Simulink® Support Package for Parrot® Minidrones

User's Guide

MATLAB® & SIMULINK®

R2020a

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

Simulink® Support Package for Parrot® Minidrones User's Guide

Revision History

July 2017	Online only	New for Version 17.1.0 (R2017a)
September 2017	Online only	Revised for Version 17.2.0 (R2017b)
January 2018	Online only	Revised for Version 17.2.2 (R2017b)
March 2018	Online only	Revised for Version 18.1.0 (R2018a)
September 2018	Online only	Revised for Version 18.2.0 (R2018b)
November 2018	Online only	Revised for Version 18.2.1 (R2018b)
March 2019	Online only	Revised for Version 19.1.0 (R2019a)
April 2019	Online only	Revised for Version 19.1.1 (R2019a)
September 2019	Online only	Revised for Version 19.2.0 (R2019b)
March 2020	Online only	Revised for Version 20.1.0 (R2020a)

Contents

1 Working with Simulink Support Package for Parrot Minidrones

2 Run on Target Hardware

Working with Simulink Support Package for Parrot Minidrones

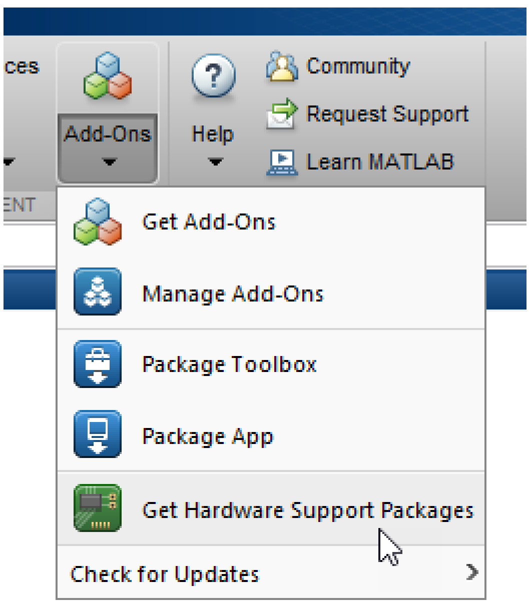
Install Support for Parrot Minidrones

You can add support for Parrot minidrones to the Simulink product by installing the Simulink Support Package for Parrot Minidrones. When you complete this process, and replace the firmware, you can build and deploy flight control algorithms on Parrot minidrones.

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 **Environment** section, select **Help** > **Check for Updates**.

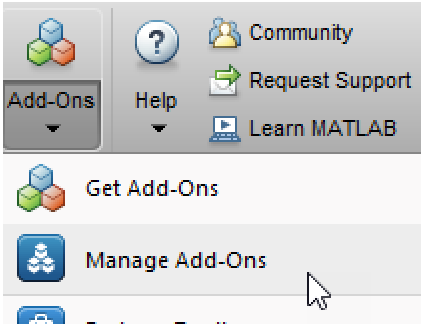
Uninstall Support Package

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

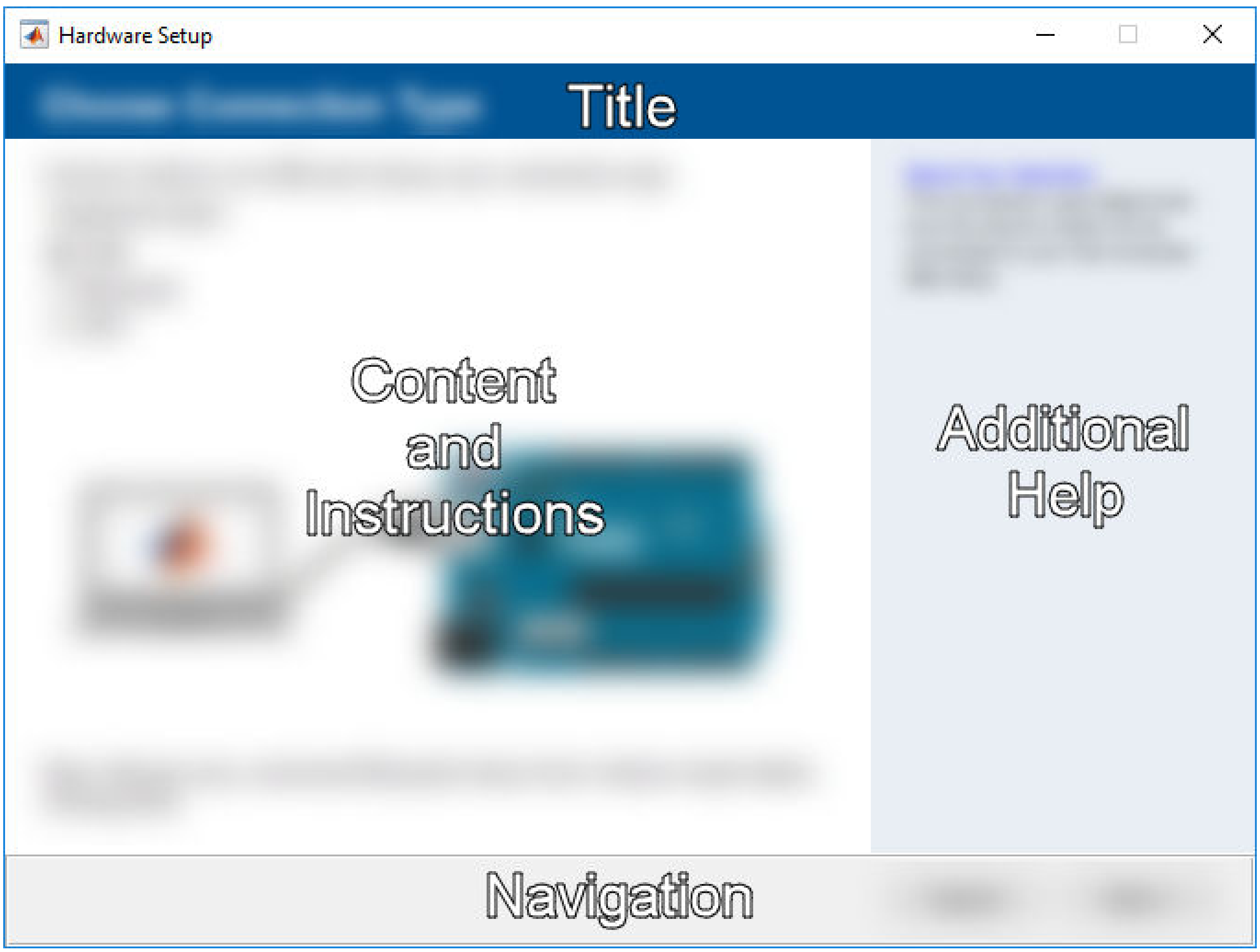
Devices supported by MathWorks® require additional configuration and setup steps to connect to MATLAB and Simulink. Each support package provides a Hardware Setup Wizard that guides you through the process of configuring and connecting to your device.

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 **Gear** icon, .

After starting, the Hardware Setup window provides instructions to configure your device. Hardware Setup uses the following format to present the instructions:



- *Title* - Current step in the setup process.

- *Content and Instructions* - Instructions for the current step in the setup process. Typically an **Action** is provided that will make changes to the current hardware or computer system required for the setup process.
- *Additional Help* - Additional help and considerations for the current step.
- *Navigation* - Click **Next** or **Back** to move to the next or previous step in the setup process. You can also click **Cancel** to exit the process.

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

Note To use Parrot minidrone on Windows computer, you must install RNDIS. For more information, see “Install RNDIS for Parrot Minidrones” on page 1-5.

See Also

“Install RNDIS for Parrot Minidrones” on page 1-5

Install RNDIS for Parrot Minidrones

If you are working on a Windows® computer, you must install Remote Network Driver Interface Specification (RNDIS) before you start working with the Simulink Support Package for Parrot Minidrones.

Note When the Parrot minidrone is connected to the host computer over USB, there is a timeout period (typically, 60 seconds) after which the drone automatically disconnects from the host computer. After the disconnection, the USB drive for Parrot minidrone disappears from the `Devices and drives` list in the Windows PC.

Therefore, ensure that you complete the RNDIS installation procedure fast enough, before the timeout occurs.

Prerequisites

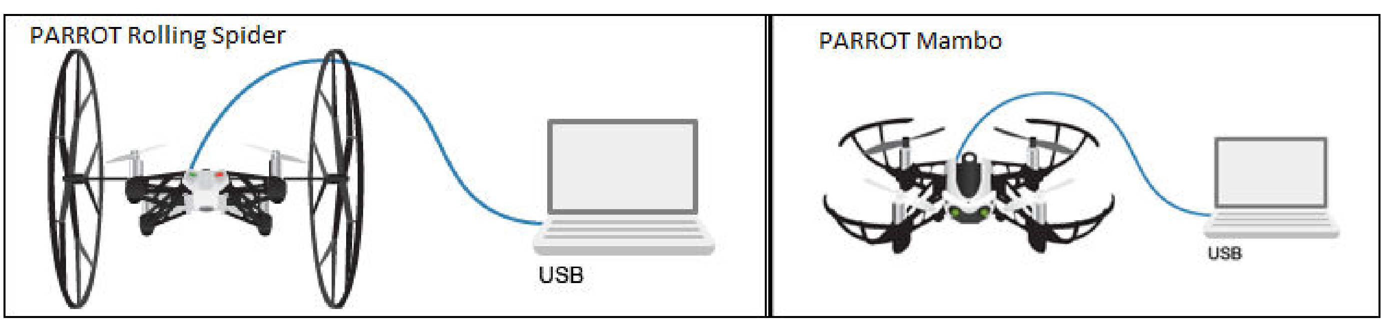
Before you begin:

1 Switch On the Parrot minidrone. The On/Off button is located under the minidrone, near the vertical camera. The LEDs glow or blink to indicate the On or Off status of the Parrot minidrone. The LED indications for Parrot minidrones are:

LED Indications Related to USB Connection and Timeout for Parrot Minidrones

Name of the Parrot Minidrone	LED Indications		
	Drone is Turned ON	**Minidrone is Connected to PC over USB**	**Connection Timeout Occurs**
Parrot Rolling Spider	Both left and right LEDs – Green and stable	Left LED – Green and stable Right LED – Red and stable	Left LED is off
Parrot Mambo	Both left and right LEDs – Green and blinking	Both left and right LEDs – Green and stable	Both LEDs are off

2 Connect the minidrone to a USB port of your computer using a micro-USB Type-B cable, and wait for the LED indications to be stable (as described in the column - Minidrone is connected to PC over USB - in the above table).



Note Ensure that the minidrone is recognized by your computer.

Install the RNDIS Driver

Before you begin, complete the steps mentioned in the “Prerequisites” on page 1-5 section.

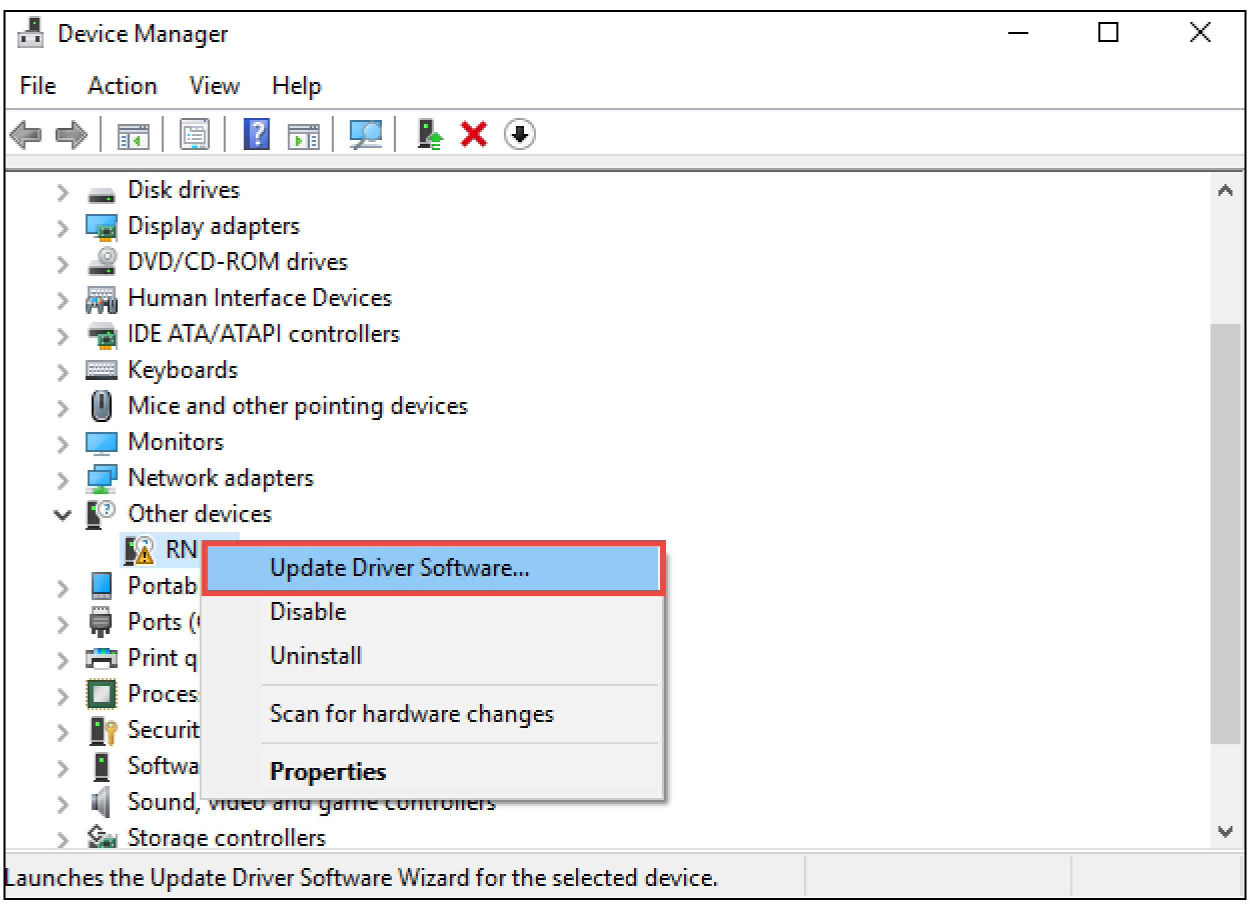
Note During the entire process, ensure that:

1. The Parrot minidrone is switched on and the battery is charged.
2. The LED indications with the USB connected are the same as described in the column - Minidrone is connected to PC over USB - in “Prerequisites” on page 1-5.

   If a timeout happens, the Parrot minidrone disconnects from the host computer. You have to then unplug the micro-USB cable from the minidrone, reconnect it, and repeat the entire process (as described below) again.

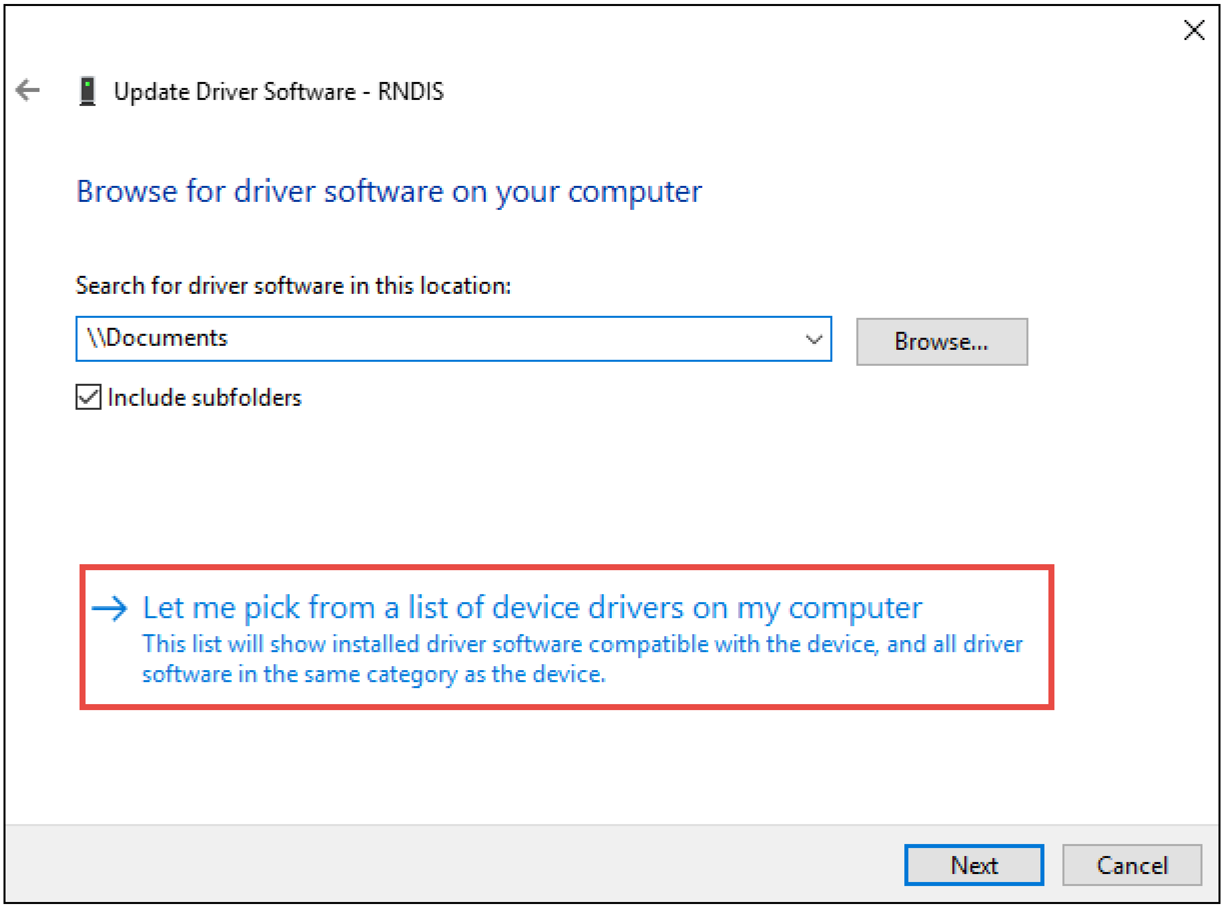
To install the RNDIS driver:

1. Open **Device Manager** and select **Other devices**. From the expanded list, right-click **RNDIS** and then click **Update Driver Software...** from the menu.

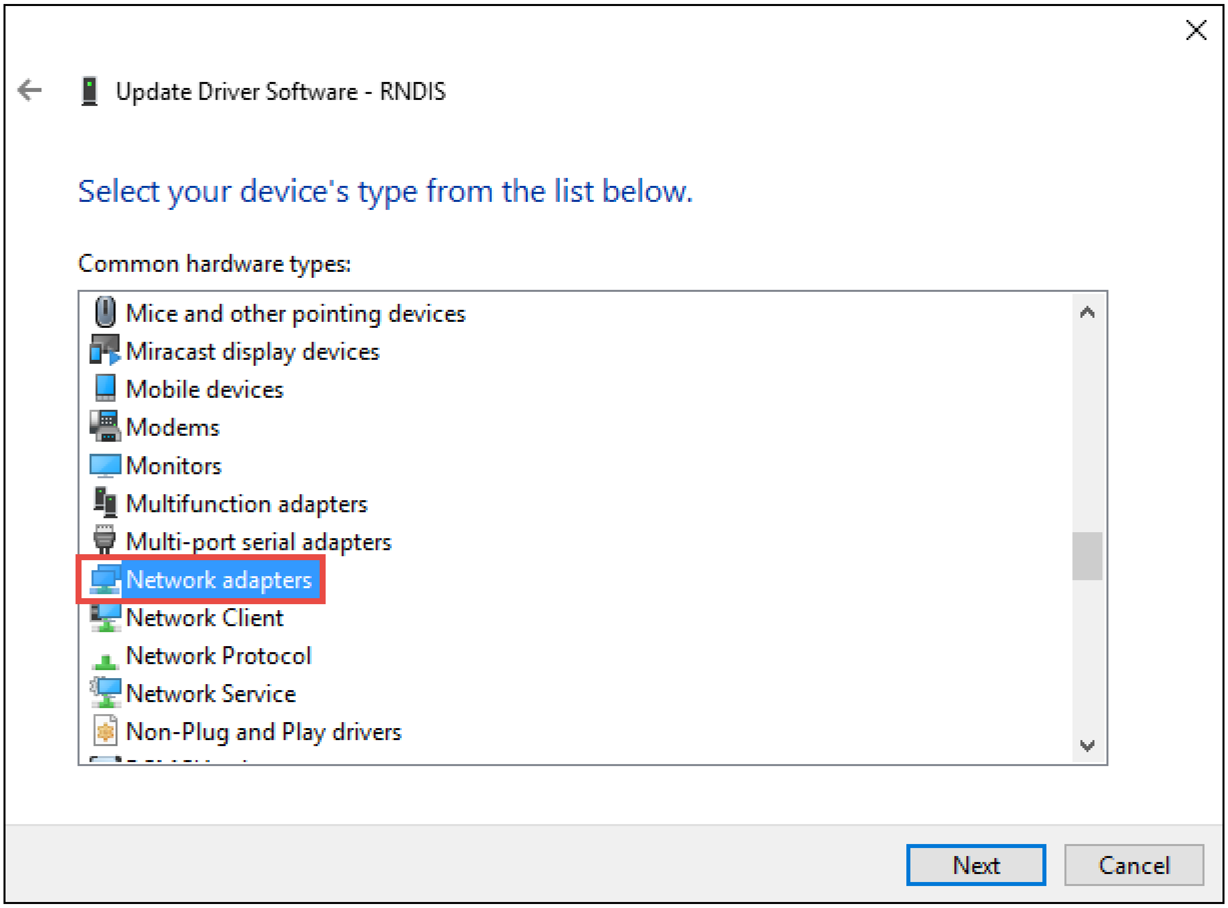


2 When prompted, click **Browse my computer for driver software**.

3 Click **Let me pick from a list of device drivers on my computer**.

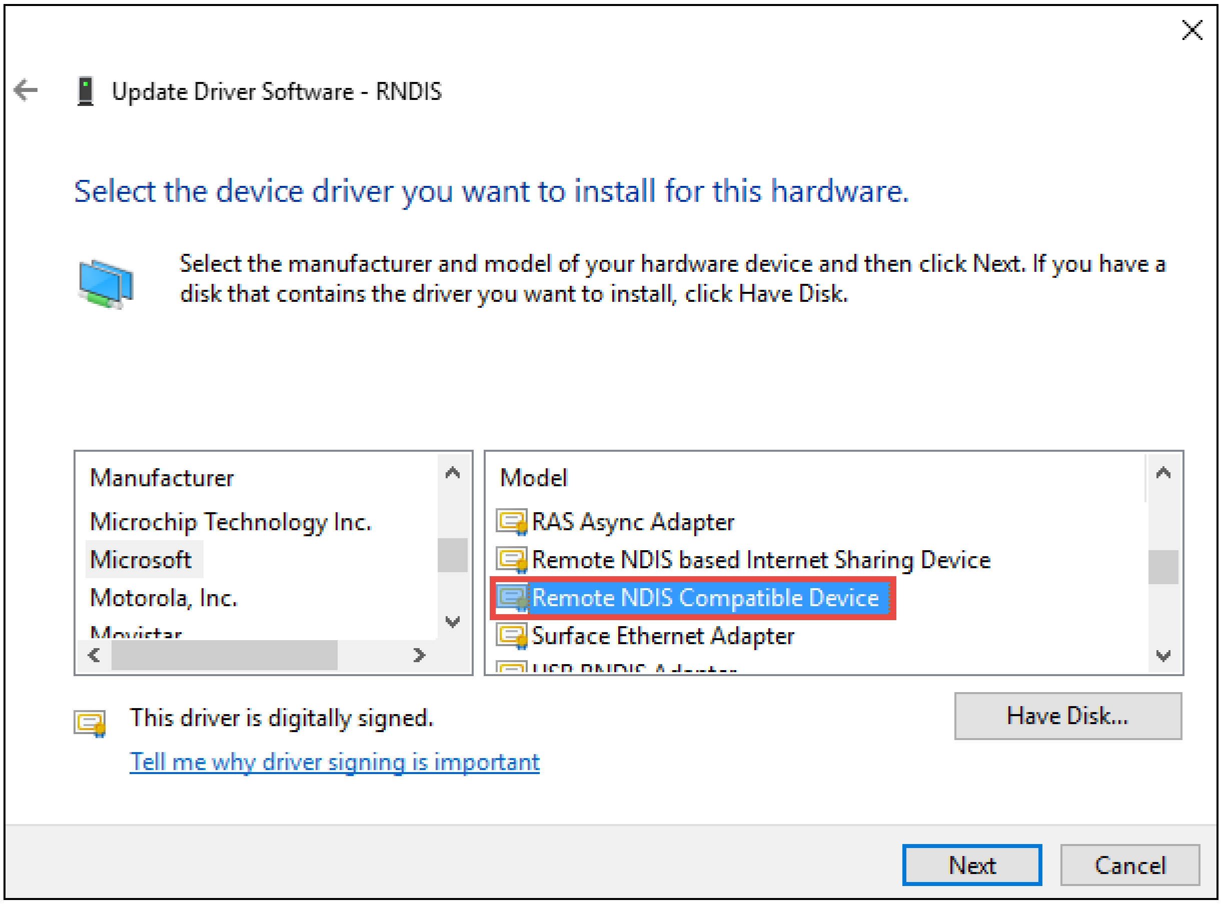


4 From the **Common hardware types** list, select **Network adapters** and click **Next**.

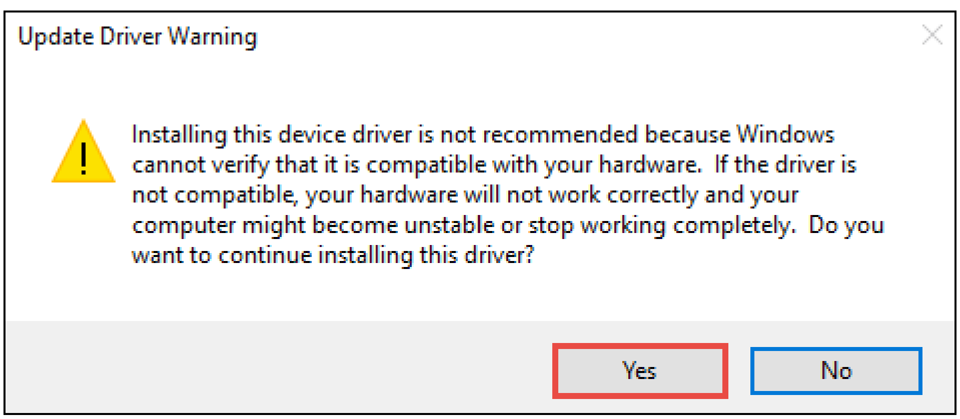


5 Select **Microsoft** in the **Manufacturer** list and **Remote NDIS Compatible Device** in the **Model** list. Click **Next**.

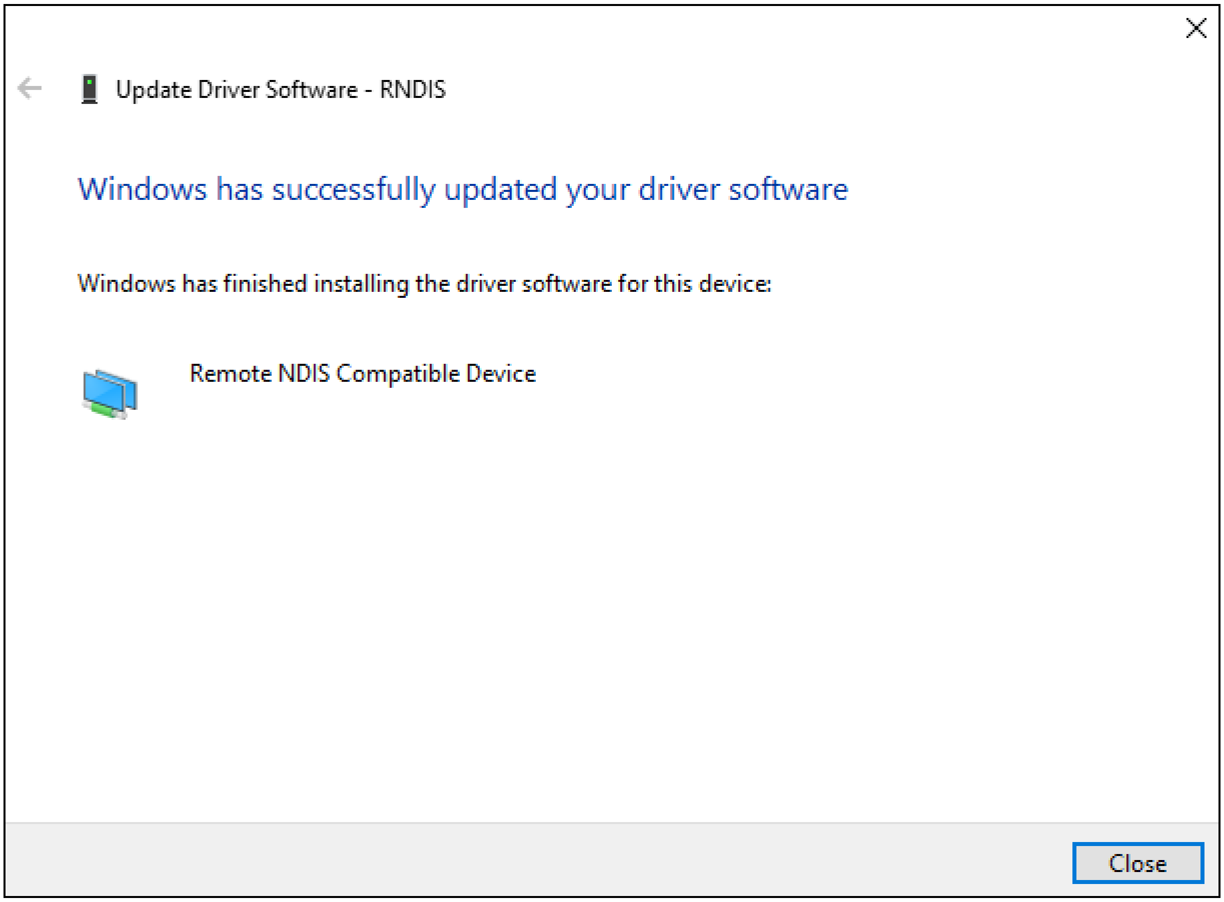
Note If you are using Windows 7, select **Microsoft Corporation** instead of **Microsoft**.



6 In the **Update Driver Warning** dialog box, click **Yes**.

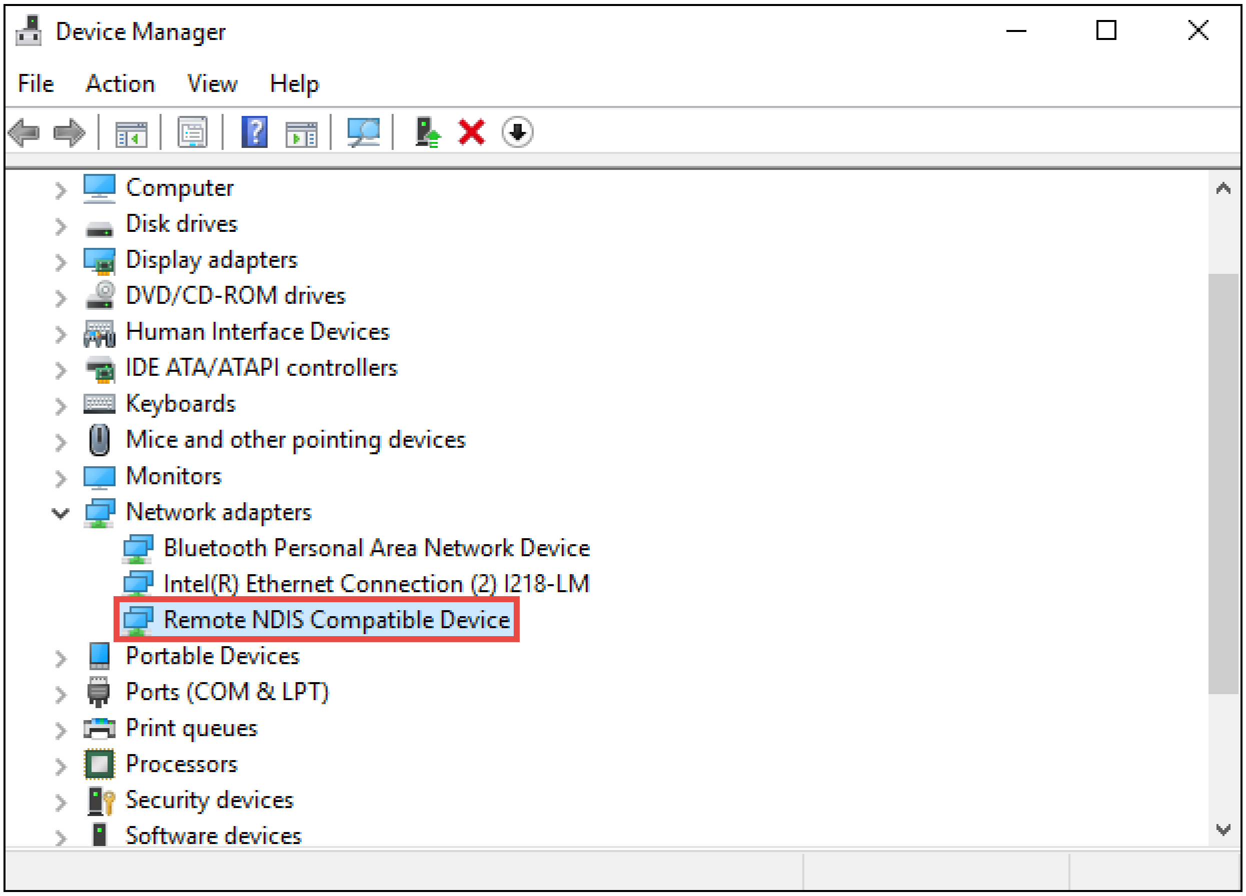


The RNDIS driver is now installed and ready for use.



After a successful installation, the RNDIS device appears on the **Device Manager** screen under **Network adapters**.

Note The RNDIS device name may be different in Windows 7 and Windows 8. For example, the device name appears as **RNDIS** in Windows 7.



Note A timeout can happen during the RNDIS installation or just after the RNDIS installation is completed. If the timeout happens, the RNDIS device does not appear under **Network adapters**. In this case:

- Disconnect the micro-USB cable and reconnect it again.
- If reconnecting the USB cable also does not list the RNDIS device, repeat the entire RNDIS installation process again, before the timeout happens.

See Also

"Connect a Parrot Rolling Spider Minidrone to a Windows System Using Bluetooth" on page 1-12

Connect a Parrot Rolling Spider Minidrone to a Windows System Using Bluetooth

Note In version 17.2.2 or later of Simulink Support Package for Parrot Minidrones, the IP address of Parrot Rolling Spider has been changed from 192.168.1.1 to 192.168.3.5. Therefore, if you have upgraded the Support Package to version 17.2.2 or later, you must perform the "Hardware Setup" on page 1-2 process again for Parrot Rolling Spider.

Note The in-built Bluetooth module of your Windows system is not supported. To connect a Parrot minidrone, you must have an adapter that uses CSR Bluetooth stack. Ensure the following when you purchase a CSR 4.0 or Cinolink Bluetooth 4.0 USB Dongle adapter:

- The package must include the Driver Installation CD for the adapter
- The adapter must also support Personal Area Networking (GN) (for connection to Parrot Rolling Spider) or Personal Area Networking (NAP) (for connection to Parrot Mambo)

After connecting the adapter, you need to install the Bluetooth driver for the Bluetooth adapter. Ensure that you install the Bluetooth driver using the Driver Installation CD. The CSR 4.0 dongle fails to communicate with the Parrot minidrone if the Bluetooth driver is not installed on the Windows system.

The configuration instructions and screen shots included are for Cinolink Bluetooth 4.0 USB adapter on Windows 10. Similar configuration steps are applicable for Bluetooth adapters from other manufacturers on Windows 7 and Windows 8.

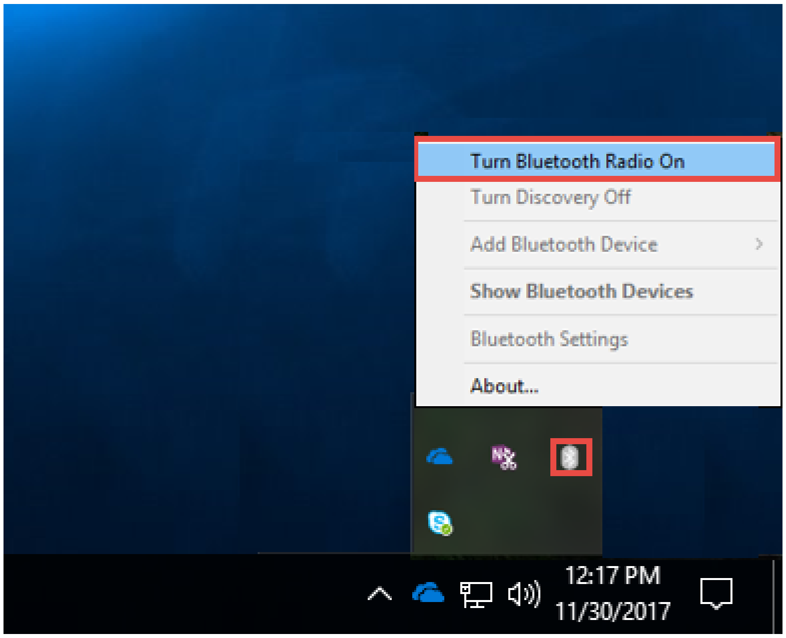
To connect the Parrot minidrone to your Microsoft® Windows computer using Bluetooth, follow these instructions:

Prerequisites

Before you begin:

1. Switch On the Parrot Rolling Spider. The On/Off button is located under the minidrone, near vertical camera. Wait for both the LEDs to be green and stable.
2. Insert your Bluetooth 4.0 adapter into a USB port on your computer. To find out if your adapter is Bluetooth Low Energy 4.0 compatible, see "Finding the LMP Version of Bluetooth Adapter on Your Windows System" on page 1-58.
3. Install the correct Bluetooth driver for your adapter. After installing the driver, restart the computer.
4. Turn on the Bluetooth support on your computer. To turn on the Bluetooth support, perform these steps:

   a. In the notification area on your taskbar, right-click the **Bluetooth** icon.
   b. Select **Turn Bluetooth Radio On** from the menu.



Add a Parrot Rolling Spider Minidrone to a Windows System

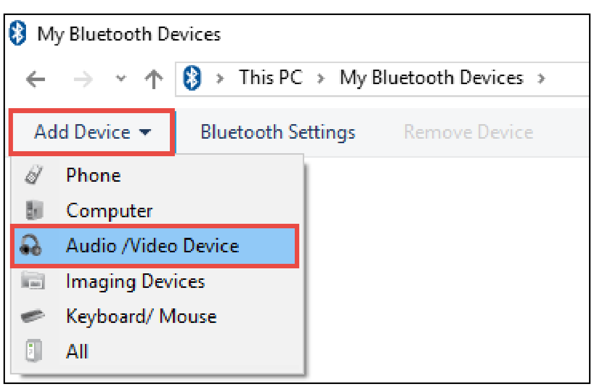
Before you begin, complete the steps mentioned in the "Prerequisites" on page 1-12 section.

Note During the entire process of connecting the minidrone to your computer, ensure that:

1. The Parrot Rolling Spider minidrone is switched on and the battery is charged.
2. Both the LEDs on the minidrone are green and stable.

To add a Parrot Rolling Spider minidrone over Bluetooth:

1. Open **Windows Explorer** and click **This PC/ My Computer**.
2. In the **Devices and drives/ Other Devices** section, double-click **My Bluetooth Devices**.
3. In the top-left corner, click **Add Device** and select **Audio /Video Device** from the menu.



Your computer searches for all the Bluetooth-enabled audio/video devices in its range. Wait a few seconds for the Parrot Rolling Spider minidrone to appear in the list.

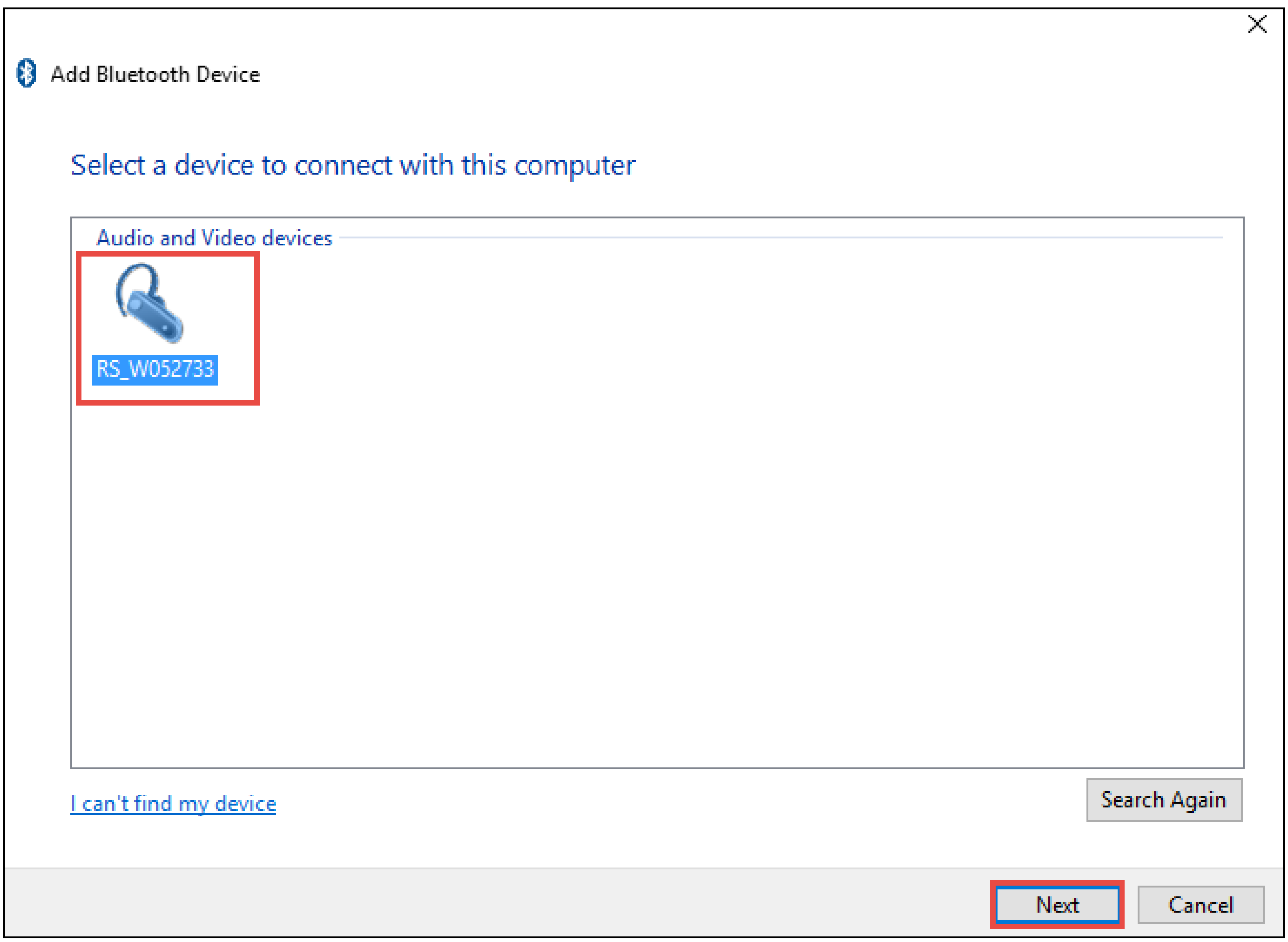
Note The minidrone must appear within a few seconds after you click **Add Device > Audio / Video Device**. If not, click the **Search Again** button to scan for the Bluetooth-enabled devices again.

The name of the minidrone appears as one of these:

- `RS_YXXXXXX`, where *Y* is the first alphabet of the color of the minidrone and each *X* is a number. For example, if the minidrone is white, the name appears as `RS_WXXXXXX`.
- the media access control (MAC) address of the minidrone (for example, a0:14:3d:4b:10:5f)

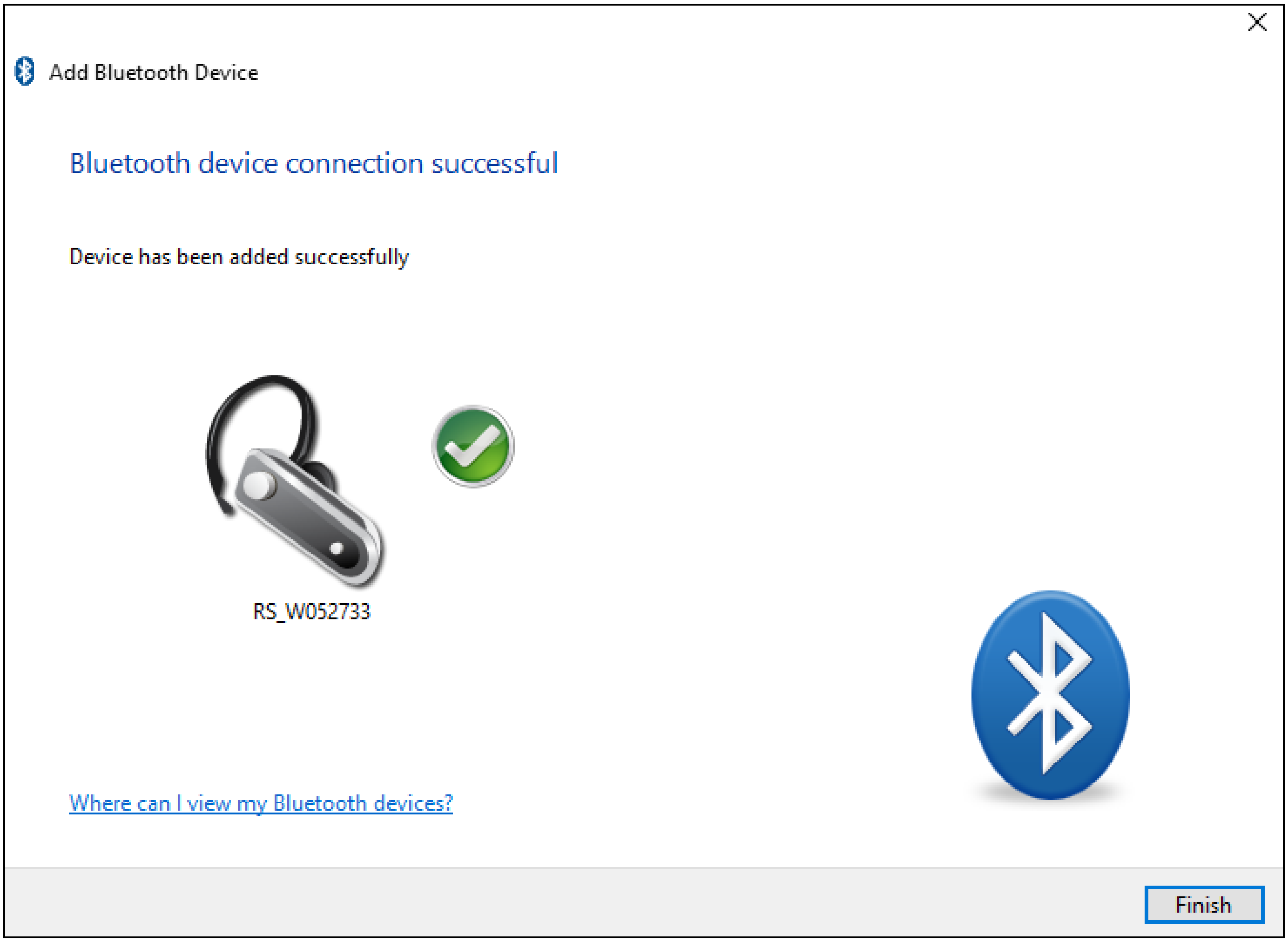
4 From the devices listed, select the minidrone that you want to connect to your computer and click **Next**.

If your minidrone is not available in the list of devices, see "Minidrone Not Found in Bluetooth Scan Result" on page 1-50.

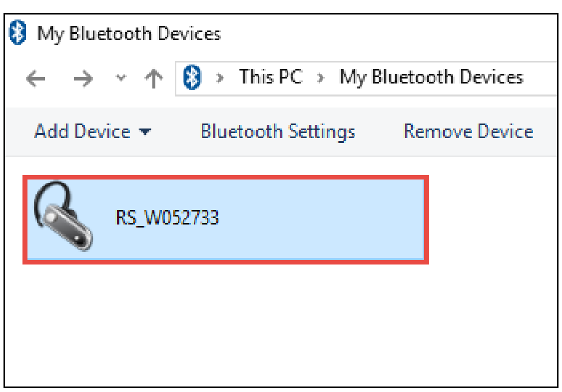


5 The minidrone is now connected and ready for use.

If the connection is unsuccessful, see “Bluetooth Connection Fails” on page 1-54.



6 After a successful connection, the minidrone appears in the **This PC/ My Computer** > **My Bluetooth Devices**.



Set Up a Bluetooth PAN Connection for Parrot Rolling Spider

Before you begin:

1. Complete the steps in the "Prerequisites" on page 1-12 section.
2. Make sure that the minidrone is connected to your computer, as described in the "Add a Parrot Rolling Spider Minidrone to a Windows System" on page 1-13 section.

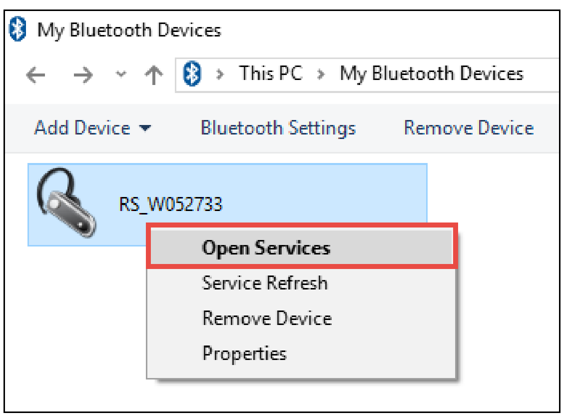
Note During the entire process, ensure that:

1. The Parrot Rolling Spider minidrone is switched on and the battery is charged.
2. Both the LEDs on the minidrone are green and stable.

To set up a PAN connection:

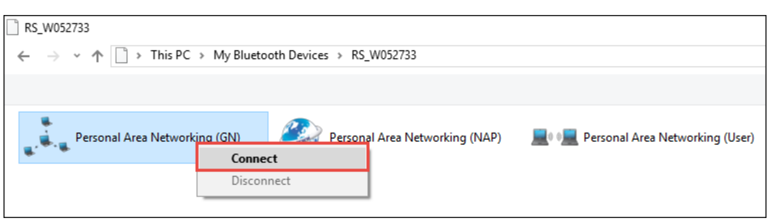
1. Browse to **This PC/ My Computer > My Bluetooth Devices**.
2. Right-click the connected Parrot Rolling Spider minidrone, and select **Open Services** from the menu.

   If the **Open Services** option is not available, ensure that a correct Bluetooth Personal Area Network driver is installed on your computer.

3. Right-click **Personal Area Networking (GN)** and select **Connect**.

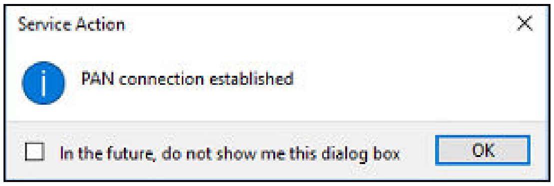
   If the **Personal Area Networking (GN)** option is not available, ensure that a relevant Bluetooth Personal Area Network driver is installed on your computer.

4. As the computer attempts to establish the PAN connection, it prompts you twice to continue pairing. Click the prompt every time it appears. Otherwise, the Bluetooth connection may fail.

5 A 'PAN connection established' message appears after a successful connection.

If the PAN connection is unsuccessful, see “Personal Area Networking (PAN) Connection Fails” on page 1-56.



See Also

“Install RNDIS for Parrot Minidrones” on page 1-5 | “Troubleshooting Connection Settings in Windows” on page 1-59

Connect a Parrot Mambo Minidrone to a Windows System Using Bluetooth

Note The in-built Bluetooth module of your Windows system is not supported. To connect a Parrot minidrone, you must have an adapter that uses CSR Bluetooth stack. Ensure the following when you purchase a CSR 4.0 or Cinolink Bluetooth 4.0 USB Dongle adapter:

- The package must include the Driver Installation CD for the adapter
- The adapter must also support Personal Area Networking (GN) (for connection to Parrot Rolling Spider) or Personal Area Networking (NAP) (for connection to Parrot Mambo)

After connecting the adapter, you need to install the Bluetooth driver for the Bluetooth adapter. Ensure that you install the Bluetooth driver using the Driver Installation CD. The CSR 4.0 dongle fails to communicate with the Parrot minidrone if the Bluetooth driver is not installed on the Windows system.

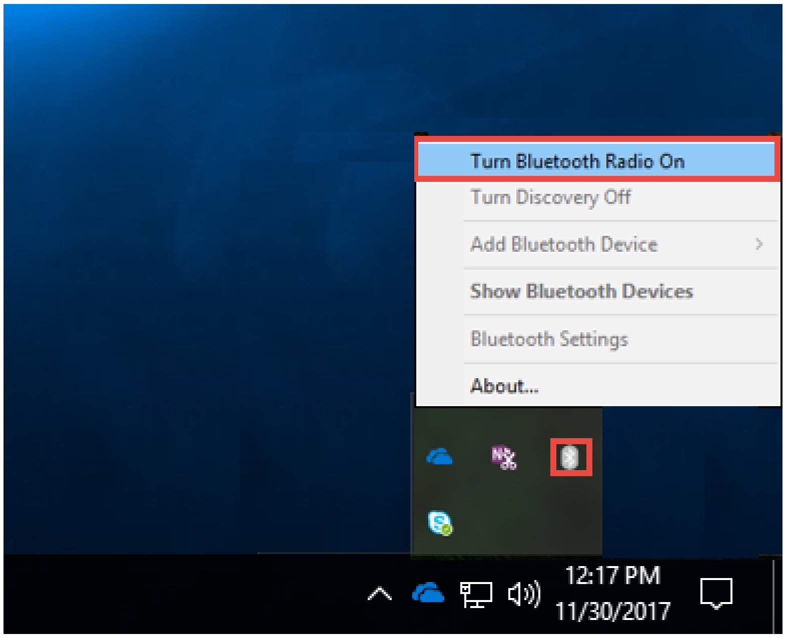
The configuration instructions and screen shots included are for Cinolink Bluetooth 4.0 USB adapter on Windows 10. Similar configuration steps are applicable for Bluetooth adapters from other manufacturers on Windows 7 and Windows 8.

To connect the Parrot Mambo minidrone to your Microsoft Windows computer using Bluetooth, follow these instructions.

Prerequisites

1. Switch on the Parrot Mambo. The On/Off button is located under the minidrone, near the vertical camera. Wait until both LEDs are green and blinking before proceeding to the next step.
2. Insert your Bluetooth 4.0 adapter into a USB port on your computer. To find out if your adapter is Bluetooth Low Energy 4.0 compatible, see “Finding the LMP Version of Bluetooth Adapter on Your Windows System” on page 1-58.
3. Install the correct Bluetooth driver for your adapter. After installing the driver, restart the computer.
4. Turn on the Bluetooth support on your computer. To turn on the Bluetooth support, perform these steps:

   a. In the notification area on your taskbar, right-click the **Bluetooth** icon.
   b. Select **Turn Bluetooth Radio On** from the menu.



Add a Parrot Mambo Minidrone to a Windows System

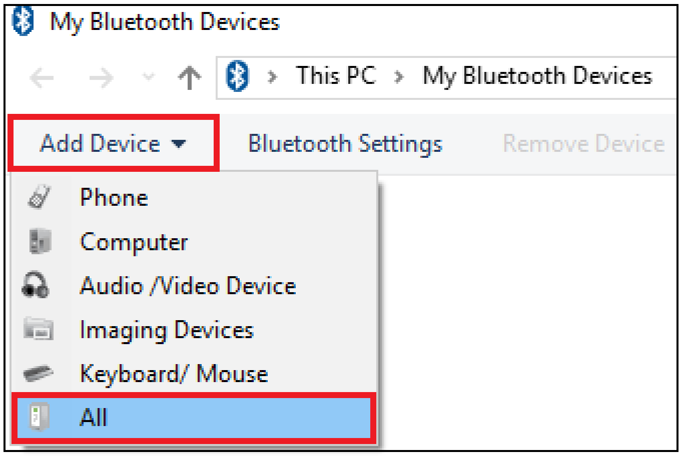
Before you begin, complete the steps mentioned in the “Prerequisites” on page 1-12 section.

Note During the entire process of connecting the minidrone to your computer, ensure that:

1 The Parrot Mambo minidrone is switched on and the battery is charged.
2 Both the LEDs on the minidrone are green and blinking.

To add a minidrone:

1 Open **Windows Explorer** and select **This PC/ My Computer**.
2 In the **Devices and drives/ Other Devices** section, double-click **My Bluetooth Devices**.
3 In the top-left corner, click **Add Device** and select **All** from the menu.

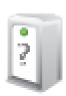


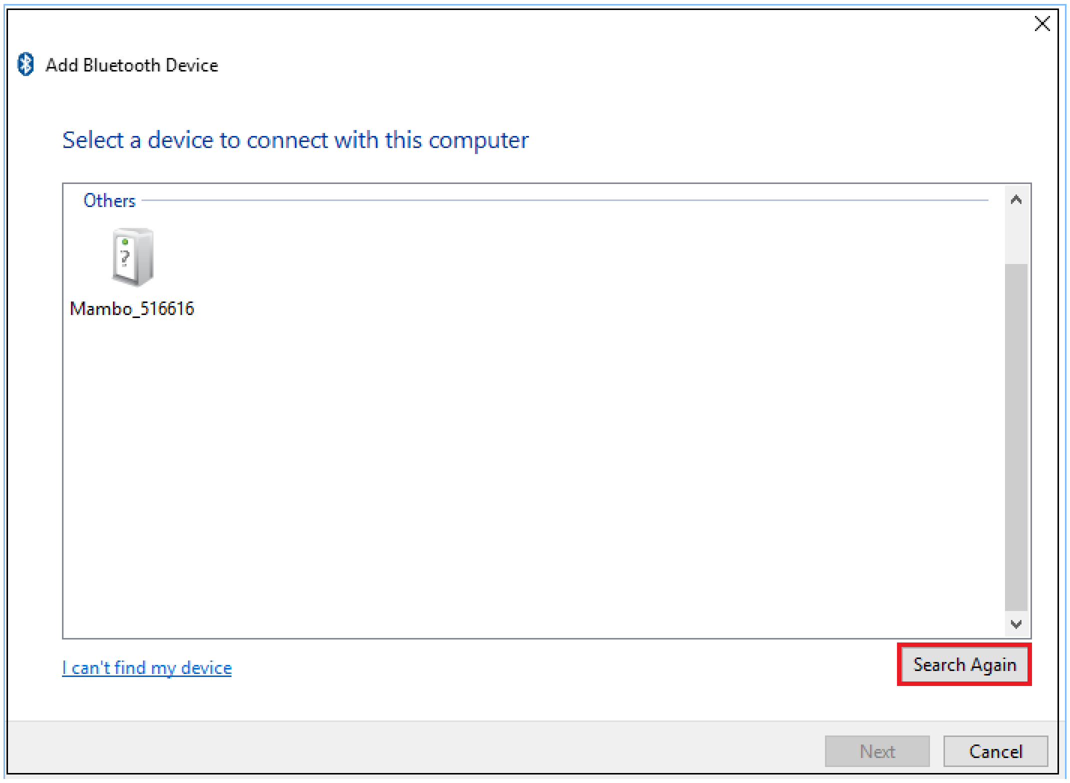
Your computer searches for all the Bluetooth-enabled devices in its range. Wait a few seconds for the Parrot Mambo minidrone to appear in the list under the **Others** category.

The name of the minidrone appears as one of these:

- `Mambo_XXXXXX`, where each *X* is a number.
- the media access control (MAC) address of the minidrone (for example, a0:14:3d:3e:20:6f).

You need to identify the Parrot Mambo minidrone that appears in the list with the correct icon,  . If a device with this icon does not appear after a few seconds, wait for the scan to be completed, and then click **Search Again** to scan for the Bluetooth-enabled devices again.

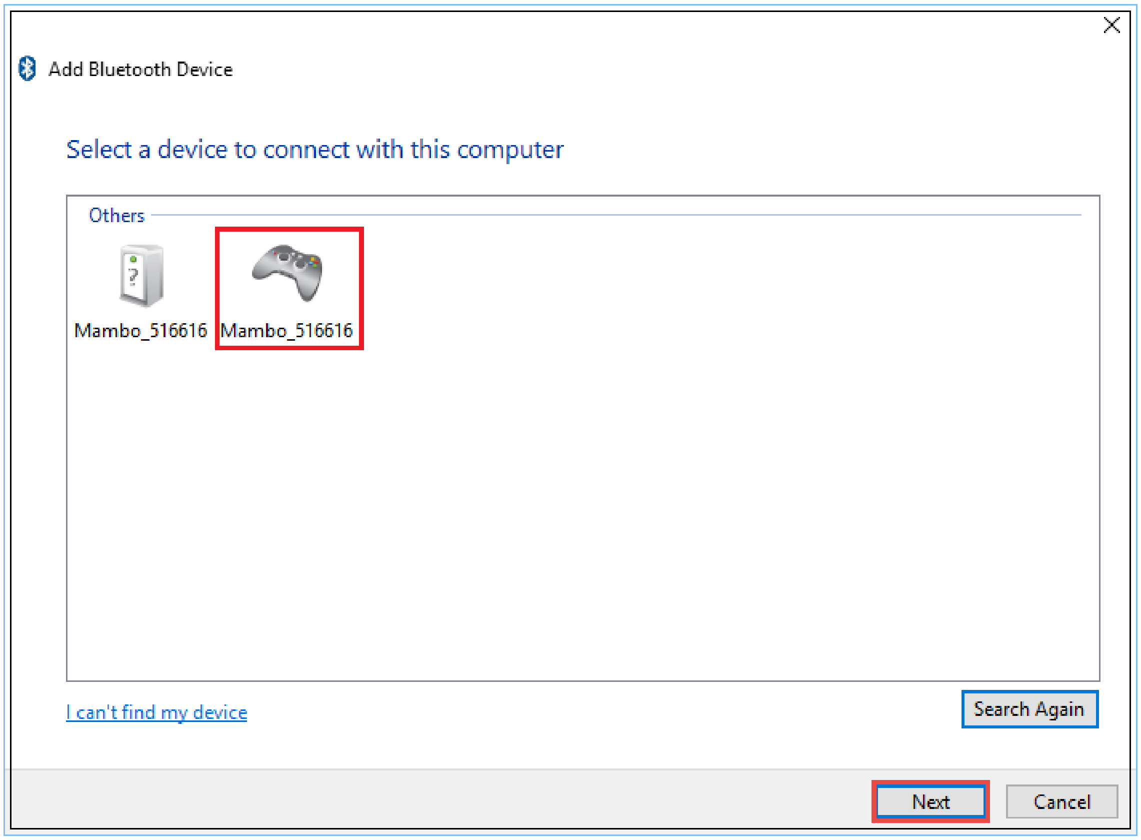
Another icon  with the title `Mambo_XXXXXX` may appear in the scan result. DO NOT click on this icon.



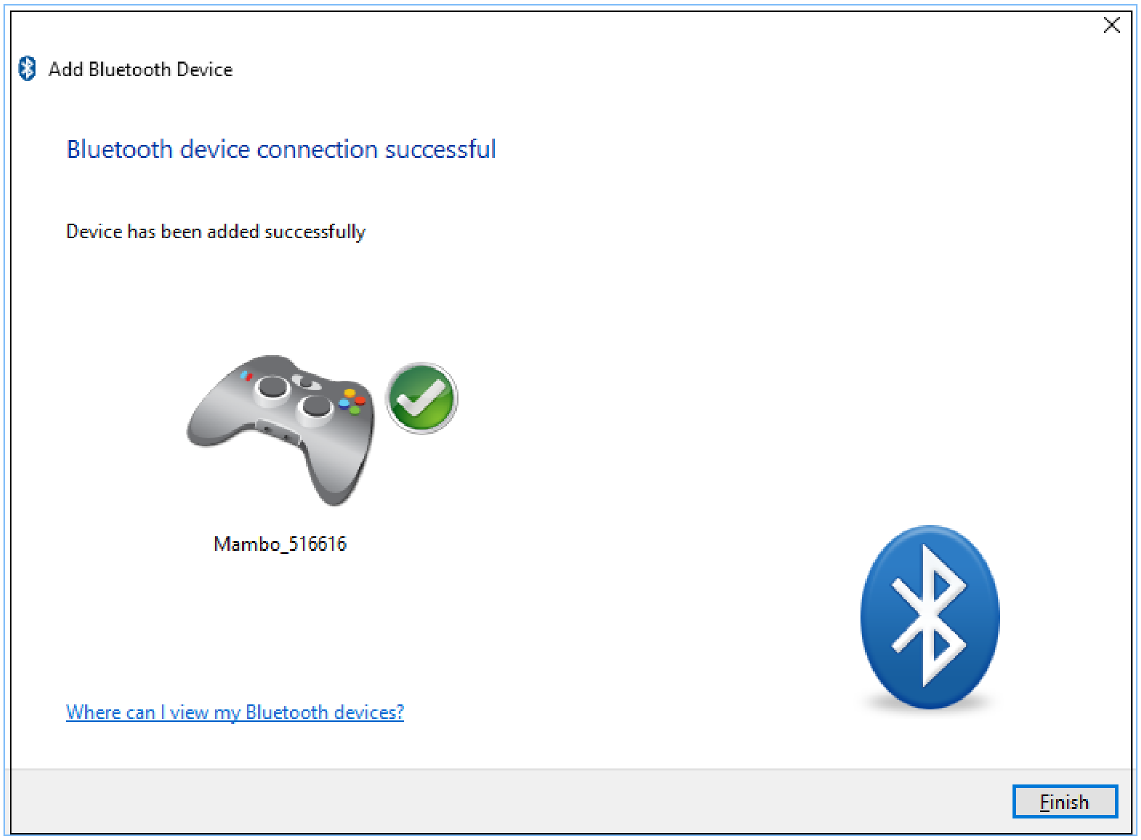
Click **Search Again** until the correct icon  appears. Click **Search Again** multiple times for the correct icon to appear.

If your minidrone is still not available in the list of devices, see “Minidrone Not Found in Bluetooth Scan Result” on page 1-50.

4 Select the Parrot Mambo minidrone (which appears with the correct icon), and click **Next**.

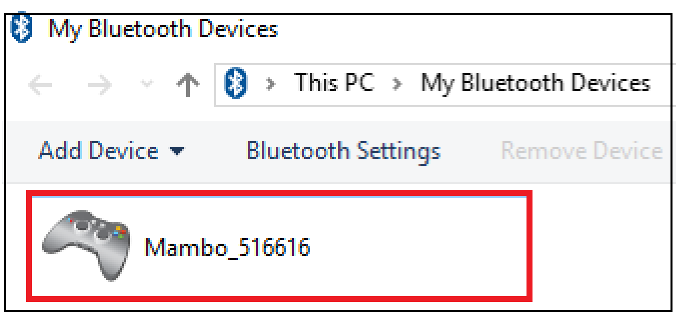


5 The minidrone is now connected and ready for use. Click **Finish**.



If the connection is unsuccessful, see “Bluetooth Connection Fails” on page 1-54.

6 After a successful connection, the Parrot Mambo minidrone appears in the **This PC/ My Computer > My Bluetooth Devices**.



Set Up a Bluetooth NAP Connection for Parrot Mambo

Before you begin:

1 Complete the steps in the “Prerequisites” on page 1-12 section.

2 Make sure that the minidrone is connected to your computer, as described in the “Add a Parrot Mambo Minidrone to a Windows System” on page 1-19 section, and the minidrone appears in the **This PC/ My Computer > My Bluetooth Devices**.

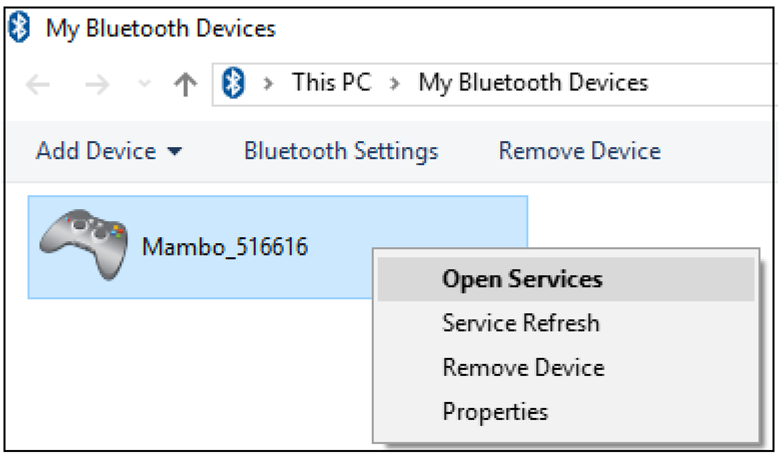
Note During the entire process, ensure that:

1 The Parrot Mambo minidrone is switched on and the battery is charged.
2 Both the LEDs on the minidrone are green and blinking.

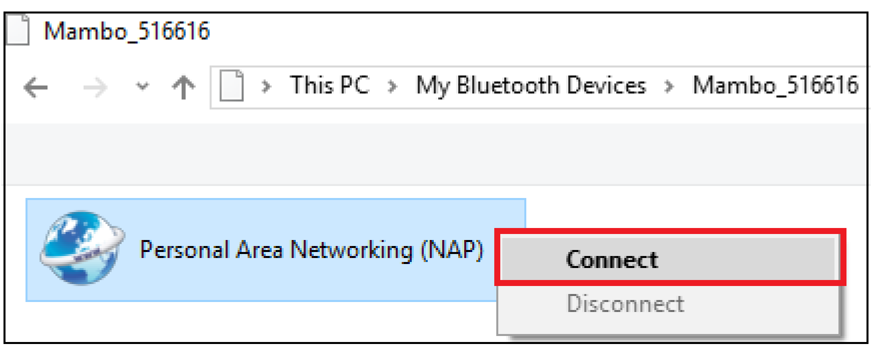
To set up a NAP connection:

1 Browse to **This PC/ My Computer > My Bluetooth Devices**.
2 Right-click the connected Parrot Mambo minidrone, and select **Open Services** from the menu.

   If the **Open Services** option is not available, ensure that a correct Bluetooth Personal Area Network driver is installed on your computer.

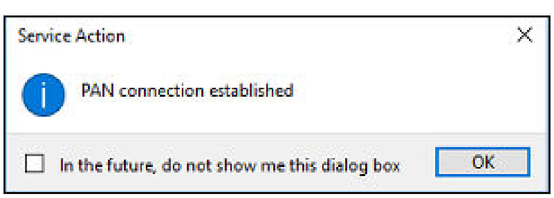


3 Right-click **Personal Area Networking (NAP)** and select **Connect**.



4 A "PAN connection established" message appears after a successful connection.

   If the NAP connection is unsuccessful, see “Personal Area Networking (PAN) Connection Fails” on page 1-56.



See Also

Connect a Parrot Rolling Spider Minidrone to a Mac Using Bluetooth

Note In version 17.2.2 of Simulink Support Package for Parrot Minidrones, the IP address of Parrot Rolling Spider has been changed from 192.168.1.1 to 192.168.3.5. Therefore, if you have upgraded the Support Package to version 17.2.2 or later, you must perform the "Hardware Setup" on page 1-2 process again for Parrot Rolling Spider.

To connect the Parrot Rolling Spider minidrone to your Mac using Bluetooth, follow these instructions:

Prerequisites

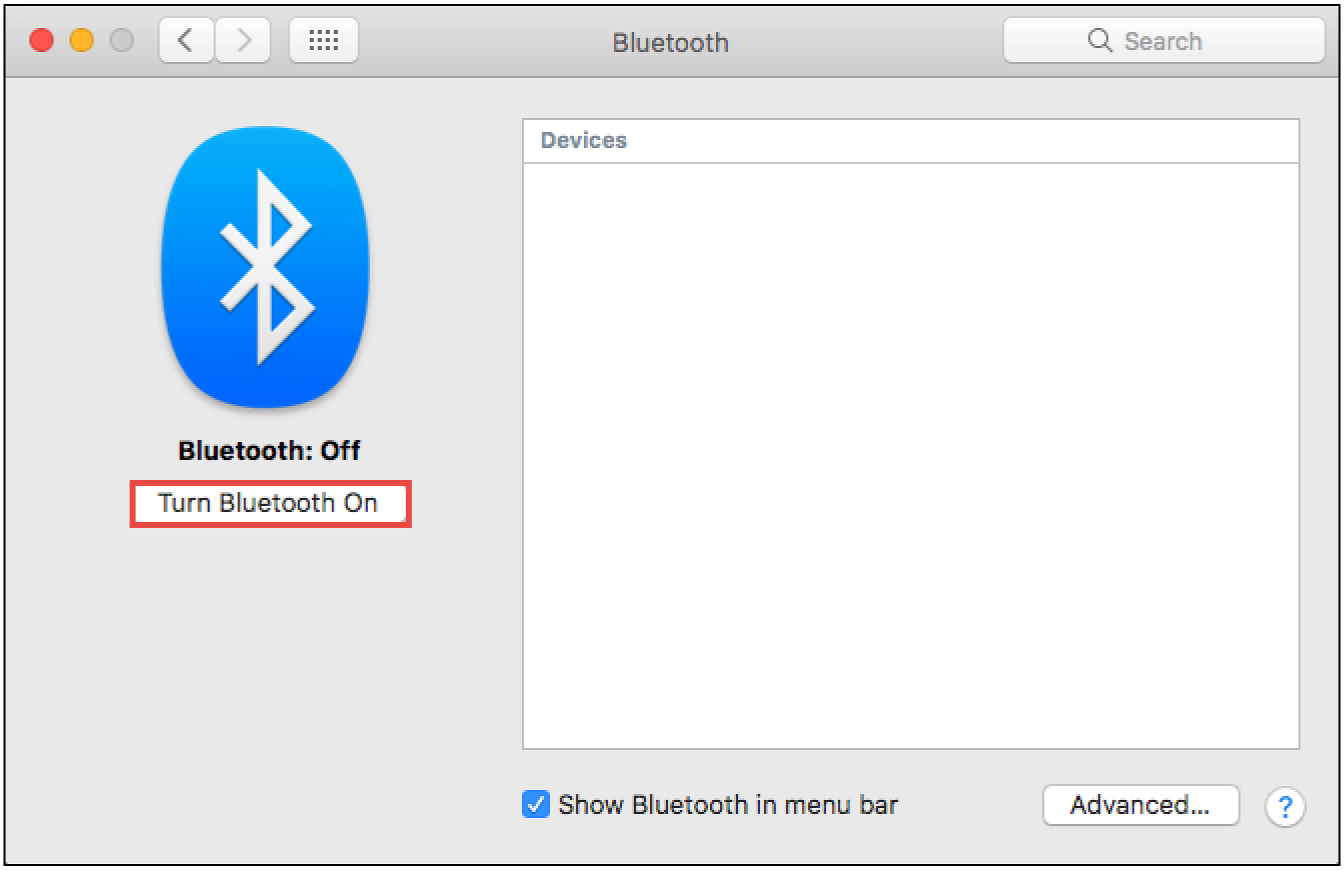
Before you begin:

1. Turn on the Parrot Rolling Spider minidrone. The On/Off button is located under the minidrone, near the vertical camera.

   Wait until both the LEDs turn green and are stable.

2. Ensure that your Mac has a Bluetooth 4.0 adapter. To check the Bluetooth version on your Mac, see "Finding the LMP Version of Bluetooth Adapter on Your Mac System" on page 1-65.
3. Turn on the Bluetooth support on your Mac. To turn on the Bluetooth support:
   a. Browse to **System Preferences** > **Bluetooth**.
   b. Click the **Turn Bluetooth On** button.

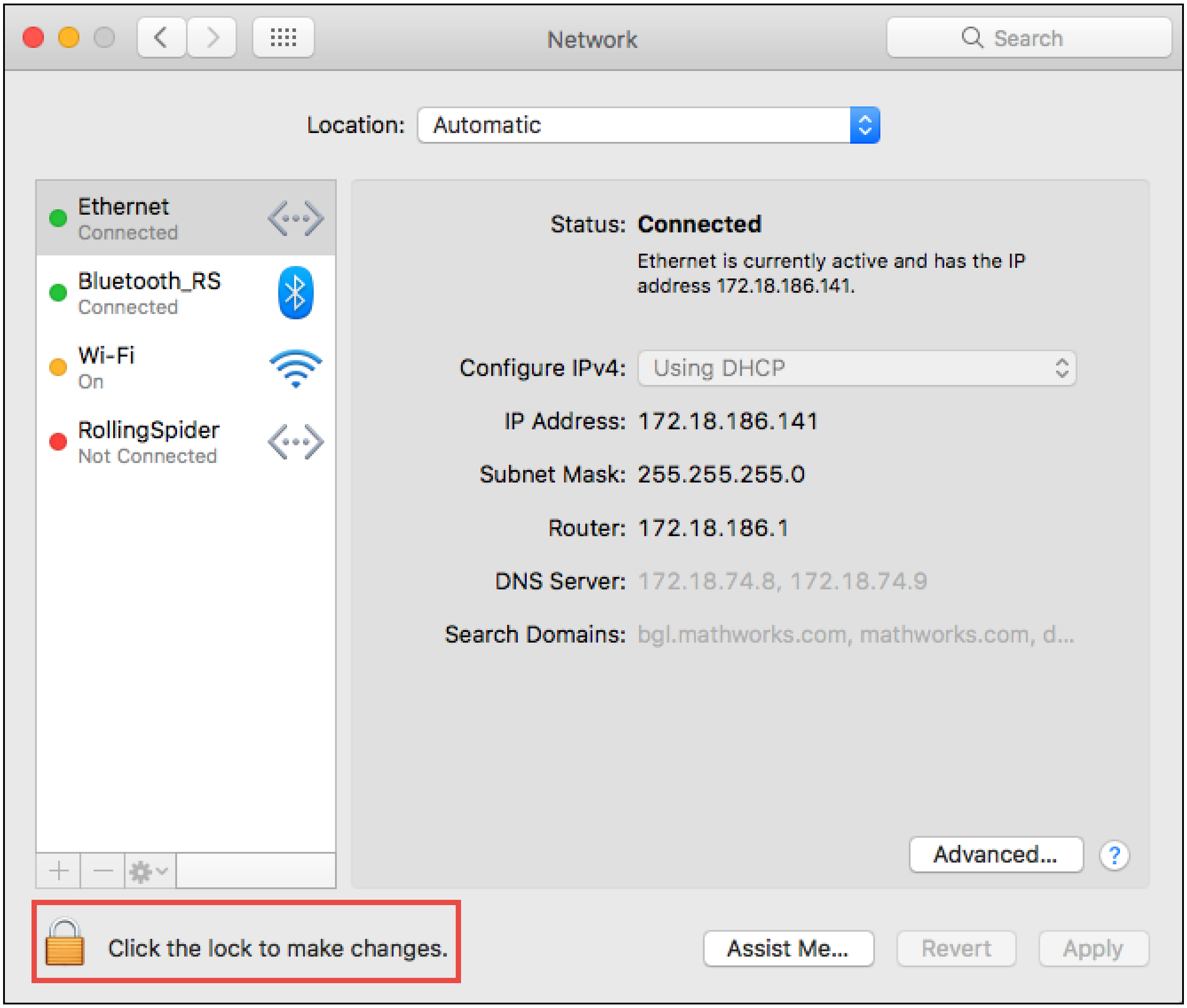


Unlock the Network Preferences

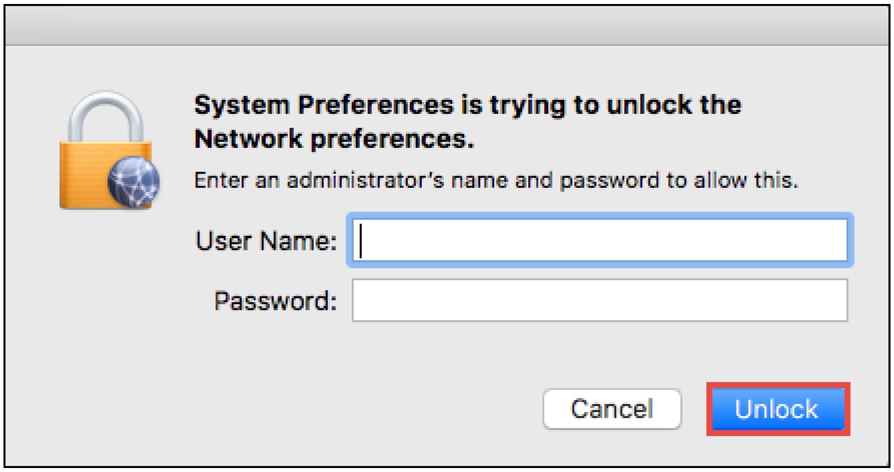
To edit the network preferences, you must have administrative privileges. If you are already logged in as an administrator, you can skip the steps in this section.

1 In **System Preferences**, browse to **Network**.

2 Click the lock symbol to edit the network preferences.



3 Enter the administrator credentials and click **Unlock**.



Pair the Minidrone with Your Mac

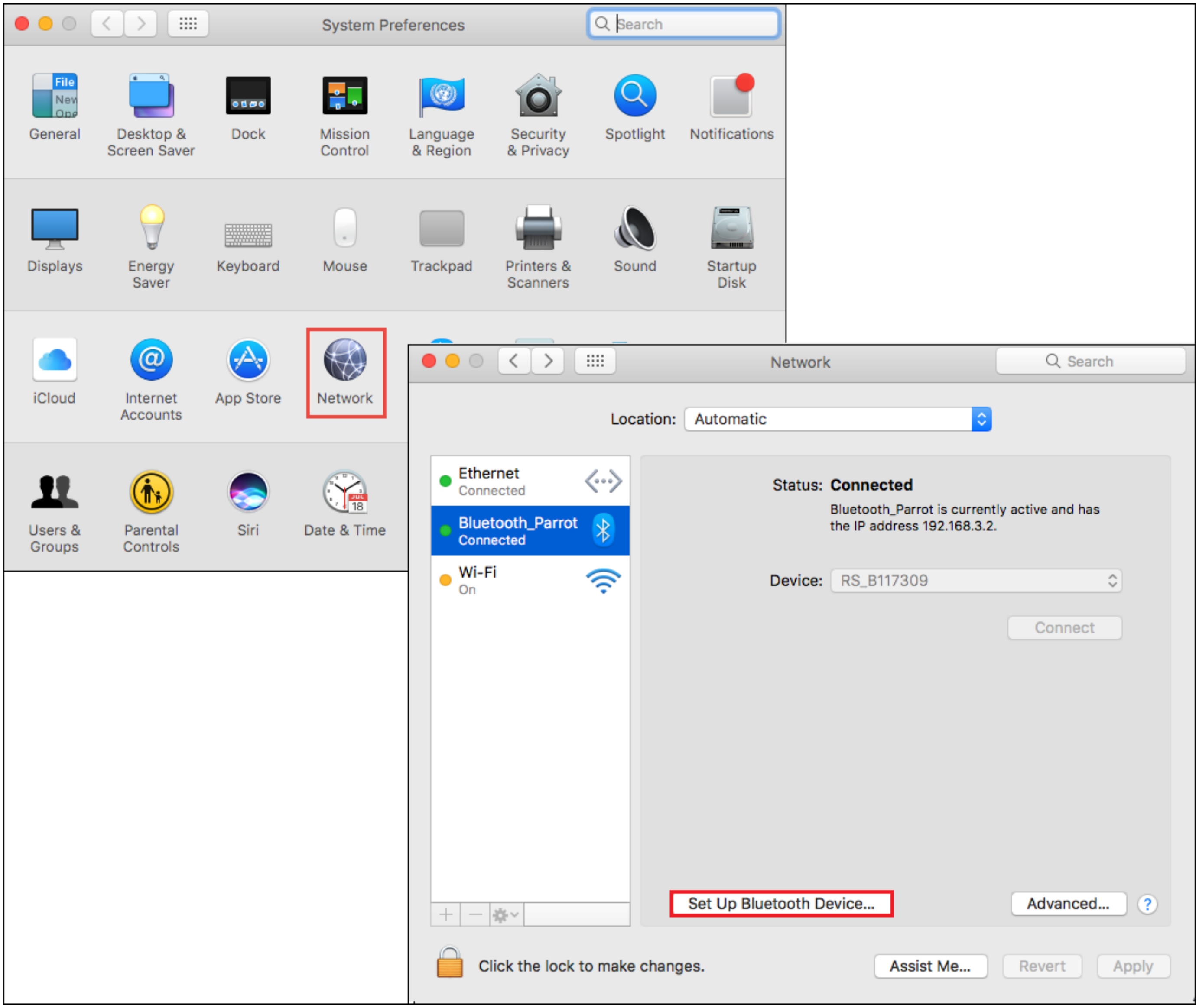
Before you begin, complete the steps mentioned in the "Prerequisites" on page 1-26 section.

Note During the entire process, ensure that:

1. The Parrot minidrone is switched on.
2. Both the LEDs on the minidrone are green.

To pair the minidrone:

1. Browse to **System Preferences** > **Network** > **Bluetooth_RS** and click the **Set Up Bluetooth Device...** button. Your computer searches for all the Bluetooth-enabled devices in its range. Wait a few seconds for the minidrone to appear.

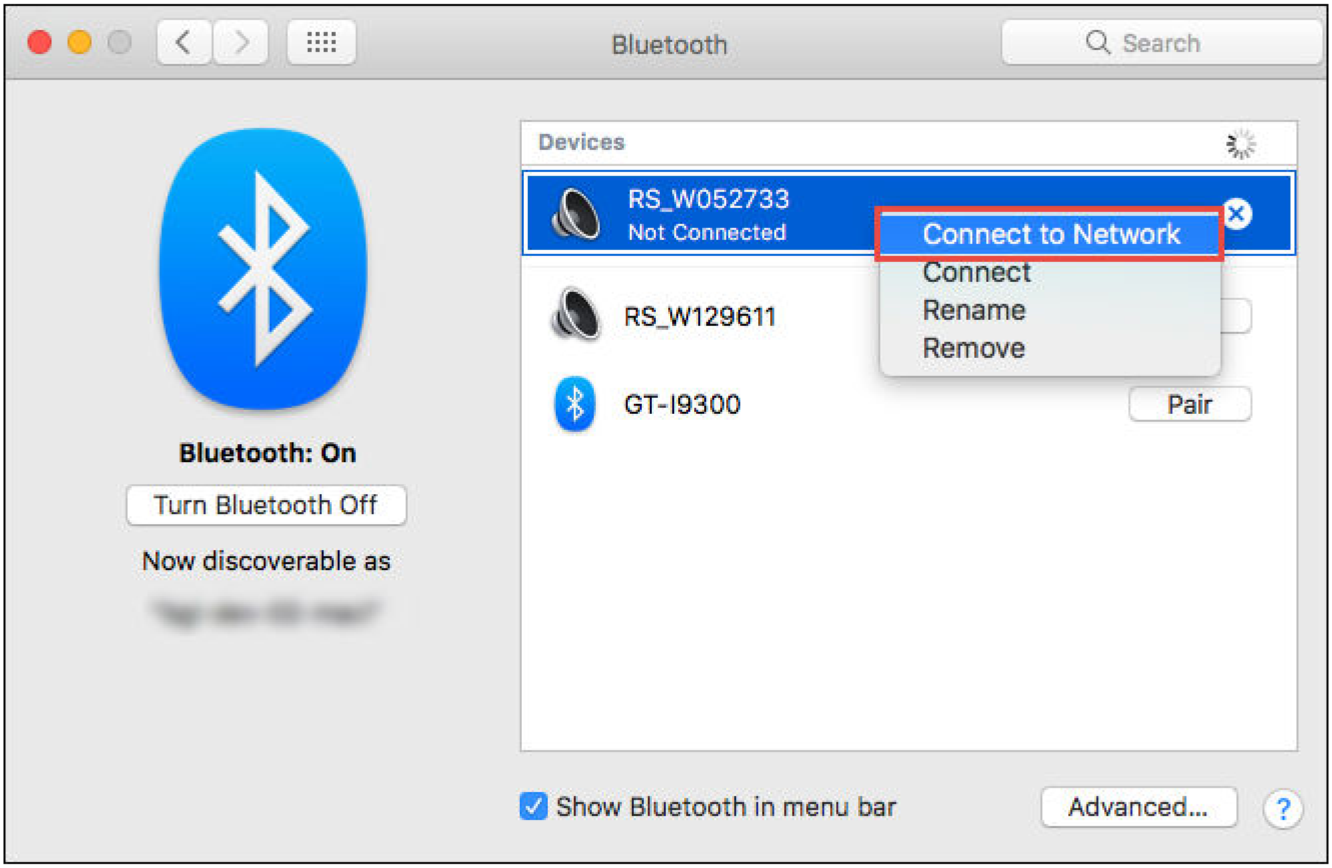


2 From the devices listed, right-click the minidrone that you want to pair with your computer and select **Connect to Network**.

The name of the minidrone appears as any one of these:

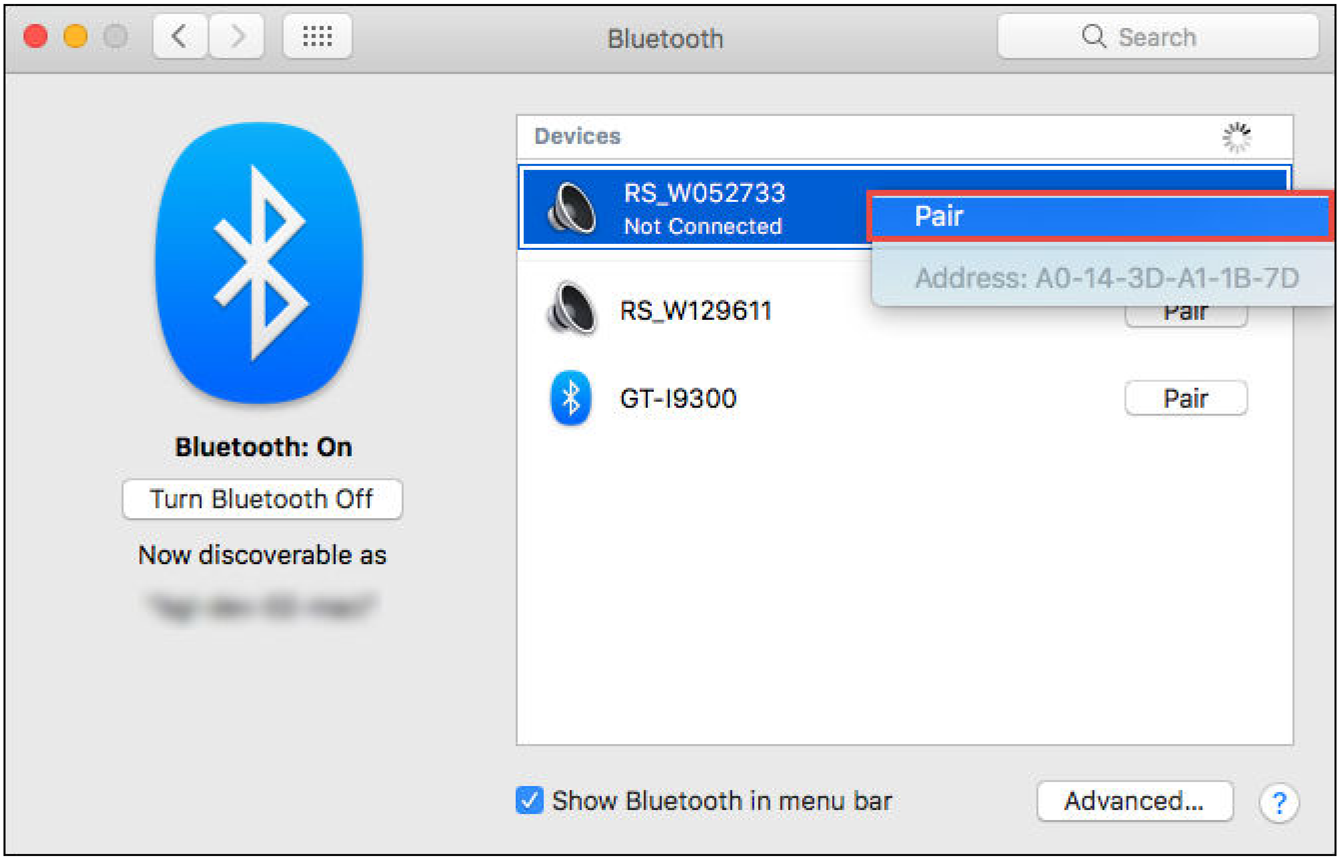
- RS_YXXXXXX, where *Y* is the first alphabet of the color of the minidrone and each *X* is a number. For example, if the minidrone is white, the name appears as RS_WXXXXXX.
- the media access control (MAC) address of the minidrone (for example, a0:14:3d:4b:10:5f)

If your minidrone is not available in the list of devices, see “Minidrone Not Found in Bluetooth Scan Result” on page 1-60.



3 Right-click the minidrone again and select **Pair** from the menu. The **Pair** option might not appear immediately.

When you right-click the minidrone, the **Not Connected** status below the minidrone changes to **Connected**. The minidrone is now paired with your computer.



Connect the Paired Minidrone to Your Mac

Before you begin, complete the steps mentioned in the “Prerequisites” on page 1-26 section.

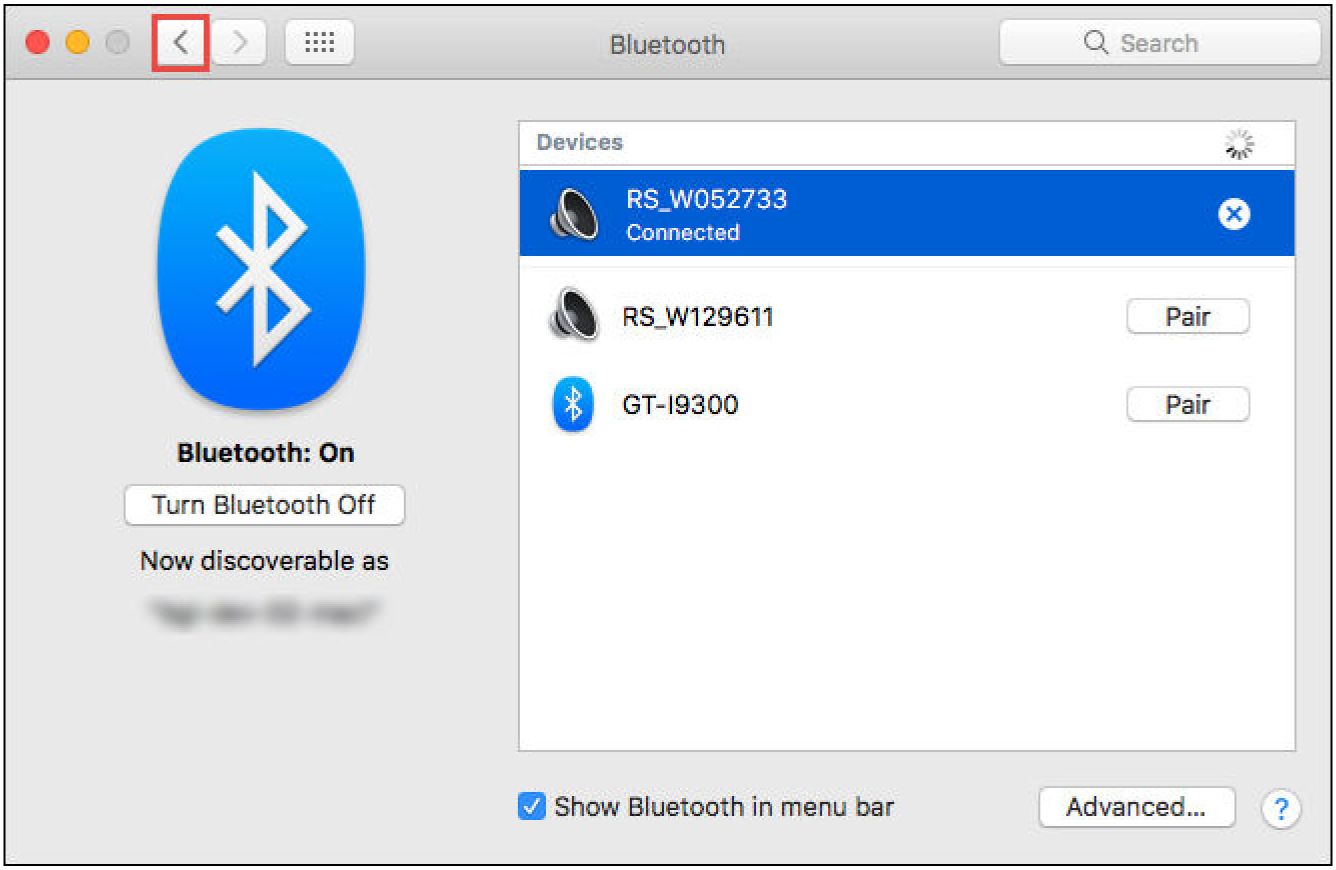
Note During the entire process, ensure that:

1. The Parrot minidrone is switched on.
2. Both the LEDs on the minidrone are green.

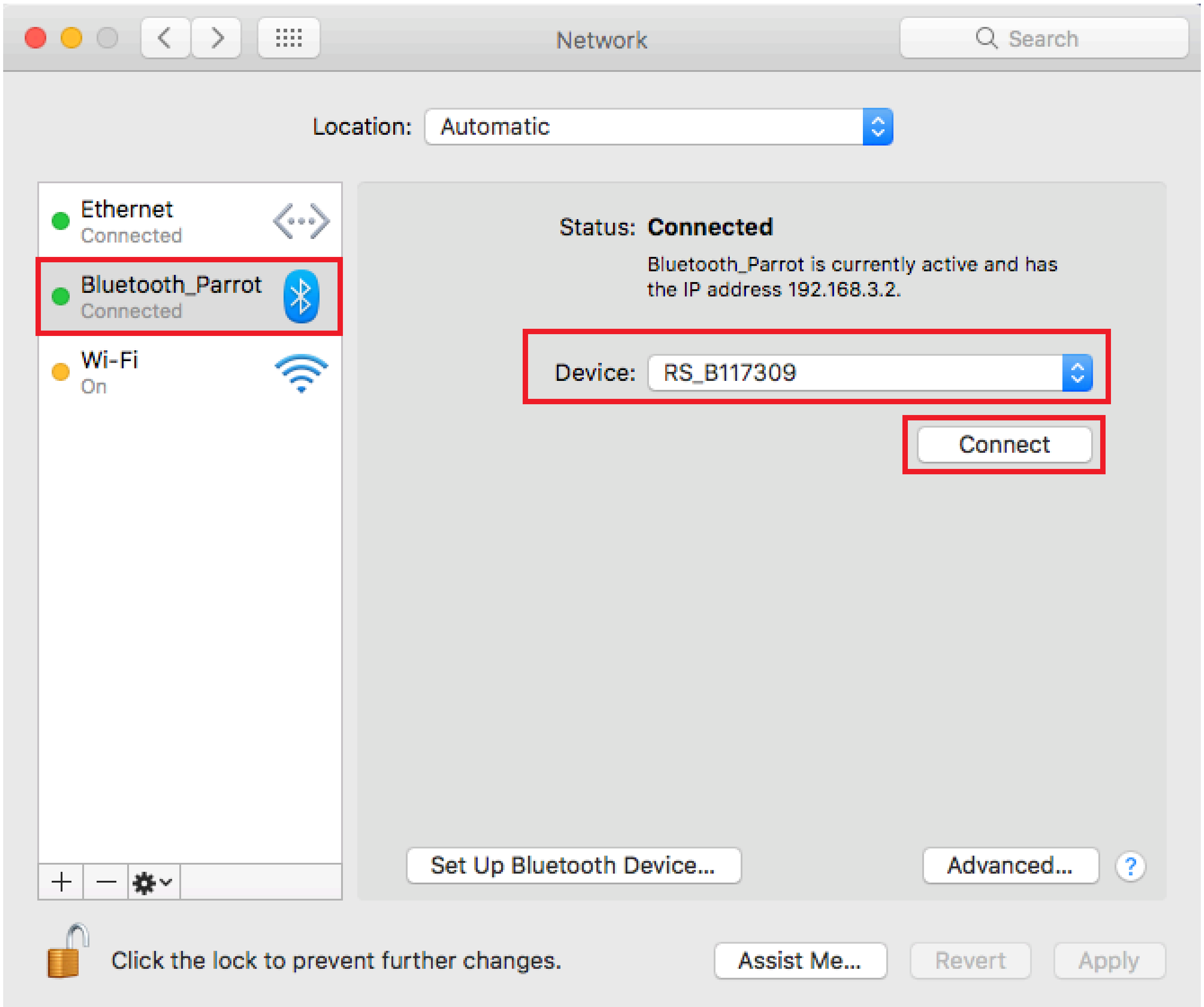
Complete the following procedure immediately after the minidrone is paired. Otherwise, the pairing may time out and you have to repeat the entire process again.

To connect the paired minidrone:

1. In the **Bluetooth** window, click the **Back** button.

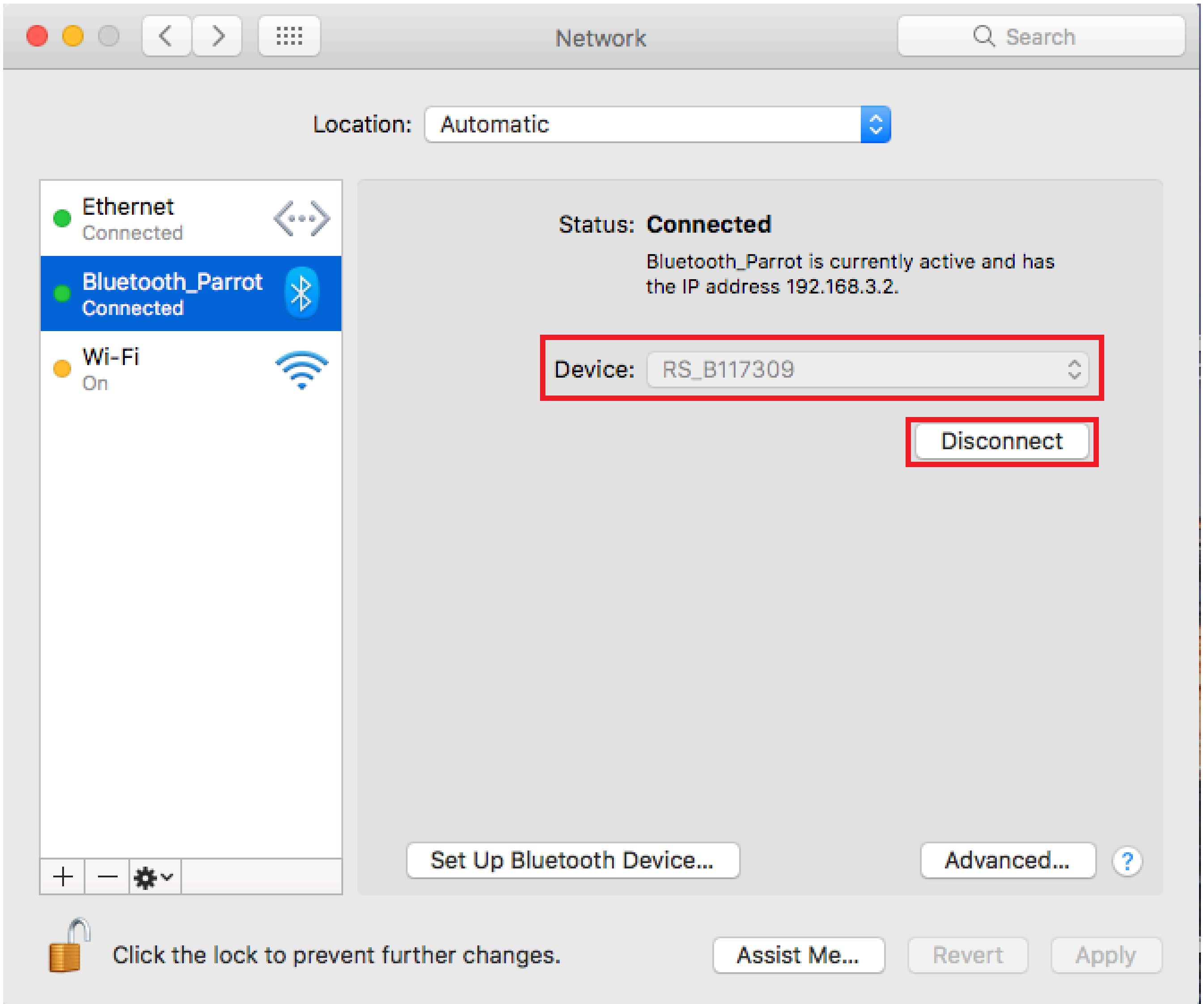


2 In the left pane, select **Bluetooth_Parrot**.
3 From the **Device** list, select the minidrone that you paired and click **Connect**.



4 After a successful connection, the **Connect** button changes to **Disconnect** and the **Device** list is disabled. The minidrone is now connected and ready to use.

Note If the **Connect** button does not change to **Disconnect** and the **Device** list is not disabled, it indicates that the connection has timed out. In such a scenario, repeat the pair on page 1-30 and connect on page 1-33 procedure again. When repeating the steps, ensure that you complete the connection procedure immediately after the minidrone is paired. Otherwise, the pairing may time out and you will have to pair and connect the minidrone again.



See Also

"Troubleshooting Connection Settings in Mac" on page 1-66

Connect a Parrot Mambo Minidrone to a Mac Using Bluetooth

To connect the Parrot Mambo minidrone to your Mac using Bluetooth, follow these instructions:

Prerequisites

1 Turn on the Parrot Mambo minidrone. The On/Off button is located under the minidrone, near the vertical camera.

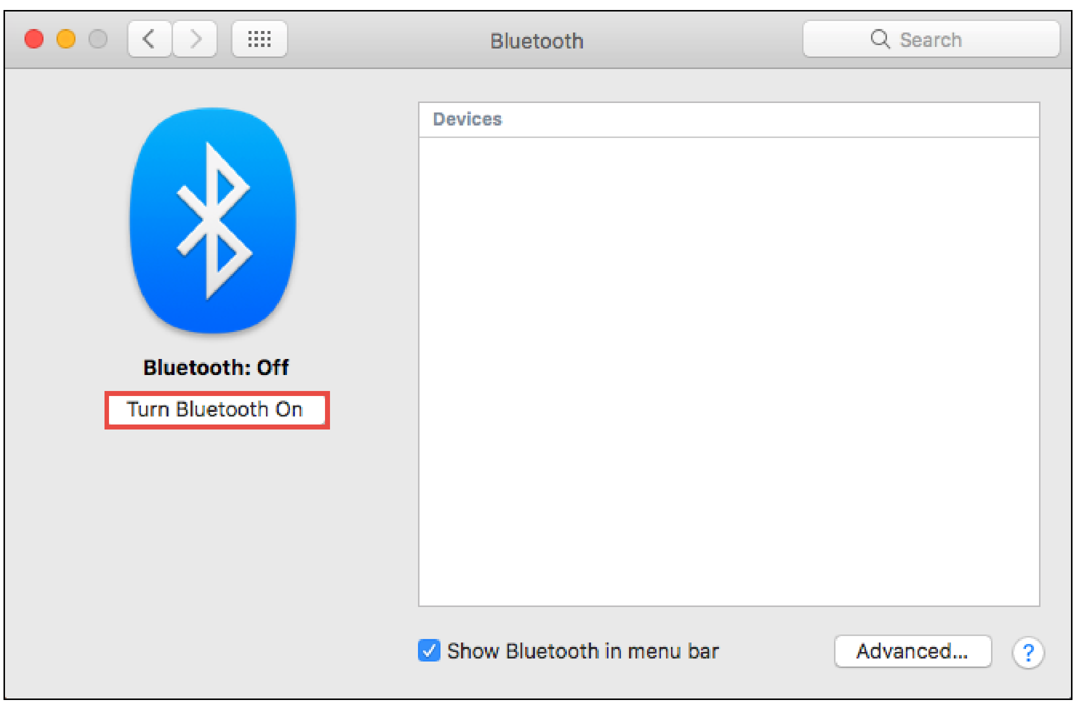
Wait until both the LEDs turn green and are blinking before proceeding to the next step.

2 Ensure that your Mac has a Bluetooth 4.0 adapter. To check the Bluetooth version on your Mac, see "Finding the LMP Version of Bluetooth Adapter on Your Mac System" on page 1-65.

3 Turn on the Bluetooth support on your Mac:

a Browse to **System Preferences** > **Bluetooth**.

b Click the **Turn Bluetooth On** button.



Pair the Parrot Mambo Minidrone with Your Mac

Before you begin, complete the steps mentioned in the “Prerequisites” on page 1-26 section.

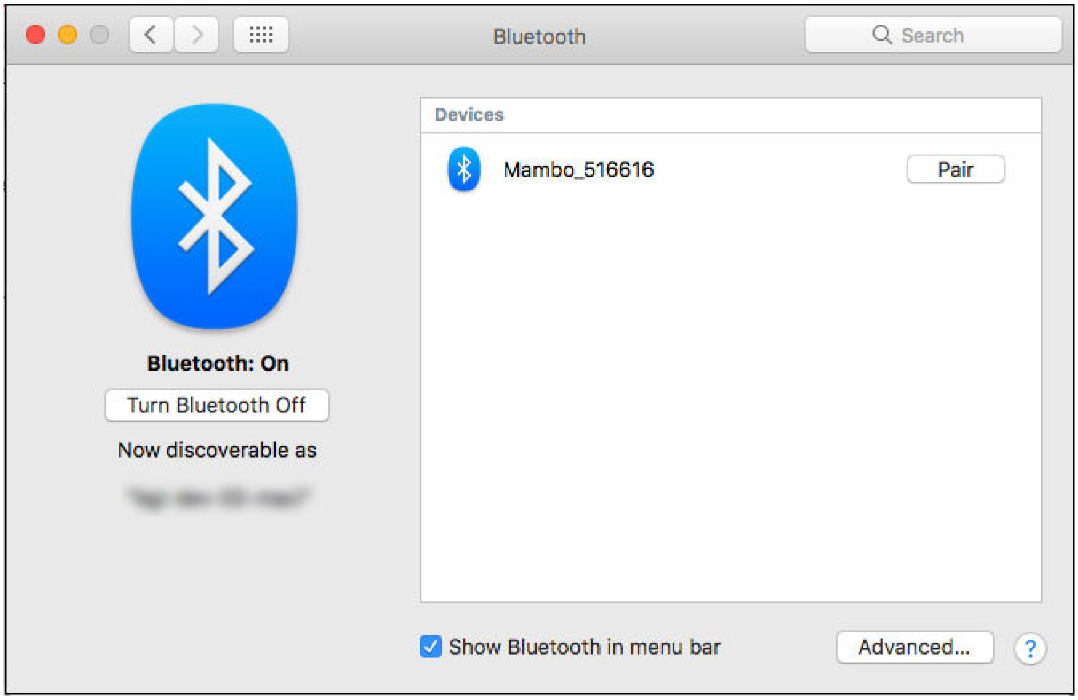
Note During the entire process, ensure that:

1. The Parrot Mambo minidrone is switched on.
2. Both the LEDs on the minidrone are green and blinking.

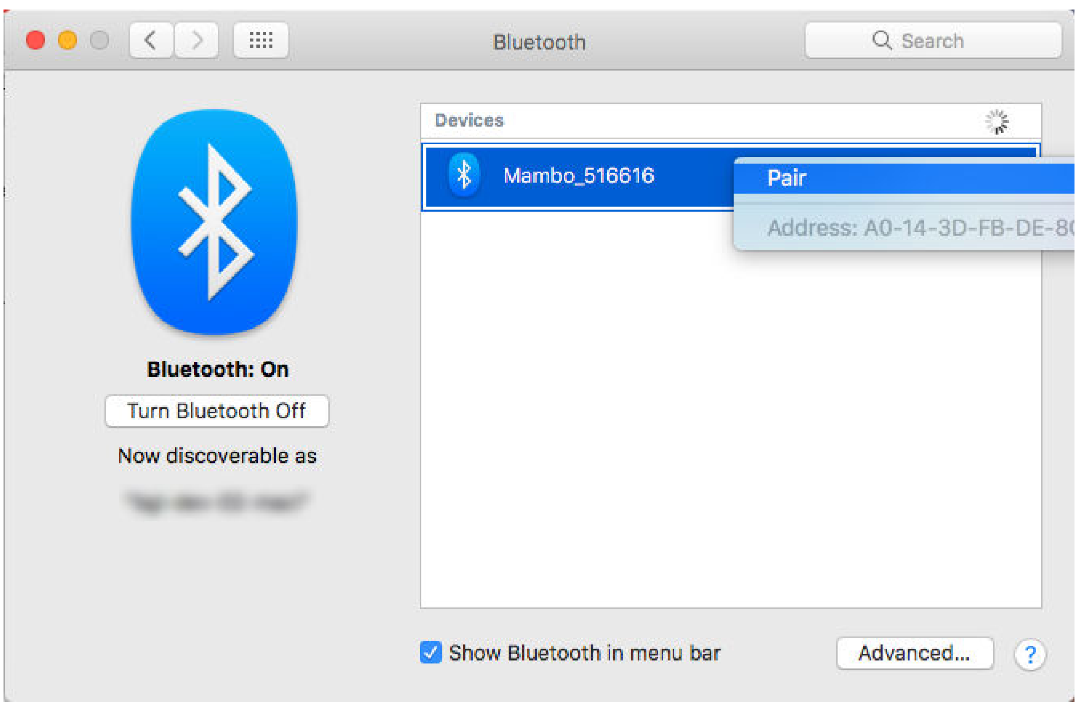
To pair the minidrone:

1. Browse to **System Preferences** and click **Bluetooth**.

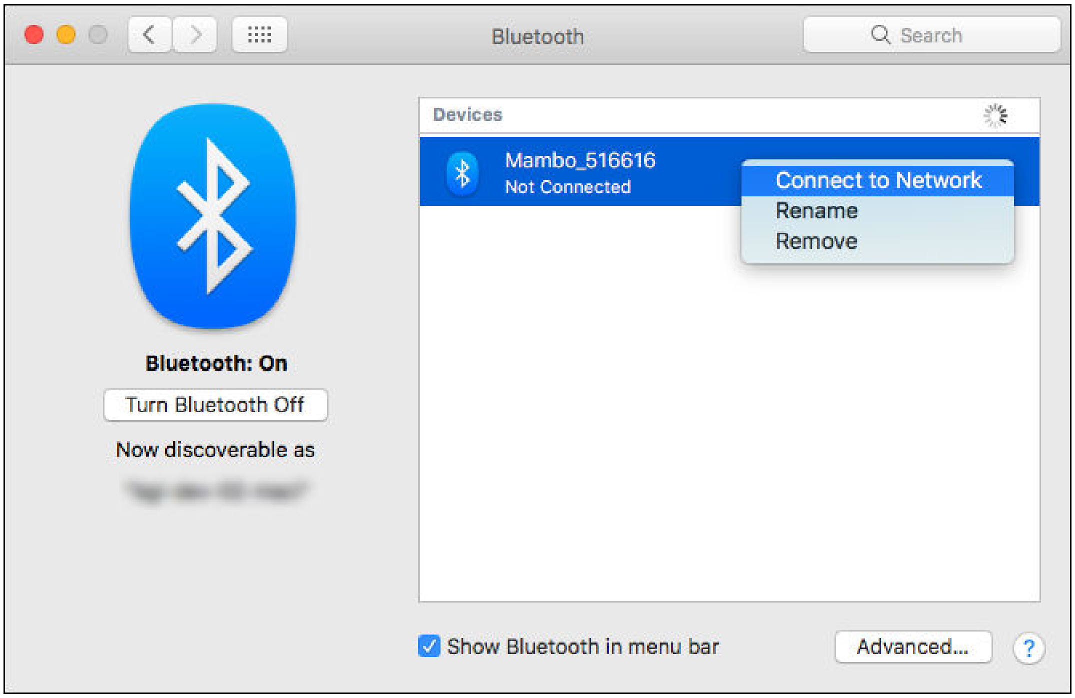
Your computer searches for all the Bluetooth-enabled devices in its range. Wait a few seconds for the minidrone to appear. The name of the minidrone appears as `Mambo_XXXXXX`, where each *X* is a number.



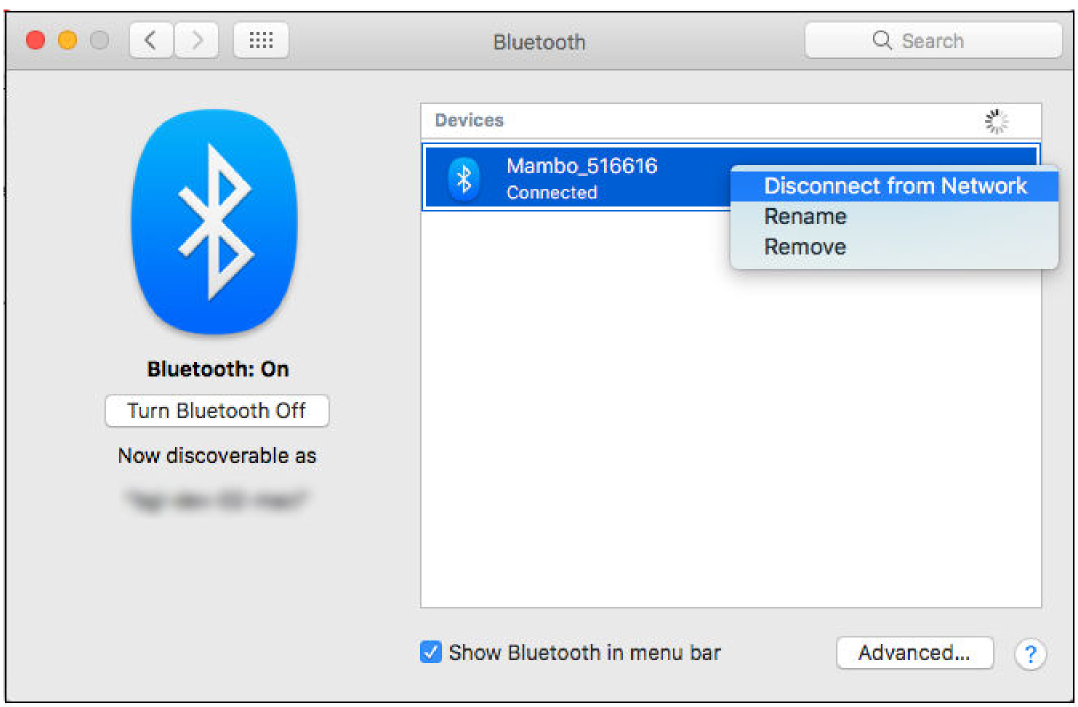
2 From the devices listed, right-click the minidrone that you want to pair with your computer and select **Pair** from the menu. The **Pair** option may not appear immediately after you right-click the name of minidrone.



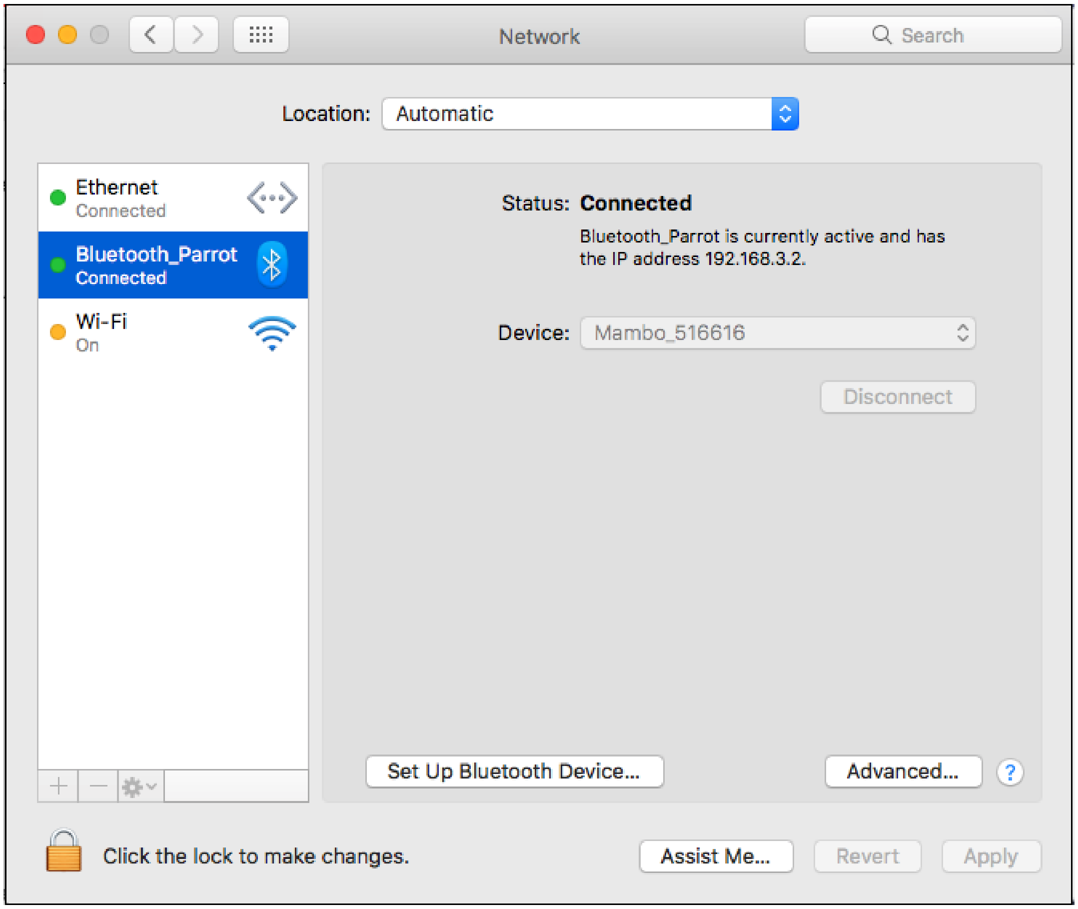
3 After the pairing is successful, right-click the minidrone again and select **Connect to Network**.



After the connection is successful, the **Not Connected** status below the minidrone changes to **Connected**, and when you right-click the minidrone, the **Disconnect from Network** option appears. The minidrone is now paired with your computer.



To verify the connection, you can also browse to **System Preferences** > **Network**, and verify that the Parrot Mambo minidrone is listed with the status as **Connected**.



See Also

"Troubleshooting Connection Settings in Mac" on page 1-66

Connect a Parrot Minidrone to a Linux System Using Bluetooth

Note You must have root privileges on the Linux system to perform tasks for establishing the Bluetooth connection.

To connect the Parrot minidrone to your Linux computer over Bluetooth, perform either of the following tasks:

- "Perform Hardware Setup Process" on page 1-42 – After you successfully complete the Hardware Setup process on your Linux computer, the Parrot minidrone is automatically connected over Bluetooth.
- "Connect to the Minidrone from MATLAB" on page 1-42 – This task can be performed to re-establish the Bluetooth connection to the Parrot minidrone, instead of performing the Hardware Setup process again, provided that the Hardware Setup process has been performed earlier, at least once, for the required minidrone.

Prerequisites

Before you begin:

1. Turn on the Parrot minidrone. The On/Off button is located under the minidrone, near the vertical camera. Wait until both the LEDs turn green.
2. Insert your Bluetooth 4.0 adapter into a USB port on your computer. To know if your adapter is Bluetooth Low Energy 4.0 compatible, see "Finding the LMP Version of Bluetooth Adapter on Your Linux System" on page 1-77.
3. Turn on the Bluetooth support on your Linux computer.
4. During the entire process, ensure that the Parrot minidrone remains switched on, and both the LEDs on the minidrone are green.

Perform Hardware Setup Process

Before you begin, complete the steps mentioned in the "Prerequisites" on page 1-42 section. You also need a micro-USB Type-B cable to complete the Hardware Setup process.

To perform the Hardware Setup process of a Parrot minidrone on the Linux computer:

1. On the MATLAB **Home** tab, in the **Environment** section, click **Add-Ons** > **Manage Add-Ons**.
2. When the **Add-On Manager** opens, click **Setup** and follow the onscreen instructions.

At the end of Hardware Setup process, the Bluetooth connection to the Parrot minidrone is automatically established.

Connect to the Minidrone from MATLAB

Before you begin, complete the steps mentioned in the "Prerequisites" on page 1-42 section. Ensure that the Hardware Setup process has been performed earlier, at least once, for the required minidrone.

To establish the Bluetooth connection to a Parrot minidrone from MATLAB on the Linux computer:

1 At the MATLAB prompt, type:

```
parrot.util.ConnectDroneToBlueTooth
```

The Linux computer searches for all Parrot minidrones nearby over Bluetooth.

2 The Bluetooth connection is established based on the number of minidrones detected during the search:

- If there is only one Parrot minidrone that is detected, the Bluetooth connection is automatically established to that minidrone.
- If there are multiple minidrones detected during the search, the search results are displayed along with the name and MAC address of the minidrones. For example, the following figure shows the results with a Rolling Spider and a Mambo detected during the search.

```
>> parrot.util.ConnectDroneToBlueTooth
Searching for PARROT Minidrone ....
The following drones have been detected. Please keep only one drone powered on. You can also provide the drone name and MAC address as an argument.
    'Mambo_560253 A0:14:3D:FD:F6:30'

    'RS_B117309 A0:14:3D:4F:14:BF'
```

Search Results for Bluetooth

Tip You can also switch-off the other minidrones and keep the required minidrone only in switched-on condition, so that `parrot.util.ConnectDroneToBlueTooth` command directly establishes connection to the required minidrone.

If your minidrone is not available in the list of devices, see "Minidrone Not Found in Bluetooth Scan Result" on page 1-73.

To connect to a specific Parrot minidrone, you need to enter the command again with proper arguments based on the search results:

- To connect to a Parrot Mambo, enter the command along with the arguments `'Mambo'` and its MAC address from the search results. For example:

  ```
  parrot.util.ConnectDroneToBlueTooth('Mambo', 'A0:14:3D:FD:F6:30')
  ```
- To connect to a Parrot Rolling Spider, enter the command along with the arguments `'Rolling Spider'` and its MAC address from the search results. For example:

  ```
  parrot.util.ConnectDroneToBlueTooth('Rolling Spider', 'A0:14:3D:4F:14:BF')
  ```

After you enter the command, the MATLAB window displays the status of the Bluetooth connection being established to the minidrone. For example, the following figure shows the status of a successful connection to a Parrot Mambo.

```
>> parrot.util.ConnectDroneToBlueTooth('Mambo', 'A0:14:3D:FD:F6:30')
Setting environmental variables....
Disconnecting drone (if already connected)....
Connecting to Mambo drone with MAC address A0:14:3D:FD:F6:30....
Creating Bluetooth network interface : bnep0.
Verifying Connection ....
Drone Connected successfully.
```

Bluetooth Connection Successful

If the above MATLAB command fails to establish a Bluetooth connection between the Parrot minidrone and the Linux computer, see “Bluetooth Connection to the Minidrone Fails” on page 1-74.

See Also

“Troubleshooting Connection Settings in Linux” on page 1-78

Restore the Original Firmware

Description

Restore the original firmware for your Parrot minidrone.

Action

To restore the original firmware of Parrot Rolling Spider, click here and follow the onscreen instructions.

To restore the original firmware of Parrot Mambo, click here and follow the onscreen instructions.

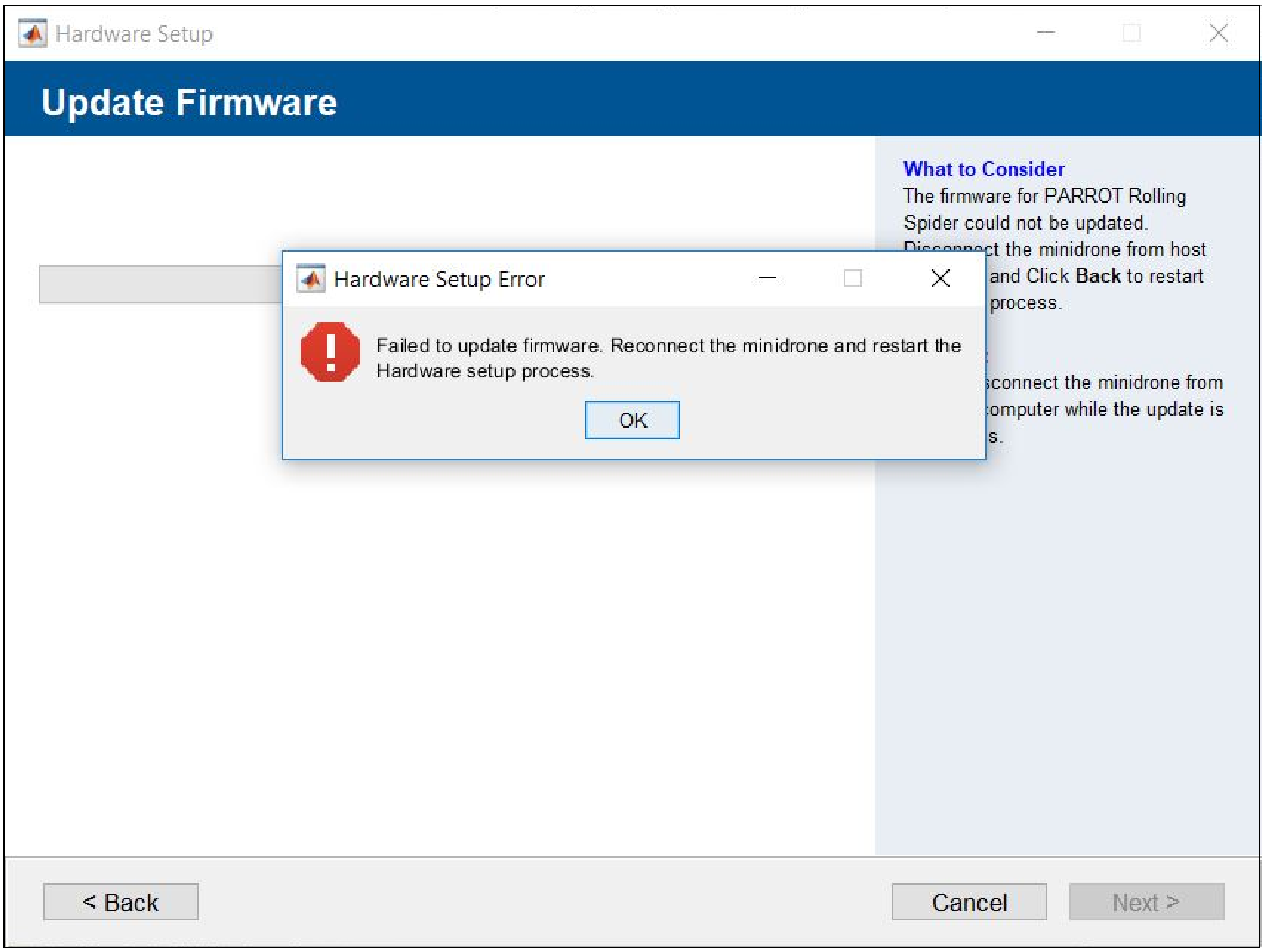
See Also

"Connect a Parrot Rolling Spider Minidrone to a Windows System Using Bluetooth" on page 1-12 | "Connect a Parrot Rolling Spider Minidrone to a Mac Using Bluetooth" on page 1-26 | "Connect a Parrot Minidrone to a Linux System Using Bluetooth" on page 1-42

Troubleshooting Firmware Update Failure

Description

If you perform Hardware Setup of Parrot Rolling Spider, the following error message may sometimes appear in the Hardware Setup screen:



Action

To troubleshoot the issue, try one of these three options sequentially until the Hardware Setup process is successful:

- Disconnect the Parrot Rolling Spider from the host computer USB port, restart the drone, reconnect the drone to the USB port, and start the Hardware Setup process again.
- Connect the Parrot Rolling Spider to a different USB port on your host computer using a different Micro-USB type B cable, and start the Hardware Setup process again.
- If the above troubleshooting tasks do not resolve the issue, perform the following steps:

1 Connect the Parrot Rolling Spider to the USB port on your host computer and wait for the two LEDs on the minidrone to stabilize to red and green.

2 If the host computer is running Windows, perform the RNDIS installation (see "Install RNDIS for Parrot Minidrones" on page 1-5).

   If the host computer is running on a Mac, ensure that the installation of all Third-Party Packages is successfully completed during the Support Package installation.

   If the host computer is running Linux:

   a Disconnect any other USB mass-storage devices; ensure that the connection to the drone is the only USB connection from the host computer.

   b Open the Linux Terminal.

   c Enter the following command:

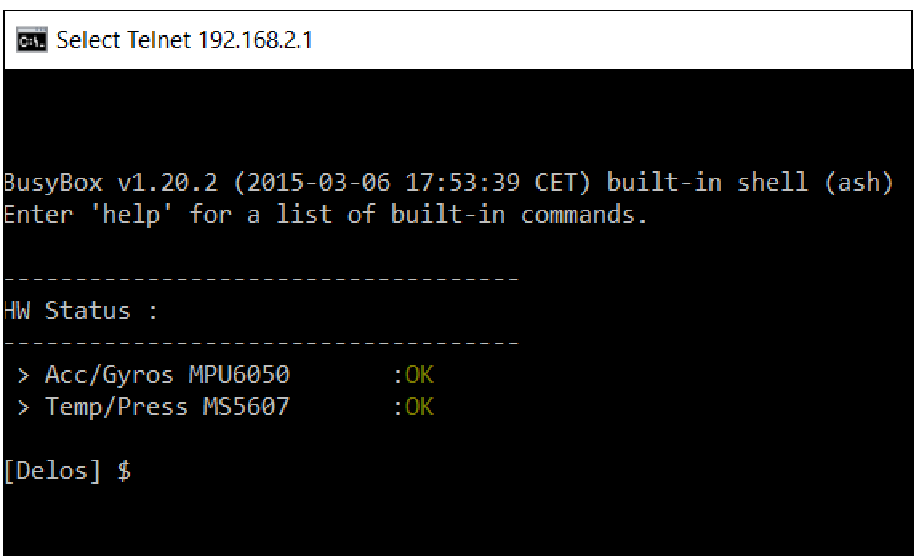
```
sudo ifconfig usb0 192.168.2.2 up
```

3 From the host computer, connect to the minidrone using Telnet (use Command Prompt (in Windows) or Terminal (in Mac and Linux)):

```
telnet 192.168.2.1
```

   If this Telnet command fails, see "Telnet Connection to Minidrone is Failing" on page 1-71 to troubleshoot the Telnet connection based on the operating system of the host computer.

   If the Telnet connection is successful, the Delos prompt appears:



4 Enter the following command at the Delos prompt:

```
echo "1.99.2" > /version.txt
```

   This command creates the file `version.txt` in the root folder of Parrot Rolling Spider. This file is required for the firmware update to be successful.

   If you get any write permission error while executing this command, reboot the minidrone and repeat the above steps.

5 Start the Hardware Setup process again.

See Also

"Connect a Parrot Rolling Spider Minidrone to a Windows System Using Bluetooth" on page 1-12 | "Connect a Parrot Rolling Spider Minidrone to a Mac Using Bluetooth" on page 1-26 | "Connect a Parrot Minidrone to a Linux System Using Bluetooth" on page 1-42

Factory Reset of Parrot Mambo Minidrone

Description

A factory reset of Parrot Mambo minidrone clears all the data that was deployed on the minidrone. The factory reset may be required if you encounter the following problems:

- The connection to Parrot Mambo minidrone over Bluetooth is failing, even though the connection was successful earlier.
- Parrot Mambo minidrone is powering off immediately after a reboot.
- You observe unsatisfactory performance of the code running on the Parrot Mambo minidrone.

Action

To perform a factory reset of Parrot Mambo minidrone, perform these steps:

1. Switch On the Parrot minidrone, and wait for the LEDs to turn green.
2. Press the Power button again and keep it pressed for about 30 seconds. During this period:
   a. The two LEDs turn red alternatively and start blinking.
   b. The frequency of blinking increases indicating the last stage of the reset process.
   c. Keep the Power button pressed until the LEDs turn green.
3. Clear the memory of the minidrone (see “Troubleshooting by Clearing Parrot Minidrone Memory” on page 1-70).

After performing the factory reset of Parrot Mambo minidrone, connect the minidrone to the host computer over Bluetooth (see “Setup and Configuration”).

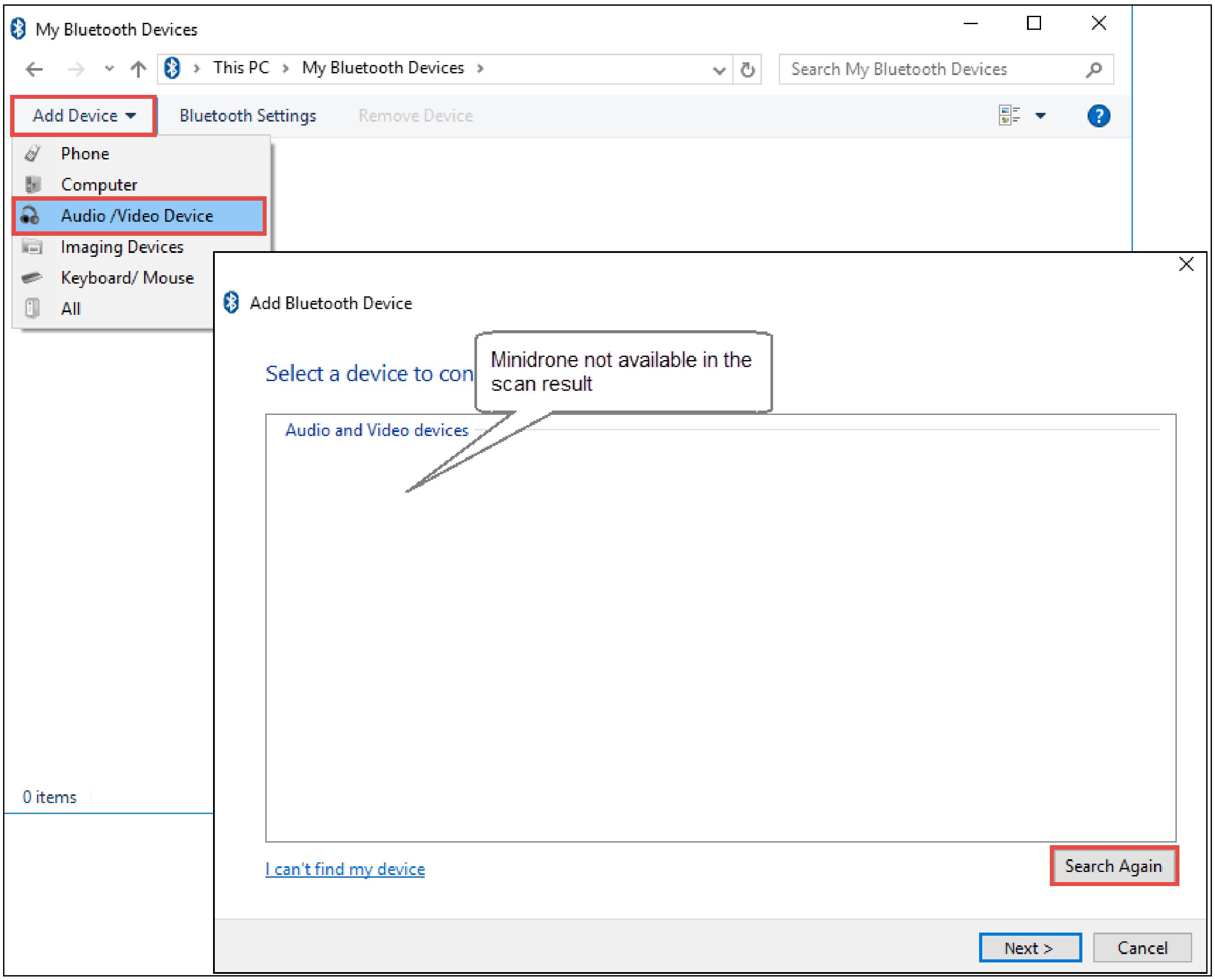
See Also

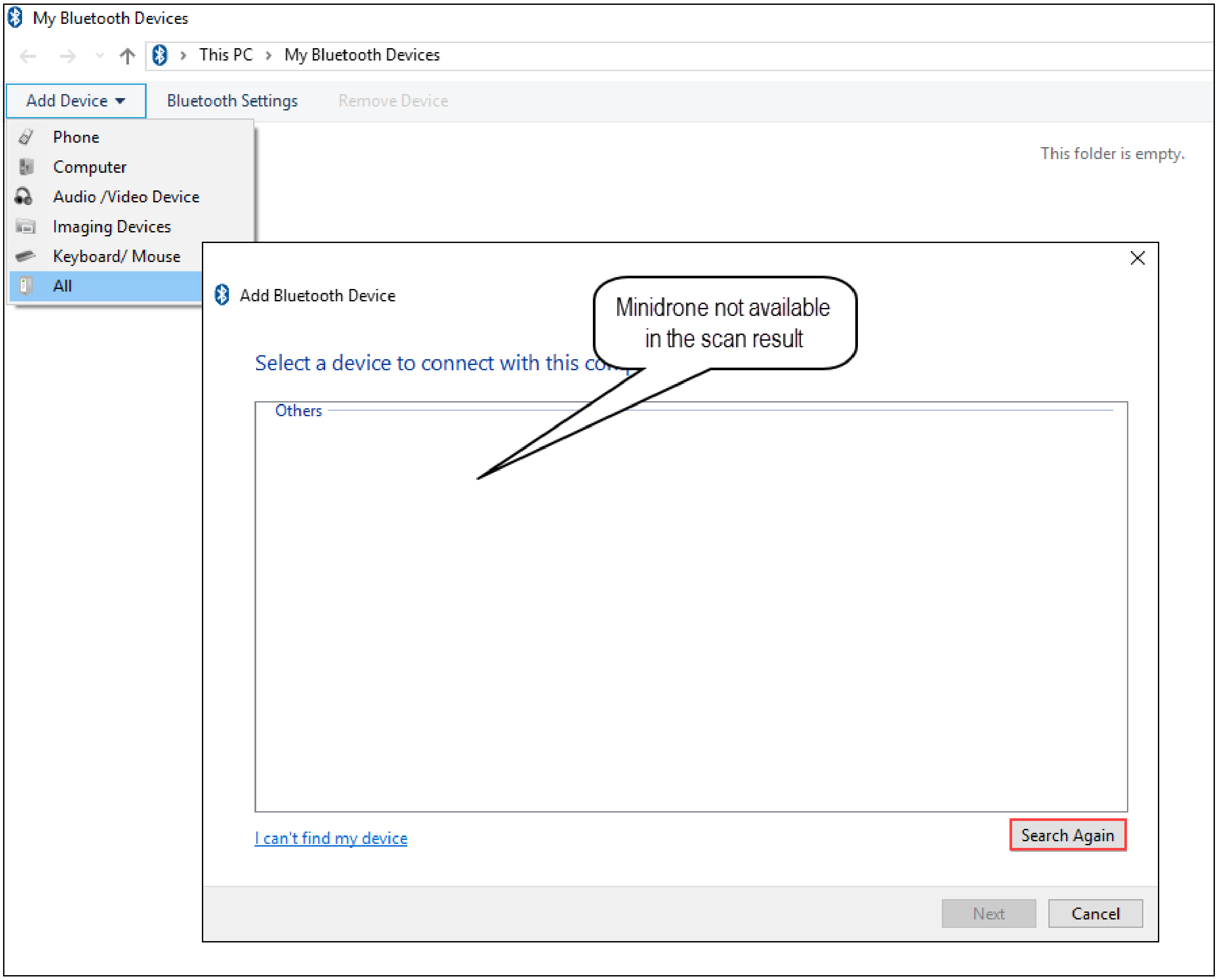
“Connect a Parrot Rolling Spider Minidrone to a Windows System Using Bluetooth” on page 1-12 | “Connect a Parrot Rolling Spider Minidrone to a Mac Using Bluetooth” on page 1-26 | “Connect a Parrot Minidrone to a Linux System Using Bluetooth” on page 1-42

Minidrone Not Found in Bluetooth Scan Result

Description

The minidrone does not appear in the list of nearby Bluetooth-enabled devices. This issue occurs after you search for `Audio/Video Device` (for Parrot Rolling Spider) or when you search for `All` (for Parrot Mambo).





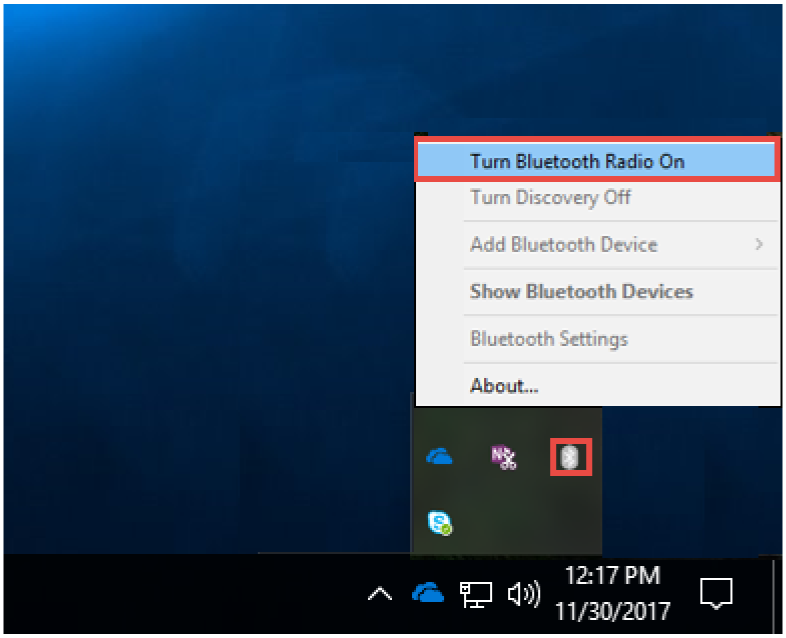
Action

Try one of these four options sequentially until the minidrone appears in the scan results:

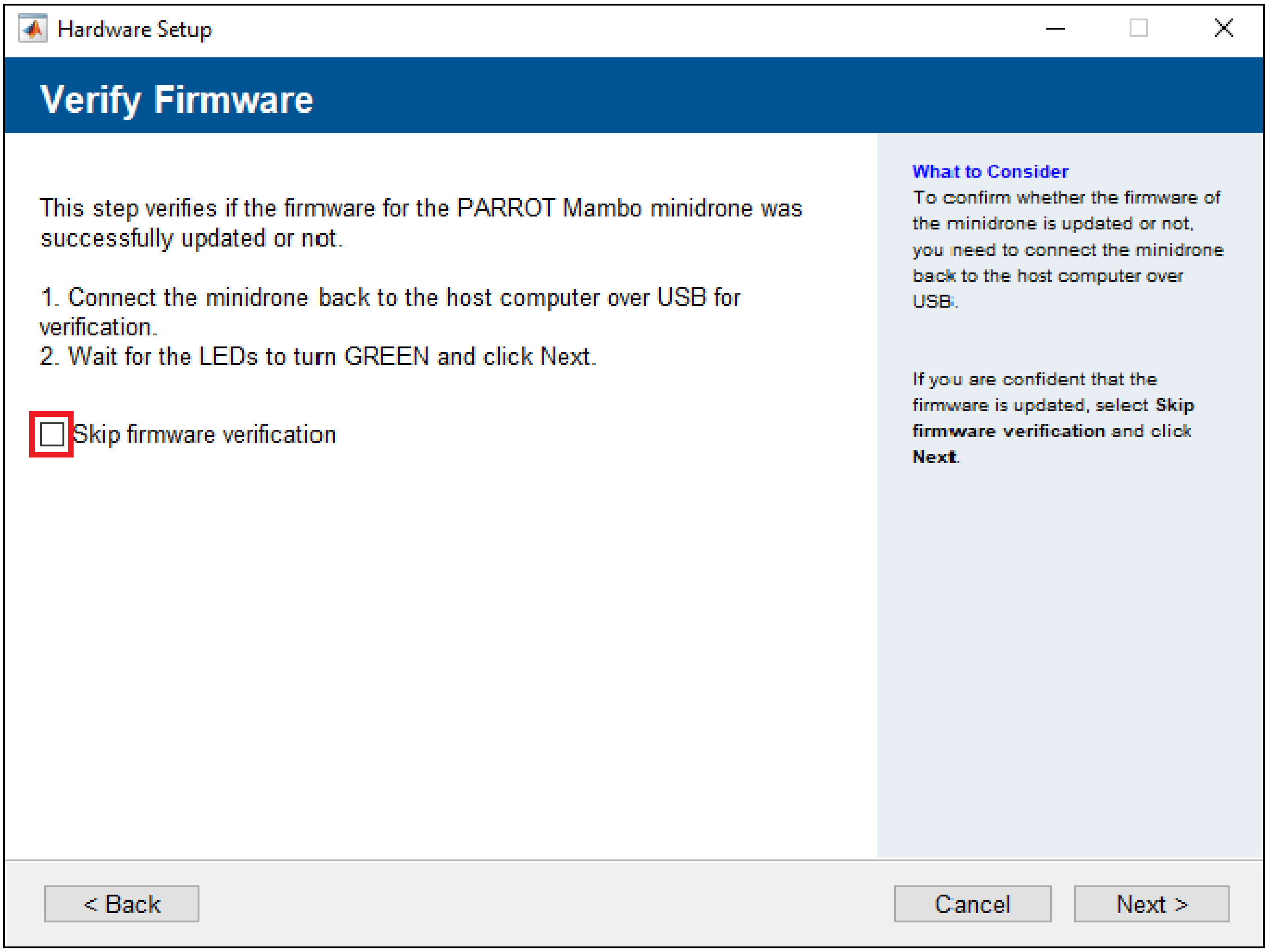
1 If the minidrone does not appear in the list, it indicates that the Bluetooth driver installation was not successful. In that case, install the driver again (see Step 4 in "Prerequisites" on page 1-12) and follow the steps in "Connect a Parrot Rolling Spider Minidrone to a Windows System Using Bluetooth" on page 1-12 or "Connect a Parrot Mambo Minidrone to a Windows System Using Bluetooth" on page 1-18.

2 Check these settings and click the **Search Again** button:

- **a** The minidrone is switched on and both the LEDs on the minidrone are green.
- **b** The Bluetooth support on your computer is turned on. To turn on the Bluetooth support, right-click the Bluetooth icon in your taskbar, and select **Turn Bluetooth Radio On**.



3. Restart the minidrone, wait until both LEDs on the minidrone turn green, and follow the steps in "Add a Parrot Rolling Spider Minidrone to a Windows System" on page 1-13 or "Add a Parrot Mambo Minidrone to a Windows System" on page 1-19.
4. Restart your computer and follow the steps in "Connect a Parrot Rolling Spider Minidrone to a Windows System Using Bluetooth" on page 1-12 or "Connect a Parrot Mambo Minidrone to a Windows System Using Bluetooth" on page 1-18.
5. If the minidrone still does not appear in the list, it indicates that the Bluetooth driver installation was not successful. In that case, install the driver again (see Step 4 in "Prerequisites" on page 1-12) and follow the steps in "Connect a Parrot Rolling Spider Minidrone to a Windows System Using Bluetooth" on page 1-12.
6. Parrot Mambo does not appear in the scan results if the firmware is not updated during the Hardware Setup process. To verify that the firmware is updated, perform the Hardware Setup process again for Parrot Mambo, and ensure that the **Skip firmware verification** option is not checked.



The Hardware Setup screen indicates if the firmware is updated or not. If the firmware is updated, perform Step 3 and Step 4 as listed in this section.

See Also

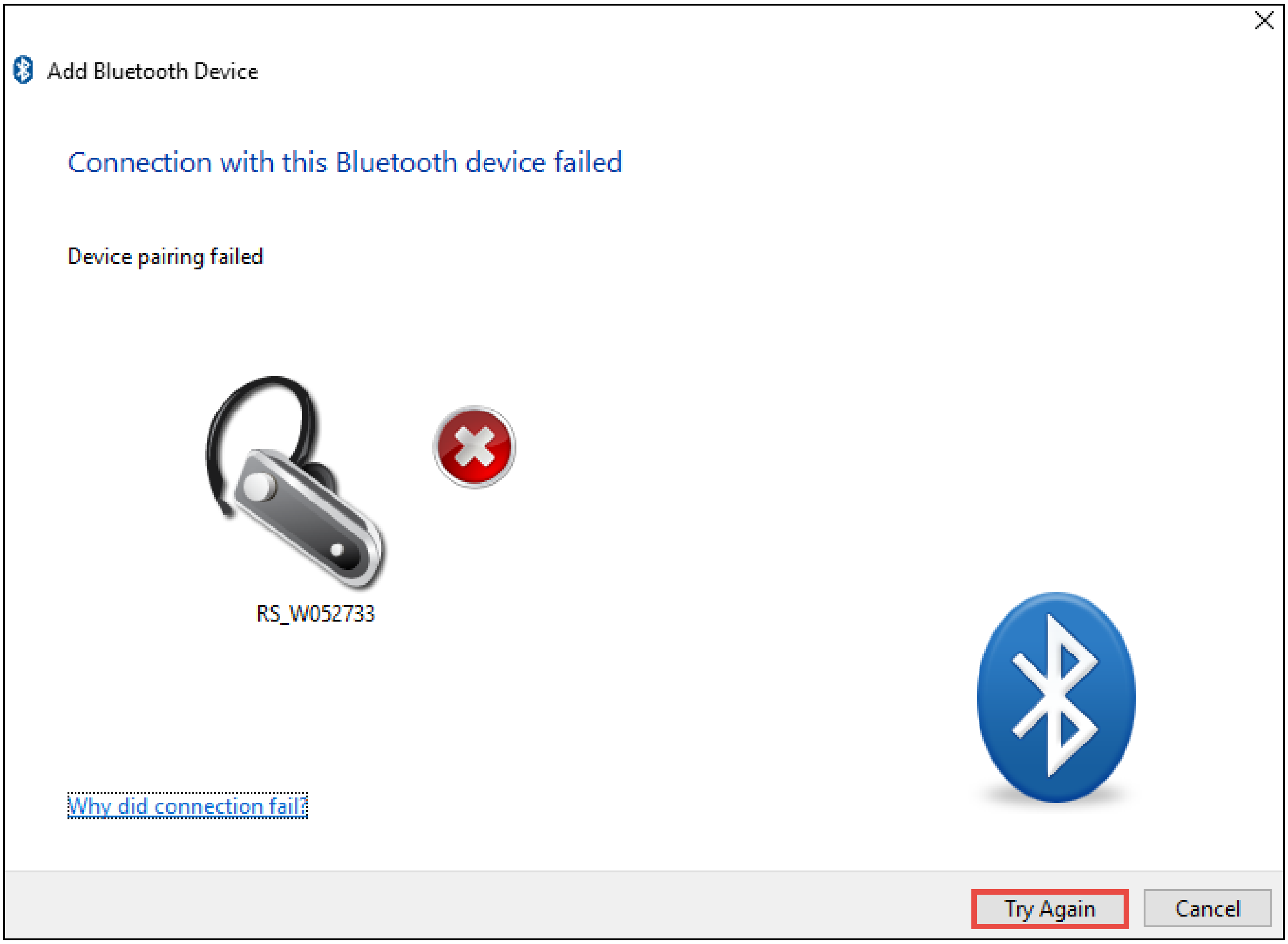
"Troubleshooting Connection Settings in Windows" on page 1-59 | "Connect a Parrot Rolling Spider Minidrone to a Windows System Using Bluetooth" on page 1-12

Bluetooth Connection Fails

Description

The Bluetooth connection between your computer and Parrot minidrone fails.

For example, for Parrot Rolling Spider, the following error message is displayed:



Action

Try these options sequentially until the connection is established:

1. Check these settings and click the **Try Again** button:
   a. The Bluetooth adapter supports Bluetooth Low Energy 4.0 version. To find out if your adapter is Bluetooth Low Energy 4.0 compatible, see "Finding the LMP Version of Bluetooth Adapter on Your Windows System" on page 1-58.
   b. The minidrone is switched on and both LEDs on the minidrone are green.

2 Restart the minidrone, wait until both the LEDs on the minidrone turn green, and follow the steps in “Add a Parrot Rolling Spider Minidrone to a Windows System” on page 1-13 or “Add a Parrot Mambo Minidrone to a Windows System” on page 1-19.

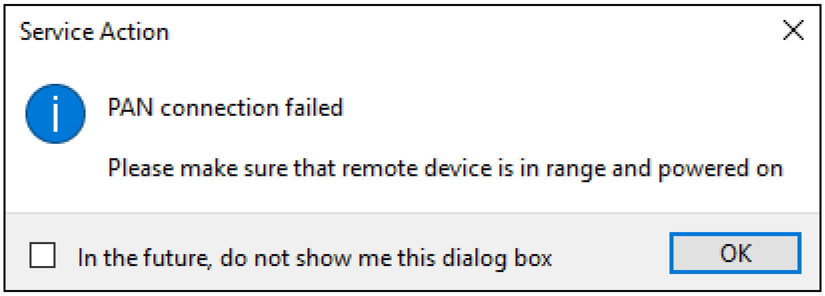
See Also

“Troubleshooting Connection Settings in Windows” on page 1-59 | “Connect a Parrot Mambo Minidrone to a Windows System Using Bluetooth” on page 1-18 | “Connect a Parrot Rolling Spider Minidrone to a Windows System Using Bluetooth” on page 1-12

Personal Area Networking (PAN) Connection Fails

Description

The PAN connection between your computer and the Parrot minidrone fails.



Action

Try these options sequentially until the connection is established:

1 If you are connecting to Parrot Rolling Spider, follow the steps in “Add a Parrot Rolling Spider Minidrone to a Windows System” on page 1-13 and “Set Up a Bluetooth PAN Connection for Parrot Rolling Spider” on page 1-16, and click the pairing prompt every time it appears.

If you are connecting to Parrot Mambo, follow the steps in “Add a Parrot Mambo Minidrone to a Windows System” on page 1-19 and “Set Up a Bluetooth NAP Connection for Parrot Mambo” on page 1-23.

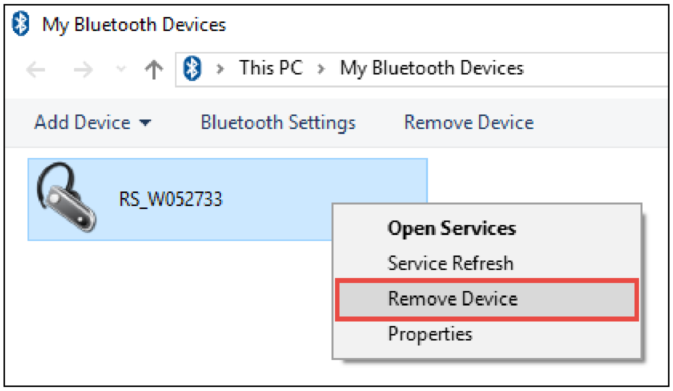
2 Perform these steps:

a Open **Windows Explorer** and click **This PC/ My Computer**.

b In the **Devices and drives/ Other Devices** section, double-click **My Bluetooth Devices**.

c From the list of available devices, right-click the minidrone, and select **Remove Device** from the menu.

For example, for Parrot Rolling Spider:



d Connect the minidrone again by following the steps in "Add a Parrot Rolling Spider Minidrone to a Windows System" on page 1-13 and "Set Up a Bluetooth PAN Connection for Parrot Rolling Spider" on page 1-16, or "Add a Parrot Mambo Minidrone to a Windows System" on page 1-19 and "Set Up a Bluetooth NAP Connection for Parrot Mambo" on page 1-23.

See Also

"Troubleshooting Connection Settings in Windows" on page 1-59 | "Connect a Parrot Rolling Spider Minidrone to a Windows System Using Bluetooth" on page 1-12

Finding the LMP Version of Bluetooth Adapter on Your Windows System

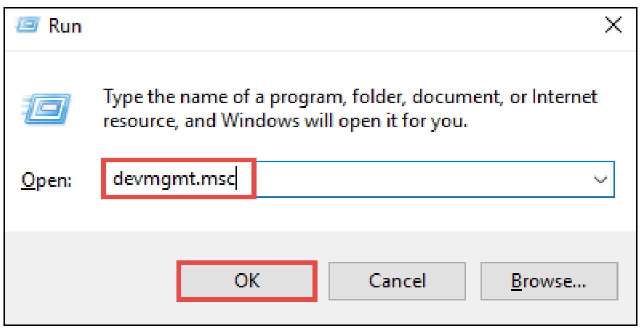
Description

In Windows, you want to know if your Bluetooth adapter is Bluetooth Low Energy 4.0 compatible.

Action

Follow these steps to find out if your Bluetooth adapter is Low Energy 4.0 compatible:

1. Click the **Search** icon in the taskbar and type `Run`.
2. In the **Run** dialog box, type `devmgmt.msc` and click **OK**.



3. In the **Device Manager** window, click **Bluetooth**. Look for `Generic Bluetooth Radio` in the expanded list.
4. Double-click the device. In the **Properties** dialog box that opens, click the **Advanced** tab. Locate the **LMP Version** information. If the version starts with 6 or higher, your adapter is compatible with Bluetooth Low Energy 4.0. Any version lower than 6 indicates an older version of Bluetooth.

See Also

"Troubleshooting Connection Settings in Windows" on page 1-59 | "Connect a Parrot Rolling Spider Minidrone to a Windows System Using Bluetooth" on page 1-12

Troubleshooting Connection Settings in Windows

This section explains how to solve problems that you might encounter when connecting the Parrot minidrone to Windows system using Bluetooth.

- “Minidrone Not Found in Bluetooth Scan Result” on page 1-50
- “Bluetooth Connection Fails” on page 1-54
- “Personal Area Networking (PAN) Connection Fails” on page 1-56
- “Finding the LMP Version of Bluetooth Adapter on Your Windows System” on page 1-58

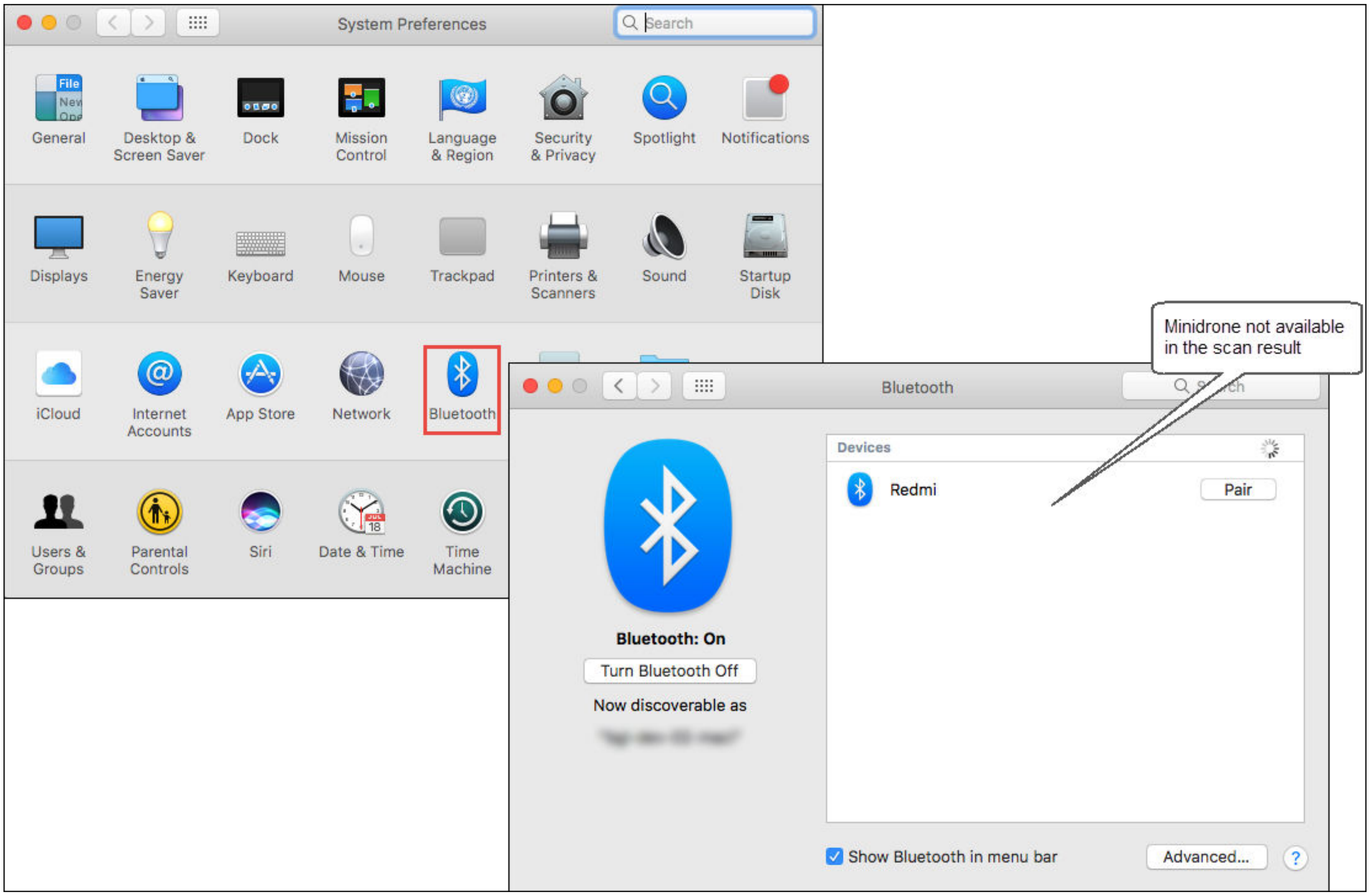
See Also

“Connect a Parrot Rolling Spider Minidrone to a Windows System Using Bluetooth” on page 1-12

Minidrone Not Found in Bluetooth Scan Result

Description

The minidrone does not appear in the list of nearby, Bluetooth-enabled devices.



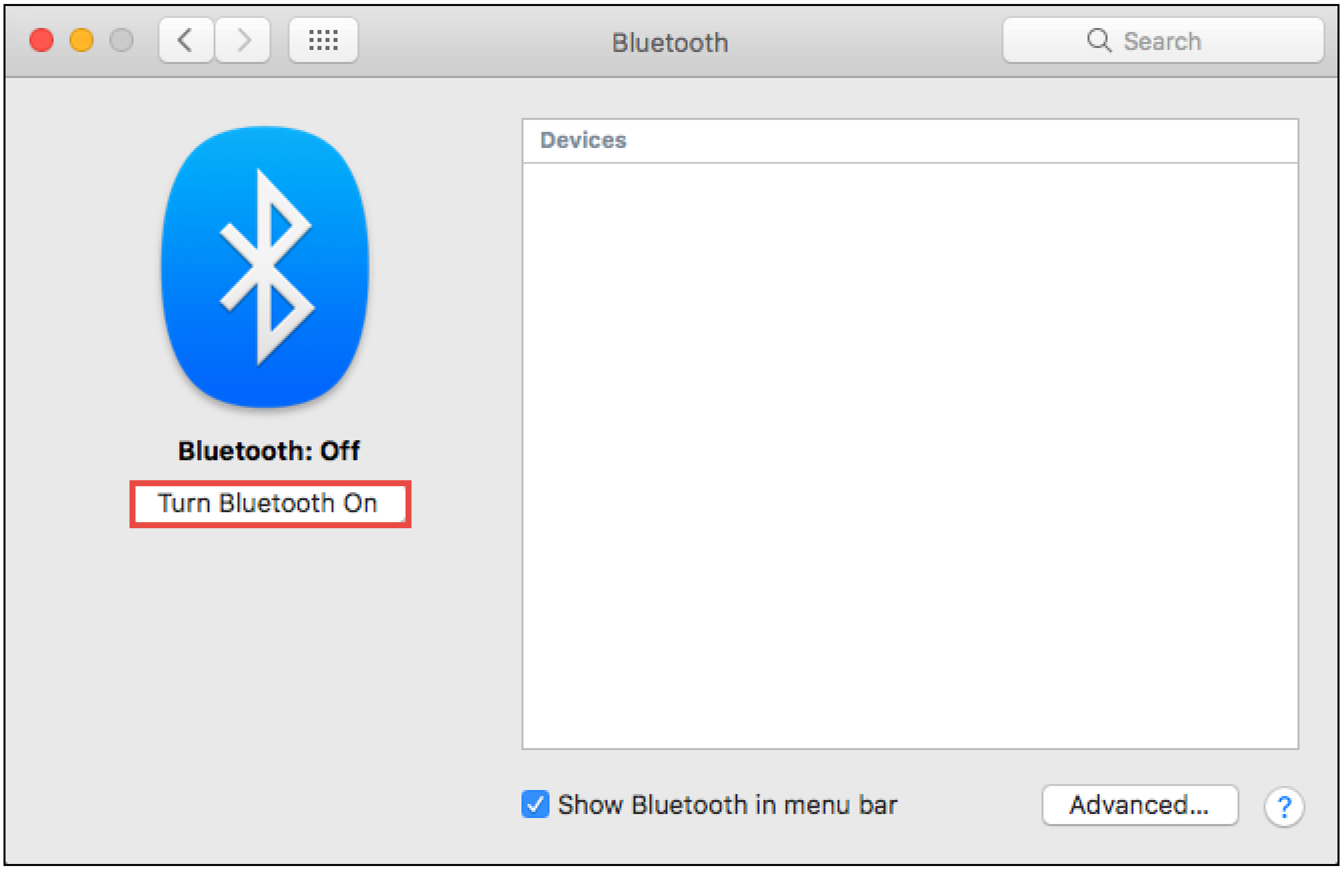
Action

Try these options sequentially until the minidrone appears in the scan result:

1 Check these settings and follow the steps in “Pair the Minidrone with Your Mac” on page 1-30:

   a The minidrone is switched on, and both the LEDs on the minidrone are green.

   b The Bluetooth support on your computer is turned on. To turn on Bluetooth support, open **System Preferences** and click **Bluetooth**. Click **Turn Bluetooth On**.



2 Restart the minidrone, wait until both LEDs on the minidrone turn green, and follow the steps in "Pair the Minidrone with Your Mac" on page 1-30.

3 Restart your computer and follow the steps in "Connect a Parrot Rolling Spider Minidrone to a Mac Using Bluetooth" on page 1-26.

See Also

"Troubleshooting Connection Settings in Mac" on page 1-66 | "Connect a Parrot Rolling Spider Minidrone to a Mac Using Bluetooth" on page 1-26

Bluetooth Connection Fails

Description

After a successful pairing, the connection between your Mac and the Parrot minidrone fails.

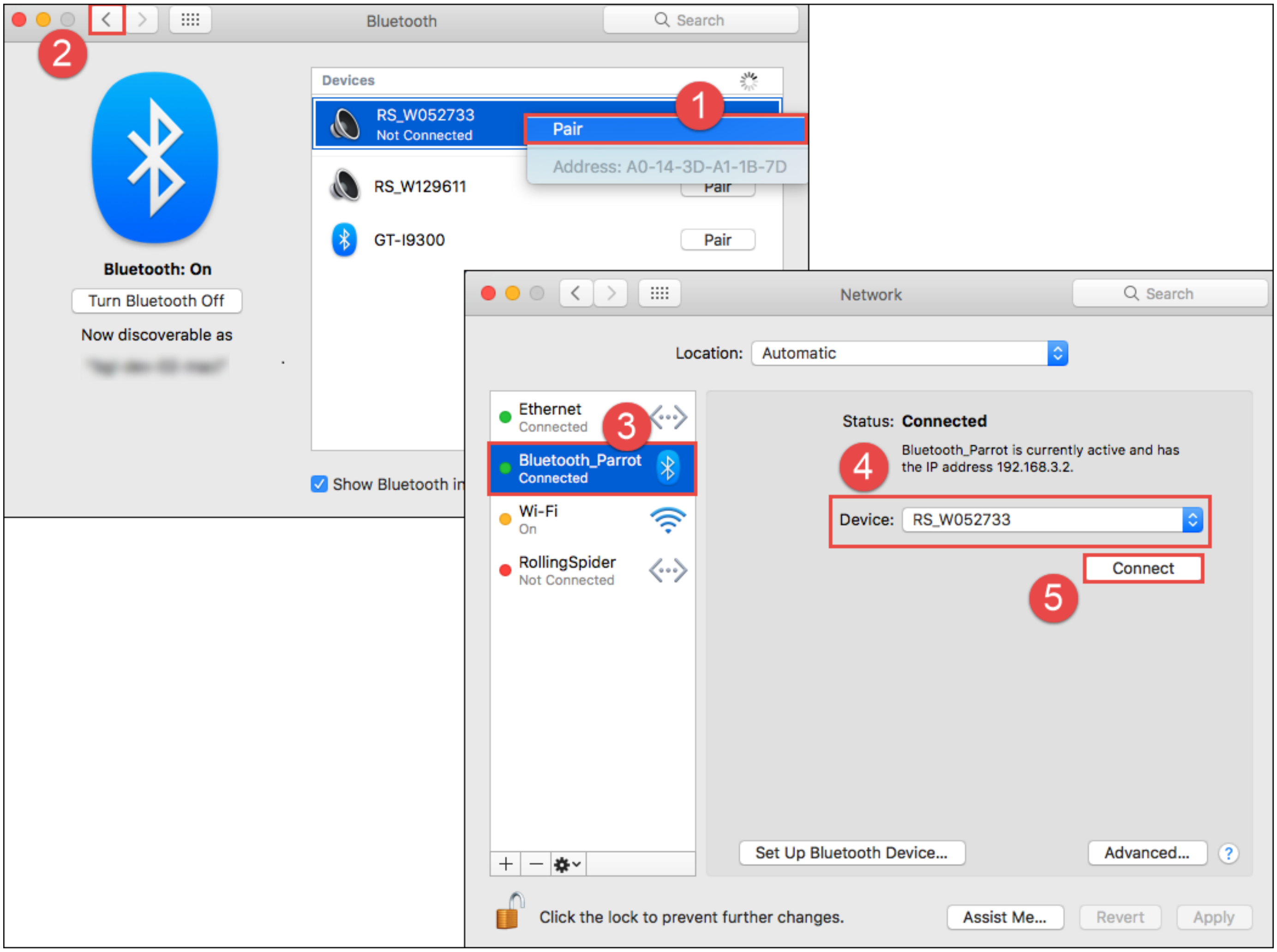
Action

Complete these steps before the pairing time-out occurs. Otherwise, the connection fails again.

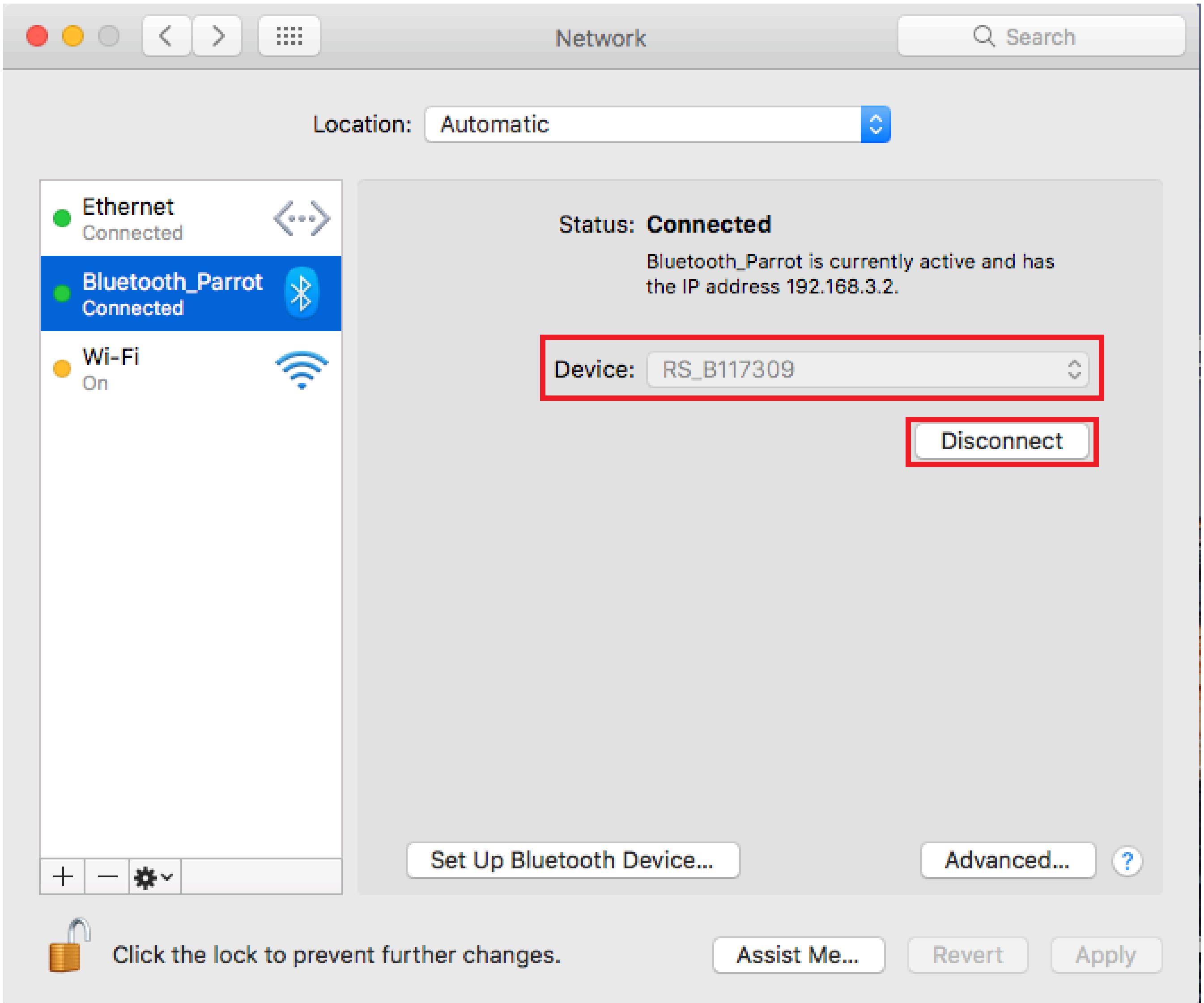
1. In the Bluetooth window, from the list of devices, right-click the minidrone. Select **Pair** from the menu.

   Note The **Pair** option might take a few seconds to appear.

2. Click the **Back** button.
3. In the left pane of the Network window, select **Bluetooth_Parrot**.
4. In the **Device** list, select the minidrone that you paired.
5. Click **Connect** to connect the minidrone.



After a successful connection, the **Connect** button changes to **Disconnect** and the **Device** list is disabled. The minidrone is now connected and ready to use.



For more information, see "Pair the Minidrone with Your Mac" on page 1-30 and "Connect the Paired Minidrone to Your Mac" on page 1-33.

See Also

"Troubleshooting Connection Settings in Mac" on page 1-66 | "Connect a Parrot Rolling Spider Minidrone to a Mac Using Bluetooth" on page 1-26

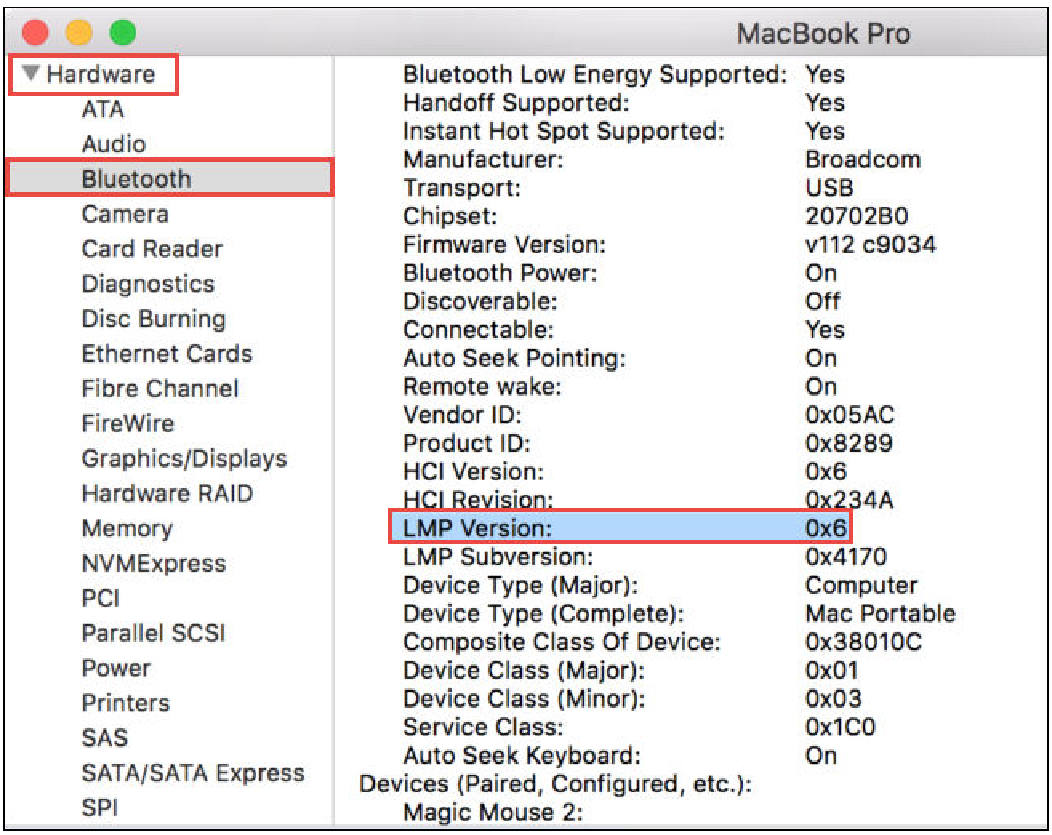
Finding the LMP Version of Bluetooth Adapter on Your Mac System

Description

You want to know if your Mac system is Bluetooth Low Energy 4.0 compatible.

Action

1. Click the Apple logo icon on the top left corner, and then select **About This Mac**.
2. Click the **System Report** button.
3. Under the **Hardware** tab, select **Bluetooth**. To find the version of Bluetooth on your Mac, find **LMP Version** in the list. If the version is `0x6` or higher, your Mac is compatible with Bluetooth Low Energy 4.0. Any version lower than that indicates an older version of Bluetooth.



See Also

"Troubleshooting Connection Settings in Mac" on page 1-66 | "Connect a Parrot Rolling Spider Minidrone to a Mac Using Bluetooth" on page 1-26

Troubleshooting Connection Settings in Mac

This section explains how to solve problems that you might encounter when connecting the Parrot minidrone to a Mac using Bluetooth.

- "Minidrone Not Found in Bluetooth Scan Result" on page 1-60
- "Bluetooth Connection Fails" on page 1-62
- "Finding the LMP Version of Bluetooth Adapter on Your Mac System" on page 1-65

See Also

"Connect a Parrot Rolling Spider Minidrone to a Mac Using Bluetooth" on page 1-26

Adapter Not Recognized by Computer

Description

You insert your Bluetooth adapter into a USB port on your computer, but the adapter is not recognized by your computer.

Command from MATLAB

```
parrot.util.ConnectDroneToBlueTooth
```

Sample

```
No Bluetooth adapter found. Please connect a Bluetooth adapter and ensure that Bluetooth is turne
```

Action

Reinsert the adapter into other types of USB ports, such as USB 1.0 and USB 2.0, and then turn-on the Bluetooth on the Linux computer. At the MATLAB prompt, type the following command again, and see if the scan process has started.

```
parrot.util.ConnectDroneToBlueTooth
```

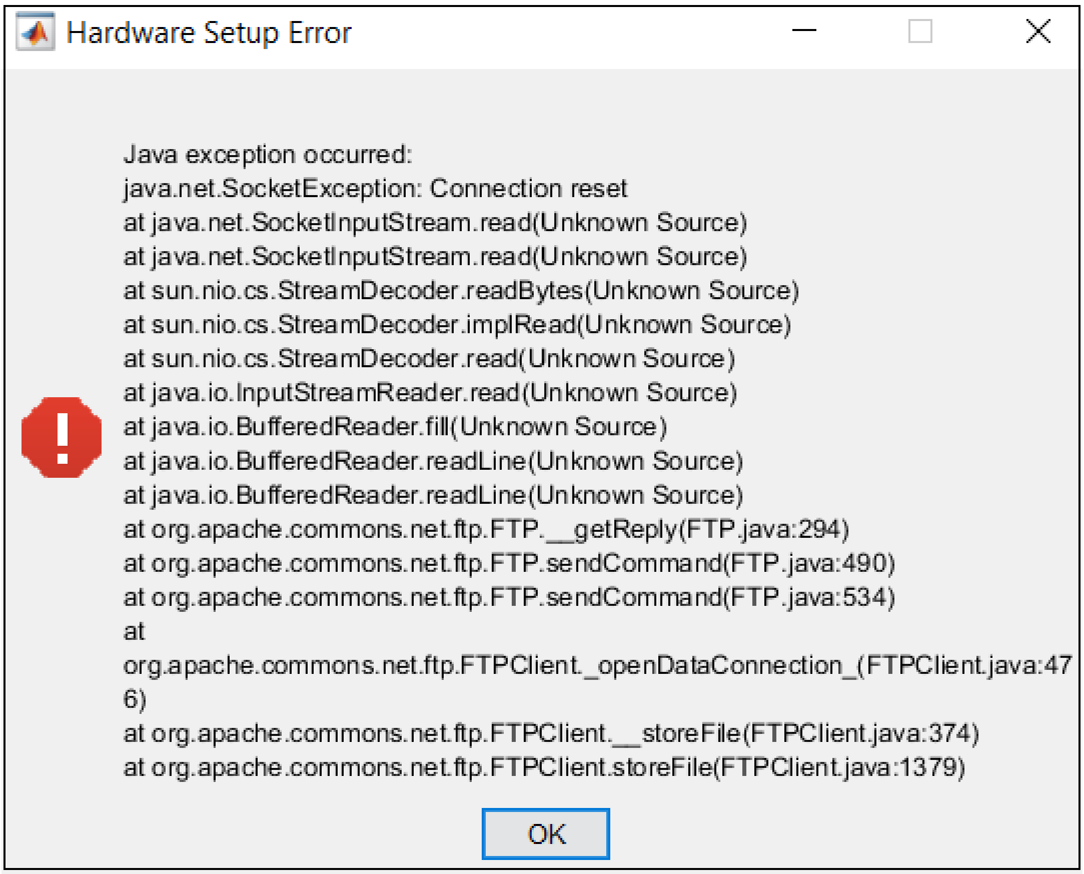
See Also

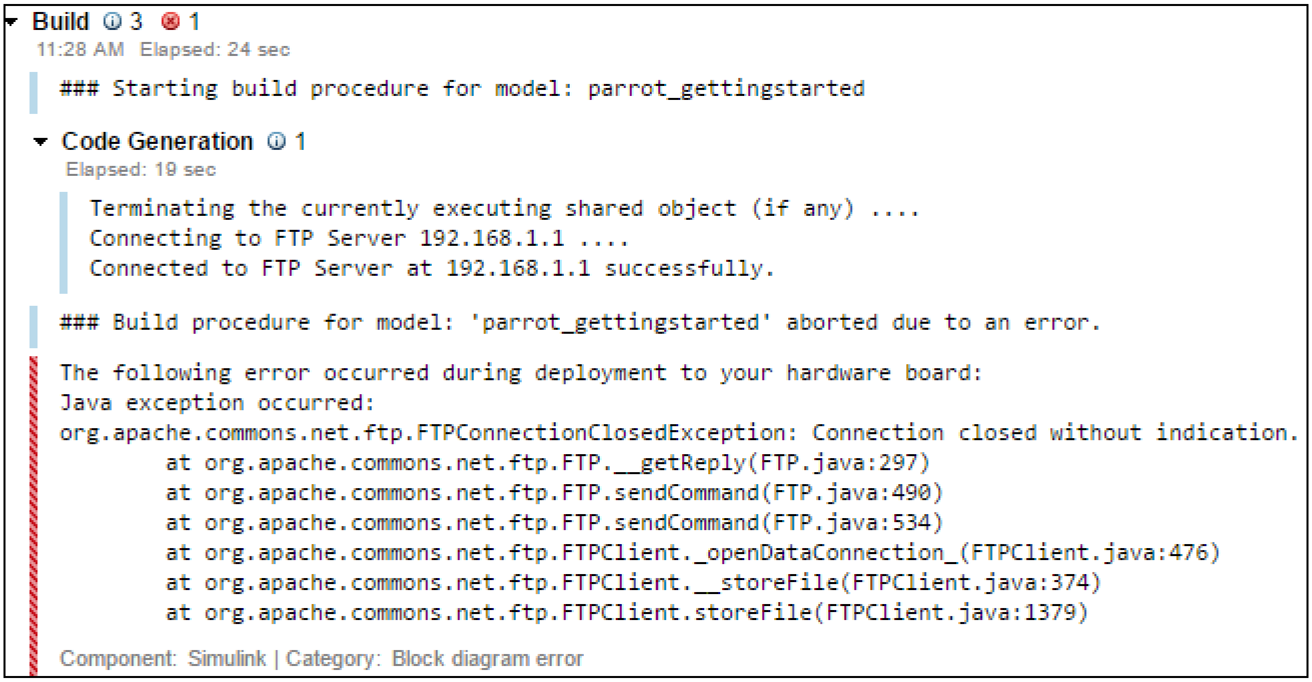
"Troubleshooting Connection Settings in Linux" on page 1-78 | "Connect a Parrot Minidrone to a Linux System Using Bluetooth" on page 1-42

Troubleshooting Java Exception Failures

Description

Java exception failures related to FTP can occur either during the Hardware Setup process of Parrot minidrone on your host computer or during the deployment of code from Simulink to the Parrot minidrone.





Action

To troubleshoot the Java exception failures related to FTP, try one of these two options:

- Modify the settings of the firewall software running on the host computer. Add an exception for the IP addresses `192.168.3.2` and `192.168.2.2` on the firewall software.
- Clear the internal memory of Parrot minidrone (see "Troubleshooting by Clearing Parrot Minidrone Memory" on page 1-70).

See Also

"Troubleshooting by Clearing Parrot Minidrone Memory" on page 1-70

Troubleshooting by Clearing Parrot Minidrone Memory

Description

The internal memory of Parrot minidrone needs to be cleared if you encounter the following problems:

- FTP transfer between the drone and host computer fails either during firmware update or during the deployment of Simulink model to the drone.

  The error message shows the following Java Exception failures:

  ```
  at org.apache.commons.net.ftp.FTP.__getReply(FTP.java:297)
  at org.apache.commons.net.ftp.FTP.sendCommand(FTP.java:490)
  at org.apache.commons.net.ftp.FTP.sendCommand(FTP.java:534)
  at org.apache.commons.net.ftp.FTPClient._openDataConnection_(FTPClient.java:476)
  at org.apache.commons.net.ftp.FTPClient.__storeFile(FTPClient.java:374)
  at org.apache.commons.net.ftp.FTPClient.storeFile(FTPClient.java:1379)
  ```
- The communication over Bluetooth connection is very slow.
- The flight log indicates that a memory allocation error has occurred.
- The LEDs on the Parrot Rolling Spider minidrone do not turn green after you switch on the drone; the LEDs remain yellow even after a few seconds.

Action

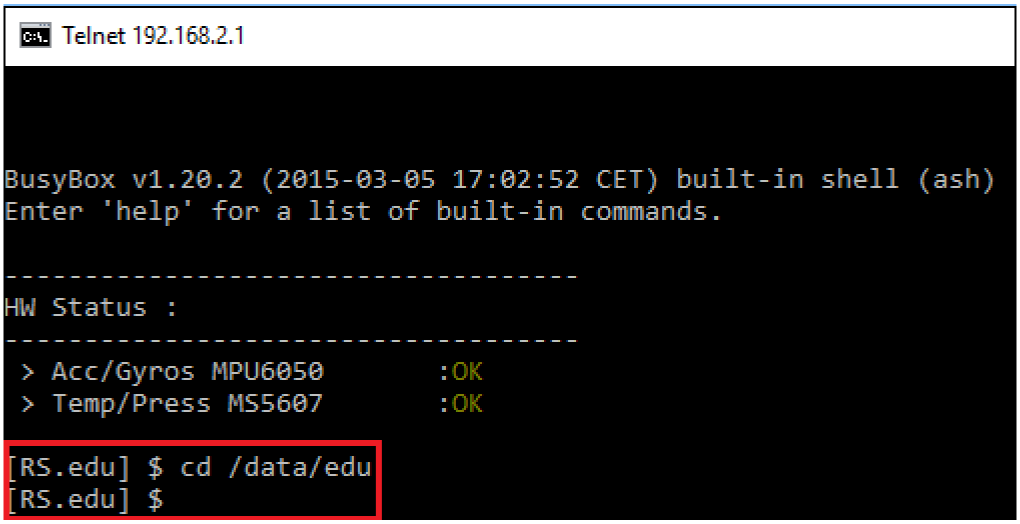
To clear the memory:

1. Connect the Parrot minidrone to the host computer using USB.
2. If the host computer is running on Windows, ensure that RNDIS is installed (see "Install RNDIS for Parrot Minidrones" on page 1-5).
3. Wait for the LEDs on the drone to stabilize to either green or yellow.
4. Open the command prompt on the host computer, and connect to the drone using Telnet:

   ```
   telnet 192.168.2.1
   ```

   If the Telnet command fails, see "Telnet Connection to Minidrone is Failing" on page 1-71.
5. After successful Telnet connection, go to `/data/edu` directory.

   ```
   cd /data/edu
   ```



6 Delete all the text files, MAT files, and shared object files.

```
rm *.txt
rm *.mat
rm *.so
```

7 Disconnect the USB connection between the drone and host computer.

Telnet Connection to Minidrone is Failing

If the Telnet connection to the minidrone is failing, first reboot the host computer and try the Telnet connection again. Ensure that the minidrone remains switched on while the Telnet connection is being established.

If the Telnet connection still fails, check the platform-specific errors described here.

Windows

If the host computer is running Windows and the Telnet connection to the drone fails, check for these error conditions:

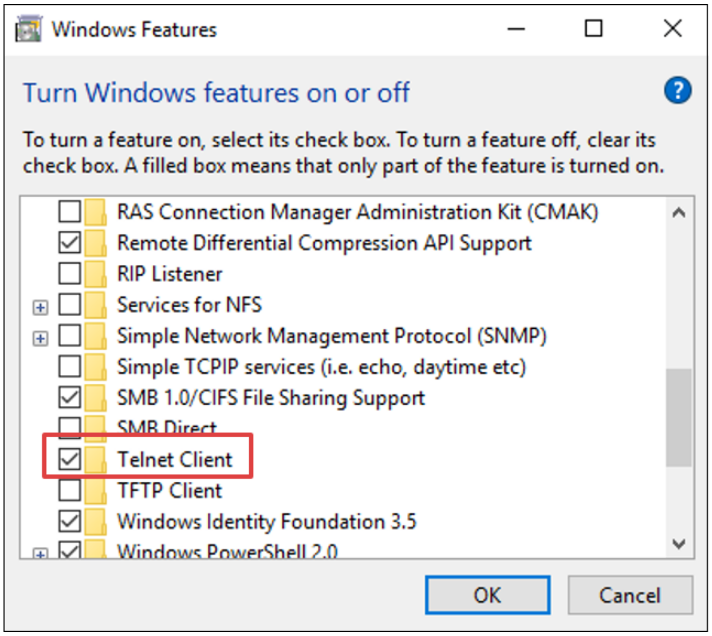
- **Error**: The command prompt displays the message "`Connecting to 192.168.2.1......`" for a long time.

  Action: Install RNDIS on the host computer (as described in "Install RNDIS for Parrot Minidrones" on page 1-5), and perform the setup and configuration again.
- **Error**: The command prompt displays the message "`'telnet' is not recognized as an internal or external command, operable program or batch file`".

  Action: Enable the Telnet Client on Windows:

  1 On the Search bar of Windows 10, search `Turn Windows features on or off`.

  2 Select **Telnet Client** and click **OK**.



Reinstall the RNDIS if you are using a different USB port than earlier to connect to the Parrot minidrone.

Mac

If the host computer is a Mac and the Telnet connection to the drone fails:

1 Reinstall the Support Package on the host computer.
2 Reboot the host computer.
3 Perform the setup and configuration activities again for Parrot minidrone.

Linux

If the host computer is running Linux and the Telnet connection to drone fails:

1 Disconnect any other USB mass-storage devices; ensure that the connection to the drone is the only USB connection from the host computer.
2 Open the Terminal.
3 Enter the following command:

```
sudo ifconfig usb0 192.168.2.2 up
```

See Also

"Troubleshooting Connection Settings in Linux" on page 1-78 | "Connect a Parrot Minidrone to a Linux System Using Bluetooth" on page 1-42

Minidrone Not Found in Bluetooth Scan Result

Description

The required Parrot minidrone is not found in the scan result of nearby Bluetooth-enabled devices, when you run the `parrot.util.ConnectDroneToBlueTooth` command from MATLAB on the Linux computer.

Command from MATLAB

```
parrot.util.ConnectDroneToBlueTooth
```

Sample

```
Unable to detect a Parrot Minidrone. Please ensure that the drone is powered on and try again.
```

Action

Check that the minidrone is switched on and both the LEDs on the minidrone are green. After these checks, follow the steps in “Connect to the Minidrone from MATLAB” on page 1-42.

See Also

“Troubleshooting Connection Settings in Linux” on page 1-78 | “Connect a Parrot Minidrone to a Linux System Using Bluetooth” on page 1-42

Bluetooth Connection to the Minidrone Fails

Description

The MATLAB command, `parrot.util.ConnectDroneToBlueTooth` (along with the proper arguments), fails to establish a Bluetooth connection from the Linux computer to the Parrot minidrone.

Action

Try these options sequentially, until the Bluetooth connection is re-established (using the procedure described in "Connect to the Minidrone from MATLAB" on page 1-42):

1. Check that the minidrone is switched on and both the LEDs on the minidrone are green.
2. If the minidrone was paired previously, remove any existing pairing in the system.
3. Ensure that there is no other Parrot minidrone connected to the host computer over Bluetooth. If any other minidrone is connected, switch-off that drone.
4. Check if the Bluetooth adapter on the Linux computer supports Bluetooth Low Energy 4.0 version. To find out if your adapter is Bluetooth Low Energy 4.0 compatible, see the instructions in "Finding the LMP Version of Bluetooth Adapter on Your Linux System" on page 1-77.
5. Disconnect the Bluetooth Low Energy adapter and reconnect it after a while to the host computer.
6. Turn-off and turn-on the Bluetooth support on the Linux computer.

After you try these options, reboot the minidrone, wait until both LEDs on the minidrone turn green, and follow the steps in "Connect to the Minidrone from MATLAB" on page 1-42.

See Also

"Troubleshooting Connection Settings in Linux" on page 1-78 | "Connect a Parrot Minidrone to a Linux System Using Bluetooth" on page 1-42

Personal Area Networking (PAN) Connection Fails

Description

The PAN connection between your computer and the minidrone fails.

Action

Check these settings and follow the steps in "Connect to the Minidrone from MATLAB" on page 1-42:

1. The minidrone is switched on and both LEDs on the minidrone are green.
2. The Bluetooth support on your computer is turned on.

See Also

"Troubleshooting Connection Settings in Linux" on page 1-78 | "Connect a Parrot Minidrone to a Linux System Using Bluetooth" on page 1-42

Ping Command Fails or Bluetooth Connection Unsuccessful

Description

The ping command fails because of no reply from your computer.

Command

```
ping -c 4 -S 192.168.3.2 192.168.3.5
```

Sample

```
PING 192.168.3.5 (192.168.3.5) 56(84) bytes of data.
---  192.168.3.5 ping statistics  ---
4 packets transmitted, 0 received, 100% packet loss
```

Action

Try these options sequentially until the minidrone appears in the scan result:

1. Check these settings and follow the steps in “Connect to the Minidrone from MATLAB” on page 1-42:

   a. The drone battery is charged.

   b. The `<bluetooth interface IP address>` set using the `sudo ifconfig bnepx <bluetooth interface IP address> up` command is not assigned to any other interface.

      If the IP address is already assigned to any other interface, specify a new Bluetooth interface IP address according to these guidelines:

      - The subnet address, typically the first 3 bytes of the interface IP address, must be the same as the first 3 bytes of the drone IP address. The drone IP address for Rolling Spider is `192.168.3.5` by default.
      - The last byte of the interface IP address must be different from the last byte of the drone IP address.

2. Restart the minidrone, wait until both LEDs on the minidrone turn green, and follow the steps in “Connect a Parrot Minidrone to a Linux System Using Bluetooth” on page 1-42.

See Also

“Troubleshooting Connection Settings in Linux” on page 1-78 | “Connect a Parrot Minidrone to a Linux System Using Bluetooth” on page 1-42

Finding the LMP Version of Bluetooth Adapter on Your Linux System

Description

On Linux, you want to know if your Bluetooth adapter is Bluetooth Low Energy 4.0 compatible.

Action

1 To find the version of Bluetooth adapter on your Linux, open the terminal and use this command:

```
sudo hcitool -a
```

2 Find **LMP Version**. If the version is `0x6` or higher, your system is compatible with Bluetooth Low Energy 4.0. Any version lower than that indicates an older version of Bluetooth.

See Also

"Troubleshooting Connection Settings in Linux" on page 1-78 | "Connect a Parrot Minidrone to a Linux System Using Bluetooth" on page 1-42

Troubleshooting Connection Settings in Linux

This section explains how to solve problems that you might encounter when connecting the Parrot minidrone to Linux system using Bluetooth.

- “Adapter Not Recognized by Computer” on page 1-67
- “Minidrone Not Found in Bluetooth Scan Result” on page 1-73
- “Finding the LMP Version of Bluetooth Adapter on Your Linux System” on page 1-77
- “Bluetooth Connection to the Minidrone Fails” on page 1-74

See Also

“Connect a Parrot Minidrone to a Linux System Using Bluetooth” on page 1-42

Run on Target Hardware

Important Test Flight Considerations

Before you run Simulink models on Parrot minidrones, consider the general safety precautions and the dependencies on the surrounding objects that may affect the flight stability.

All Flights Are Test Flights

All the flights that you design with Simulink Support Package for Parrot Minidrones are test flights. You must adhere to the following safety precautions:

- Test the flight controller designs and inputs using simulation so that they meet the performance criteria.
- Perform test flights in a room that is large enough – typically, 20 x 20 x 10 feet (length x width x height).
- Before flying the minidrone, take reasonable care to prevent damage to the minidrone and the surroundings (including the people, animals, and property in the vicinity of the flight).

Flight Stability Based on Ultrasound Sensitivity

If the data received from ultrasonic sensors on Parrot minidrone is incorrect, the flight stability is affected. Therefore, consider the ultrasound sensitivity while flying the drone:

- The ultrasound signals bouncing on walls and furniture of a small room may cause instability of the flight.
- Ground materials (for example, carpet) may absorb ultrasound signals from the minidrone and affect flight stability.

Flight Stability Based on Camera Sensitivity

If you enable optical flow in the Simulink model, consider the sensitivity of Parrot minidrone's vertical camera:

- Patterns on the ground captured through the camera may affect optical flow performance and flight stability.
- Lighting conditions may also affect optical flow and image processing performance and affect flight stability.

See Also

“Code Generation Simulink Template for Parrot Minidrone” on page 2-3 | “Fly a Parrot Minidrone Using the Quadcopter Simulink Model” on page 2-27

Code Generation Simulink Template for Parrot Minidrone

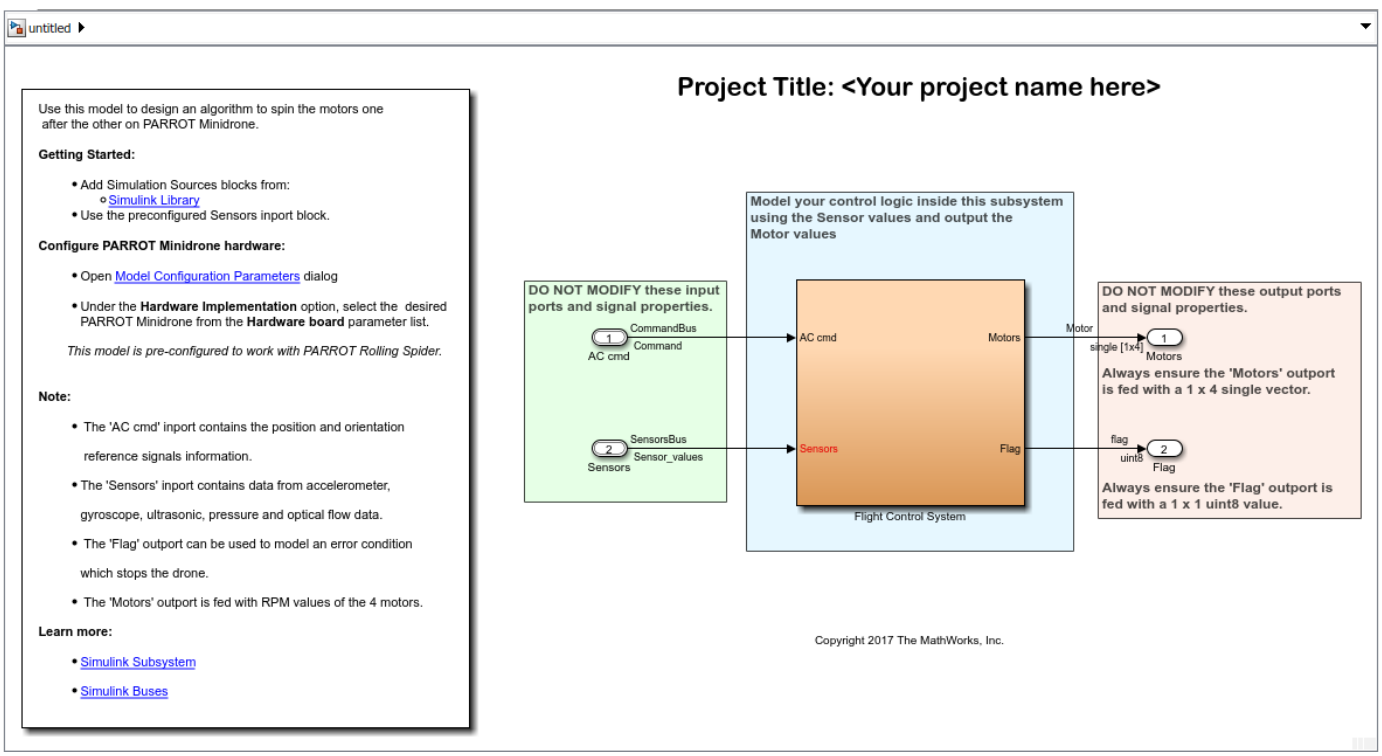
After you have completed “Setup and Configuration”, use the Code Generation Simulink template to create a model to be deployed on your Parrot minidrone hardware. This template is designed to spin the four motors on the drone cyclically for 2 seconds each. You can update this template to add your own logic to start the drone.

Open the Code Generation Template

1. At the MATLAB command prompt, type `simulink`.
2. On the Simulink Start Page, navigate to **Simulink Support Package for Parrot Minidrones**, and select **Code Generation Template**.

   A Simulink model opens. This model is configured for code generation and contains two inports, two outports, and a subsystem block. For more information, see “Inports in Code Generation Template” on page 2-3 and “Outports and Signals in Code Generation Template” on page 2-6.

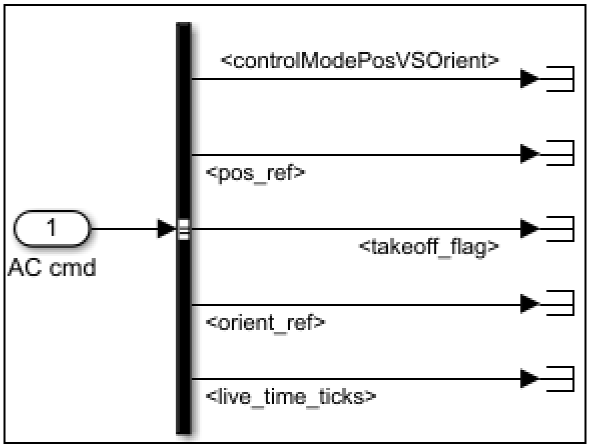
Note The **Code Generation Template** is available only if you have installed Simulink Support Package for Parrot Minidrones. For more information, see “Setup and Configuration”.



Inports in Code Generation Template

The Code Generation Template model contains two inport blocks, `AC cmd` and `Sensors`:

- AC cmd – This inport is connected to a Bus Selector block that provides five output signals:

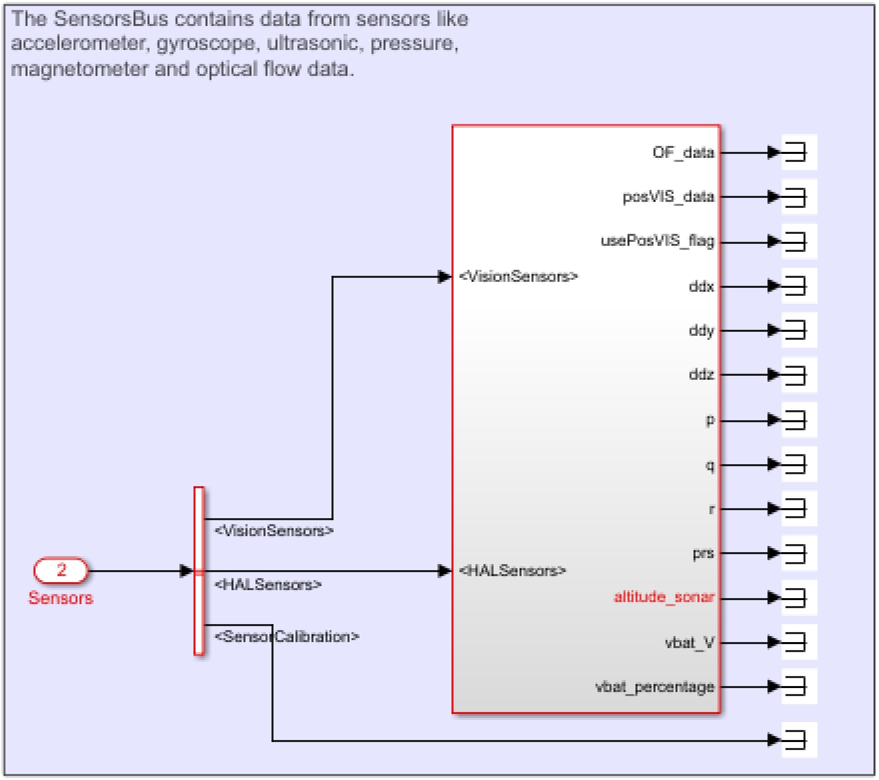


1. <controlModePosVSOrient> – Checks for the presence of pitch and roll values in the orient_ref signal. This signal is set to true if both pitch and roll values are zero.

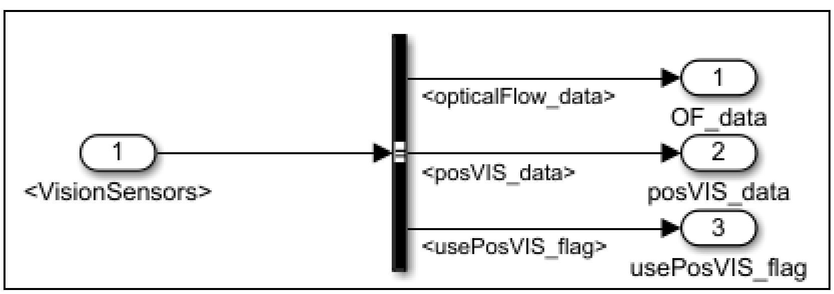
   The <controlModePosVSOrient> signal can be used to determine if the controller logic that drives the drone uses orientation or position control. If one or both of the pitch value or roll value is present, the drone will fly based on those orientation values only, and it will ignore any reference to the desired position of the drone specified using the <pos_ref> signal.
2. <pos_ref> – An array representing *x*-, *y*-, and *z*-axes values (in meters) of the desired position of the drone.
3. <takeoff_flag> – A flag that is set to true for 1 second. After you execute the start command, there is a calibration period for 2 seconds, after which the <takeoff_flag> is set to true for 1 second. This flag is disabled after that 1-second period. This flag can be used to include any additional logic that needs to be considered when the motors on the drone start (for example, to separate the logic required for take-off and hovering).
4. <orient_ref> – An array representing the yaw, pitch, and roll values (in radians per second) that determine the orientation of the drone.
5. <live_time_ticks> – This signal represents a continuous counter that is incremented at the rate of 5 ms (200 Hz frequency). This counter is present in the control code that is running in the drone at every 5 ms. In other words, the <live_time_ticks> value is incremented by 200 each second.

Note In the Code Generation Template, the signals from the AC cmd block are terminated. However, you can connect the signals when you design a model based on the template.

- Sensors – This inport is connected to a Bus Selector that provides output signals that are derived from the different sensors on the drone. The Sensors inport is further divided into VisionSensors, HALSensors, and SensorCalibration.



- `VisionSensors` connects to a Bus Selector block that provides three output signals:



1 `<opticalFlow_data>` – Obtains the values of horizontal displacement of the drone based on optical flow data. This signal can be used to estimate the speed of the drone.

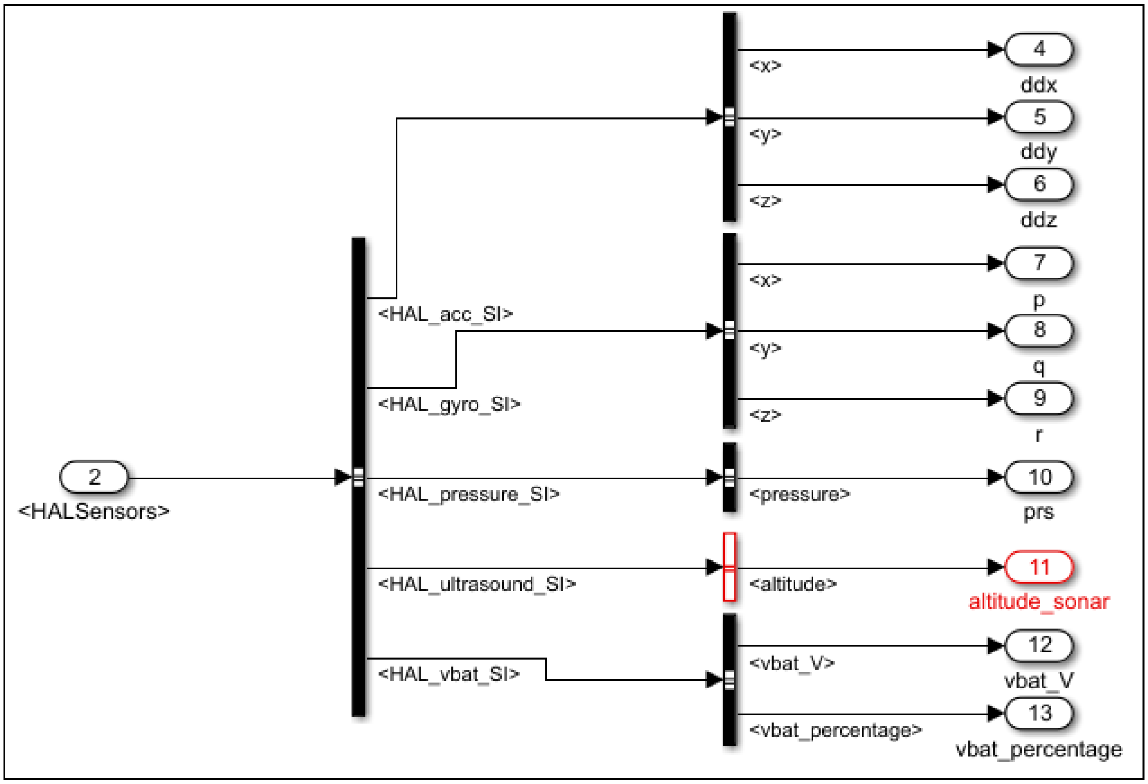
2 `<posVIS_data>` – Data from the image processing thread that is running inside the drone, which in turn is based on the data from the camera on the drone.

Currently, the data from the image processing thread is not passed to the `<posVIS_data>` signal, so this signal returns zero values.

3 `<usePOSVIS_flag>` – A flag that determines whether the `<posVIS_data>` signal is to be used in the controller logic or not.

Currently, the `<usePOSVIS_flag>` is always set to false.

- `HALSensors` connects to a Bus Selector block that derives values from the Hardware Abstraction Layer (HAL) of the drone. The signal values are obtained from the accelerometer, gyroscope, pressure sensor, ultrasonic sensor, and battery:



1 `<HAL_acc_SI>` – An array representing the acceleration of the drone (in m/s^2) along *x*-, *y*-, and *z*-axes.

2 `<HAL_gyro_SI>` – An array representing the angular velocity of the drone (in radians per second) along *x*-, *y*-, and *z*-axes.

3 `<HAL_pressure_SI>` – Atmospheric pressure (in pascals) measured by the drone.

4 `<HAL_ultrasound_SI>` – Altitude of the drone (in meters) measured using sonar.

5 `<HAL_vbat_SI>` – An array representing the voltage of the battery (in volts) and the percentage of charge remaining.

- `SensorCalibration` provides calibrated values of the different sensor values. The calibration is performed over 400 cycles (2 seconds), and you can obtain eight values that include:

  - calibrated *x*, *y*, and *z* values of acceleration
  - calibrated *x*, *y*, and *z* values of angular velocity
  - calibrated atmospheric pressure
  - calibrated altitude

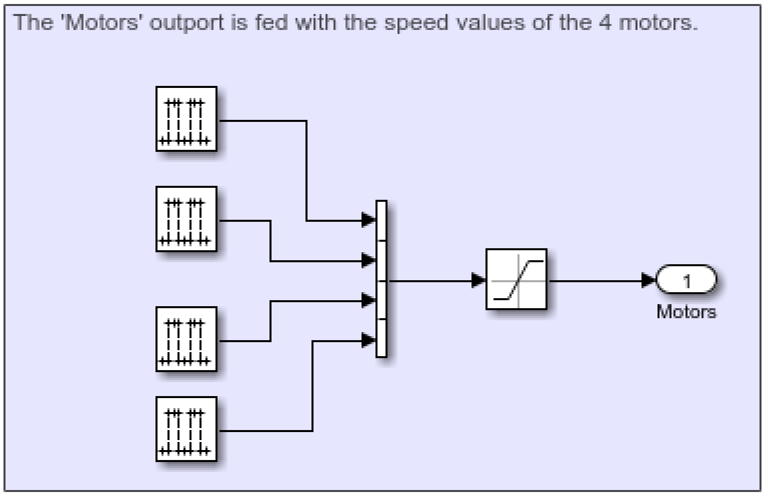
Outports and Signals in Code Generation Template

The outports in the Code Generation template are:

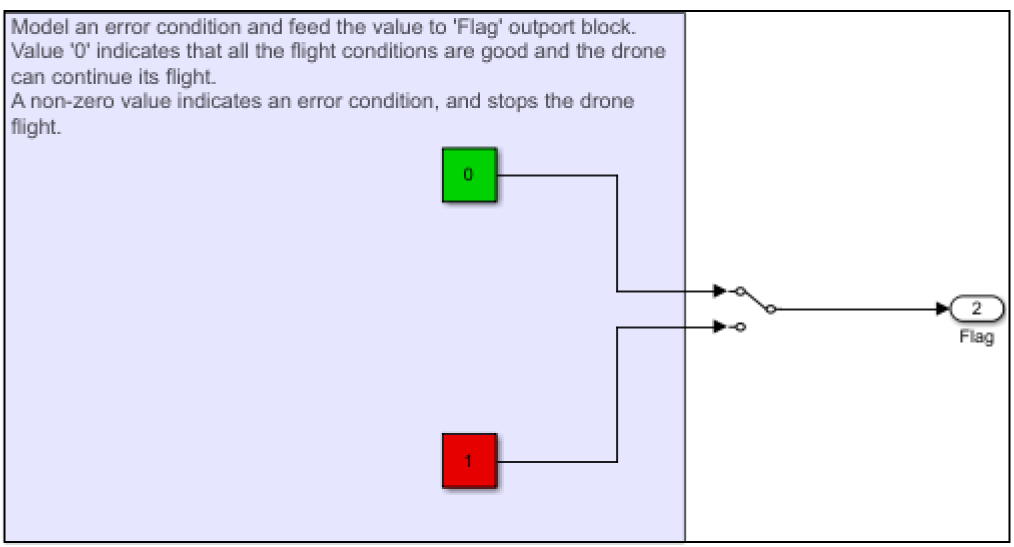
- `Motors` – This outport is used to send signals to start the four motors on the drone.

  In the Code Generation template, four Pulse Generator blocks are used to send signals to the four motors. The amplitude values in these blocks are set to 400, which spins the motors at the corresponding speed. A phase shift is also provided in the Pulse Generator so that the four motors spin cyclically for 2 seconds each.

When you design your own controller logic, the signals to the motors can be derived from the output of controller logic. (Use 1-by-4 vector as the input to the `Motors` outport.)



- `Flag` – This outport is used to set error conditions that can be used to stop the flight of the drone (stop the motors). A value of 0 indicates that there are no errors, and any nonzero value indicates an error. You can model multiple error conditions in the controller logic that correspond to multiple nonzero values that can be fed as input to this flag.



For details about setting the hardware and deploying the model, see “Setting Up the Hardware and Deploying the Model” on page 2-12.

See Also

“Setting Up the Hardware and Deploying the Model” on page 2-12 | “Flight Simulation Simulink Template for Parrot Minidrone” on page 2-25

Code Generation Simulink Template for Image Processing for Parrot Minidrone

After you have completed “Setup and Configuration”, you can use the Code Generation Simulink Template for Image Processing to create a model that processes the image from the downward-facing camera of Parrot minidrone. When you deploy the model on a Parrot minidrone, the output from the image-processing algorithm is an additional input to control the drone's flight. You can update the template to add your own logic.

Open the Code Generation Template for Image Processing

1. At the MATLAB command prompt, type `simulink`.
2. On the Simulink Start Page, navigate to **Simulink Support Package for Parrot Minidrones**, and select **Code Generation Template for Image Processing and Control**.

   A Simulink model opens. This model is configured for code generation, and it uses image processing as an additional input to control the drone's flight. The template contains two subsystem blocks – `Image Processing System` and `Flight Control System` – with corresponding input ports and output ports. For more information, see “Image Processing System” on page 2-8 and “Flight Control System” on page 2-10.

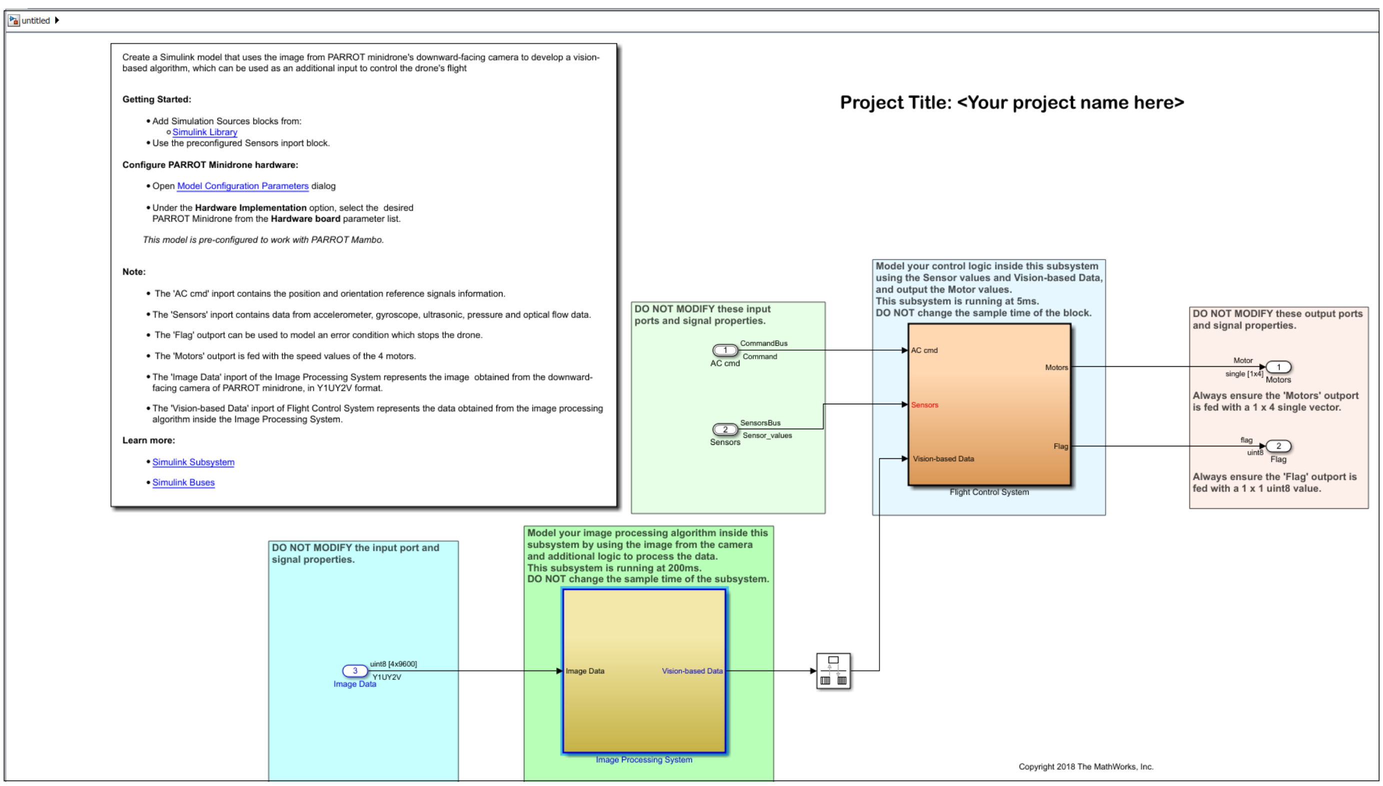


Image Processing System

You can use the `Image Processing System` to develop an image-processing algorithm. This subsystem contains an input port, an output port, and an optional block:

- **Image Data** – This input port obtains data from the downward-facing camera of a Parrot minidrone, in Y1UY2V format, as a 4-by-9600 array of type `uint8`.
- **Vision-based Data** – This output port transfers the processed data based on the image-processing algorithm modeled in the subsystem.
- PARROT Image Conversion – This is an optional block that helps you to convert the **Image Data** (in Y1UY2V format) to either YUV or RGB format. There are three outputs for this block, which correspond to the three color components of the converted image. Each color component is a 120-by-160 array of type `uint8`. You can use these output color components to create an image-processing algorithm, and then connect the output of the algorithm to the **Vision-based Data** output port of the subsystem.

Model an Image-Processing Algorithm in the Subsystem

You can model an image-processing algorithm in the `Image Processing System` by using the image obtained from Parrot minidrone's downward-facing camera. To create a typical image-processing algorithm:

1. Extract features of interest from the input image, which is captured by the drone's camera every 200 ms. For example:

   - Use the color components – Convert the input image, which is already in Y1UY2V format (YUV422), to another color space format that has color components that are simpler to use (for example, RGB888 or YUV444).
   - Identify the shape of objects in the image – Use methods like corner detection, template matching, and so on.
2. Use the output from image extraction to perform further image analysis. For example:

   - Compute the statistics of the colored region in the image.
   - Set threshold values that help to identify certain patterns on the image.
3. Based on the output from the final image analysis, set the values that can be used as input to control the position and orientation of drone. For example, you can use a stateflow in the same subsystem to design logic that uses the data from image analysis to further generate the pitch and roll values that control the drone. The final values are passed to the `Flight Control System` using the **Vision-based Data** output port.

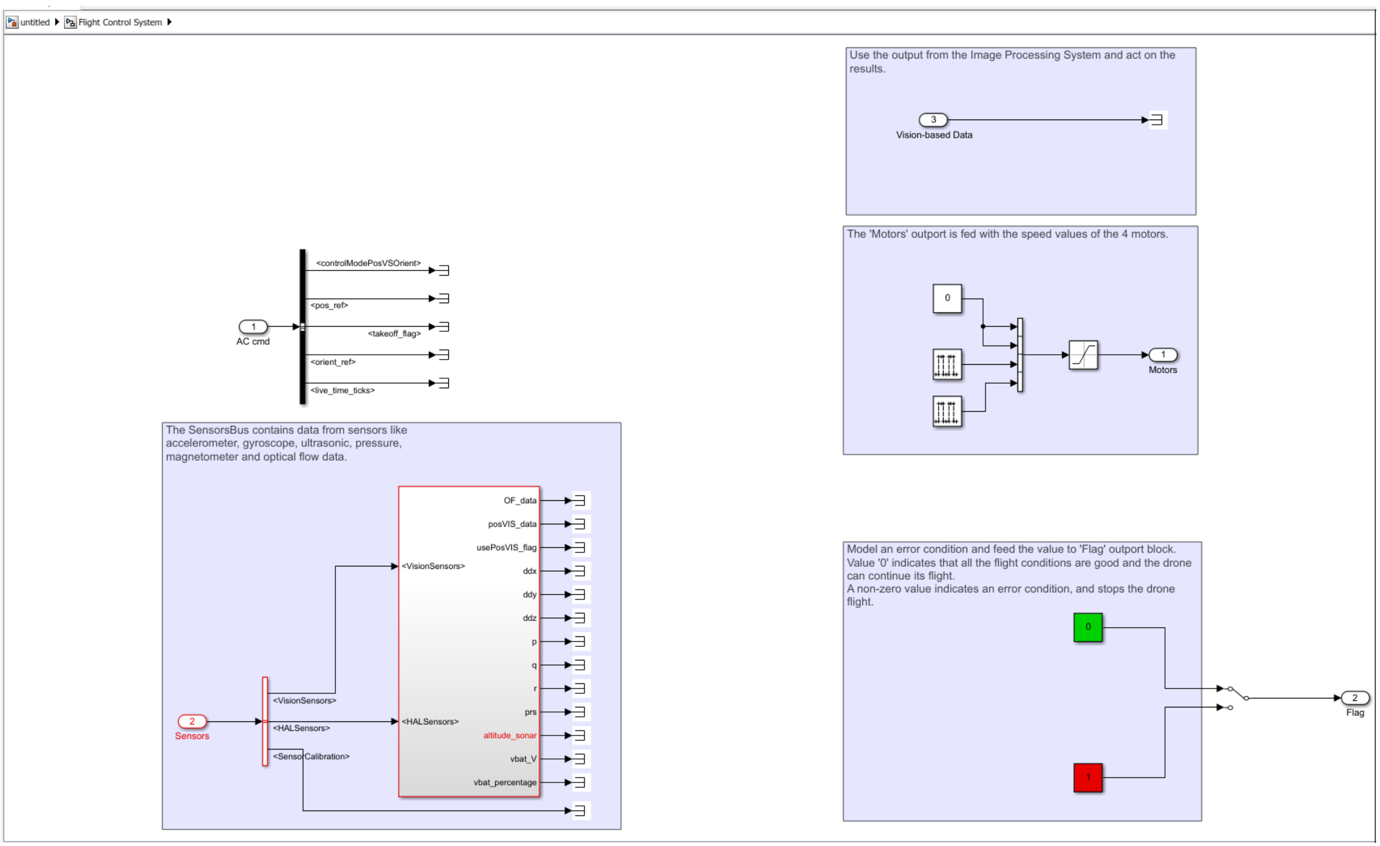
Note In the template, the Sample Time of the `Image Processing System` subsystem is 200 ms. Do not change this Sample Time.

Flight Control System

`Flight Control System` is the main subsystem that integrates the input data from the `Image Processing System` and the data from the different sensors on the drone. The control logic modeled in this subsystem starts the motors on the drone, controls the drone's flight, and stops the drone.

The Flight Control System subsystem contains three input ports – **AC cmd**, **Sensors**, and **Vision-based Data**. For details about **AC cmd** and **Sensors** input ports, see “Inports in Code Generation Template” on page 2-3.

The **Vision-based Data** input port obtains data from the Image Processing System. For details, see “Image Processing System” on page 2-8.



In the template, **Vision-based Data** in the Flight Control System subsystem is terminated. However, if the **Vision-based Data** signal contains the appropriate values, you can use it for flight control by connecting that same as input to feed values to the **Motors** output port.

The output ports in the Code Generation template are:

- **Motors** – This output port is used to send signals to start the four motors on the drone.

  In the Code Generation Template for Image Processing and Control, the input speed values for two of the four motors are set to zero. For the other two motors, two Pulse Generator blocks are connected to send signals. The amplitude values in these blocks are set to `400`, which spins the motors at the corresponding speed. A phase shift is also incorporated into the Pulse Generator so that the two motors spin cyclically for 2 seconds each.

When you design your own controller logic, each of the signals to the motors can be derived from the output of controller logic or from the **Vision-based Data**. (Use 1-by-4 vector as the input to the **Motors** output port.)

- **Flag** – This output port is used to set error conditions that can be used to stop the flight of the drone (stop the motors). A value of `0` indicates that there are no errors, and any nonzero value indicates an error. You can model multiple error conditions in the controller logic with multiple nonzero values that can be fed as input to this flag.

  Note Do not use the values `1`, `69`, `88`, and `99` for setting error conditions using the Flag output port (these values are reserved for other error conditions).

Note In the template, the Sample Time of the `Flight Control System` subsystem is 5 ms. Do not change this Sample Time.

For details about setting up the hardware and deploying the model, see “Setting Up the Hardware and Deploying the Model” on page 2-12.

See Also

“Setting Up the Hardware and Deploying the Model” on page 2-12 | “Flight Simulation Simulink Template for Parrot Minidrone” on page 2-25

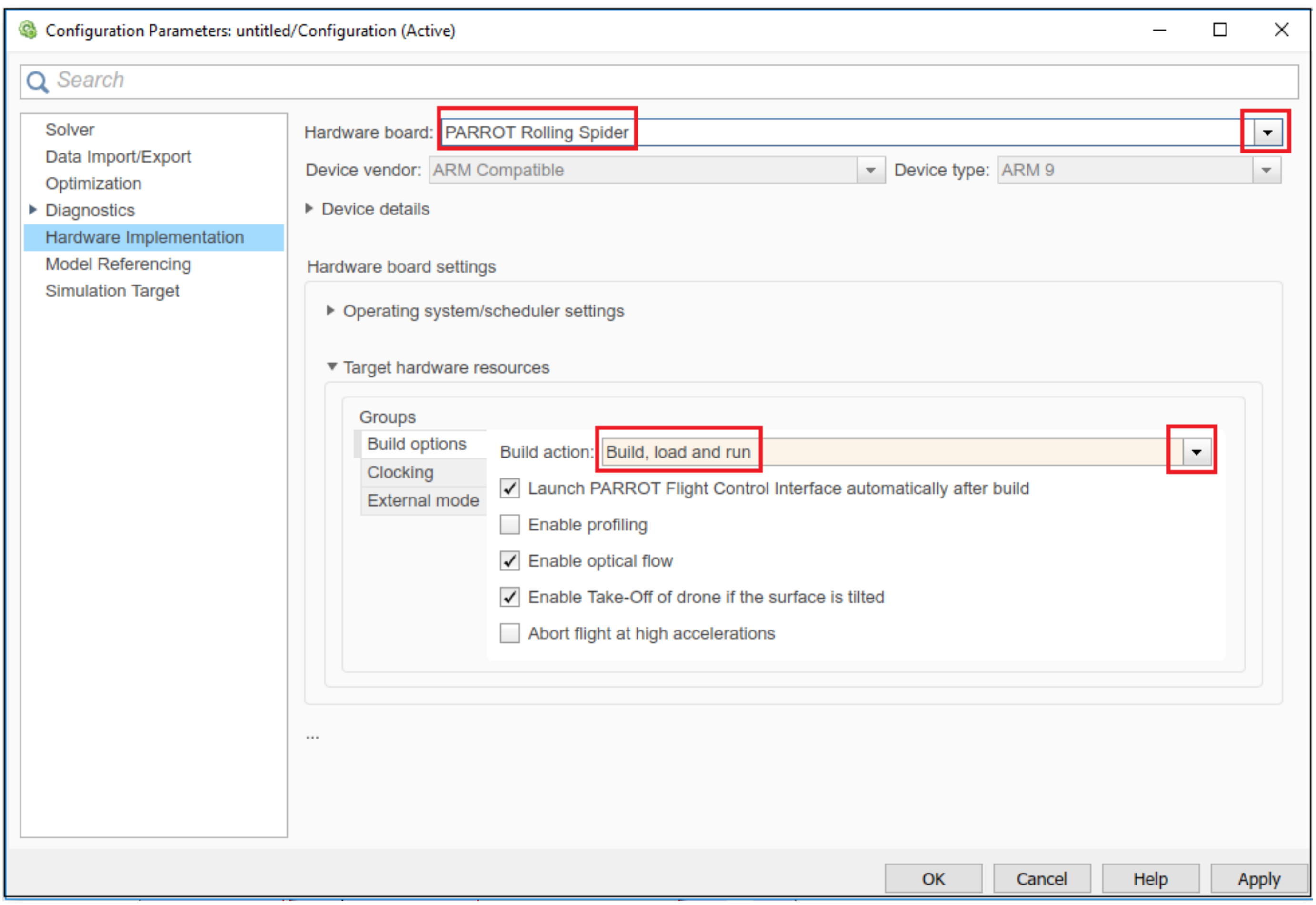
Setting Up the Hardware and Deploying the Model

Set up the hardware board from Simulink and deploy the model on the Parrot minidrone hardware.

Note Before you set up the hardware from Simulink, connect the Parrot minidrone to the host computer over Bluetooth (see "Setup and Configuration").

Open Model Configuration Parameters

1. In the Simulink model, go to the **Modeling** tab and click **Model Settings**.
2. In the **Configuration Parameters** dialog box, select **Hardware Implementation**, and set the **Hardware board** to `Parrot Rolling Spider` or `Parrot Mambo`, based on the type of minidrone you have connected.
3. In the **Build Options**, select the `Build, load and run` option.
4. Click **Apply** and then **OK** to save your changes.



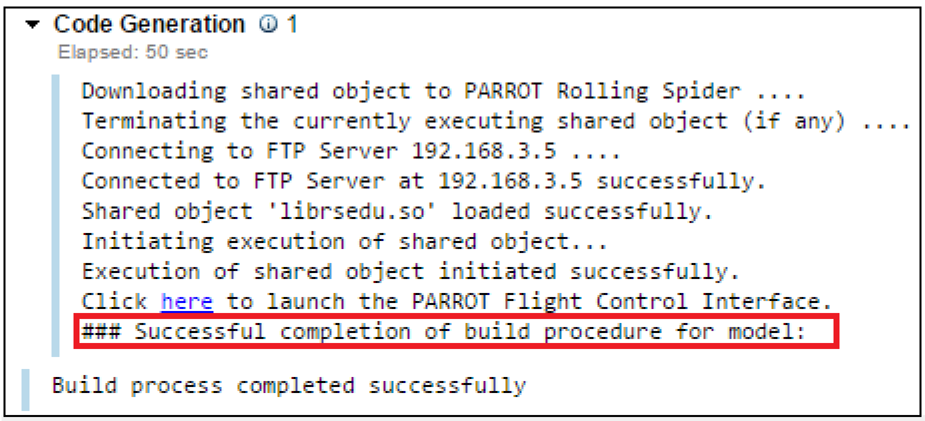
For details about the options in the Hardware Implementation pane, see "Model Configuration Parameters for Parrot Minidrone".

Note Additionally, if you are using image-processing algorithm in the Simulink model, multitasking mode needs to be enabled. Therefore, before deploying such a model, open the Configuration Parameters dialog box, navigate to the Solver pane, and select the option **Treat each discrete rate as a separate task**.

Deploy Simulink Model

To deploy the Simulink model, go to the **Hardware** tab and click **Build, Deploy & Start**. The Diagnostic Viewer displays the progress of the build.

After the build process and deployment is successful, the Diagnostic Viewer displays the following message:



After you have successfully deployed the model, you can use the Flight Control Interface to start the drone (see "Using the Flight Control Interface to Start the Drone" on page 2-14).

See Also

"Code Generation Simulink Template for Parrot Minidrone" on page 2-3 | "Flight Simulation Simulink Template for Parrot Minidrone" on page 2-25

Using the Flight Control Interface to Start the Drone

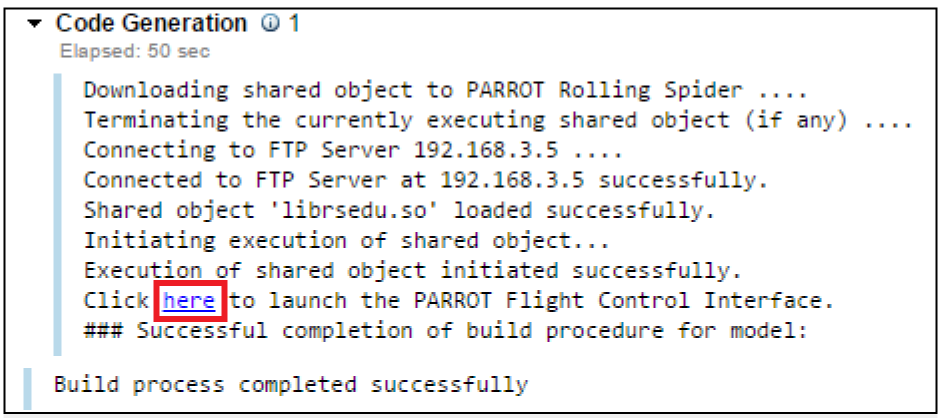
After you have successfully deployed the Simulink model on the Parrot minidrone, you can use the Flight Control Interface to start and stop the motors on the drone. The Flight Control Interface also provides additional options to control the drone flight, and it also displays some of the errors (if any) after the drone starts.

Note The Parrot Flight Control Interface uses port `24099` for TCP/IP communication with the drone to obtain its status and errors (if any). Therefore, do not use port `24099` for other TCP/IP and UDP communication using the TCP/IP and UDP blocks in the Simulink model.

Step 1: Open the Flight Control Interface

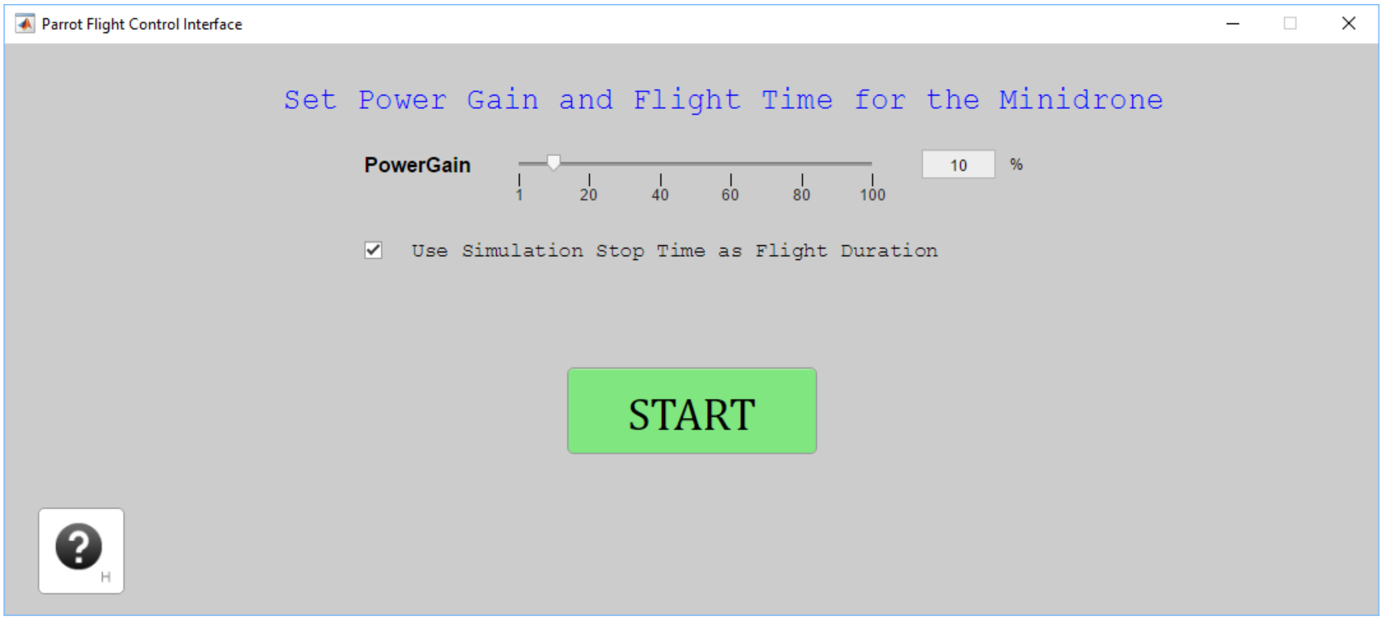
The Flight Control Interface opens automatically after the deployment of the model on Parrot minidrone is successful, if you had selected the option **Launch Parrot Flight Control Interface automatically after build** in the Configuration Parameters dialog box in Simulink. However, you can also open this interface by one of these options:

- Click the hyperlinked text (`here`) available in the Diagnostic Viewer at the end of a successful deployment.

- At the MATLAB command prompt, type:

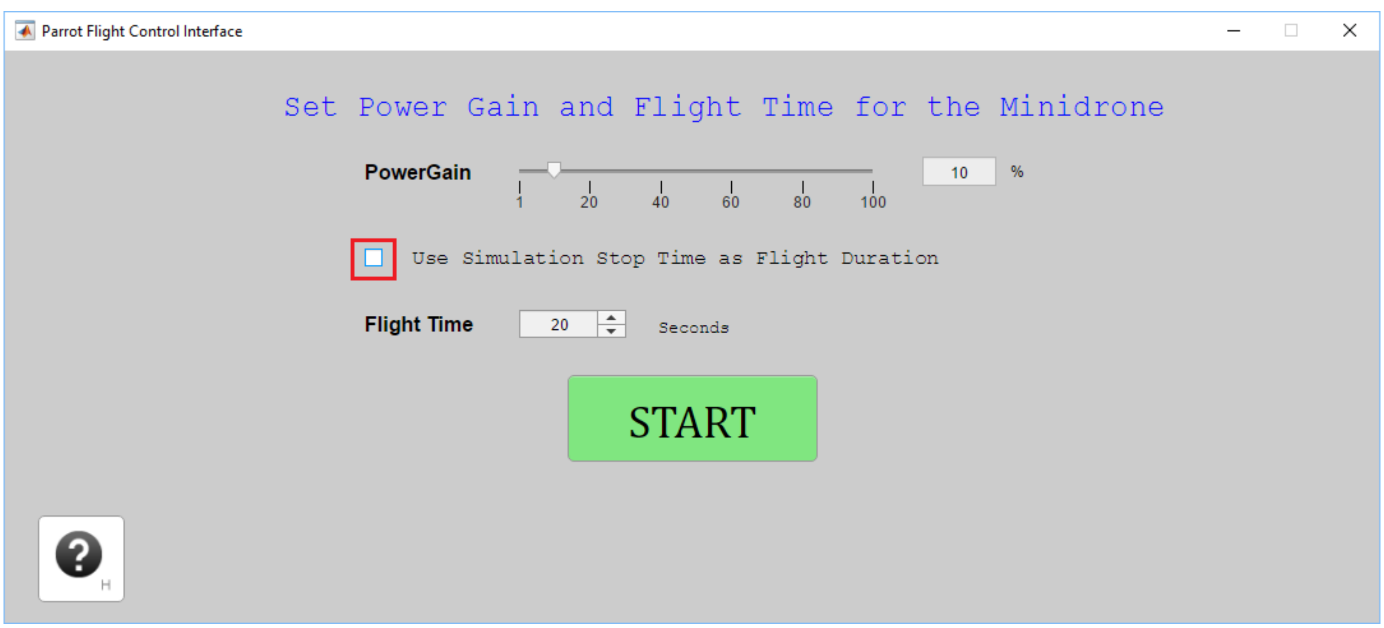
  ```
  Parrot_FlightInterface
  ```



Step 2: Start Parrot Minidrone Motors

1 Drag the **PowerGain** slider to change the power gain of motors. For example, if you want the speed of motors to be 20% of the value that is set in the controller logic, drag the slider to `20`. By default, the power gain is set at 10%. For an initial test, a lower power gain is always recommended.

2 By default, the duration of flight is set to the Simulation Stop Time of the Simulink model. However, you can change the flight duration in the Flight Control Interface. To do this, clear the **Use Simulation Stop Time as Flight Duration** check box and specify a value in the **Flight Time** field.

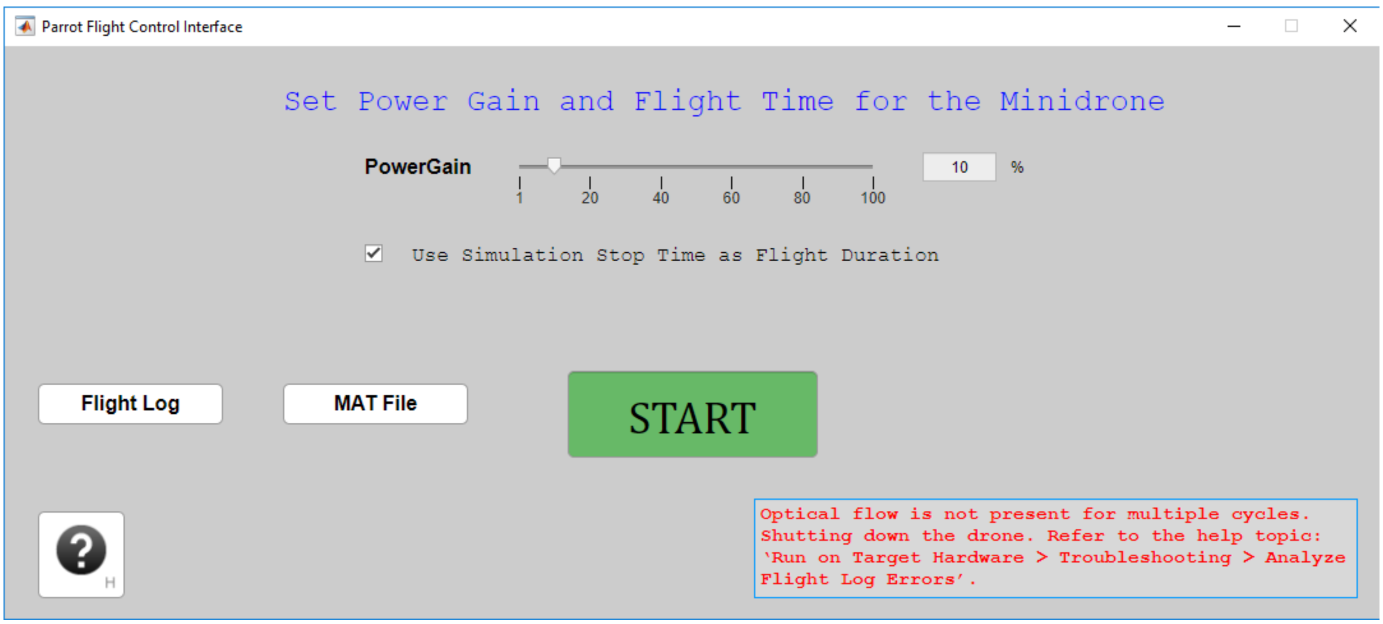


3 Click **START**. The motors on the Parrot minidrone start after a few seconds.

Note The motors do not start if there is a Bluetooth connection error between the Parrot minidrone and the host computer.

The motors stop after the flight duration time is reached.

The Flight Control Interface also displays some of the errors, if any, that resulted in a shutdown of the drone after you had clicked the **START** button. For example, see the below image that shows the interface after the drone shut down due to optical flow errors:



Land the Parrot Minidrone or Force Stop the Motors

When the motors on the Parrot minidrone start, the Flight Control Interface changes to display the **LAND** and **STOP** buttons:

- To land the drone anytime while its is flying, click **LAND**. The drone descends slowly along the z-axis, maintaining the current X-Y coordinates. Once the drone reaches a height of 0.3 meters above the ground, the Flag is set to 255, the model stops running, and no error is displayed on the Parrot Flight Interface.

  Note The **LAND** button appears in the interface only for Simulink projects that fly the drone. This button does not appear for example models and templates that does not take-off the drone.

- For an emergency shutdown of the drone (force stop the motors before the completion of the flight duration), click **STOP**. The motors of the drone immediately shut down and the drone falls to the ground.

After the motors stop (either after you click the **LAND** button or when the flight duration time is reached), proceed with the next step, “Using the Flight Control Interface to Obtain the Log Files” on page 2-18.

Note The detection of optical flow errors is disabled after you click the **LAND** button.

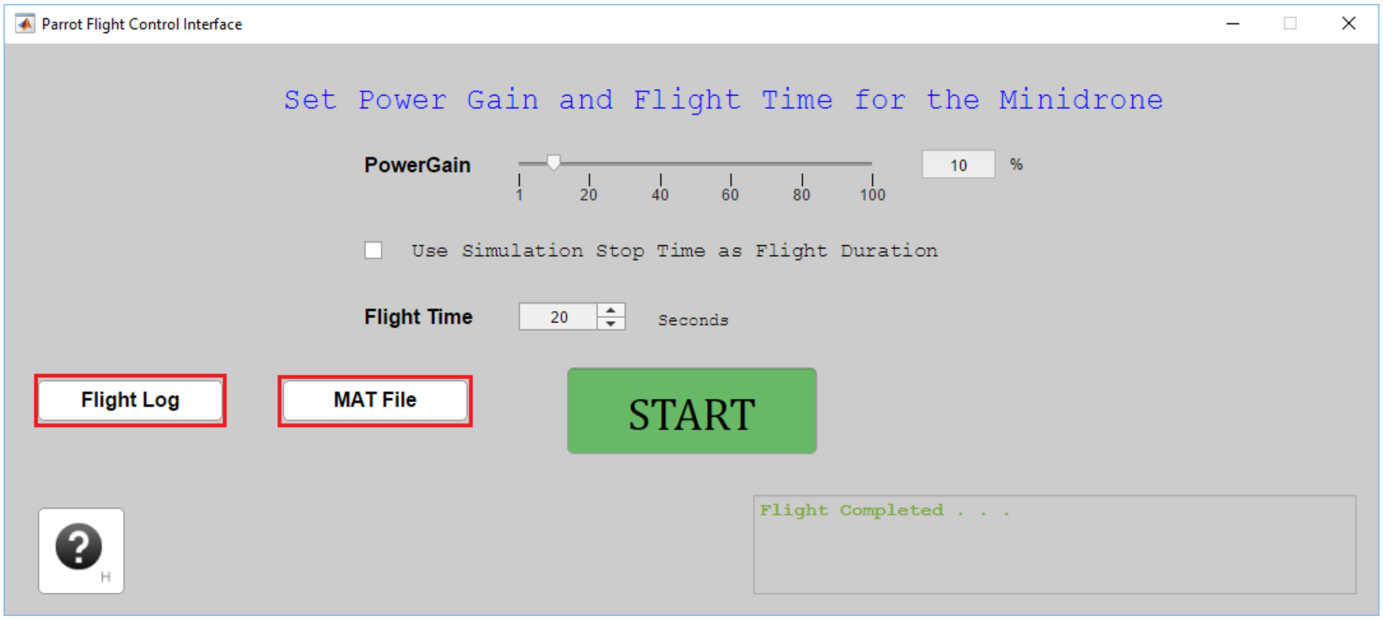
See Also

“Using the Flight Control Interface to Obtain the Log Files” on page 2-18

Using the Flight Control Interface to Obtain the Log Files

Obtaining the Flight Log and MAT File

The Flight Control Interface displays the option to retrieve the flight log and MAT-file after the you click **STOP**.



- Flight log: Click **Flight Log**. The file `droneFlight.txt` is downloaded to the Current Folder in MATLAB.
- MAT-file: Click **MAT File**. The file `RSdata.mat` is downloaded to the Current Folder in MATLAB

Note

- The existing flight log and MAT-file in the Current Folder are replaced everytime you download the files.
- To obtain the MAT-file, Simulink Coder™ must be installed on the host computer. Additionally, you must enable the options **Use Simulink Coder features** and **MAT-file logging** in the Configuration Parameters dialog box in Simulink model (see Simulink documentation).

Flight Log

The flight log (`droneFlight.txt`) is retrieved from the Parrot minidrone after the flight. This log file contains details and data captured from three threads — `rsedu_control` (the main control thread running at 200 Hz), `rsedu_of` (optical flow thread running at 60 Hz), and `rsedu_vis` (image processing thread running at 60 Hz). The events are logged every second, and they include information about:

- Threads that are executing on the drone and initialization checks

```
rsedu_of(): SUCCESS optical flow FIFO exists!
rsedu_vis(): Init fifo-communication...
rsedu_control(): Waiting for connection request from host!
```

- Creation of a successful TCP/IP connection between the drone and the host computer. (After initialization, the drone is the TCP/IP server, and it waits for connection from the host computer, which is the TCP/IP client.)
- All the sensor values captured and logged. These values are captured at every second until the end of the flight, and they include:
  - Altitude values measured from the sonar
  - Pressure measured from the pressure sensors
  - *x*, *y*, and *z* angular velocity values measured from the gyroscope
  - *x*, *y*, and *z* acceleration values measured from the accelerometer
  - Battery output voltage
  - Motor values

```
 **** Sensor Values when counter = 1400 ****
rsedu_control(): sonar altitude: 0.961149meters, pressure: 91356.406250pascal, gyrotemp: 57.685883celsius
rsedu_control(): gyro x: -0.514551rad/s, y: 0.100112rad/s, z: 0.018099rad/s
rsedu_control(): accel x: 1.630244m/s2, y: -0.372977m/s2, z: -12.451280m/s2
rsedu_control(): Battery output voltage Integrated:  3.61V - 76 percents
rsedu_control(): of: 0.003230 0.000430 0.000000
rsedu_control(): motor values : 303.670013 -267.161957 308.328613 -304.495789
```

- Initialization of data calibration

```
rsedu_control(): Calibrating Data . . .
rsedu_of(): flow: (0.000029, 0.000028, 0.000000, 1.000000, 0.500000)
rsedu_control(): Batterylevel: 100.000000
rsedu_control(): Sensorcal: 0.587689 :: -0.118883 :: -12.936934 :: -0.524371 :: 0.116673 :: 0.012041 :: 91360.812500 :: 0.000000
```

- Optical flow values

```
rsedu_of(): flow: (-0.001362, -0.000360, 0.000000, 1.000000, 0.500000)
```

- Execution of commands based on the controller logic or based on any user actions while the drone is flying (like initiating a stop)

```
 Emergency Landing Sequence Initiated !!
rsedu_control(): Flight abort request: shutting down motors now
rsedu_control(): Flight Completed after 1608 cycles ... DONE
```

- Errors, if any, that led to the stopping of motors. For details, see “Analyze Flight Log Errors” on page 2-21.

```
rsedu_control(): Flight crash about to happen, mismatch optical flow (-0.001362, -0.000360) and state estimate
 (27.352, -32.213): shutting down motors now
```

MAT File

The MAT-file (`RSdata.mat`) contains data that is captured every 5 milliseconds from the time the start command is executed on the drone until the motors on the drone stop. The data is captured in various workspace variables which you can analyze with MATLAB.

Note Before deploying the Simulink model to the drone, you need to connect the required variables to the `To Workspace` block in Simulink so that they are captured in the MAT-file.

See Also

"Using the Flight Control Interface to Start the Drone" on page 2-14 | "Analyze Flight Log Errors" on page 2-21

Analyze Flight Log Errors

If the Parrot minidrone stops unexpectedly before completion of the flight time, you can download the flight log and analyze the errors. The errors that occurred during the flight of the minidrone are captured in the flight log (`droneFlight.txt`).

Note Some of the errors are also displayed in the Parrot Flight Control Interface (in the text box below the **START** button, after the drone has stopped).

The most common errors displayed in the flight log and troubleshooting steps are explained in the below table.

Errors Captured in the Flight Log and Troubleshooting Steps

Type of Error	Error Message in Flight Log	Reasons for the Error and St
High acceleration	`rsedu_control(): Flight crash detected (accelerometer): shutting down motors now`	This error occurs when high ac detected during the flight. To troubleshoot the issue, perfo • Understand the instance at looking at the flight log, and avoid the error. • Provide a higher value for a **Maximum acceleration of** the Configuration Paramete "Model Configuration Paran

Memory allocation error before take-off	`rsedu_control(): ERROR: Error from simulink model @ counter=401`
	
	`Memory allocation errorrsedu_control():Saving logged data after 401 cycles...`
		This error occurs only if MAT-fi
		
		MAT-file logging tries to allocat minidrone. The amount of mem signals that are being logged a required memory is not availab occurs.
		
		If MAT-file logging is not requir **logging**.
		
		If you require MAT-file logging,
		
		• Reduce the simulation time required memory (as compu is reduced.
		• Reduce the number of signa less memory is required.
		• Clear the memory of the dro Clearing Parrot Minidrone M

Low battery voltage	`rsedu_control(): Flight aborted due to low voltage (29 %%): shutting down motors now, charge battery!`	This error occurs if a low batte take-off or during the flight. Th of battery voltage are 50% for t flight. To troubleshoot the issue, ensu the drone is fully charged befor from the Flight Control Interfac is reconnected, perform the tas minidrone to host computer ove Configuration").

Type of Error	Error Message in Flight Log	Reasons for the Error and St
Optical flow and state estimate mismatch	`rsedu_control(): Flight crash about to happen, mismatch optical flow (-0.001362, -0.000360) and state estimate (27.352, -32.213): shutting down motors now`	This error occurs if there is a m estimation and the state estima the drone is flying. This error occurs for the `parro` (part of the `Hover Parrot Mi` To troubleshoot the issue, rede the drone.
Optical flow is not present (for the `ParrotMinidroneHover` project and `ParrotMinidroneCompetition` project)	`rsedu_control(): Flight aborted due to error condition 69 !`	This error occurs if the optical more than 50 cycles. This is co `Prediction Logic` subsyster `ParrotMinidroneHover` proj `ParrotMinidroneCompetiti` To troubleshoot the issue, perfo • Ensure that there is necessa you are flying the drone, so properly captured for the op • If you do not wish to use the controller logic, disable the `Prediction Logic` subsys Flow with Parrot Minidrone deploy the Simulink model a
Optical flow is not present (for `asbQuadcopter` project and any Simulink models created in R2018b or earlier releases)	`rsedu_control(): Problem with optical flow, there has been no flow for 50 cycles in cycle 723: shutting down motors now`	This error occurs if the optical more than 50 cycles. To troubleshoot the issue, ensu illumination in the area where the image or video frame is pro flow.
User-defined errors	`rsedu_control(): Flight aborted due to error condition 5 !`	This error occurs if you have m the Simulink template model. W the corresponding error is pass then logged in the flight log. For example, in this error mess that the error logic modeled by 5, has occurred. To troubleshoot the issue, go to corresponding error condition, to resolve the error.

Type of Error	Error Message in Flight Log	Reasons for the Error and St
Altitude estimation errors	`rsedu_control(): Flight aborted due to error condition 88 !`	This error occurs when the ultr values while you are flying the `ParrotMinidroneHover` proj drone is flying close to a wall o reflect the ultrasound waves, w sensor producing bad values. To troubleshoot the issue, perfo • Ensure that you do not fly th object that can reflect the u • Disable the check for ultras `Logic` subsystem (part of th the `parrotMinidroneHove` go to `Crash Prediction` `improper` subsystem, and s block to `0`.

See Also

“Setting Up the Hardware and Deploying the Model” on page 2-12 | “Using the Flight Control Interface to Start the Drone” on page 2-14

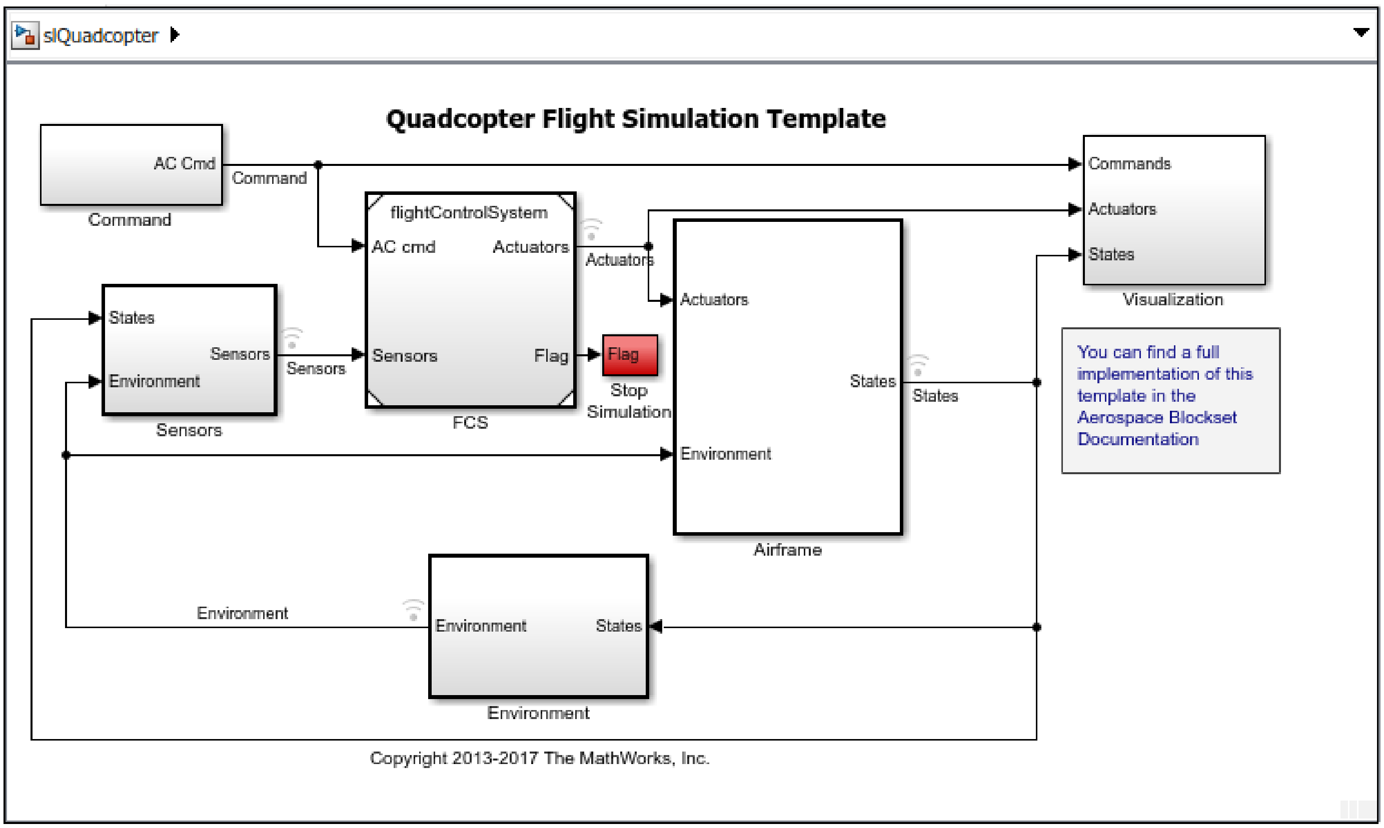
Flight Simulation Simulink Template for Parrot Minidrone

You can use the Flight Simulation Simulink template to create a model to simulate a drone's behavior. After you are satisfied with the simulation, you can deploy the model on the Parrot minidrone hardware. This template is designed to spin the two clockwise propellers and the two counterclockwise propellers for 2 seconds each. You can update this template to add different logic.

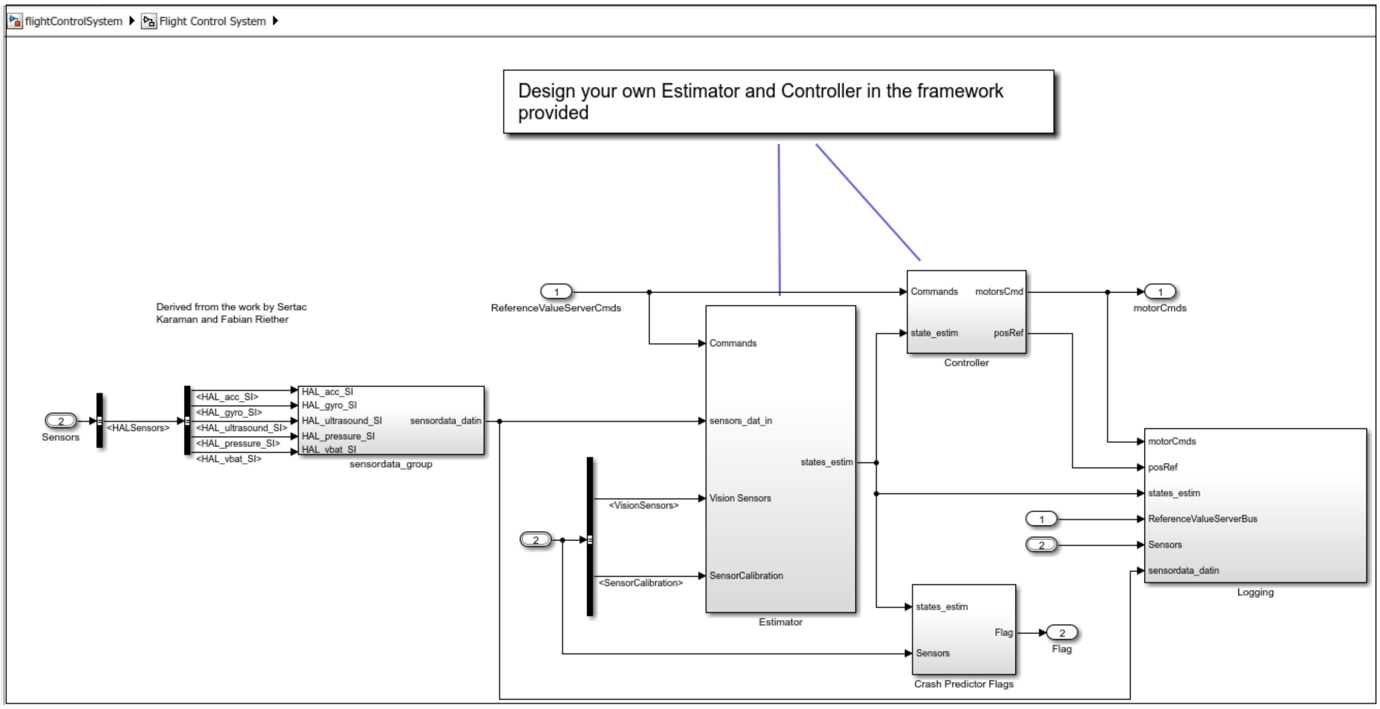
Open the Flight Simulation Template

1. At the MATLAB command prompt, type `simulink`.
2. On the Simulink Start Page, navigate to **Simulink Support Package for Parrot Minidrones**, and select **Flight Simulation Template**.

   Specify a name for the project. A new Simulink model named `slQuadcopter` opens. This model is configured for flight simulation.



   The `slQuadcopter` model contains the `FCS` (Flight Control System) subsystem, which takes input values from the different sensors on the drone and provides output signals to the motors based on the logic in the `Estimator` and `Controller` subsystems.



You can modify the logic inside `Estimator` and `Controller` subsystems to create your own model to fly the Parrot minidrone.

Deploy the Flight Simulation Template and Start the Motors

The procedure to deploy the Flight Simulation template and start the motors are the same as for the Code Generation template (see “Setting Up the Hardware and Deploying the Model” on page 2-12 and “Using the Flight Control Interface to Start the Drone” on page 2-14).

See Also

“Code Generation Simulink Template for Parrot Minidrone” on page 2-3 | “Fly a Parrot Minidrone Using the Quadcopter Simulink Model” on page 2-27

Fly a Parrot Minidrone Using the Quadcopter Simulink Model

The `asbQuadcopter` model, which is available in the Aerospace Blockset, starts the flight of a Parrot minidrone. The drone hovers at an altitude of 1.1 meters for a specified flight time before powering off.

Before you begin, make sure that you have licenses for these products:

- Aerospace Blockset
- Aerospace Toolbox
- Control System Toolbox
- Signal Processing Toolbox
- Simulink 3-D Animation
- Simulink Coder

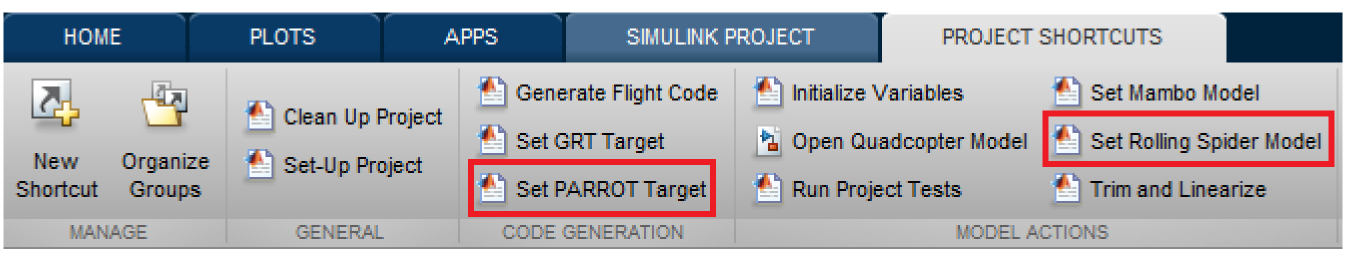
Open and Deploy the Quadcopter Model and Start the Flight

1. At the MATLAB prompt, type:

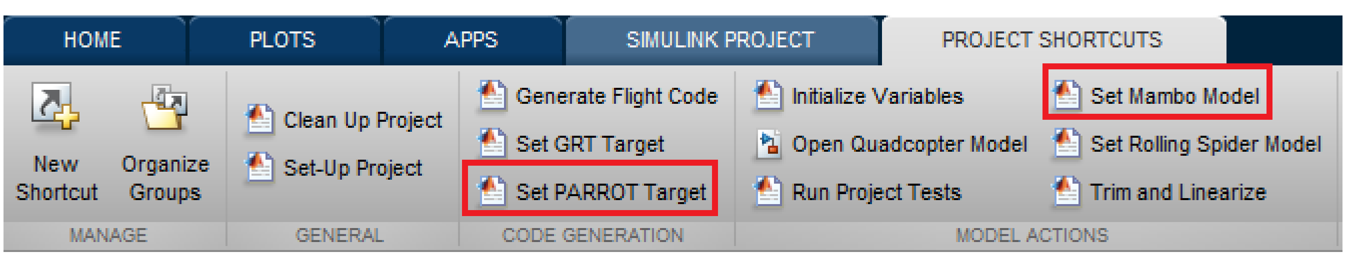
   ```
   asbQuadcopterStart
   ```

   This opens a Simulink project with the `asbQuadcopter` model and loads the required workspace variables.
2. Run the `asbQuadcopter` model in normal simulation mode. Navigate through the different subsystems to learn about the modeling hierarchy and quadcopter dynamics. You can also view the Simulink 3-D animation of the model. For more information, see Quadcopter Project.
3. With the `asbQuadcopter` Simulink project open, click the **Project Shortcuts** tab and perform one of the following tasks based on the type of minidrone connected:

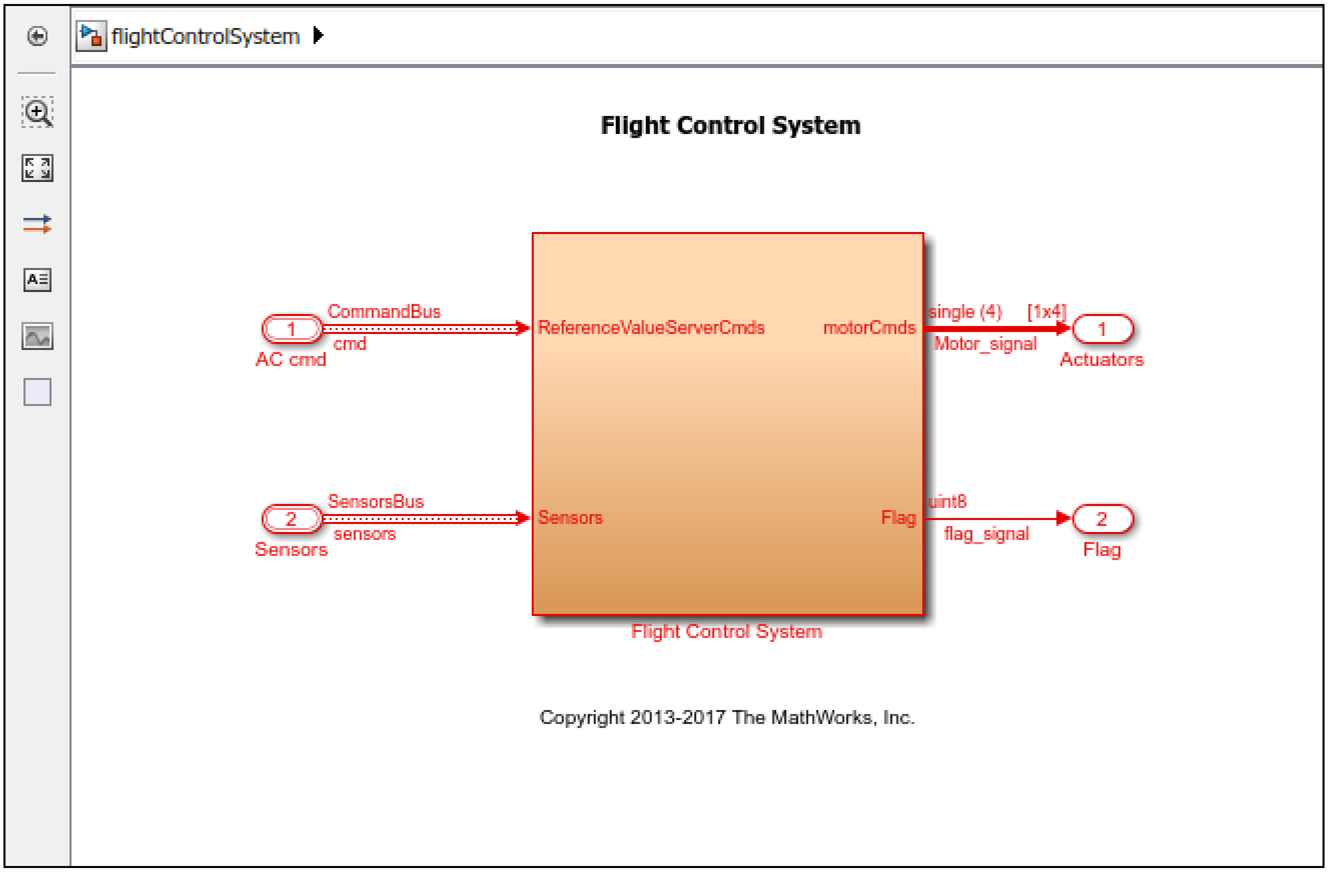
   - If the host computer is connected to a Parrot Rolling Spider, click **Set Rolling Spider Model** and then click **Set Parrot Target**.



   - If the host computer is connected to a Parrot Mambo, click **Set Mambo Model** and then click **Set Parrot Target**.



4 Double-click the Flight Control System block in the `asbQuadcopter` model in Simulink. The `flightControlSystem` model opens.



Note In the `flightControlSystem` model, do not change the root level input ports, output ports, or the signals through them.

---

5 In Simulink, click the **Deploy to Hardware** icon, . After the build process and deployment is successful, the Flight Control Interface is launched automatically if you had selected the option **Launch Parrot Flight Control Interface automatically after build** in the Configuration Parameters dialog box in Simulink (see "Model Configuration Parameters for Parrot Minidrone").

6 Before flying the Parrot Rolling Spider with full speed, test the model at low speed by spinning the motors at low power. Open the Flight Control Interface (see "Step 1: Open the Flight Control Interface" on page 2-14). Drag the **PowerGain** slider to 20, which sets the power gain of the motors to 20%.

7 Click **START** to start the drone. The motors on the Parrot minidrone start. The propellers spin for the time defined as flight duration (by default, the simulation time) and stop.

8 To prepare the Parrot minidrone for flight, set the power gain of the motors to the highest value (100%). Drag the **PowerGain** slider to 100.

Note Before flying the Parrot minidrone (as explained in the next step), adhere to the following safety procedures:

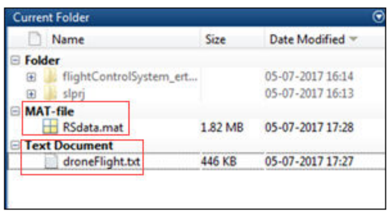
- Ensure the safety of people, animals, and property in the vicinity of the flight.
- Wear safety glasses at all times.
- Place the drone on a flat surface before starting.
- Fly the drone only indoors, with an open area greater than 10x10 feet, over a non-glossy floor.
- Always be ready to stop the flight. The Flight Control Interface displays the **STOP** button after the motors on the drone start.

---

9. Click **START** to start the flight of the drone. The motors on the Parrot minidrone start, and the drone performs a vertical take-off to an altitude of 1.1 meters. The drone hovers at this position for the time defined as flight duration (by default, the simulation time), and the motors stop after the flight duration is completed.

   To stop the flight before the flight duration is completed, click **STOP** in the Flight Control Interface.

10. Click **Flight Log** and **MAT File** to download the flight log and the MAT-file, respectively. The files `droneFlight.txt` and `RSdata.mat` are downloaded to the Current Folder in MATLAB.



The data in `RSdata.mat` includes various sensor values, motor commands, calibration values, and outputs of estimators during the flight. You can also plot the values from the downloaded MAT-file by using this command in MATLAB:

```
parrot.util.PostFlightAnalysis
```

Note The command `parrot.util.PostFlightAnalysis` is specific to the `asbQuadcopter` model.

After successfully flying the Parrot minidrone using the `asbQuadcopter` model, you can now redesign the controller logic in Simulink. Deploy your new model on the minidrone by following the same steps, and start the minidrone flight using the same commands. Always test your model with a low value of power gain (10–20%) for the motors. After you are confident about the minidrone flight, increase the power gain and run the model on the minidrone.

See Also

"Code Generation Simulink Template for Parrot Minidrone" on page 2-3 | "Flight Simulation Simulink Template for Parrot Minidrone" on page 2-25 | "Using Monitor and Tune with Image Processing for Parrot Minidrone" on page 2-34

Fly a Parrot Minidrone Using Hover Parrot Minidrone Simulink Template

The `Hover Parrot Minidrone` project template and the `parrotMinidroneHover` project, which are available in the Simulink Support Package for Parrot Minidrones, enable you to fly a Parrot minidrone. Once the model is deployed, the drone hovers at an altitude of 1.1 meters for a specified flight time before powering off.

Before you begin, make sure that you have licenses for these products:

- Simulink
- Aerospace Blockset
- Aerospace Toolbox
- Control System Toolbox
- Signal Processing Toolbox
- Simulink 3-D Animation

Open and Deploy the Hover Model and Start the Flight

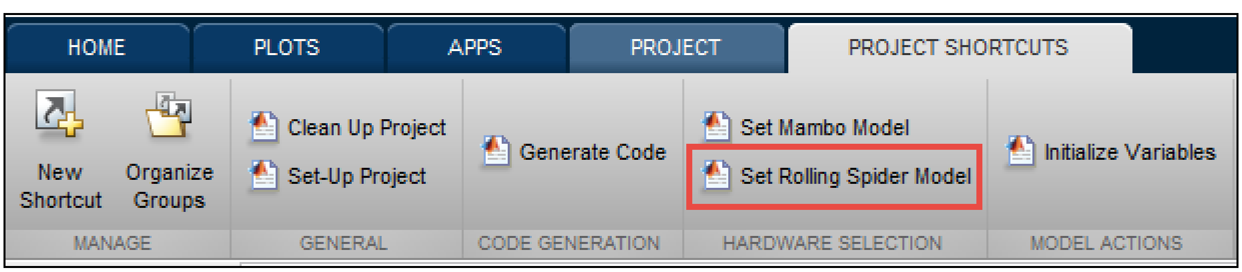
1. Start the Simulink project using either of the two methods:

   - On the Simulink Start Page, navigate to **Simulink Support Package for Parrot Minidrones**, and select **Hover Parrot Minidrone**.
   - At the MATLAB prompt, type:

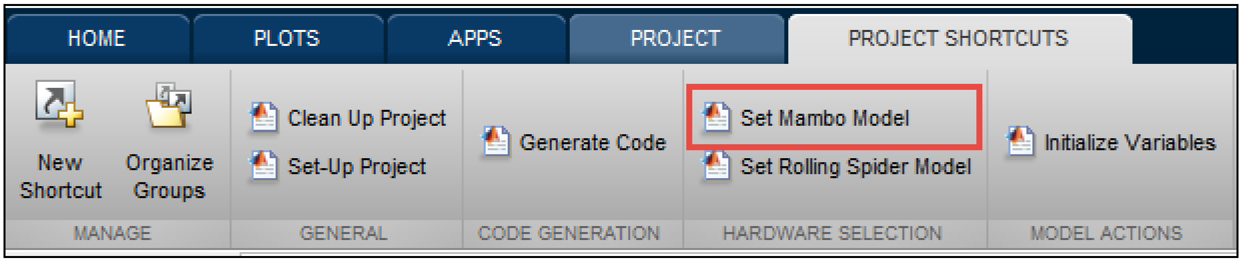
     ```
     parrotMinidroneHoverStart
     ```

     This opens a Simulink project with the `parrotMinidroneHover` model and loads the required workspace variables.
2. Run the `parrotMinidroneHover` model in normal simulation mode. Navigate through the different subsystems to learn about the modeling hierarchy and quadcopter dynamics. You can also view the Simulink 3-D animation of the model. For more information, see "Quadcopter Dynamics" on page 2-39.
3. Once the Simulink project is open, click the **Project Shortcuts** tab and perform one of the following tasks based on the type of Parrot minidrone connected:

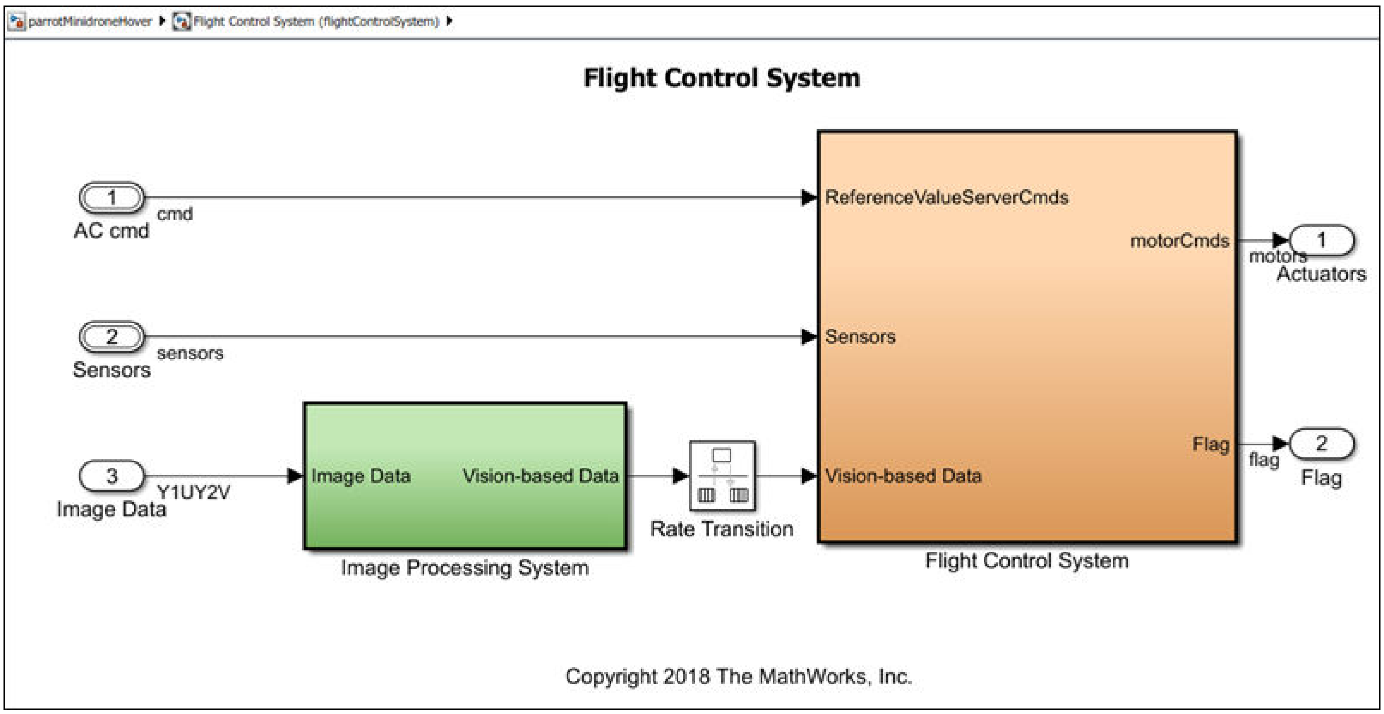
   - If the host computer is connected to a Parrot Rolling Spider, click **Set Rolling Spider Model**.



   - If the host computer is connected to a Parrot Mambo, click **Set Mambo Model**.



4 In the `parrotMinidroneHover` model, right-click the **Flight Control System** reference model, and select **Open As Top Model**. The `flightControlSystem` model opens in a separate window.



Note In the `flightControlSystem` model, do not change the root level input ports, output ports, or the signals through them.

5 In the `flightControlSystem` model, click the **Deploy to Hardware** icon, . After the build process and deployment is successful, the Flight Control Interface is launched automatically if you had selected the option **Launch Parrot Flight Control Interface automatically after build** in the Configuration Parameters dialog box in Simulink (see "Model Configuration Parameters for Parrot Minidrone").

6 Before flying the Parrot minidrone with full speed, test the model at low speed by spinning the motors at low power. Open the Flight Control Interface (see "Step 1: Open the Flight Control Interface" on page 2-14). Drag the **PowerGain** slider to 20, which sets the power gain of the motors to 20%.

7 Click **START** to start the model deployed on the drone. The motors on the Parrot minidrone start. The propellers spin for the time defined as flight duration (by default, the simulation time) and stop.

8 To prepare the Parrot minidrone for flight, set the power gain of the motors to the highest value (100%). Drag the **PowerGain** slider to 100.

Note Before flying the Parrot minidrone (as explained in the next step), adhere to the following safety procedures:

- Ensure the safety of people, animals, and property in the vicinity of the flight.
- Wear safety glasses at all times.
- Place the drone on a flat surface before starting.
- Fly the drone only indoors, with an open area greater than 10x10 feet, over a non-glossy floor.
- Always be ready to stop the flight. The Flight Control Interface displays the **STOP** button after the motors on the drone start.

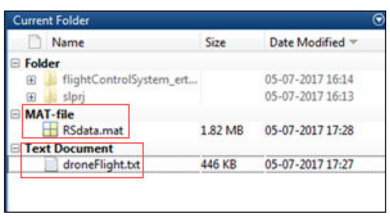
---

9 Click **START** to start the flight of the drone. The motors on the Parrot minidrone start, and the drone performs a vertical take-off to an altitude of 1.1 meters. The drone hovers at this position for the time defined as flight duration (by default, the simulation time), and the motors stop after the flight duration is completed.

To stop the flight before the flight duration is completed, click **STOP** in the Flight Control Interface.

10 Perform the flight analysis:

- Click **Flight Log** to download the flight log. The file `droneFlight.txt` is downloaded to the Current Folder in MATLAB.
- Click **MAT File** to download the MAT-file. The file `RSdata.mat` is downloaded to the Current Folder in MATLAB.



Note To enable MAT-file logging, Simulink Coder must be installed on the host computer. Additionally, you must enable the options **Use Simulink Coder features** and **MAT-file logging** in the Configuration Parameters dialog box in Simulink model (see Simulink documentation)

The data in `RSdata.mat` contains various estimated outputs.

After successfully flying the Parrot minidrone using the `parrotMinidroneHover` model, you can now redesign the controller logic in Simulink. Deploy the new model on the minidrone by following the same steps, and start the minidrone flight using the same commands. Always test your model with a low value of power gain (10–20%) for the motors. After you are confident about the flight, increase the power gain and run the model on the minidrone.

Landing Logic in parrotMinidroneHover Model

The control logic to land the drone is present in Flight Control System > Path Planning subsystem.

Two fields have been added to the Command bus (highlighted in the below figure):

- `land_drone` - This flag will be set when the **Land** button on the Parrot Flight Interface is clicked.
- `time_remaining` - This is the time remaining for the simulation to finish.

These two fields will be used to enable the landing of the drone.

This landing is achieved using two subsystems - `Landing Enable` and `Landing Logic`.

The `Landing Enable` subsystem defines the logic required to enable the landing. The `Land` output of this subsystem is set when either of these occurs:

- The `Land Flag` flag is set.
- The time remaining is less than or equal to 5 seconds (this value can be changed).

The `Landing Logic` subsystem defines the logic required to land the Parrot minidrone. This subsystem is used in the *parrotMinidroneHover* model for the following:

- Obtain the landing status from the *Landing Enable* subsystem, and initiate the landing logic
- Define the landing step size that starts the descent of the drone along the z-axis. In this example, the landing step size is 0.2 meters. If you increase the value of landing step size, the drone descents faster along the z-axis.
- Stop the model when the drone reaches 0.3 meters above the ground.

  Flag `255` is set in this case, and no error will be shown in the Parrot Flight Interface.

See Also

"Code Generation Simulink Template for Parrot Minidrone" on page 2-3 | "Flight Simulation Simulink Template for Parrot Minidrone" on page 2-25 | "Using Monitor and Tune with Image Processing for Parrot Minidrone" on page 2-34

Using Monitor and Tune with Image Processing for Parrot Minidrone

The image-processing algorithm can be used as an additional input to control the flight of a Parrot minidrone. In the Simulink model, this algorithm is part of a separate subsystem (Image Processing System) which integrates with the main subsystem (Flight Control System). The tasks that are being executed in the two subsystems run at different rates.

The image-processing algorithm can be analyzed by using the Monitor and Tune feature. In this case, the signal monitoring and image-processing run at the same priority, which results in a slow-down of the image-processing algorithm. To ensure that image-processing runs at real-time, you may need to disable the Monitor and Tune action.

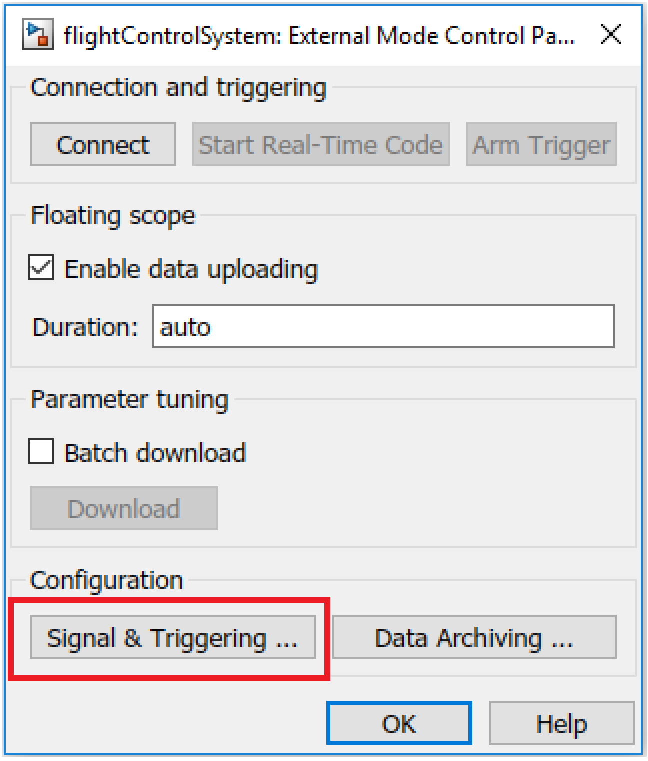
Note The performance limitations are due to the limited computational power of the Parrot minidrone when signal monitoring and parameter tuning is used along with image-processing from the drone's camera.

However, if you want to use Monitor and Tune feature to analyze the image-processing algorithm in the Simulink model, you need to change the settings accordingly:

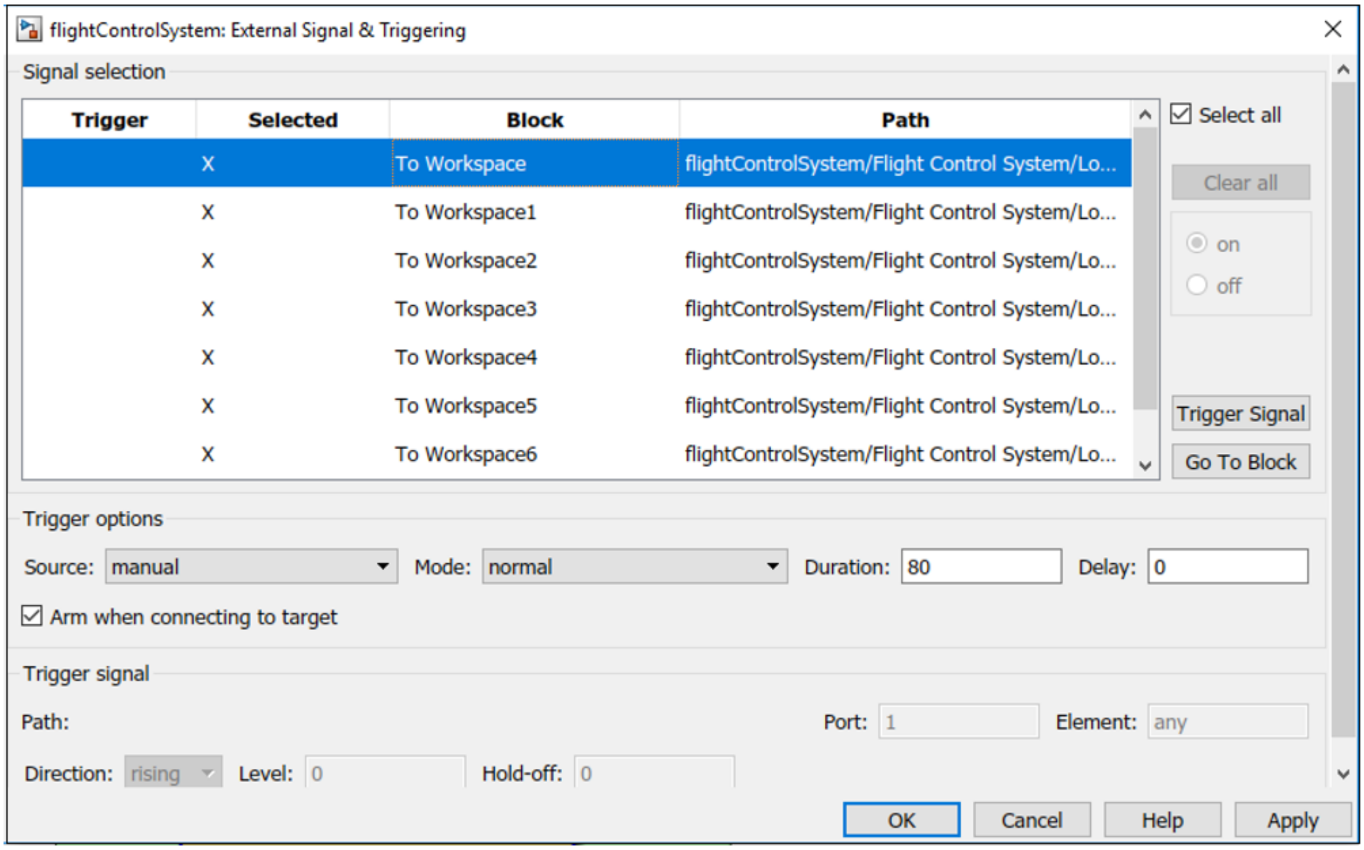
Note The guidelines explained in this section requires that you install Simulink Coder before using the Monitor and Tune feature for the Simulink model, as it provides additional options for the same.

---

1. On the Simulink toolstrip, go to **Hardware** tab, open the **Prepare** gallery and click **Control Panel**.
2. In the External Mode Control Panel, click **Signal & Triggering**.



The External Signal & Triggering dialog box displays a list of blocks and signals in the model that support data uploading and logging. By default, all the items listed are selected for viewing in the External mode and logging data (see the **x** label for each signal). (For more details about External mode, refer to documentation)



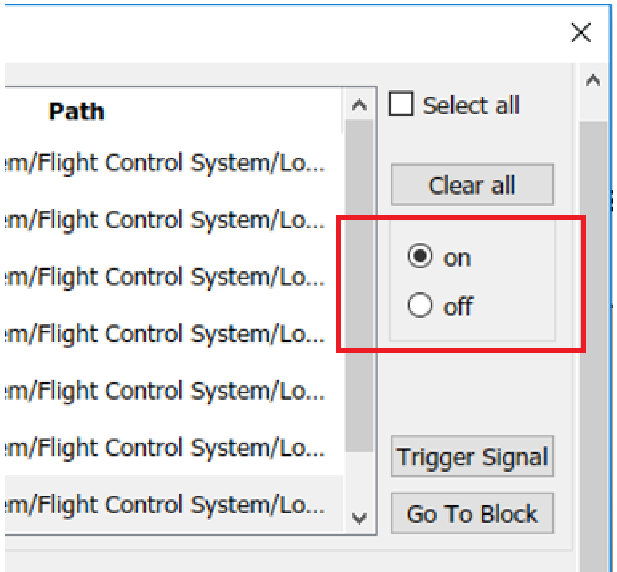
3 Change the settings for selection of signals because logging of multiple signals during image processing slows down both the image-processing algorithm and the External mode logging:

- Ensure that you select only the signal that is important for visualization. If you would like to view a signal using a Video Viewer block (from the Computer Vision Toolbox™) or a Scope/Display block, connect it at the output of the intermediate stage of the image-processing algorithm. In this case:

  a In the External Signal & Triggering dialog box, clear the **Select all** option (by default, this option is selected).

  b Click the row corresponding to the Video Viewer block or Scope/Display block.

  c Click the **on** option (which appears at the right side of the dialog box). This ensures that the Video Viewer or Scope/Display is the only block that can be viewed in External mode, and the performance of image-processing algorithm is not affected.



d Click **Apply** and then **OK**.

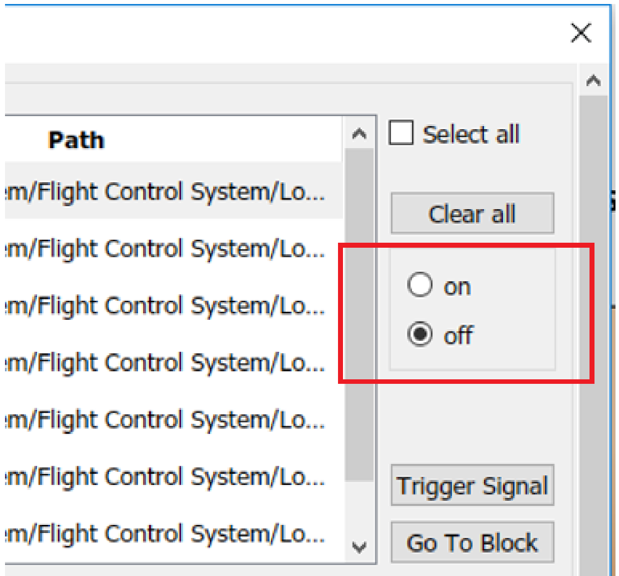
Note In the Simulink model, avoid using multiple Video Viewer blocks for visualization of image data as it slows down the image processing.

- Remove some signals from uploading any data for analysis. To do this:

  a In the External Signal & Triggering dialog box, clear the **Select all** option (by default, this option is selected).

  b Navigate through the list and identify the unwanted signals and Scope/Display blocks that are being logged from the Flight Control System subsystem.

  c Click the row corresponding to the signal or Scope/Display block, and click the **off** option (which appears at the right side of the dialog box). This ensures that these signals or Scope/Display blocks are not used for logging of any data, thereby improving the overall performance of the drone while the image-processing is in progress.



d Click **Apply** and then **OK**.

See Also

"Code Generation Simulink Template for Image Processing for Parrot Minidrone" on page 2-8 | "Fly a Parrot Minidrone Using the Quadcopter Simulink Model" on page 2-27

Quadcopter Dynamics

This section explains how the quadcopter physical characteristics and dynamics are implemented in the `parrotMinidroneHover` project and `Hover Parrot Minidrone` Simulink template.

Quadcopter Physical Characteristics

The following schematic shows the quadcopter physical characteristics:

- Axis
- Mass and Inertia
- Rotors

Axis

The quadcopter body axis is centered in the center of gravity.

- The *x*-axis starts at the center of gravity and points in the direction along the nose of the quadcopter.
- The *y*-axis starts at the center of gravity and points to the right of the quadcopter.
- The *z*-axis starts at the center of gravity and points downward from the quadcopter, following the right-hand rule.

Mass and Inertia

We assume that the whole body works as a particle. The file `vehicleVars` contains the values for the inertia and mass.

Rotors

- Rotor #1 rotates positively with respect to the *z*-axis. It is located parallel to the *xy*-plane, -45 degrees from the *x*-axis.

- Rotor #2 rotates negatively with respect to the body's *z*-axis. It is located parallel to the *xy*-plane, -135 degrees from the *x*-axis.
- Rotor #3 has the same rotation direction as rotor #1. It is located parallel to the *xy*-plane, 135 degrees from the *x*-axis.
- Rotor #4 has the same rotation direction as rotor #2. It is located parallel to the *xy*-plane, 45 degrees from the *x*-axis.

This example uses the approach defined by Prouty[1] and adapted to a heavy-lift quadcopter by Ponds et al[2].

Control

For control, the quadcopter uses a complementary filter to estimate attitude, and Kalman filters to estimate position and velocity. The example implements:

- A PID controller for pitch/roll control
- A PD controller for yaw
- A PD controller for position control in North-East-Down coordinates

The `controllerVars` file contains variables pertinent to the controller. The `estimatorVars` file contains variables pertinent to the estimator.

The example implements the controller and estimators as model subsystems, enabling several combinations of estimators and controllers to be evaluated for design.

Sensors

The example uses a set of sensors to determine its states:

- An Inertial Measurement Unit (IMU) to measure the angular rates and translational accelerations.
- A camera for optical flow estimation.
- A sonar for altitude measurement.

The example stores the characteristics for the sensors in the file `sensorVars`. To include sensor dynamics with these measurements, you can change the `VSS_SENSORS` variable in the workspace.

Environment

The models implement several Aerospace Blockset™ environment blocks, including those for atmosphere and gravity models. To include these models, you can change the `VSS_ENVIRONMENT` variable in the workspace to toggle between variable and fixed environment models.

Linearization

The model uses the `trimLinearizeOpPoint` to linearize the nonlinear model of the quadcopter using Simulink Control Design (R).

Visualization

You can visualize the variables for the quadcopter in one of the following ways:

- Using Simulation Data Inspector.
- Using the flight instrument blocks.
- Toggling between the different visualization variant subsystems. You can toggle between the different variant subsystems by changing the `VSS_VISUALIZATION` variable. Note that one of these variants is a FlightGear animation. To use this animation, you must add a FlightGear compatible model of the quadcopter to the project. The software does not include this model.

References

[1] Prouty, R. Helicopter Performance, Stability, and Control. PWS Publishers, 2005.

[2] Ponds, P., Mahony, R., Corke, P. Modelling and control of a large quadrotor robot. Control Engineering Practice. 2010.

Keyboard Control of Parrot Minidrones

Simulink Support Package for Parrot Minidrones helps you to remotely control the behavior of the drone. This is achieved by forwarding the key presses on the keyboard of the host computer to the drone.

The workflow for implementing keyboard control of Parrot minidrone includes the following steps:

1. Preparing Simulink model for keyboard control of drone
2. Deploying the Simulink model to the drone
3. Using the Keyboard Control Panel after the drone has started

Preparing Simulink Model for Keyboard Control

Simulink Support Package for Parrot Minidrones contains the Keyboard Read block that can be used to receive key presses from the host computer. This block outputs the ASCII code of the key that was pressed on the keyboard of the host computer, which can be used to perform actions on the drone.

Note Keyboard keys can be used for controlling the drone only after the Simulink model has started running in the drone (you still need to press the **START** button on the Flight Control Interface to start the drone).

To prepare the Simulink model for keyboard control of drone, follow these steps:

1. Add the Keyboard Read block in the Simulink model.

   Note Only one Keyboard Read block is allowed in the entire Simulink model. Any number of connections can be derived from the output signal of this block.

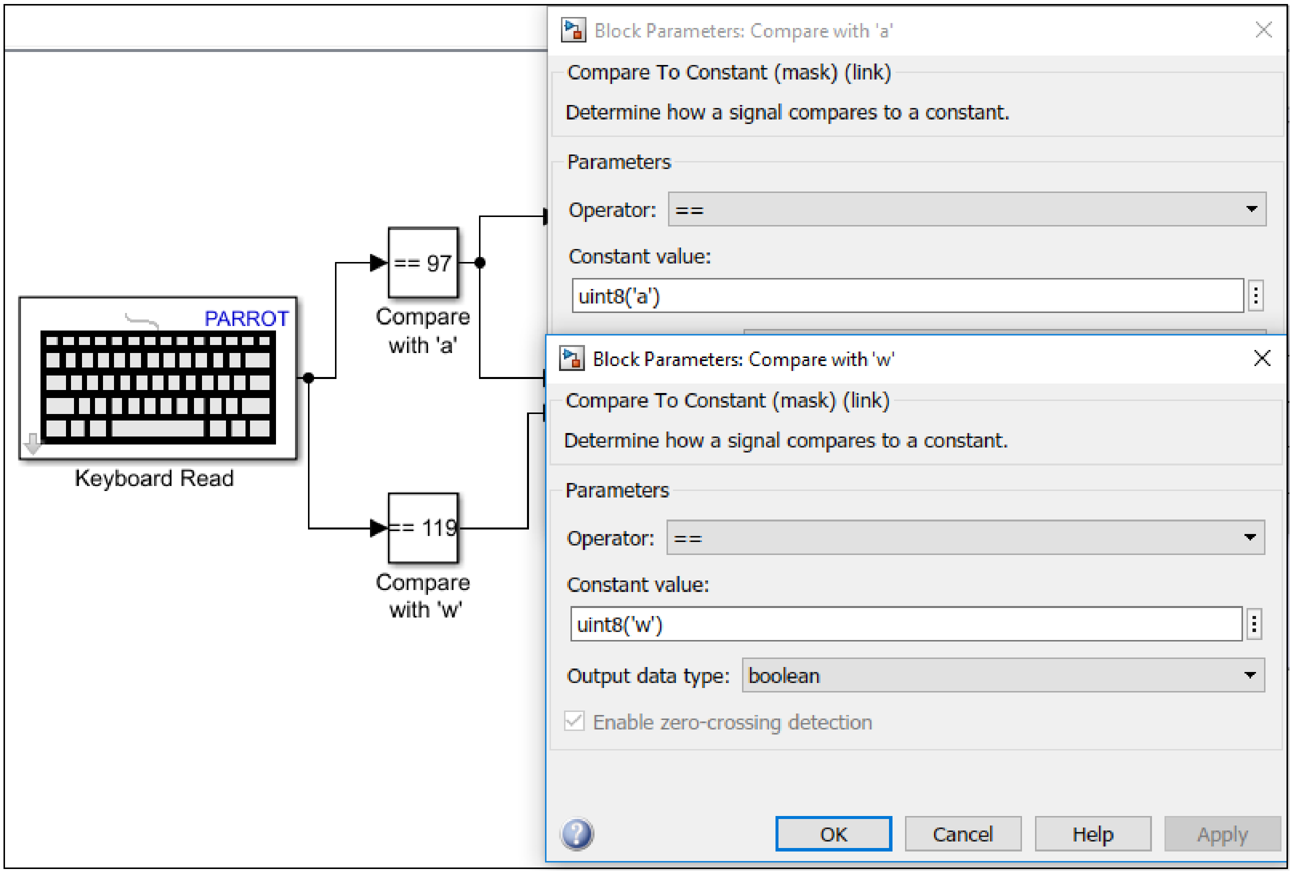
2. Identify the action that you would like to trigger by pressing a key on the keyboard. For example:

   - Changing the pitch and yaw values to reach to a desired position
   - Changing the speed of the motors on the drone
   - Landing the drone
   - Emergency shutdown of the drone

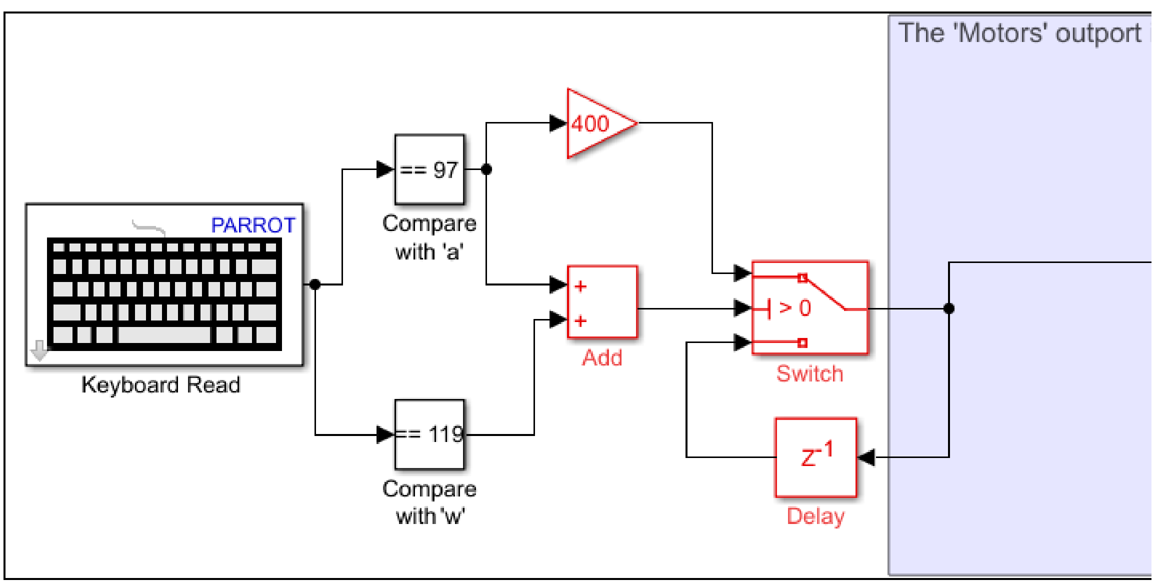
3. Identify a character on the keyboard to trigger this action.

   Note The identified characters can include all the printable characters – letters, numbers, punctuation marks, and common symbols present on the keyboard. Do not consider any key that does not have the corresponding ASCII character code available (for example – do not use arrow keys, Function keys, OS-specific keys and so on).

4. Use the output of the Keyboard Read block to trigger a particular action. For example, add a Compare to Constant block to check for a particular key press (based on the ASCII code of the key pressed).



5 Design the logic for executing the action. For example, you can add additional blocks like Add and Switch to control the motors of the drone based on key presses.



Example of Keyboard Read block used in a modified Code Generation Template

Using Keyboard Control Panel

After the Simulink model is successfully deployed on the Parrot minidrone, you need to start the model by clicking the **START** button on the Flight Control Interface.

The Keyboard Control Panel appears as part of the Flight Control Interface after the Simulink model has started running on the drone.

To use the Keyboard Control Panel while the model is running on the drone, follow these steps:

1. On the Flight Control Interface, click the **Show Keyboard Control Panel** checkbox.
2. Click **Enable Keyboard Control**.

3. Place the cursor in the field below the **Enable Keyboard Control** button, and press the required key on the keyboard.

   The drone executes the action that you have mapped to the pressed key.

Using MATLAB Scripts for Keyboard Control

The keyboard control of the Parrot minidrone using the Flight Control Interface can be extended to a MATLAB script also. The Keyboard Control interface uses a TCP/IP interface to send the key presses to the Keyboard Read block in the Simulink model. The Keyboard Read block listens to the port `26061` for any incoming connections. The Flight Control Interface opens up a TCP client and connects to this port.

Once a model having a Keyboard Read block is running on the drone, run the MATLAB script to send characters to the drone. The script needs to:

- Open a TCP/IP handle to the IP address `192.168.3.1`, and connect to the port `26061`.

  Note The Keyboard Read block listens at the port **26061 ONLY**.

- Write a single byte to the TCP socket.

For example, to send the ASCII a to the Keyboard Read block in the model running on the drone, use the MATLAB script as below (the IP address 192.168.3.1 corresponds to Parrot Mambo; for Parrot Rolling Spider the IP address is 192.168.3.5):

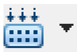
```
tcpHandle = tcpclient('192.168.3.1',26061);
write(tcpHandle, uint8('a'));
```

See Also

“Getting Started with Keyboard Control of Parrot Minidrones”

Troubleshooting Code Deployment Failure on Parrot Rolling Spider

Description

The deployment of the Simulink model to a Parrot Rolling Spider (using the Deploy to Hardware icon, ) sometimes fails. In this case, the Diagnostics Viewer displays the following error message:

```
The following error occurred during deployment to your hardware board:
The PARROT Rolling Spider minidrone is not connected over Bluetooth to host computer. Connect the
minidrone and build the model again.

Component: Simulink | Category: Block diagram error
```

This error is due to one of the following reasons:

- The Bluetooth connection between the Parrot Rolling Spider and the host computer is lost.
- You upgraded the Simulink Support Package for Parrot Minidrones to version 17.2.2 or later, and you did not perform the Hardware Setup process after the upgrade.

  Version 17.2.2 or later of Simulink Support Package for Parrot Minidrones uses the IP address 192.168.3.5 for the Parrot Rolling Spider. In earlier versions, this IP address is 192.168.1.1.

Action

To troubleshoot the issue, try one of these two options:

- Perform the steps required to connect the Parrot Roling Spider to the host computer using Bluetooth (see the related topics in “Setup and Configuration” based on the operating system running in the host computer).

  After the Bluetooth connection is established, deploy the Simulink model again to the Parrot Rolling Spider.
- If you did not perform the Hardware Setup process after you had upgraded to version 17.2.2 or later, run the Hardware Setup process at least once for the Parrot Rolling Spider after the upgrade (see “Hardware Setup” on page 1-2). The setup process updates the IP address of Parrot Rolling Spider to 192.168.3.5.

  After the hardware setup is completed, connect the drone to the host computer over Bluetooth, and deploy the Simulink model again to the Parrot Rolling Spider.

See Also

“Using the Flight Control Interface to Start the Drone” on page 2-14

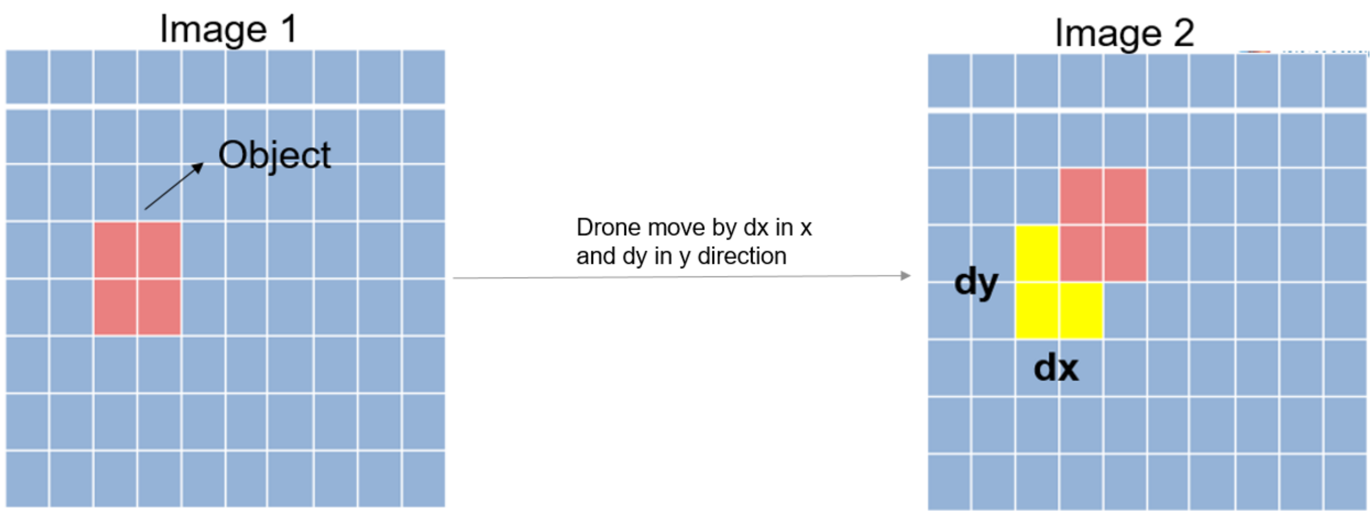
Optical Flow with Parrot Minidrones

Introduction

The Parrot minidrone uses various onboard sensors like Accelerometer and Gyroscope to stabilize itself. But, due to minor imperfections, the drone does not stay in place and tends to drift. This situation can be avoided by using images from the downward-facing camera of the drone. The camera continuously captures the images and checks for motion in X and Y directions. The flight controller uses the camera data, in addition to Accelerometer and Gyroscope, to stabilize itself in space and hold the x-y position of the drone.

Optical Flow

To understand optical flow, consider a case where the drone drifts by a distance of dx in x and dy in y direction, due to some noise in the sensors. In this case, the controller needs to know the distances moved in both x and y directions to bring it back to the original position.



This can be obtained using the downward-facing camera by comparing consecutive images as the drone moves. The downward facing camera fixes onto objects/patterns on the ground and if there is any change in position of that object/pattern, the camera reports it as a movement in terms of velocity – which is optical flow. This velocity measurement can be used to calculate the distance moved by the drone.

Troubleshooting Optical Flow Issues

There are cases in which the camera cannot detect the horizontal movement. If the texture of the surface below the drone is uniform with no patterns, the camera might not detect movement. Also, when the lighting is low, the camera might not detect movement. In such cases, the optical flow values will be zero resulting in drift and possible crash of the drone.

To avoid such situations, we have modelled a logic (crash prediction logic) in the new Simulink project template, `Hover Parrot Minidrone`. The `parrotMinidroneHover` project (which is part of the template) contains the `Crash Predictor Flags` subsystem (in the `flightControlSystem`

Simulink model). This subsystem is used to shut down the drone if there is no optical flow measurement for 50 continuous sample times.

If the drone shuts-down due to optical flow issues, some fixes can be done to the surface to increase contrast. Increasing contrast on the surface ensures that your camera produces non-zero optical flow values.

The below image shows a scenario of a surface where you may see optical flow errors:

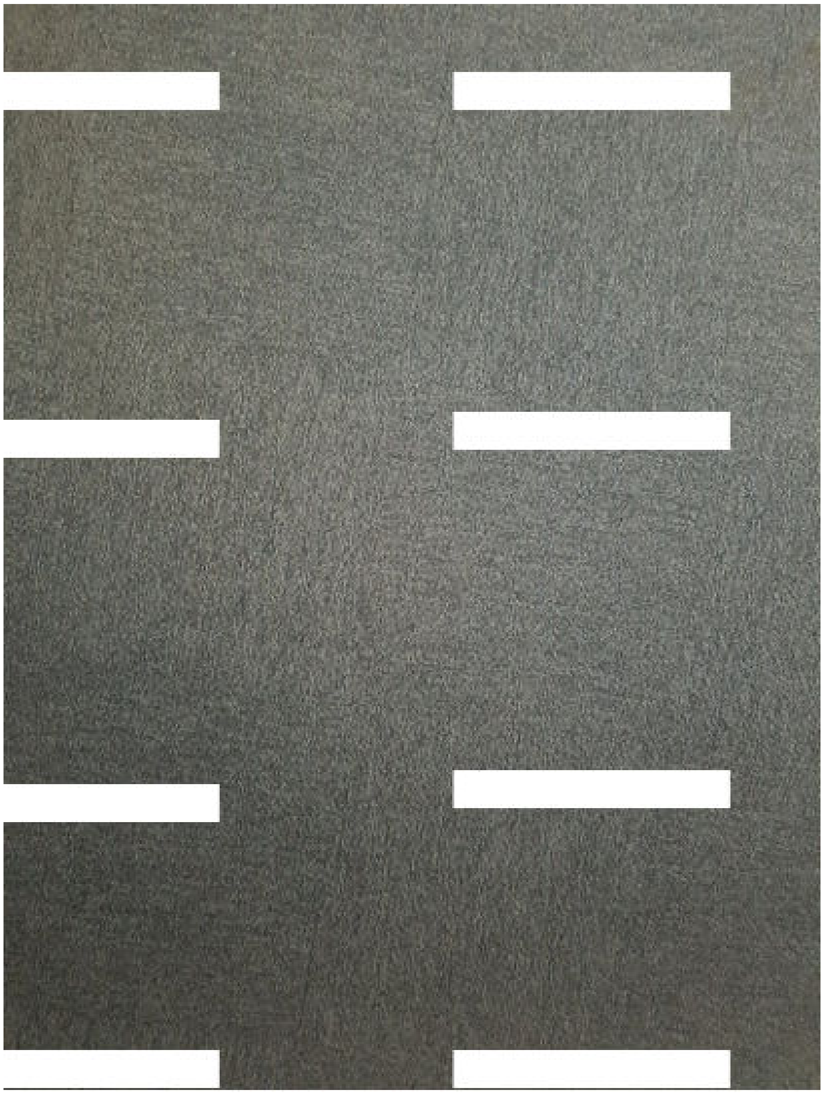
The region marked in red is a surface of uniform texture. When the drone is flying over the surface, it receives zero optical flow information. This is what the drone sees:

In this case, the drone will drift and eventually crash.

To increase contrast, add tapes on the surface over which the drone is flying. This change avoids the shutdown of drone due to optical flow issues.

The below image shows an example of adding tapes to the surface to increase contrast.

This is what the drone sees now:



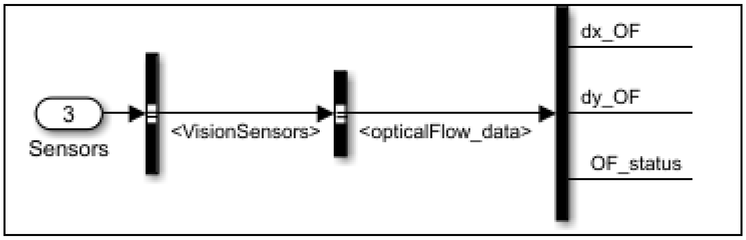
The texture on the surface need not be as shown in the above image. It can be any random pattern as long as there is a contrast on the surface.

In cases where there are optical flow issues even though there is contrast on the surface, the lighting can be an issue. In this case, increase the room's lighting so that the drone can get non-zero optical flow values.

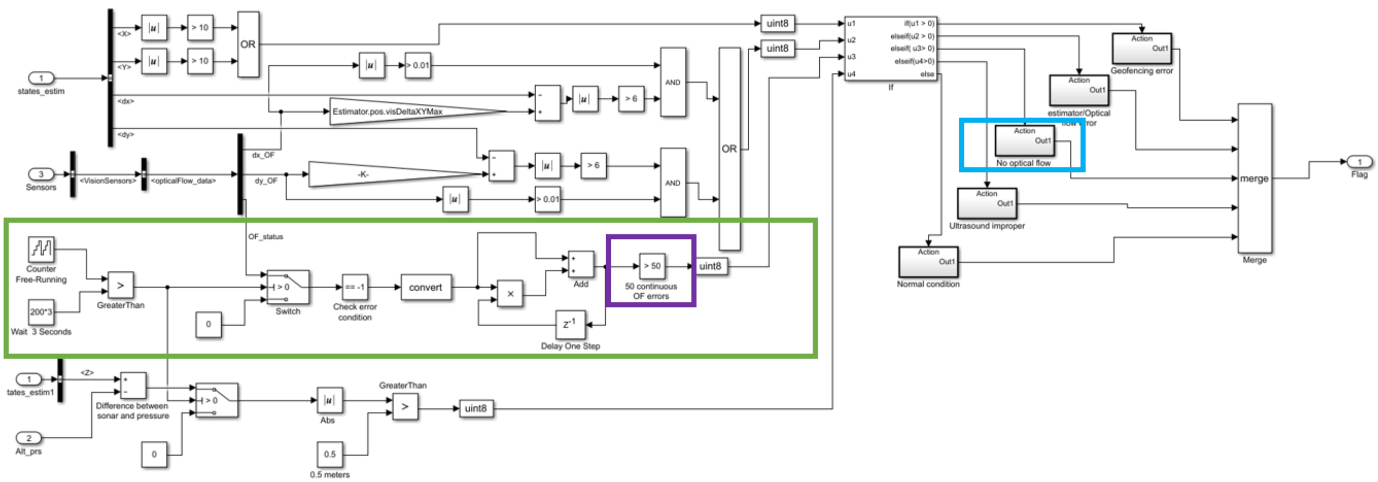
Simulink Model for Optical Flow

In the `parrotMinidroneHover` project (which is part of `Hover Parrot Minidrone` Simulink template), the crash prediction logic to shut down the drone is modeled in the `Crash Prediction Flags` subsystem in the `flightControlSystem` model.

In the `Crash Prediction Flags` subsystem, the <opticalFlow_data> is a vector obtained from `VisionSensors`, and it has the following fields: `dx_OF` (velocity in x-direction), `dy_OF` (velocity in y-direction) and `OF_status`. The `OF_status` is ‘0’ when the drone is unable to detect optical flow and ‘1’ when the drone detects optical flow.



The maximum number of cycles for which there is no optical flow can be changed by changing the value of the `Compare to constant` block, highlighted in purple. Once the specified value is reached, the drone shuts down.



To disable the Optical flow, perform one of the following options:

- Increase the value of the `Compare to constant` block (highlighted in purple) to a very large value.

  Note Increasing the count (maximum number of cycles for which there is no optical flow) to a very large value may cause the drone to drift.

- Open the `No optical flow` subsystem (highlighted in blue) and set the value of the Constant block inside the subsystem to `0`.

Removal of Enable Optical Flow Option from Configuration Parameters

The **Enable Optical Flow** option in the Configuration Parameters dialog box is now removed, and the functionality is exposed in the `flightControlSystem` model as shown above. The option is deprecated from R2019a onwards.

Limitations/ Backward Compatibility

For the older Simulink models (before R2019a), the optical flow behavior is unchanged (the Parrot drone shuts down after 50 cycles without optical flow). However, it is recommended to move to the new template (`Hover Parrot Minidrone`) introduced in R2019a.

See Also

"Quadcopter Dynamics" on page 2-39