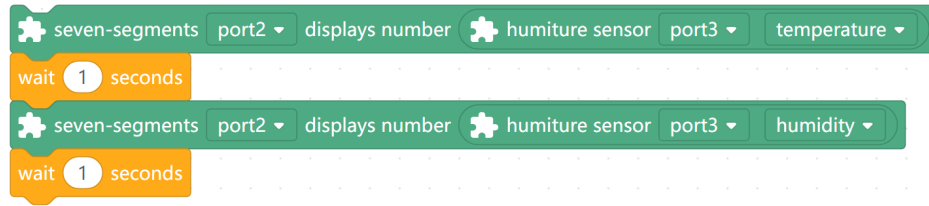


FIGURE 8 Combine "seven-segments" and "wait" blocks. Update the ports to match your robot.

- Update the settings for the "seven-segments" blocks to match the connection in your weather station:
 - The first "seven-segments" block should display temperature and the second "seven-segments" block should display humidity. Both blocks are sent to display humidity by default, so all you need to do is change the first block. To do this, click the humidity drop-down menu in the first "seven-segments" block, and select temperature (see Figure 8).
 - For both "seven-segments" blocks, update the port selections to match the ports to which the 7-Segment Display and the Temperature and Humidity Sensor are connected. The 7-Segment Display is connected to port 2 and the Temperature and Humidity Sensor is connected to port 3 (see Figure 8).
- Drag the set of "seven segments" and "wait" blocks into the "forever" block (see Figure 9).

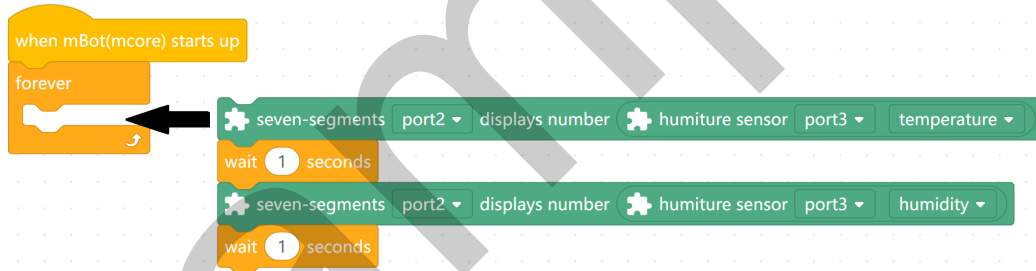


FIGURE 9 Blocks can be inserted into slots in the "forever" block

- Your program should look like the program in Figure 10. Save your program with a unique name. Saving your work often avoids losing your work, and later on, you may be able to use your code in future programs.

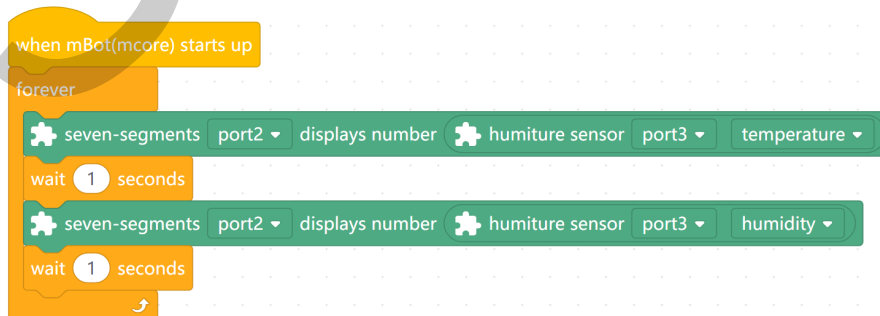




FIGURE 10 mBlock program configured to display temperature and humidity on the 7-Segment Display

TRY IT OUT

Once you have built the program and set the ports properly, try out the program with mBot:

1. Turn on mBot and connect mBot to a computer or Chromebook with a USB cable.
2. Click Connect, , in the Devices pane. mBlock 5 suggests the COM port to which mBot is connected. Click Connect again.
3. To upload the program, click Upload, , in mBlock 5.
4. Wait for the program to upload. If the program uploads successfully, a message indicates that the upload is complete. The program starts running as soon as it has uploaded. Do you see the temperature and humidity values on the display?

PART III Simple improvements

Notice that when the program is running on mBot, alternating numbers can be seen on the display, but the meanings of the numbers may not be obvious. Let's improve the *user interface*, that is the interaction you have with the robot platform, to make the robot more helpful.

The 7-Segment Display only displays numbers, but we can use the LED Matrix to give the user more information about the numbers that are being displayed. To do this we will need to insert a block that controls the LED Matrix.

15. Select the Looks palette and drag two of the "LED panel shows image" blocks into the programming area. Many blocks in the Looks palette are similar; use the block in Figure 11.



FIGURE 11

16. The Temperature and Humidity Sensor measures temperature in Celsius degrees (°C) and humidity in percent (%). Modify the "LED panel shows image" block to display a sensor abbreviation and the units. In our program, we modified the display to show **T•C** for temperature and **H%** for humidity, but you can use whatever will make sense to you and the users of your weather station.

To modify what is displayed on the LED Matrix, click the image of the two blue "eyes" in the "LED panel shows image" block. In the LED editor, use the pencil and the eraser tools to modify which LEDs light up (see Figure 12). When you are done, click the green check mark to exit the LED editor; your design is automatically saved.

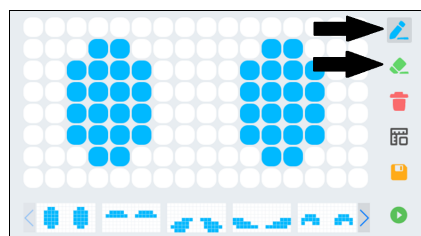


FIGURE 12 In the LED editor, use the pencil and eraser tools to change the display.

17. Once you have the display finished, drag the blocks under the "seven segments" blocks so your program is similar to the program in Figure 13.

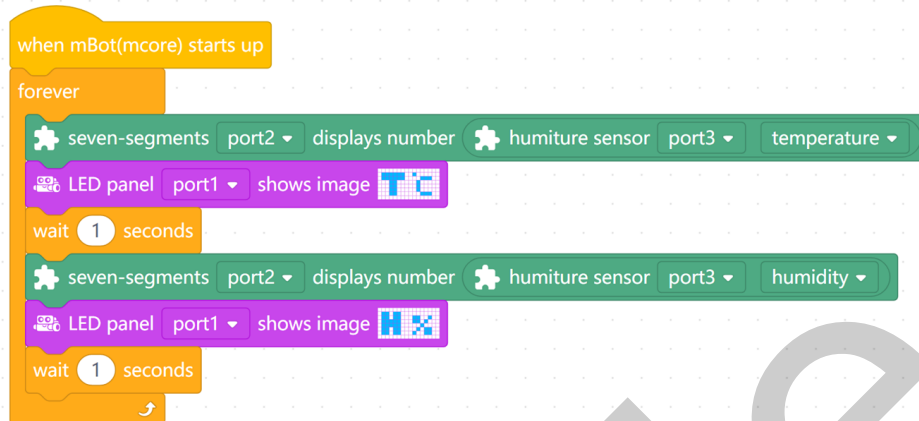


FIGURE 13 Adding two LED panel blocks to our program with labels and units

18. If you were to run the Figure 13 program on mBot, the program would run "forever" (or until you turn off mBot). The LED Matrix would also stay on "forever," which could quickly run down the mBot battery. So, let's incorporate an energy saving feature by modifying this program to display the weather information for a few cycles after a user presses the Onboard Button.

See if you can find the additional blocks to modify your program so that the program continues to run "forever", but only displays information for a few cycles after the Onboard Button has been pressed. You can see how we modified our program in Figure 14; the LED Matrix displays the values 10 times and then waits for the Onboard Button to be pressed again.

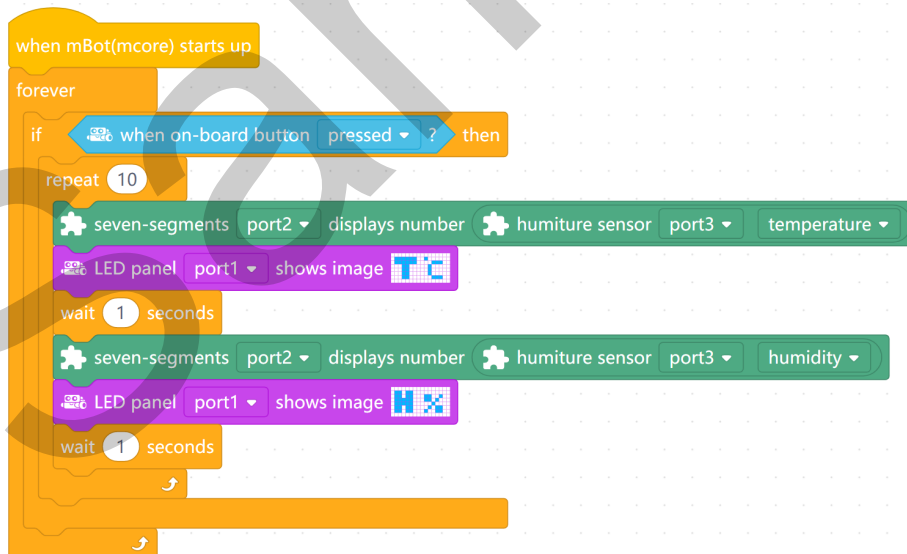


FIGURE 14 Include an "if-then" block into the program so that the weather displays 10 times if the Onboard Button has been pressed.

19. As a final improvement, modify the code so the program clears the screen of both the 7-Segment Display and the LED Matrix after the program runs through the 10 cycles. Ask your teacher if you need a hint.

TRY IT OUT

Upload the program to mBot. Are you able to use the Onboard Button to start the program and display the results? Do the 7-Segment Display and the LED Matrix screens clear after 10 cycles?

PART IV Add comments

As a programmer, it is important to frequently add comments, or short explanations, to your code. When you are writing a program, you think you know exactly what is going on, and it may feel as though comments are unnecessary. However, if you close the program and do not think about it for a few weeks, you may not remember the logic you used to create the program. Also, you may want to share your program with other people so they can use it.

mBlock 5 makes it easy to add comments to your program. To create a comment, right-click (Windows or Chrome OS) or CTRL-click (macOS) in the programming area or on a block and choose "add comment".

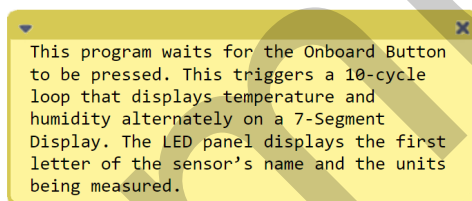


FIGURE 15 Comments help you remember your logic and allow others to understand the program

Comments become increasingly helpful as your programs become more complex. It takes time to learn how to write comments that will be useful to you and others. Practice adding comments to your programs for these activities so you can do it more easily and effectively in the future.

When you upload a program to mBot, comments do not impact how the program runs. However, comments are saved within the program itself, so they will always be available when you open the program in mBlock 5.

Construction

Now that you have brought all of the components for your weather station together, you should create a physical structure that makes it easy to use. Consider the following design criteria for your weather station:

- It should be easy to see and clear what is being displayed.
- It should be obvious how to start the program.

You may wish to include items that are not part of the STEM Classroom Kit to make it clear it is a weather station.

Challenge Extensions

EXTENSION 1 Convert temperature to Fahrenheit

You can modify the number that is displayed on the 7-Segment Display by applying a formula to convert °C to °F. You will find math blocks in the Operators palette. Your teacher may have some tips for you if you are not sure how to proceed.

EXTENSION 2 Initiate the program based on a voice command

Although mBot cannot interpret language, you can modify your program to start displaying the weather when the noise reaches a certain level.

EXTENSION 3 Investigate heat index and wind chill

The calculations of heat index and wind chill are based on complex models, and it is not realistic to recreate the models in mBlock 5. However, the heat index and wind chill are often included in a weather report, so it is useful to learn about them. Conduct research to understand what goes into the calculations for heat index and wind chill and why they are good indicators that deserve our attention. Design a process to capture your weather station data so the data could be used to calculate heat index and wind chill.

Sample

Personal Weather Assistant

Overview

In this initial activity, students create a simple program to collect and display data from a sensor. Students are introduced to mBlock programming as they create their first program. Students learn how to connect sensors and displays from the mBot STEM Classroom Kit to the mBot, and interact with these devices through mBlock programming. These skills build an important foundation for the rest of the *Life Hacks* activities.

Objectives

- Practice writing code, uploading and running a program on mBot, and troubleshooting code when problems arise.
- Practice using "if-then" blocks, and loops in mBlock.
- Understand the importance of adding comments to programs.
- Engineer the physical design of a robot, including consideration for the user interface.
- Use a variable and customize the user experience (extensions only).

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

mBot STEM Classroom Kit
 mBot battery (Makeblock's 3.7 V rechargeable battery or battery pack with 4 AA batteries)
 USB cable (included with the STEM Classroom Kit)
 computer (Windows or macOS) or Chromebook running mBlock 5 software

Key Concepts

Consider covering the following topics as you introduce the activity to students:

- Inputs and outputs, specifically sensor data as the input and display of data at the output
- Starting programs with a block from the Events palette
- Using loops and conditional statements
- How to set a variable and when you should use one (Extension 1 only)

Tips

The tips for this activity also apply to the subsequent activities.

1. The Electronic Resources include a PDF of the student activity as well as sample mBlock programs. The PDF can be printed or distributed to students electronically. The sample programs are fully functioning and can be found in the Supplemental Materials folder. Open a program in mBlock 5 and then upload the program to mBot. Sign in to your account at www.vernier.com/account to access the Electronic Resources.

In the Electronic Resources, you can also find the student edition of *Coding with mBot: Life Hacks*. This PDF includes an Introduction written specifically for students as well as all the student activities; the instructor information pages are not included. The student edition works well as a standalone text and gives students an easy way to search through previous activities for specific programs or information.

2. This set of activities is intended to appeal to the widest range of students. A certain emphasis is also placed on taking advantage of the add-on packs included in the STEM Classroom Kit. Some students, however, may wish to explore the mBot as a mobile robot, and you may consider allowing your students time to play with mBot in Default Program mode. The "Getting Started With the mBot and mBlock 5" guide, which can be found at www.vernier.com/mbot-kit or in the Electronic Resources, includes additional information about the Default Program mode.
3. The iterative process of testing mBot programs is most efficient if the robot-testing area is close to the students' computer or Chromebook. Ideally student groups have space enough for their computer/Chromebook, mBot STEM Classroom Kit, and extra table space for building/prototyping their creations.
4. The mBots will run for a surprisingly long time on one battery charge. You can typically run mBot for many hours on a single charge.
5. We recommend using rechargeable AA batteries in mBot or the rechargeable battery made specifically for mBot (www.vernier.com/mbot-bat, order code: MBOT-BAT).
6. Your students may discover that the mBlock software translates the block code into Arduino C code. The Arduino C code is text-based code (opposed to blocked-based code) and is the code that is loaded onto mBot. Students can view the Arduino C code by clicking the `</>` icon on the right side of the programming area. As students become more skilled at programming in mBlock, they may wish to learn another programming language such as Arduino C.

Example Construction Solutions

See Figures 1 and 2 for examples of the mBot weather stations that we designed.

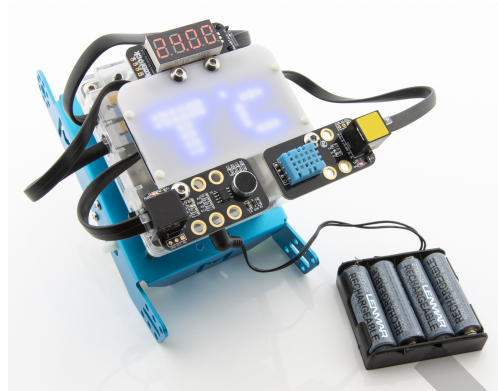


FIGURE 1 Using the mBot vehicle as a base for the weather station

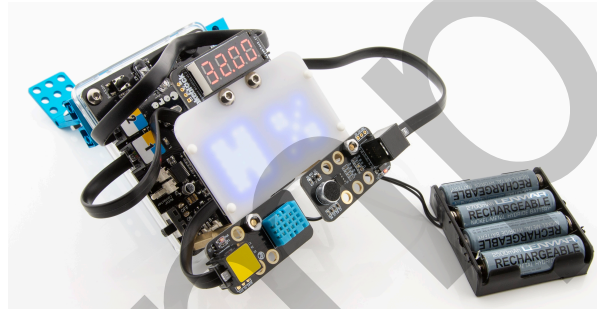


FIGURE 2 Constructing a weather station with mCore as the base

Solutions to Challenge Extensions

EXTENSION 1 Convert temperature to Fahrenheit

Use the multiply and add math functions from the Operators palette to convert the temperature reading from Celsius to Fahrenheit (multiply the Celsius reading by 1.8 and then add 32).

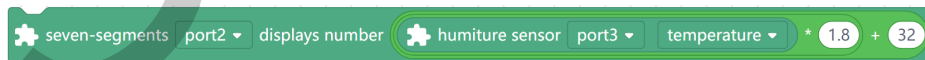


FIGURE 3

Another way to accomplish this is to use a variable named "Fahrenheit" and two math functions from the Operators palette to convert the temperature sensor reading from Celsius to Fahrenheit (see Figure 4).

```

when mBot(mcore) starts up
forever
  if when on-board button pressed? then
    repeat 10
      set Temperature (F) to humiture sensor port3 temperature * 1.8
      set Temperature (F) to Temperature (F) + 32
      seven-segments port2 displays number Temperature (F)
      LED panel port1 shows image F
      wait 1 seconds
      seven-segments port2 displays number humiture sensor port3 humidity
      LED panel port1 shows image H
      wait 1 seconds
    LED panel port1 clears screen
    seven-segments port2 displays number 0000
  
```

FIGURE 4 This program converts the Celsius reading to Fahrenheit. Note that we also changed the LED panel to show "F" for Fahrenheit.

EXTENSION 2 Initiate the program based on a voice command

In this example solution, the robot is triggered to display the weather when a sound threshold is met. The Sound Sensor outputs values from 0 to 1023. We had to do some testing to find an appropriate threshold; the value had to be high enough that the program wasn't triggered by ambient sounds but low enough that the user didn't need to yell to trigger the program. We created a simple program to determine the appropriate trigger, then modified our program as follows:

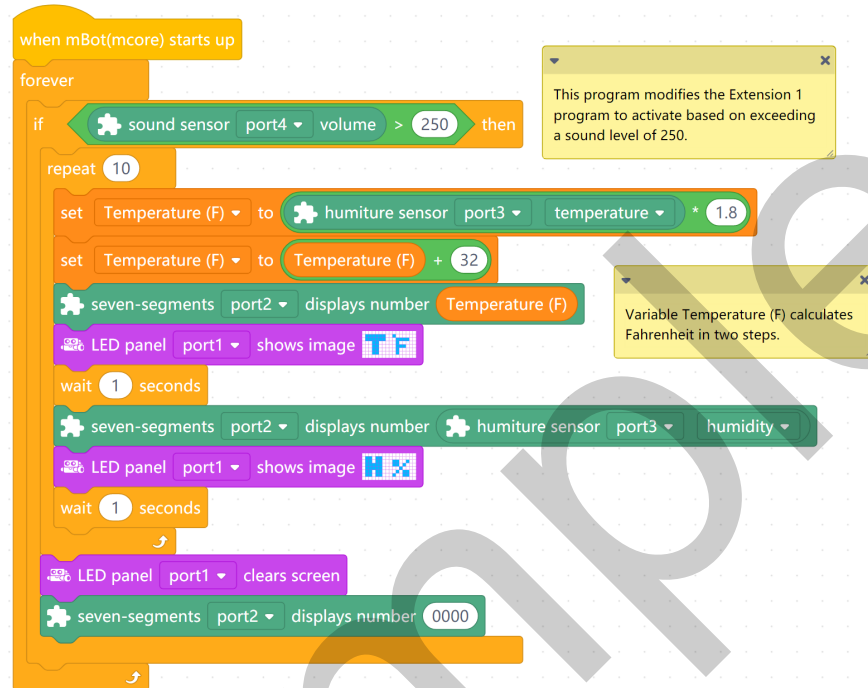


FIGURE 5 This program monitors the Sound Sensor. If the volume goes above a threshold value, the temperature and humidity are displayed.

EXTENSION 3 Investigate heat index and wind chill

This extension does not require additional coding, but is instead intended to help students gain an appreciation of how heat index and wind chill are determined. There is a host of resources on the Internet. We recommend www.weather.gov as a reputable resource.