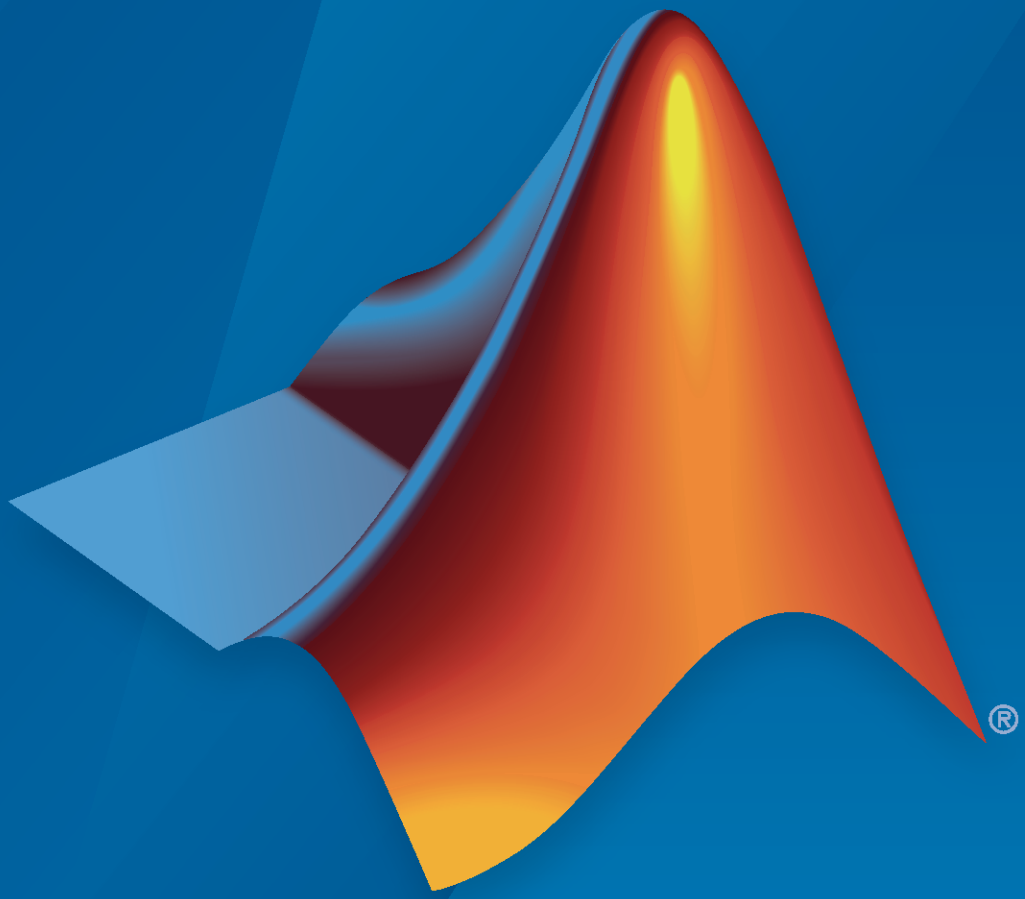
MATLAB® Support Package for LEGO® MINDSTORMS® EV3 Hardware

User's Guide



MATLAB® & SIMULINK®

R2023b

MathWorks®

How to Contact MathWorks

Latest news: `www.mathworks.com`

Sales and services: `www.mathworks.com/sales_and_services`

User community: `www.mathworks.com/matlabcentral`

Technical support: `www.mathworks.com/support/contact_us`

Phone: 508-647-7000

The MathWorks, Inc.
1 Apple Hill Drive
Natick, MA 01760-2098

MATLAB® Support Package for LEGO® MINDSTORMS® EV3 Hardware User's Guide

Trademarks

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See `www.mathworks.com/trademarks` for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

Patents

MathWorks products are protected by one or more U.S. patents. Please see `www.mathworks.com/patents` for more information.

Revision History

September 2014	Online only	Revised for Version 1.0 (R2014b)
March 2015	Online only	Revised for Version 15.1.0 (R2015a)
September 2015	Online only	Revised for Version 15.2.0 (R2015b)
March 2016	Online only	Revised for Version 16.1.0 (R2016a)
September 2016	Online only	Revised for Version 16.2.0 (R2016b)
March 2017	Online only	Revised for Version 17.1.0 (R2017a)
September 2017	Online only	Revised for Version 17.2.0 (R2017b)
March 2018	Online only	Revised for Version 18.1.0 (R2018a)
September 2018	Online only	Revised for Version 18.2.0 (R2018b)
March 2019	Online only	Revised for Version 19.1.0 (R2019a)
September 2019	Online only	Revised for Version 19.2.0 (R2019b)
March 2020	Online only	Revised for Version 20.1.0 (R2020a)
September 2020	Online only	Revised for Version 20.2.0 (R2020b)
March 2021	Online only	Revised for Version 21.1.0 (R2021a)
September 2021	Online only	Revised for Version 21.2.0 (R2021b)
March 2022	Online only	Revised for Version 22.1.0 (R2022a)
September 2022	Online only	Revised for Version 22.2.0 (R2022b)
March 2023	Online only	Revised for Version 23.1.0 (R2023a)
September 2023	Online only	Revised for Version 23.2.0 (R2023b)

Contents

1 Install and Use MATLAB Support Package for LEGO MINDSTORMS EV3 Hardware

Install and Use MATLAB Support Package for LEGO MINDSTORMS EV3 Hardware

Adding Support for LEGO MINDSTORMS EV3 Hardware

In this section...

Add support for LEGO MINDSTORMS EV3 hardware to the MATLAB product by installing the MATLAB Support Package for LEGO MINDSTORMS EV3 Hardware.

This process installs the following items on your host computer:

- Third-party software development tools
- MATLAB commands
- Examples

This process does not install a customized version of the EV3 operating system (firmware) on the EV3 brick.

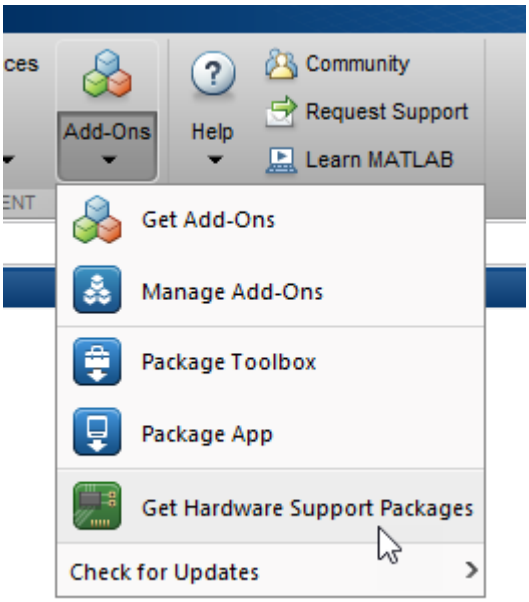
When you complete this installation, you can use MATLAB commands to control and retrieve data from LEGO MINDSTORMS EV3 hardware and peripherals.

The LEGO MINDSTORMS EV3 hardware is also referred to as a brick or as target hardware.

Install, Update, or Uninstall Support Package

Install Support Package

1 On the MATLAB **Home** tab, in the **Environment** section, select **Add-Ons > Get Hardware Support Packages**.



2 In the Add-On Explorer window, click the support package and then click **Install**.

Update Support Package

On the MATLAB **Home** tab, in the **Resources** section, select **Help > Check for Updates**.

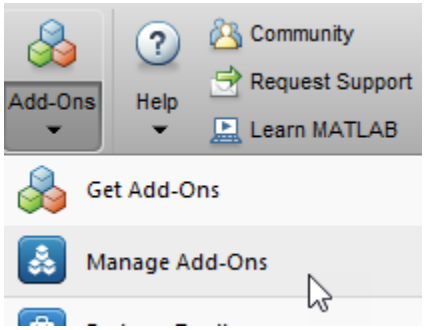
Uninstall Support Package

1. On the MATLAB **Home** tab, in the **Environment** section, click **Add-Ons > Manage Add-Ons**.
2. In the **Add-On Manager** window, find and click the support package, and then click **Uninstall**.

Hardware Setup

Hardware boards and devices supported by MathWorks® require additional configuration and setup steps to connect to MATLAB and Simulink®. Each support package provides a hardware setup process that guides you through registering, configuring, and connecting to your hardware board.

If the support package is already installed, you can start the hardware setup by opening the Add-On Manager.



In the Add-On Manager, start the hardware setup process by clicking the **Setup** button,  .

After starting, the Hardware Setup window provides instructions for configuring the support package to work with your hardware.

Follow the instructions on each page of the Hardware Setup window. When the hardware setup process completes, you can open the examples to get familiar with the product and its features.

Connect the Host Computer to an EV3 Brick

Connect the host computer to the EV3 brick using a USB, wireless network, or Bluetooth® connection. For more information, see:

- “Connect to EV3 Brick with USB Cable”
- “Connect to EV3 Brick Over a Wireless Network”
- “Connect from Windows to EV3 Brick Over Bluetooth” or “Connect a Host Computer Running Windows to an EV3 Brick Using Bluetooth” on page 1-16

Reconnect the Host Computer to an EV3 Brick

This example shows how to reconnect from MATLAB software to an EV3 brick using the same settings as the previous connection.

In the Command Window, use the `legoev3` function to connect to the EV3.

```
myev3 = legoev3;
```

The resulting `legoev3` object, *`myev3`*, represents the EV3 brick.

Connect Host Computer to EV3 Brick over USB Cable

This example shows how to connect from MATLAB software to an EV3 brick over a USB cable.

Required hardware: The USB cable supplied with the EV3 brick.

1. Connect the USB ports on the host computer and the EV3 brick.
2. In the Command Window, use the `legoev3` function to connect from the MATLAB software to the EV3. Specify the connection type, USB.

```
myev3 = legoev3('USB');
```

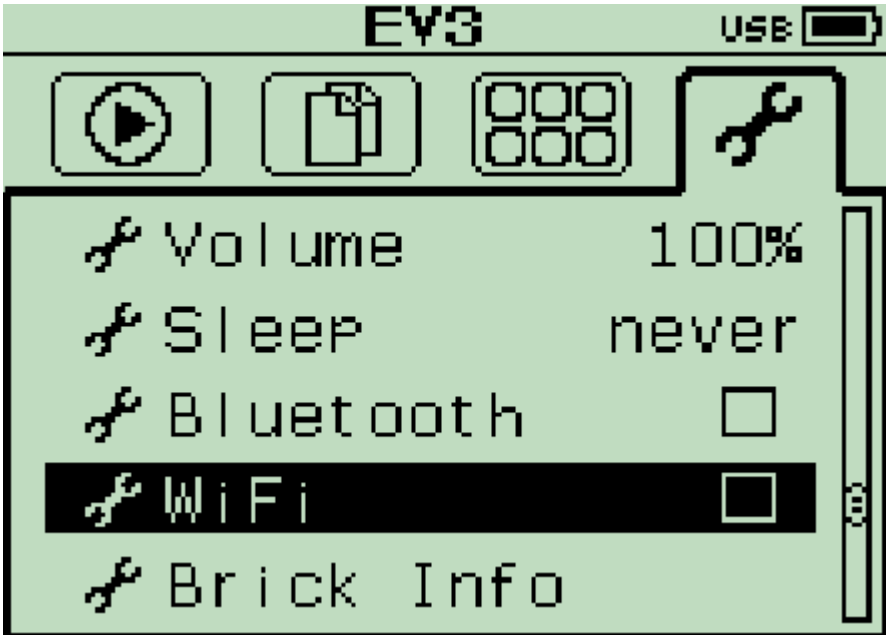
Connect Host Computer to EV3 Brick over Wireless Network

This example shows how to connect from MATLAB software to an EV3 brick using a wireless network.

Required hardware:

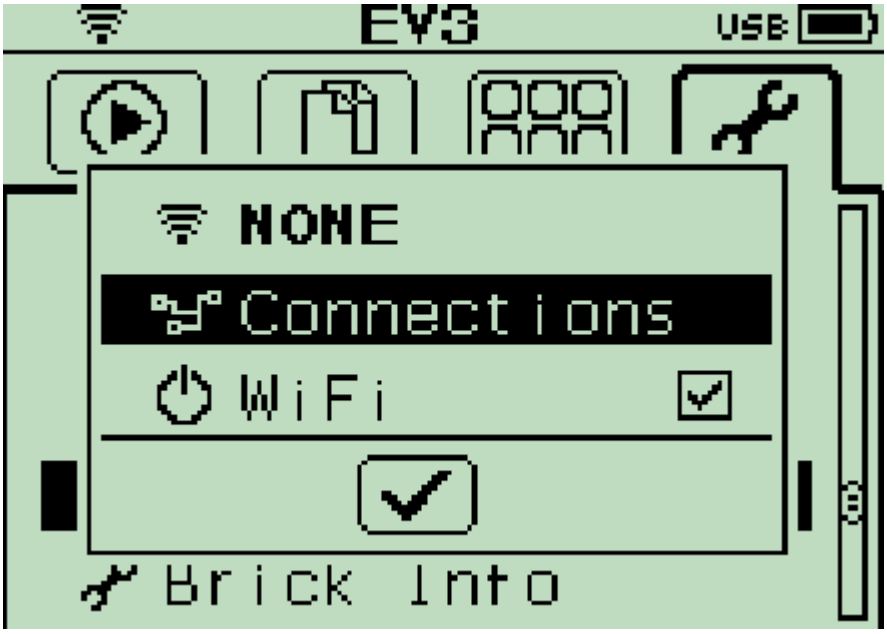
- A wireless network router with encryption set to None or WPA2.
- A wireless network dongle that is compatible with the EV3 brick.

1. Create a wireless network connection from your host computer to the wireless network router.
2. Attach a wireless network dongle to the USB port on the EV3 brick.
3. Shut down and restart the EV3 brick.
4. In the EV3 interface, under settings, select **WiFi**.

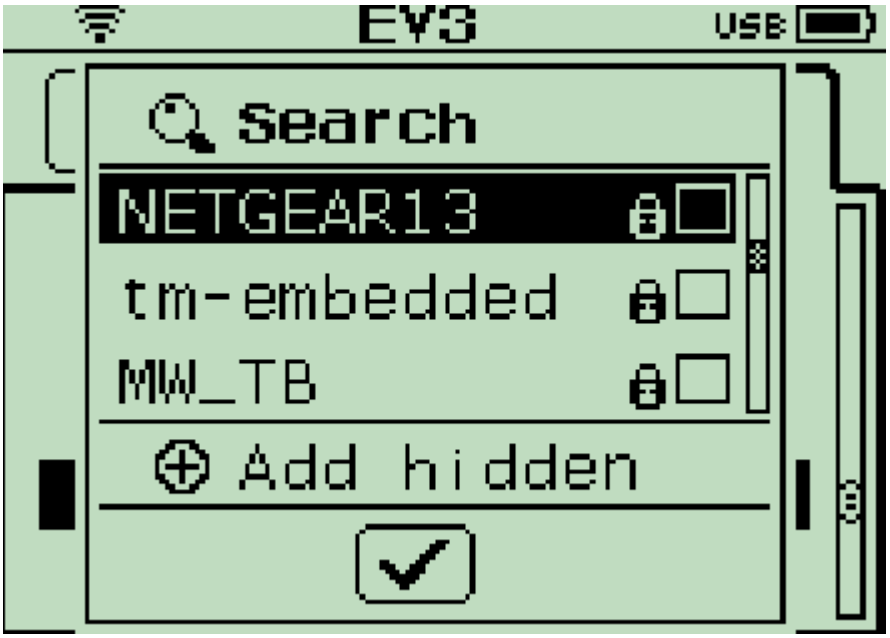


5. Select **WiFi** so that a check mark appears next to it.

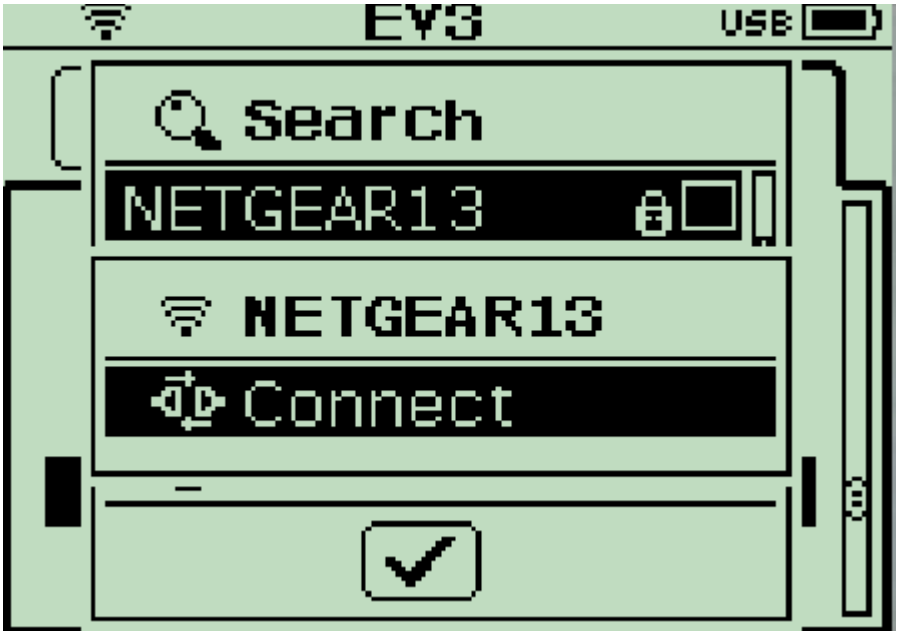
6 Select **Connections**. The EV3 brick scan for wireless network access points.



7 Select a wireless network access point.



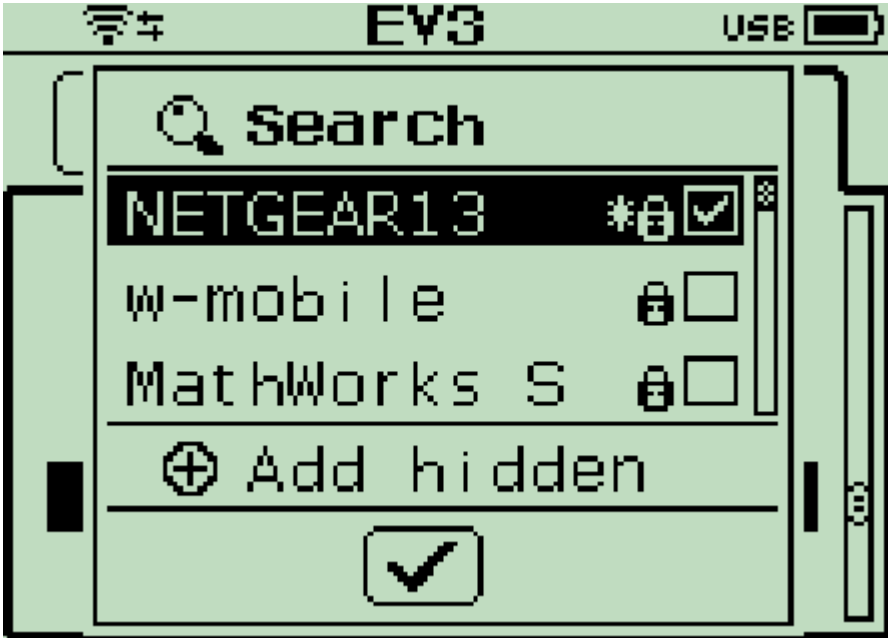
8 Select **Connect**.



9 Select the type of encryption the wireless network access point uses, **WPA2** or **None**.

10 If the wireless network access point uses WPA2 encryption, enter a password.

11 Confirm that the wireless network access point displays a check mark. Then, press the back button or select the large check mark.



12 Confirm that wireless network displays a check mark. The EV3 is connected to the wireless network access point.

13 Select **Brick Info**. Then, write down the **IP address** and the **ID** of the EV3 brick.

14 In the Command Window, use the `legoev3` function to connect from the MATLAB software to the EV3. Specify the connection type, wireless network, the IP address, and the ID of the EV3 brick.

```
myev3 = legoev3('wifi','192.168.1.2','00165340e49b');
```

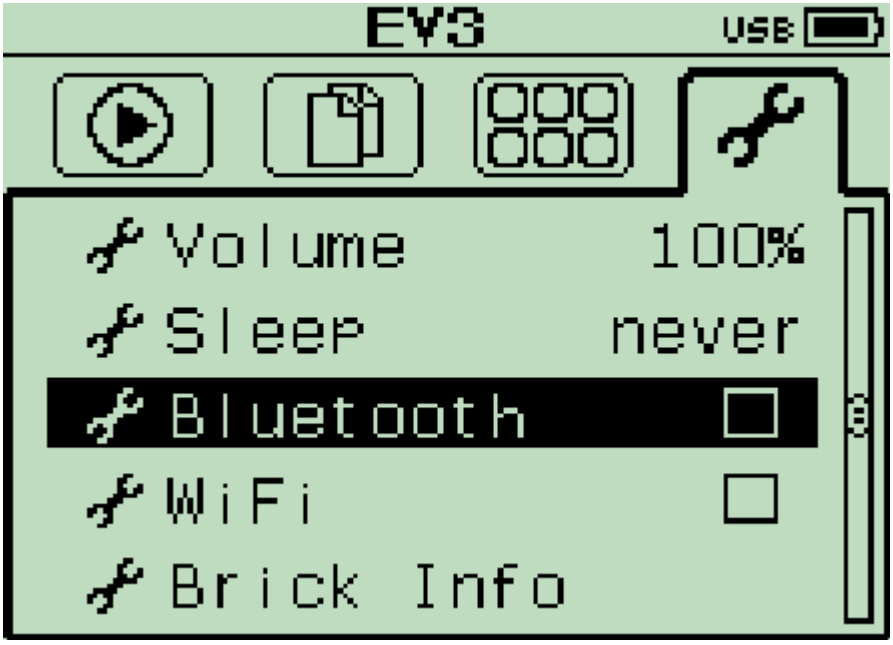
Connect a Host Computer Running macOS to an EV3 Brick Using Bluetooth

This example shows how to connect the MATLAB software running on a Mac computer to an EV3 brick over a Bluetooth connection.

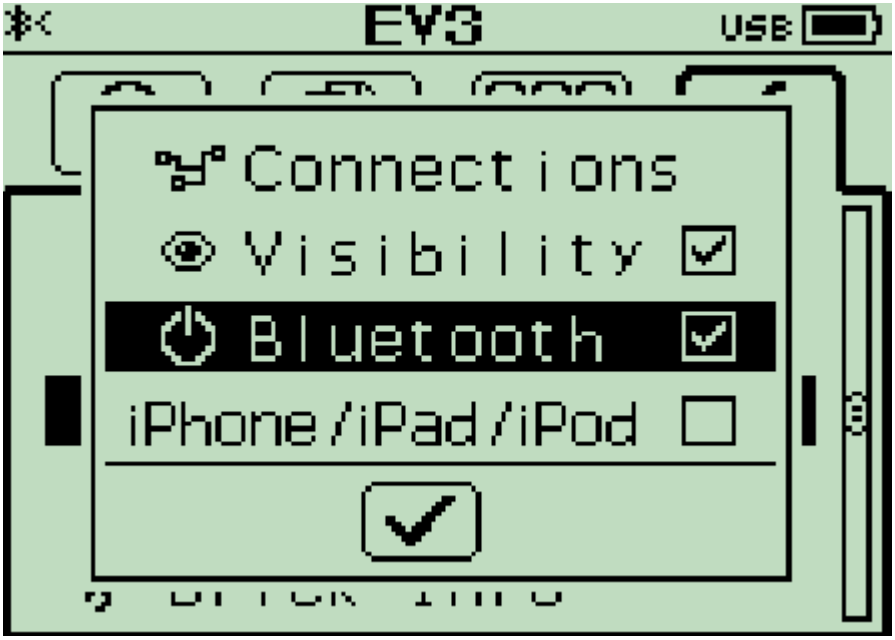
Required hardware: A compatible USB Bluetooth dongle or a host computer with built-in Bluetooth.

Note For an example that uses Windows®, see “Connect a Host Computer Running Windows to an EV3 Brick Using Bluetooth” on page 1-16.

1. On the Mac computer, edit the Bluetooth settings. Enable Bluetooth and make the computer visible to other devices.
2. In the EV3 interface, under settings, select **Bluetooth**.



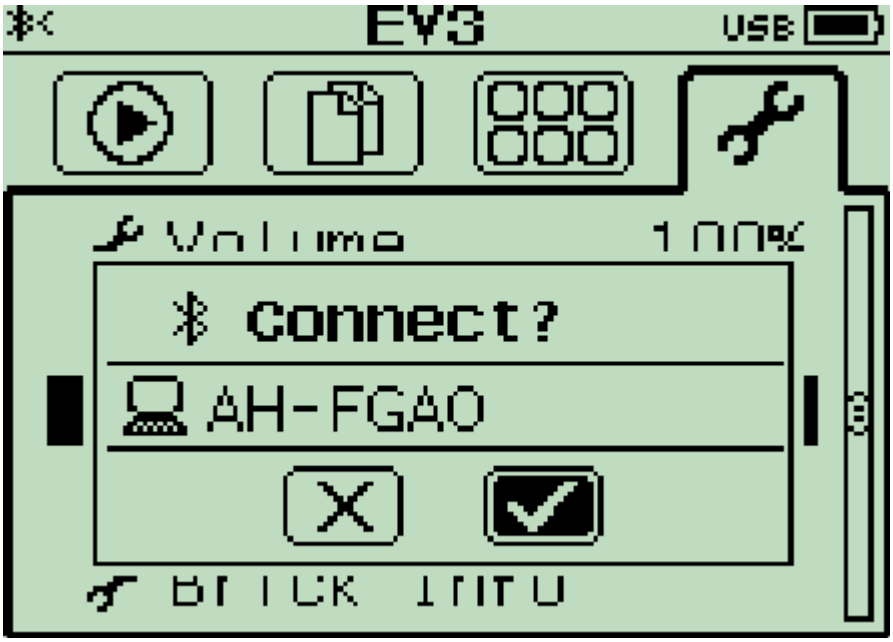
3. Enable **Bluetooth** and **Visibility**. Then, select the large check mark.



4 Confirm that **Bluetooth** is enabled.

5 On the Mac computer, add the EV3 brick to the list of Bluetooth devices.

6 On the EV3 interface, when you see **Connect?**, select the check mark.



7 On the EV3 interface, read the **PASSKEY** value (default: 1234) and select the check mark.

Ignore the underscore character at the end of the passkey value.

8 On the Mac computer, enter the passkey value as the Bluetooth pairing key.

9 Get the device name of the serial port connection to the EV3 brick. Open Terminal in macOS and enter:

```
ls /dev/tty*
```

```
/dev/tty.EV3-SerialPort
```

10 In the MATLAB Command Window, use the `legoev3` function to connect to the EV3. Specify the connection type, Bluetooth, and the device name of the serial port connection to the EV3 brick.

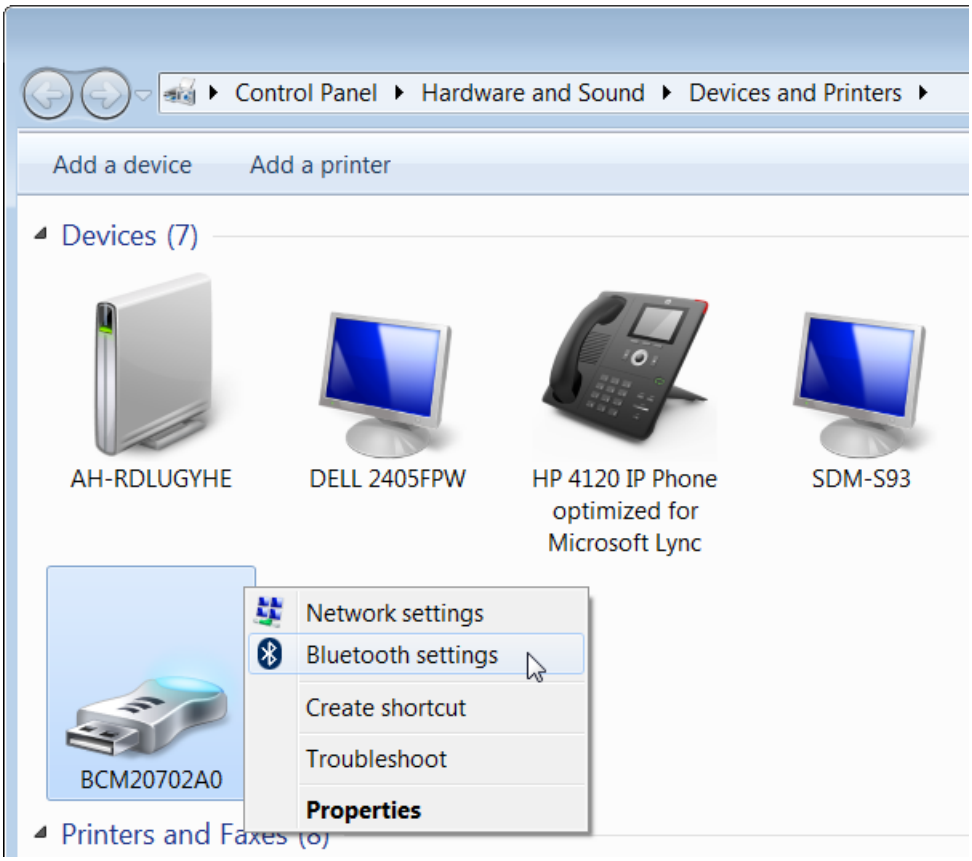
```
myev3 = legoev3('Bluetooth','/dev/tty.EV3-SerialPort');
```

Connect a Host Computer Running Windows to an EV3 Brick Using Bluetooth

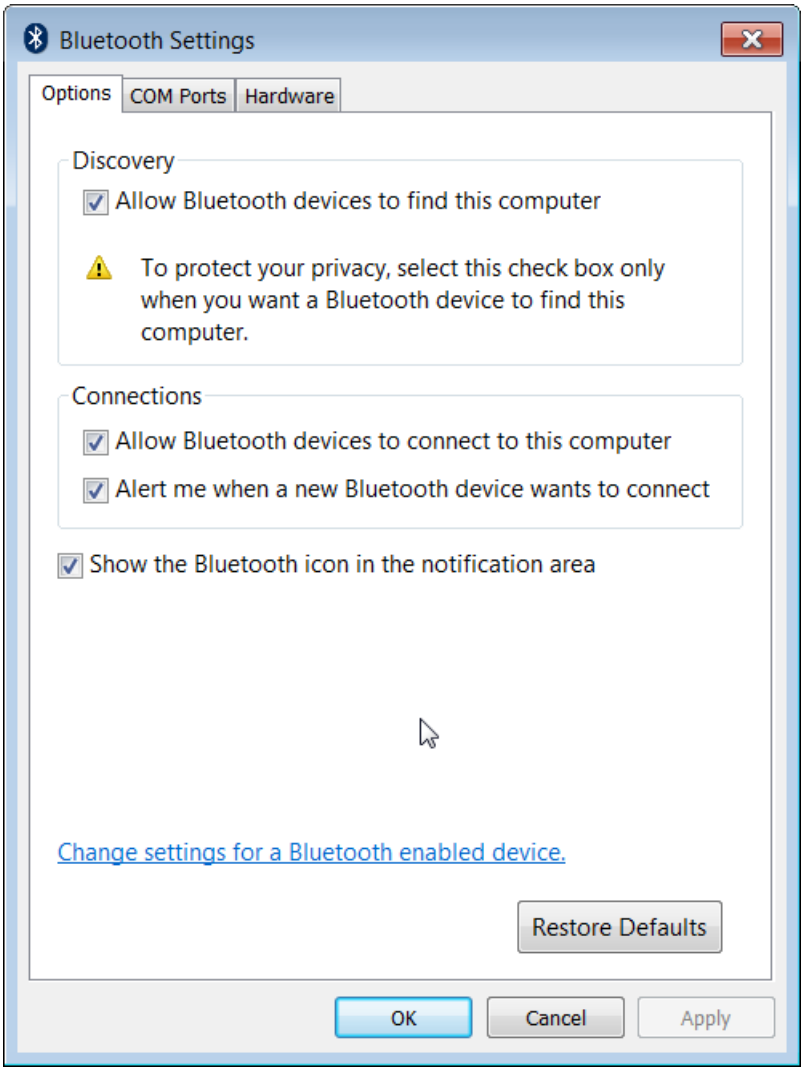
This example shows how to connect to an EV3 brick over a Bluetooth connection from a host computer that is running Windows.

Required hardware: A compatible USB Bluetooth dongle or a host computer with built-in Bluetooth.

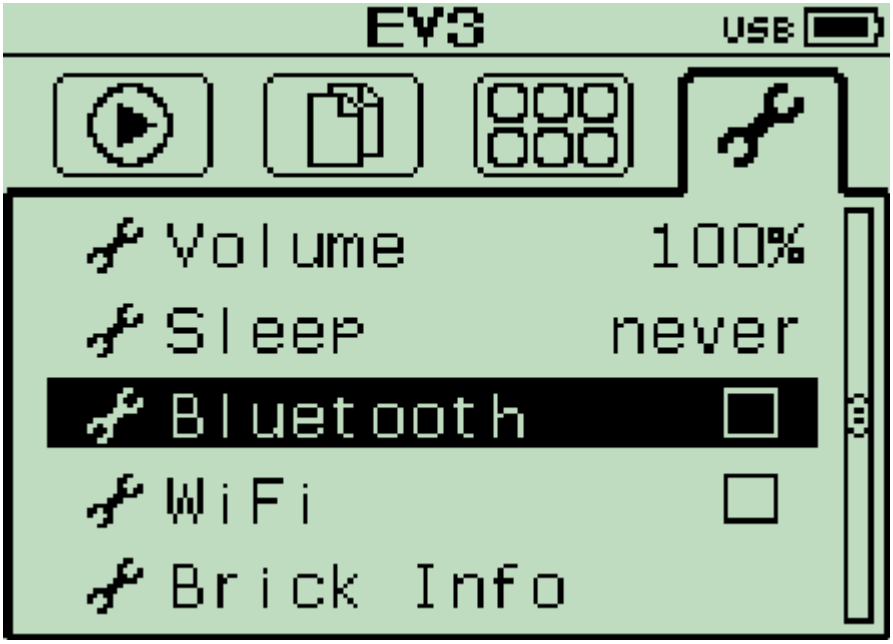
1. If your host computer does not have Bluetooth built-in, attach a compatible Bluetooth dongle to the host computer.
2. In Windows, click **Start > Control Panel > Hardware and Sound > Devices and Printers**.
3. Right-click the Bluetooth device and select **Bluetooth Settings**.



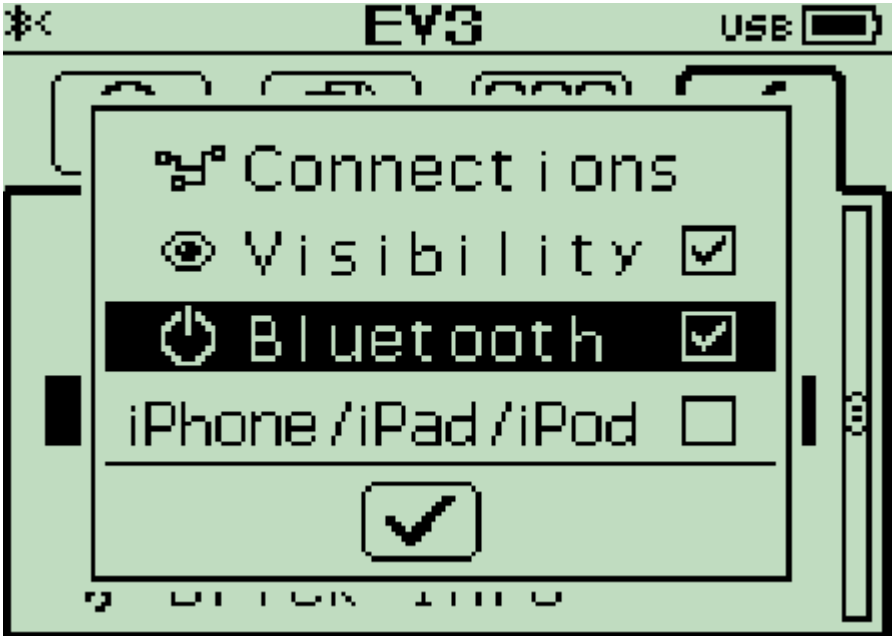
4. Enable the **Discovery** and **Connections** settings.



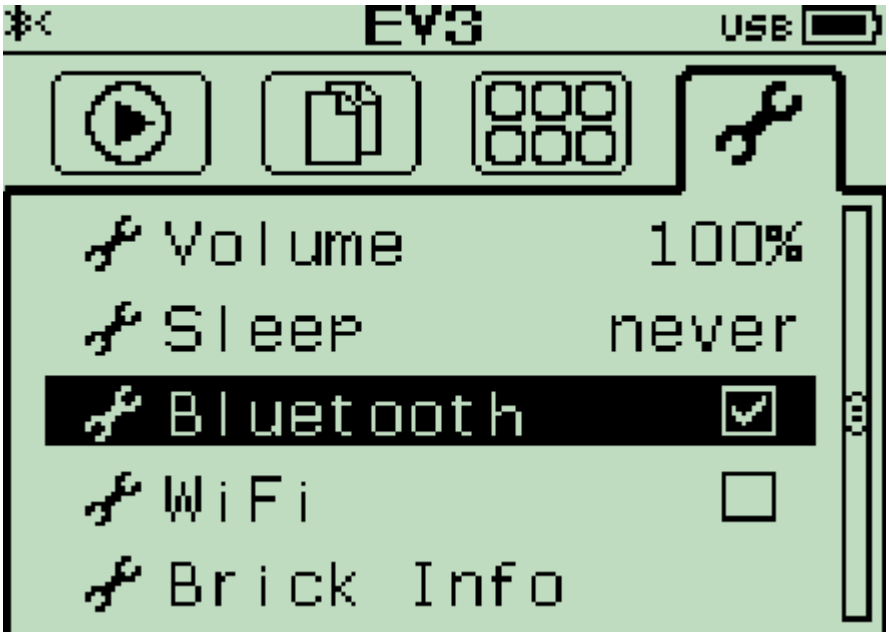
5 In the EV3 interface, under settings, select **Bluetooth**.



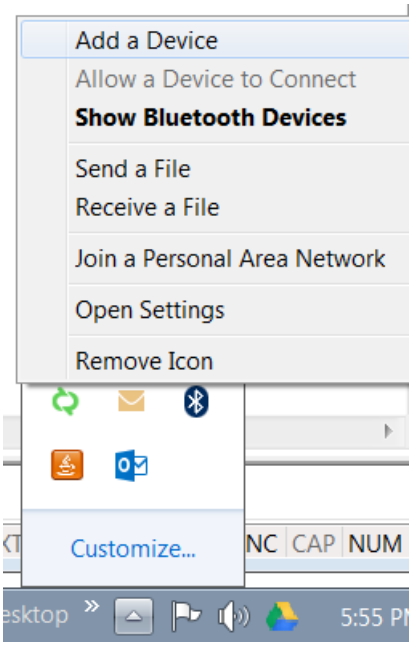
6 Enable **Bluetooth** and **Visibility**. Then, select the large check mark.



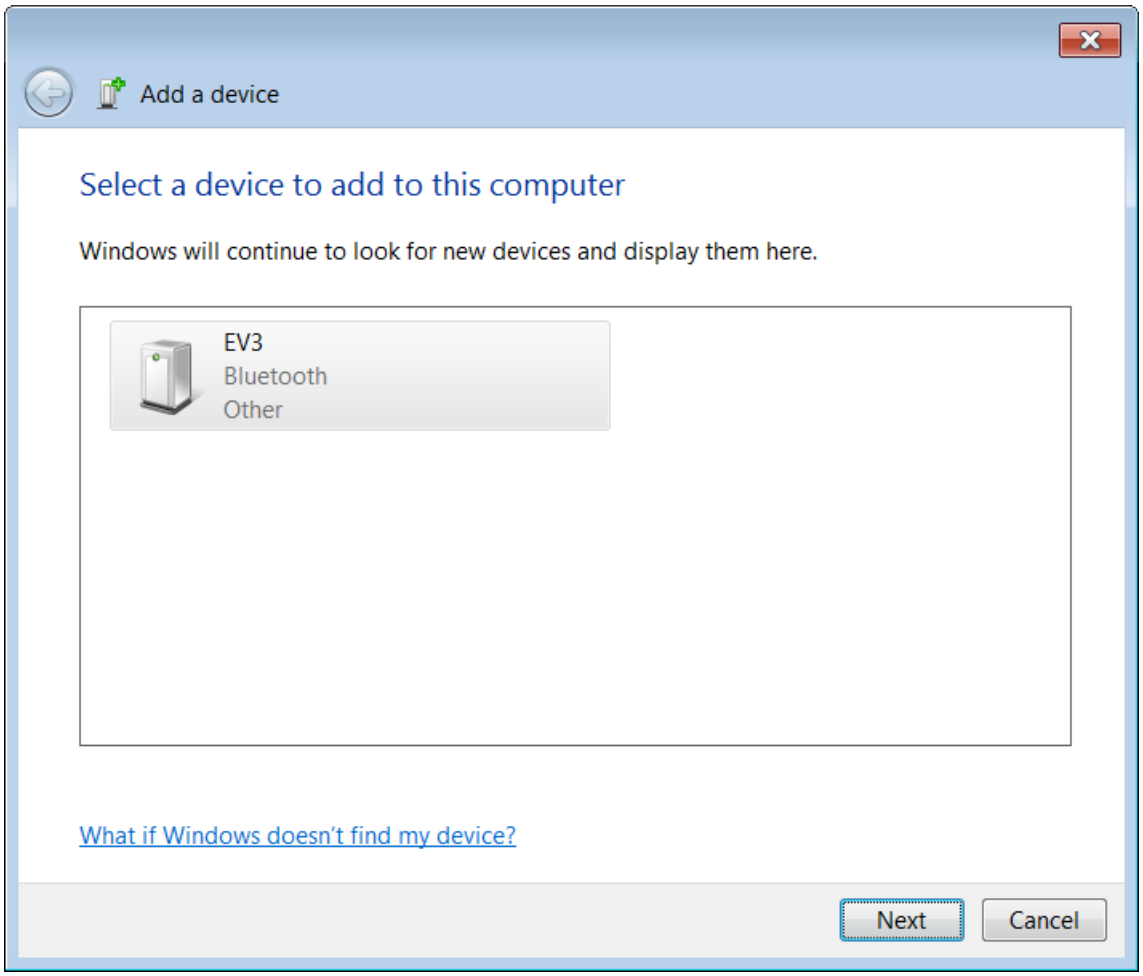
7 Confirm that **Bluetooth** is enabled.



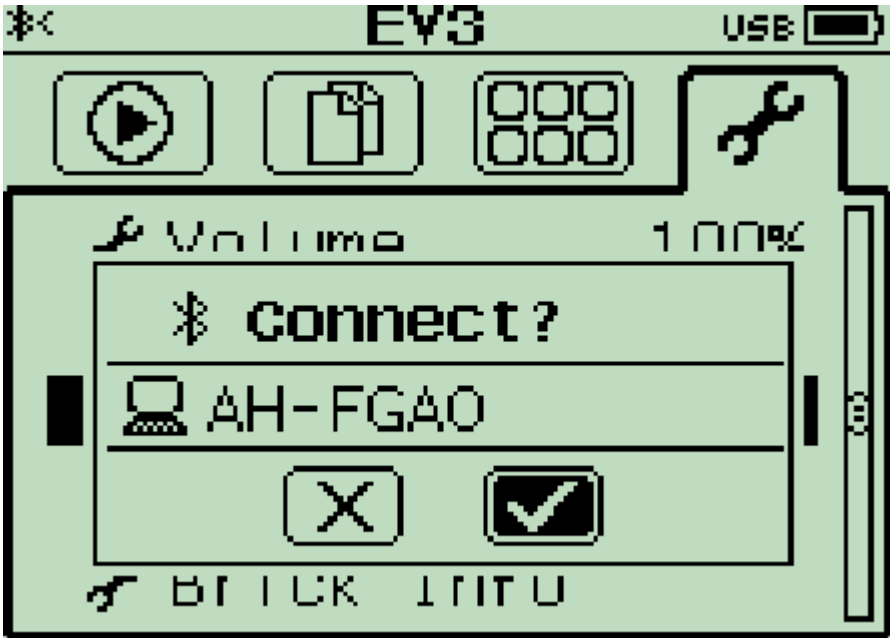
8 In Windows, open the System Tray. Right-click **Bluetooth Devices** and select **Add a Device**.



9 In the **Add a device** dialog box, select **EV3** and click **Next**.



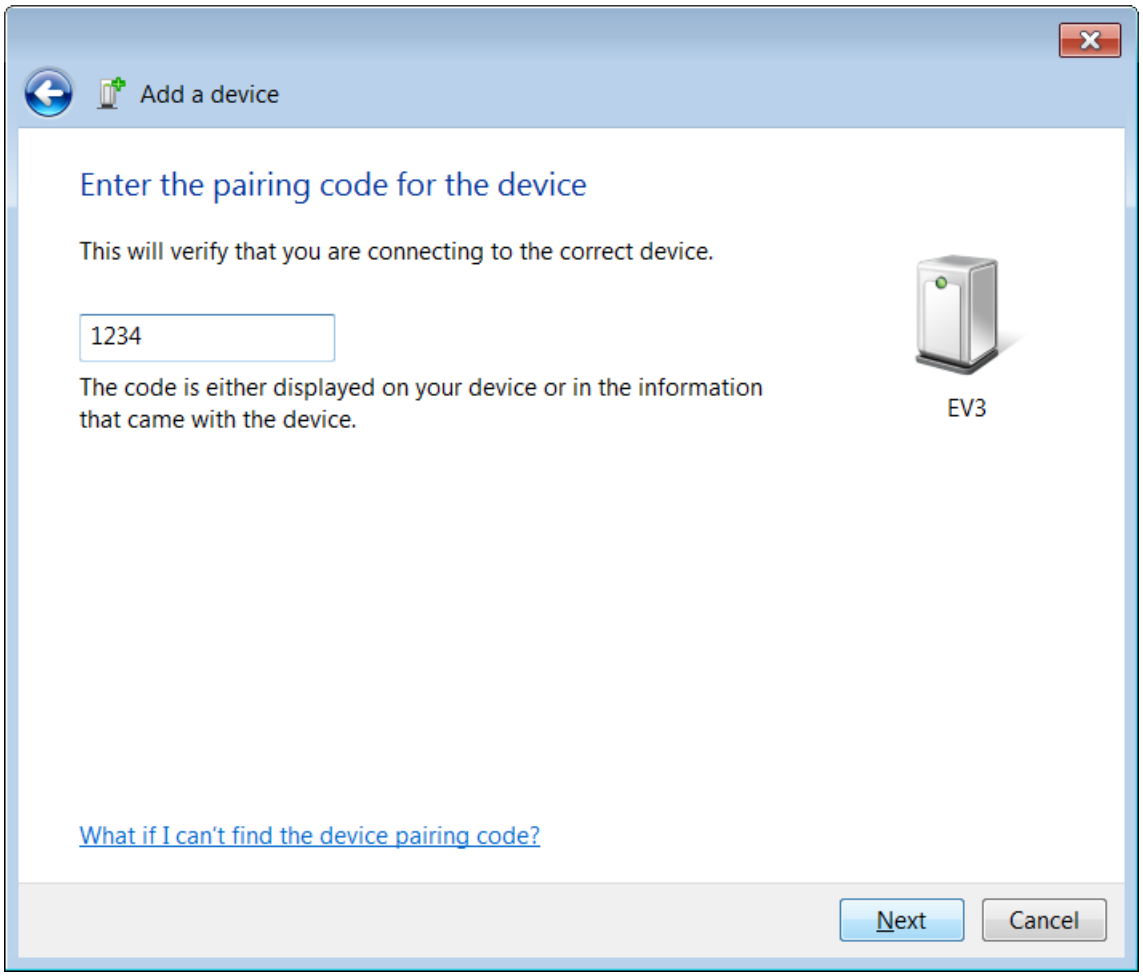
10 On the EV3 interface, when you see **Connect?**, select the check mark.



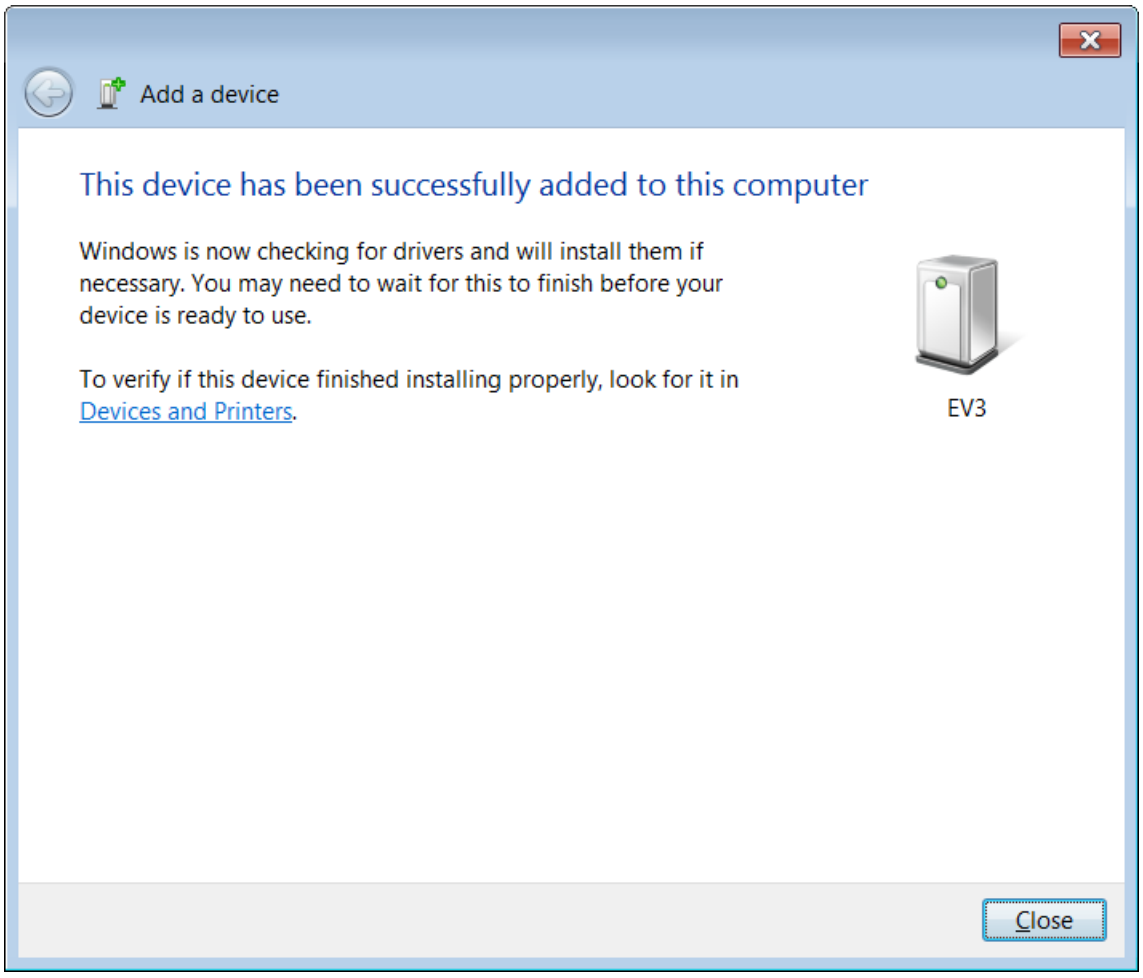
11 On the EV3 interface, read the **PASSKEY** value (default: 1234) and select the check mark.

Ignore the underscore character at the end of the passkey value.

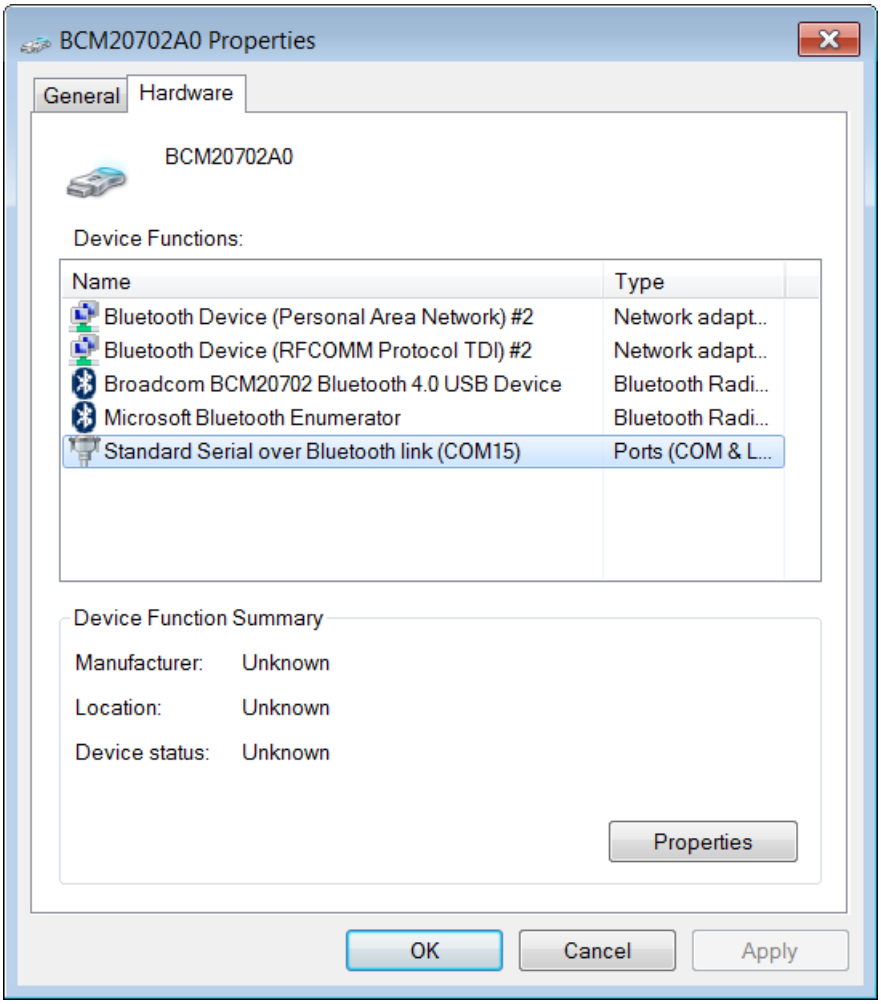
12 On Windows, in the Add a device dialog box, enter the passkey value as the pairing code. Click **Next**.



13 Click **Close**.



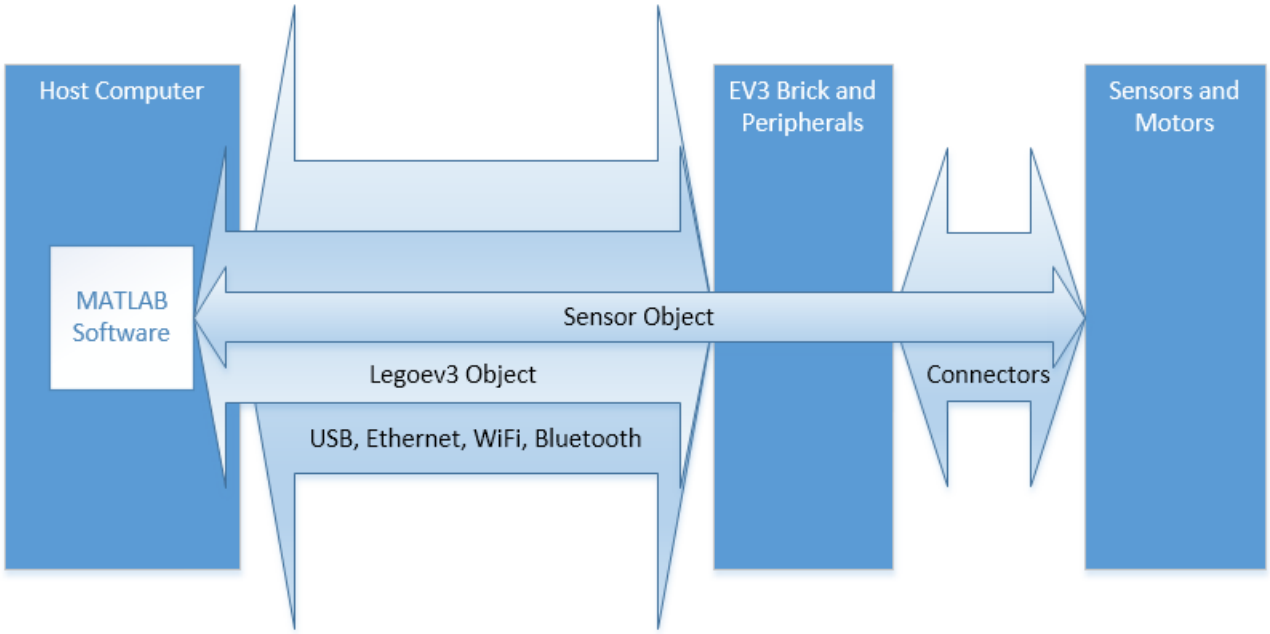
14 In the Properties dialog box for the Bluetooth device, read the COM port number.



15 In the MATLAB Command Window, use the `legoev3` function to connect to the EV3. Specify the connection type, Bluetooth, and the COM port.

```
myev3 = legoev3('Bluetooth','COM15');
```

Connections to EV3 Hardware



When you use MATLAB to interact with the EV3 hardware, you rely on several types of connections:

- Flat black connectors that attach sensors and motors to the EV3 brick.
- A single USB, wireless network, or Bluetooth communication channel between the host computer and the EV3 brick.
- A single `legoev3` object, which connects the MATLAB software to the EV3 brick. You can use this object to interact with EV3 peripherals and to create sensor and motor objects.
- One or more `colorSensor`, `gyroSensor`, `irSensor`, `motor`, `sonicSensor`, and `touchSensor` objects, which you can use to control and get data from sensors and motors.

EV3 Brick Hardware

After installing the MATLAB Support Package for LEGO MINDSTORMS EV3 Hardware, you can use MATLAB commands to control or get the data from:

- The buttons, LCD, speaker, and status light on the EV3 brick.
- Sensors that connect to input ports on the EV3 brick: Color, Infrared, Gyroscopic, Ultrasonic, and Touch.
- EV3 Motors that connect to output ports on the EV3 brick.

The MATLAB software communicates with the EV3 brick by:

- USB cable.
- Wireless network (requires a wireless network dongle attached to the EV3 brick).
- Bluetooth (may require a Bluetooth dongle attached to the host computer).

With this support package, you cannot build an executable from MATLAB code and then run that executable on the EV3 brick.

Note A Simulink support package is available that enables you to run models on the EV3 hardware. For more information, see https://www.mathworks.com/hardware-support/lego-mindstorms-ev3-simulink.html.

Color Sensors

To connect from the MATLAB software to the EV3 Color Sensor (item number 45506), use the `colorSensor` function to create a `colorSensor` object.

- To get the color of an object in front of the sensor, use the `readColor` function with the `colorSensor` object. The sensor evaluates the color of the object in from of the sensor and returns one of the following values: black, blue, brown, green, red, white, yellow, or no color. For example, you can use this capability to sort colored objects.
- To measure reflected light intensity, use the `readLightIntensity` function with the `colorSensor` object, and set the mode to `reflected`. In this mode, the sensor emits red light, and measures the amount of light nearby objects reflect back to the sensor. It represents this measurement as a range, from 0 to 100 (dark to light). For example, you can use this capability to detect a dark line on a light surface.
- To measure ambient light intensity, use the `readLightIntensity` function with the `colorSensor` object, and set the mode to `ambient`. In this mode, the sensor measures the amount of light it receives from the surrounding environment. It represents this measurement as a range, from 0 to 100 (dark to light). For example, you can use this capability to measure sunlight, or to detect when someone turns a light off.

See Also

`colorSensor` | `readColor` | `readLightIntensity`

Gyro Sensors

The gyroscopic sensor measures rotation around a single vertical axis that extends from the top and bottom of the EV3 Gyro Sensor (item number 45505).

To connect from the MATLAB software to the gyro sensor, use the `gyroSensor` function to create a `gyroSensor` object.

- To measure the total rotation of the sensor in degrees, use the `readRotationAngle` function with the `gyroSensor` object. Rotating the sensor clockwise increases the value. Rotating the sensor counterclockwise decreases the value. For example, if you rotate the sensor 360 degrees clockwise and 60 degrees counterclockwise, the measurement value is 300.
- To reset the rotation measurement to zero, use the `resetRotationAngle` function with the `gyroSensor` object.
- To measure the sensor's rate of rotation in degrees per second, use the `readRotationRate` function with the `gyroSensor` object. Rotating the sensor clockwise produces positive values. Rotating the sensor counterclockwise produces negative values. For example, if you rotate the sensor counterclockwise one full rotation (360 degrees) every 10 seconds, the rate of rotation is -36 degrees per second.

IR Sensor and Remote Infrared Beacon

To connect from the MATLAB software to the EV3 Infrared Sensor (item number 45509), use the `irSensor` function to create a `irSensor` object.

- To get the relative distance of an object in front of the sensor, use the `readProximity` function with the `irSensor` object. In this mode, the sensor emits infrared light, and measures the amount of light nearby objects reflect back to the sensor. It represents this measurement as a range, from 0 to 100 (dark to light). The maximum range of the sensor is approximately 70 cm (27 inches).
- To get the relative proximity and heading of an EV3 Remote Infrared Beacon (item number 45508), use the `readBeaconProximity` function with the `irSensor` object. Specify the channel number of the Remote Infrared Beacon. The sensor detects infrared light from the beacon, and uses that data to estimate the heading and distance to the beacon. The sensor returns the relative heading as a value from -25 to 25 (left to right). The sensor returns the relative distance as a value from 0 to 100 (near to far).
- To get the a number that corresponds to pressed buttons on the Remote Infrared Beacon, use `readBeaconButton` function with the `irSensor` object. Specify the channel number of the Remote Infrared Beacon. For information about the values produced by various button combinations, refer to Using the Infrared Sensor Remote Mode.

Note The Remote Infrared Beacon has a red slider switch with four positions. This switch specifies the channel number, 1 through 4, the beacon uses to communicate with the infrared sensor. When you use the `readBeaconProximity` and `readBeaconButton` function, specify the channel number on the beacon. The maximum range from the sensor to the beacon is approximately 200 cm (79 inches).

Motors

To connect from the MATLAB software to the EV3 Large or Medium Motor (item number 45502 or 45503), use the `motor` function to create a `motor` object.

- To control the speed and direction of the motor, assign a value to the `Speed` property of the `motor` object. Input values range from `-100` to `100` (full reverse to full forward). The default value of the `Speed` property is `0`. You can change this value while the motor is running or stopped.
- To start and stop the motor, use the `start` and `stop` functions with the `motor` object.
- To measure the total rotation of the motor in degrees, use the `readRotation` function with the `motor` object. Driving the motor forward increases the value. Driving the motor in reverse decreases the value.
- To reset the number of degrees to zero, use the `resetRotation` function with the `motor` object.

Sonic Sensors

To connect from the MATLAB software to the EV3 Ultrasonic Sensor (item number 45504) use the `sonicSensor` function to create a `sonicSensor` object.

To measure the distance from the sensor to the nearest object, use the `readDistance` function with the `touchSensor` object. The sensor uses ultrasonic sonar to measure the distance to the nearest object, and returns the measurement value in meters.

See Also

`sonicSensor`

Touch Sensors

To connect from the MATLAB software to the EV3 Touch Sensor (item number 45507), use the `touchSensor` function to create a `touchSensor` object.

To check whether an item is pressing the front of the sensor, use the `readTouch` function with the `touchSensor` object. The touch sensor returns a `0` if nothing is pressing the sensor, and `1` if something is pressing the sensor.

See Also

`touchSensor`

Cannot Connect to the EV3 Brick

When you try to connect from the MATLAB software to the EV3 hardware, the Command Window displays an error. For example:

```
myev3 = legoev3
```

```
Error using legoev3 (line 285)
Failed to connect to EV3 through USB. Please check your USB
connection.
```

1. Verify that the EV3 power is on. If not, start the EV3 brick by pushing the power button.
2. If the EV3 power was off, any `legoev3` and sensor objects in the MATLAB Workspace are no longer valid. Clear those objects from the MATLAB Workspace. For example:

   ```
   clear all;
   ```
3. Check that the USB, wireless network, or the Bluetooth cables and hardware are set up as described in “Connect the Host Computer to an EV3 Brick” on page 1-4.
4. Use `matlabshared.supportpkg.getInstalled` to verify that you installed the support package.

   ```
   >> matlabshared.supportpkg.getInstalled
   ```

   ```
          Name         Version  Base Product
   ------------------- ------- --------------
   LEGO MINDSTORMS EV3 14.2.0  MATLAB
   ```
5. Use `matlabshared.supportpkg.checkForUpdate` to check for updates to the support package.

   ```
   matlabshared.supportpkg.checkForUpdate
   ```

   ```
   No support packages need updates.
   ```
6. If you did not install the support package, or updates are available, use `supportPackageInstaller` to install or update the support package.

   ```
   supportPackageInstaller;
   ```

Cannot Connect to a Sensor or Motor

When you try to connect from the MATLAB software to a sensor or motor, the Command Window displays an error. When you use the MATLAB software to get data from a sensor or motor, the Command Window displays unexpected values.

To resolve this issue, first complete the steps in “Cannot Connect to the EV3 Brick” on page 1-33.

Then, perform each of the following steps.

1. Check that sensors are connected to the EV3 input ports, labelled 1, 2, 3, or 4. Check that motors are connected to the EV3 output ports, labelled A, B, C, and D.
2. If you add or change a connection to a sensor or motor, shut down and restart the EV3 brick to ensure that it performs a hardware detection routine.
3. Verify that the sensor appears in the `ConnectedSensors` property of the EV3 brick. For example:

```
myev3 = legoev3

myev3 =

  legoev3 with properties:

      FirmwareVersion: 'V1.03E'
           HardwareID: []
            IPAddress: []
    CommunicationType: 'USB'
         BatteryLevel: 100
     ConnectedSensors: {'touch'  'infrared'  'color'  'sonic'}
```

4. If more than one sensor of the same type is present, specify the input port the sensor uses. For example:

```
myirsensor = irSensor(myev3,2)

myirsensor =

  irSensor with properties:

      Channel: 1
    InputPort: 2
```