

# Manual Drive

New out of the box (or reloaded with the Default Program), mBot has three pre-programmed modes of operation. You select the mode by pressing the **A**, **B**, or **C** button on the Infrared (IR) Remote Control. You can also step through the three modes by pressing the Onboard Button near the front of mBot.

## Mode A—Manual Drive

In Mode A, you can drive the robot forward or backward and turn it right or left by pressing the arrow keys on the Remote Control. Numbered keys correspond to sounds. When you switch to Mode A, the LEDs are white.

## Mode B—Obstacle Avoidance

In Mode B, mBot drives forward until it detects an obstacle such as a wall or chair via its ultrasonic range finder. Then, it will back up, turn, and resume driving forward. When you switch to Mode B, the LEDs are green.

## Mode C—Line Following

In Mode C, mBot follows a black line while driving forward. It keeps track of the black line using the Line-Follower Sensor, which is mounted to the front of mBot. When you switch to Mode C, the LEDs are blue.

Let's start learning to program by looking at the code used in Mode A.

## Understanding Mode A—Manual Drive

Infrared (IR) radiation is all around us. But what is it? IR is a form of electromagnetic radiation, just like visible light. However, because IR radiation has a lower frequency than visible light, humans are not able to see it.

Many household electronics (e.g., TVs, audio systems, and games) come with remote controls that use IR to send signals between a transmitter and a receiver. The remote control for mBot uses IR in the same way. The IR transmitter (an infrared LED) is on the end of the Remote Control, and the IR receiver is located on the front end of mBot. When you press one of the buttons on the Remote Control, the infrared LED sends out a series of IR pulses (on and off). The mCore microprocessor in mBot interprets the pulses and determines which button was pressed.

## Drive Forward Program

Let's start by writing a simple program that will allow you to manually control mBot with the IR Remote Control. In this program, which we'll call the Drive Forward program, you will make mBot move forward when the **▲** (up arrow) button on the IR Remote Control is pressed.

1. Open the mBlock 5 software on your computer or Chromebook. Near the middle of the screen you will see a vertical list of palettes, starting with "Looks" at the top. These palettes contain blocks—the raw material from which you will build your program. You can drag blocks to the right side of the screen, the programming area, where you build your program.
2. Connect mBot to your computer or Chromebook. You can do this either via a USB cable or Bluetooth® wireless technology.
  - USB cable: Use the USB cable to connect mBot to a USB port on your computer or Chromebook.
  - Bluetooth: It is also possible to connect to mBot using a Makeblock Bluetooth Dongle paired with your computer or Chromebook. Information on how to make this connection can be found in *Getting Started with mBot and mBlock 5*, available on the Vernier website, [www.vernier.com](http://www.vernier.com)
3. Once connected to mBot, you can select your device in mBlock 5. The mBlock 5 software defaults to Codey as the device, so click the + button, and then select the picture of the mBot from the Device Library list that appears.

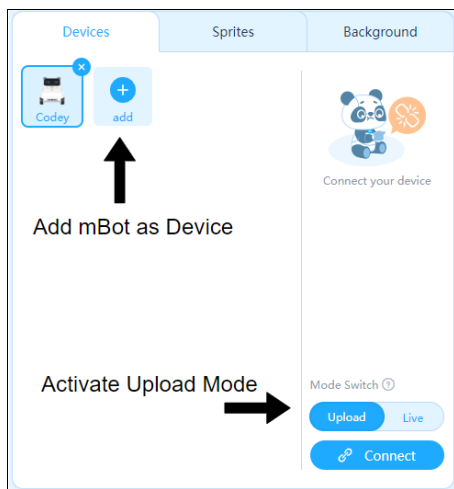



FIGURE 1 Add mBot and activate Upload mode

4. Now, click the Connect button,  **Connect**. mBlock 5 will suggest the port to which mBot is connected. Click Connect.
5. To be able to upload code to mBot, activate Upload mode by clicking the switch for Upload mode (see Figure 1).

You can now build a program for mBot by dragging blocks from the palettes (middle of the screen) to the programming area (right side of the screen). Continue following the steps to build the Drive Forward program:

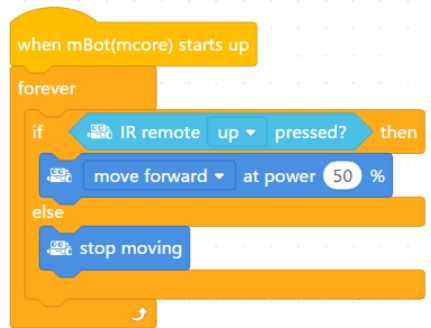


FIGURE 2 Drive Forward program

6. Click the Events palette to display all the blocks in that palette. Drag the "when mBot(mcore) starts up" block to the programming area. Note that this block has a rounded top; it is called a *hat block*. A hat block starts a program or a subroutine. **TIP** If this block is unavailable, you do not have your mBot properly connected to the computer or Chromebook.
7. Next, click the Control palette, and drag a "forever" block and an "if then else" block to the programming area. Position the blocks in the proper locations for the Drive Forward Program (see Figure 2).
8. Click the Action palette, and drag "move forward at power" and "stop moving" blocks to the programming area. Insert the blocks into your Drive Forward program.
9. Finally, click the Sensing palette, and drag the "IR remote" block into the program.
10. When you have the blocks in position, you may notice one thing that needs to be changed to match the program in Figure 2. Consider the "IR remote" block:



FIGURE 3 "IR remote" block

By default, the **A** button is selected in the "IR remote" block. For the Drive Forward program, we want to control mBot with the up arrow button (▲) on the remote. Click the **A** to view the different button options, and choose **up** (see Figure 4) from the menu.

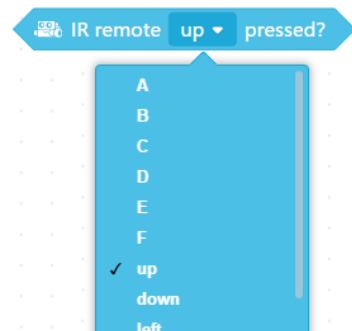



FIGURE 4 Buttons available in the "IR remote" block menu

## TRY IT OUT

Once you have built the Drive Forward program and set the menus properly, try out the program with your mBot:

1. To upload the program, click the Upload button,  Upload.
2. Wait for the program to upload. If the program uploads successfully, you will see a message telling you the upload is complete. The program will start running as soon as the program has uploaded. **NOTE** If you are using a USB cable to connect the computer and mBot, you can now disconnect the cable if you would like.
3. Use the IR Remote Control to try out the program. Does mBot move forward when you press the ▲ button?

## Forward and Backward Program

Let's modify the program so you can make mBot go both forward and backward. We'll call this the Forward and Backward program, and to build it, we need to add some blocks.

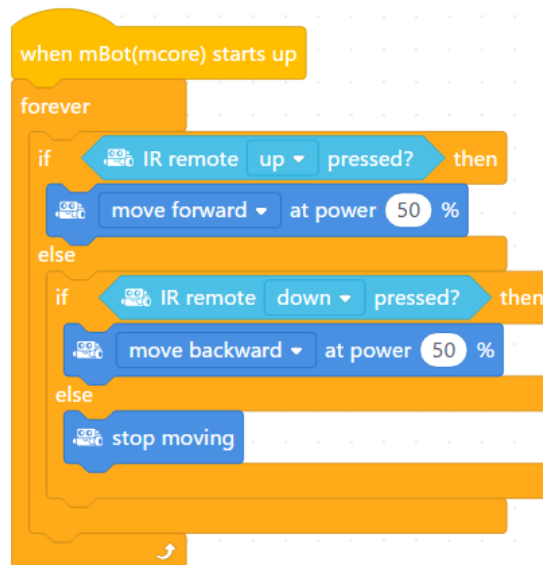


FIGURE 5 Forward and Backward program

Notice the structure of the Forward and Backward program. It has two "if then else" blocks. The second one is inside the else portion of the first one. When the ▲ button is pressed on the IR Remote Control, mBot moves forward. If the ▲ button is continually pressed (if it is held down), mBot will continue to move forward. If the button is released, mBot stops moving and checks to see if the ▼ button is pressed. If the ▼ button is pressed, mBot will move backward. If neither button is pressed, mBot stops moving.

## TRY IT OUT

Build the Forward and Backward program (Figure 5), and then upload the program to mBot. Are you able to use the ▲ and ▼ buttons to make mBot move forward and backward?

**NOTE** If you are connecting via USB and disconnected the cable to test your program, reconnect to mBot to upload the program.

## Four-Directional Movement Program

When you have the Forward and Backward program working, add a few more blocks to build the Four-Direction Movement program. This more complicated program will allow you to have four-directional control of your mBot.

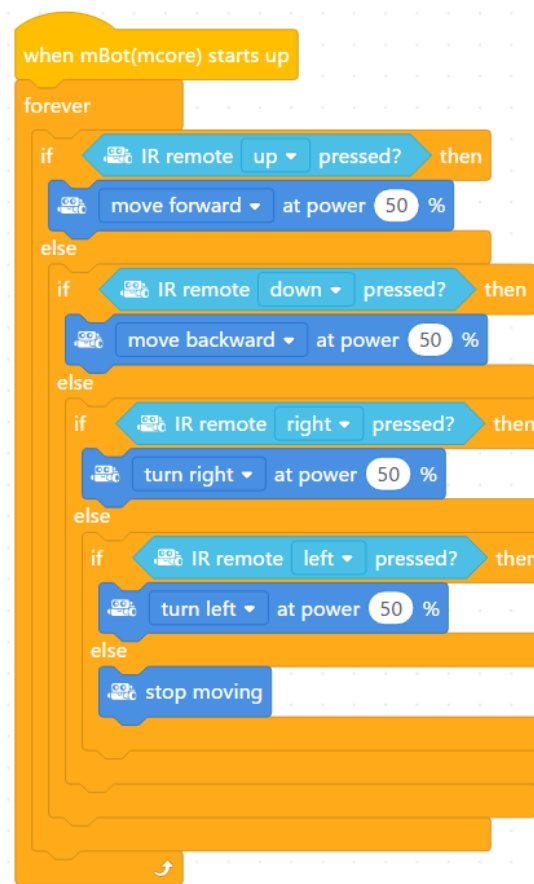


FIGURE 6 Four-Direction Movement program

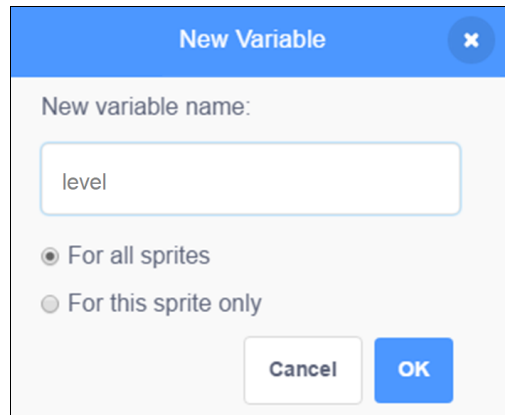
## TRY IT OUT

Upload the Four-Direction Movement program (Figure 6) to mBot. See if you can move mBot in all four directions using the arrow buttons on the IR Remote Control.

## Improving Manual Drive with Variables

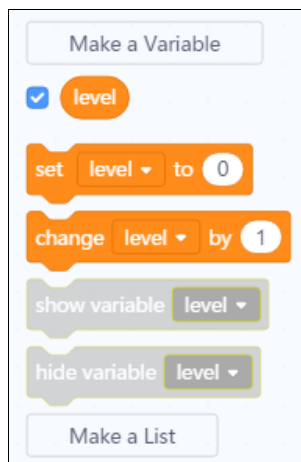
As you work with mBot, you might want to make it move more quickly or slowly. If you were to make this adjustment in the Four-Direction Movement program, you would have to change the setting in four different places. However, you can write a more elegant program—and one that is more easily modified—if you include a variable that can be used to control the power.

In mBlock, click the Variables palette, and then click the Make a Variable button. Enter a name for your variable, select "For all sprites", and click OK. In our example, we used "level" as the variable name.



**FIGURE 7** Creating a new variable

After you click OK, mBlock will add several new blocks to the Variables palette (see Figure 8). Using these blocks, you can incorporate the variable into your code and adjust its value (see Figure 9).



**FIGURE 8** Commands available when defining a variable



FIGURE 9 Four Directions with Variable program

## TRY IT OUT

In mBlock, create the Four Directions with Variable program (Figure 9), and upload the program to mBot. Does mBot drive like it did before? Change your value for the "level" variable and upload your updated program to mBot. Does mBot drive faster? How low can you set the "level" variable and still make mBot move?

**TIP** The power setting can be in the range of -100 to 100. If you set the level above 100, the value will default to 100. Negative numbers do not really make sense; a negative number in the "drive forward" block makes mBot move backwards.

## Add Comments to Your mBot Program

As a programmer, it is important to frequently add comments to your code. When you are writing a program, you know exactly how the code works, and it may feel as though comments are unnecessary. However, if you do not think about the program for a few weeks, you may not remember the situation as well as you do now. Also, it is common for programmers to share code with each other. In many of the following activities, you and your classmates will write your own programs, and you may want to share them with each other. Comments will help your classmates interpret your code and make it easier for them to build upon it.

In mBlock, it is easy to add comments to your program; comments are connected to a single block. To create a comment, right-click (Windows or ChromeOS) or CTRL-click (macOS) the block and choose Add Comment.

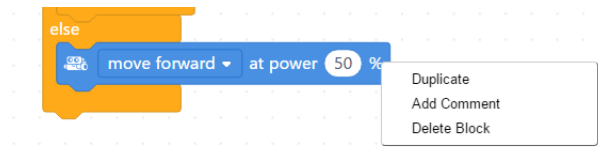


FIGURE 10 Adding comments in mBlock

Here is our program with comments:



FIGURE 11

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

**NOTE** 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.

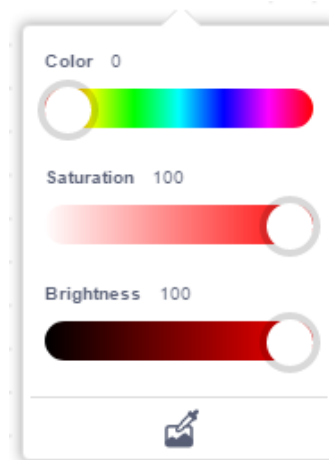


## Using Lights on mBot

If you watch mBot operating in Mode A, you may notice that there is more going on than we have incorporated in our program. One significant difference is that the mBot LEDs light up as follows:

Motion	LED color and state
Forward	Both LEDs are green.
Backward	Both LEDs are red.
Left	The left LED is green.
Right	The right LED is green.

You can use the "LED all show color" block from the Show palette to control the mBot LEDs. There are two adjustments you can make with the block. The first menu identifies which LED(s) are being controlled (all, left, or right). Click the colored circle to adjust the color, saturation, and brightness of the LED(s). You can even click the eye dropper and match the color from somewhere else in your mBlock program.



**FIGURE 12** Menu options in the "LED all show color" block

## Manual Drive Program

As a final step in the Manual Drive program, add code to control the LEDs. Use your creativity to create unique lighting pattern that you can use to identify your mBot.

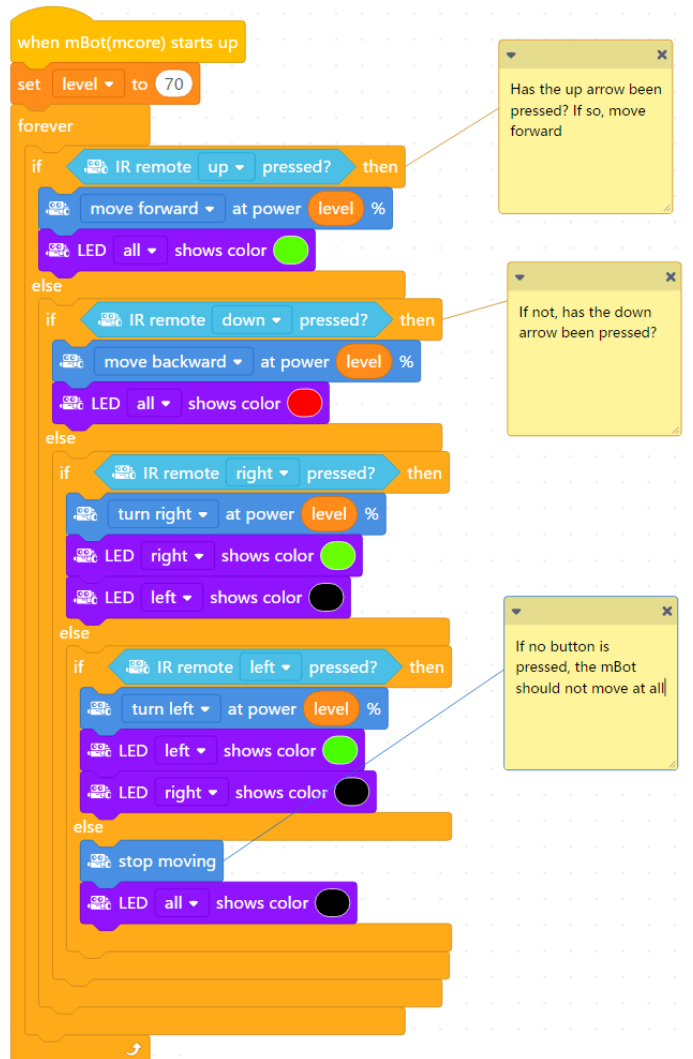


FIGURE 13 Manual Drive program with coordinated LEDs

### TRY IT OUT

Try out the Manual Drive program. Are you able to control the LEDs on your mBot? How similar is this program to Mode A of the Default Program?

## Challenge Extensions

### EXTENSION 1 Change Motor Power Setting with Numeric Buttons

Many vehicles, such as SEGWAY® and drones, have slow, medium, and fast speed settings. In the our Manual Drive program in Figure 13, we used a power level of 70%. Modify your Manual

Drive program to allow the operator to choose one of three different power levels using the IR Remote Control before driving. Be careful that you don't set the power level too low, or mBot may not move at all.

### EXTENSION 2 Backup Warning

Most construction vehicles make a warning sound when they move backward. Can you add a back-up warning to the Manual Drive program? mBot has a small Buzzer located near the front. You can use the Buzzer to make sound with the "play sound" block from the Show palette:



FIGURE 14 "play sound" block

The sound from the Buzzer is not very loud, but you have a lot of control over the duration and frequency. The frequency (vibrations per second) affects the pitch of the buzzer.

Experiment with the different sounds until you are able to produce a sound that is similar to what you hear when a vehicle is backing up. Add the sound to your Manual Drive program so a warning sound plays when mBot is moving backward.

