

## Multitouch Trackpad for Raspberry Pi

The Azoteq IQS550EV02 evaluation module can serve as a multitouch trackpad for the Raspberry Pi. The module's convenient form factor as well as the IQS550's rich feature set make the module a handy accessory for the popular embedded computing platform.



This document describes the process for enabling the module as a multitouch trackpad for the Raspberry Pi desktop environment. All steps can be done through the Raspberry Pi's command-line terminal without the need for any specialized tools or software.

### Required Materials

- Any model of Raspberry Pi with a GPIO header
- Power supply for the Raspberry Pi—refer to Raspberry Pi's [official website](#) for power supply requirements and recommendations
- HDMI monitor and USB keyboard
- Internet connection for the Raspberry Pi
- SD or microSD card depending on Raspberry Pi model
- Separate computer and SD card reader
- Azoteq IQS550EV02 evaluation module
- Five female-to-female jumper wires

## Hardware Setup

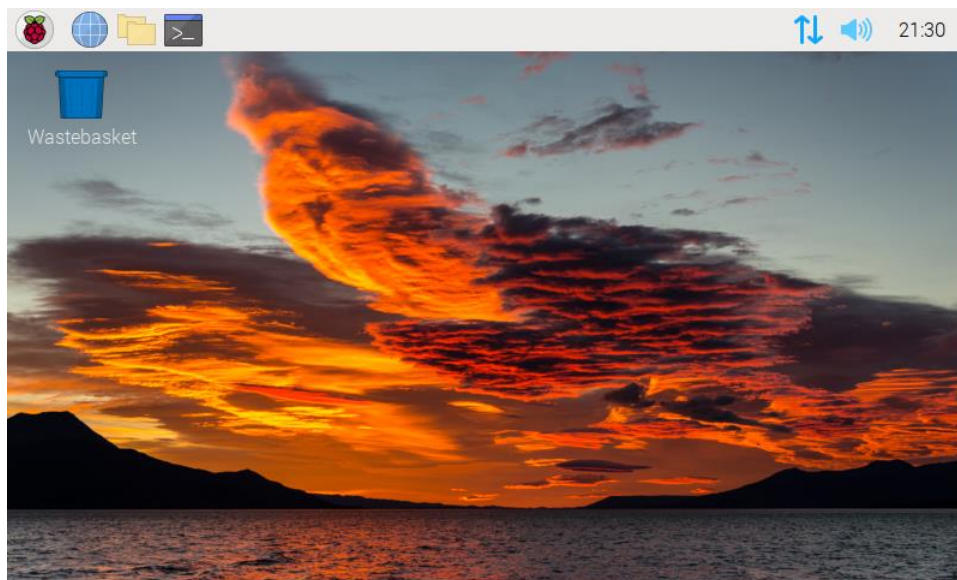
Use five female-to-female jumper wires to connect the GPIO header of the Raspberry Pi to the P1 connector of the IQS550EV02 according to the following table.

Signal Name	GPIO Header Pin	P1 Connector Pin
3.3 V	1	3
I <sup>2</sup> C Data	3	7
I <sup>2</sup> C Clock	5	9
Interrupt / Ready	7	10
Ground	9	1

## Software Setup

Download and install the Raspberry Pi Imager tool from Raspberry Pi's [official website](#) and follow its instructions to prepare the SD card. Once the imaging process is finished, insert the SD card into the Raspberry Pi's SD card slot.

Plug the HDMI monitor and USB keyboard into the Raspberry Pi. Connect the Raspberry Pi's Ethernet port to a wired internet connection and apply power to the Raspberry Pi; it will then boot into the desktop as shown in the following figure.



Use the tab, arrow and enter keys to navigate all on-screen prompts for configuring the Raspberry Pi for the first time. Refer to any number of online resources to optionally use a wireless internet connection in place of Ethernet.



Press CTRL+ALT+T to open a command-line terminal, then enter the following command to update the Raspberry Pi's firmware.

```
$ sudo rpi-update
```

Press Y to proceed past any cautionary messages. Once the update is complete, enter the following command to open the Raspberry Pi's boot configuration file.

```
$ sudo nano /boot/config.txt
```

Use the arrow keys to scroll down to the line containing `dtparam=i2c_arm=on`, then remove the pound sign to uncomment the line as shown below.

```
# Uncomment some or all of these to enable the optional hardware interfaces
dtparam=i2c_arm=on
#dtparam=i2s=on
#dtparam=spi=on
```

Scroll farther down to the line that reads `Additional overlays and parameters...` and add the line `dtoverlay=iqs550` as shown below.

```
# Additional overlays and parameters are documented /boot/overlays/README
dtoverlay=iqs550
```

Press CTRL+O followed by the enter key to save the changes. Press CTRL+X to exit the editor and return to the command prompt, then enter the following command to install the Multitouch X driver.

```
$ sudo apt-get install xserver-xorg-input-multitouch
```

Once the installation is complete, enter the following command to create a configuration file for the Multitouch X driver.

```
$ sudo nano /etc/X11/xorg.conf.d/50-multitouch.conf
```

Populate the configuration file with the following lines exactly as they are shown below.

```
Section "InputClass"
    Identifier "Multitouch Trackpad"
    MatchIsTouchscreen "on"
    MatchDevicePath "/dev/input/event*"
    Driver "multitouch"
EndSection
```

Press CTRL+O followed by the enter key to save the changes. Press CTRL+X to exit the editor and return to the command prompt, then enter the following command to reboot the Raspberry Pi.

```
$ reboot
```

Once the Raspberry Pi reboots into the desktop, the IQS550EV02 is ready to use.

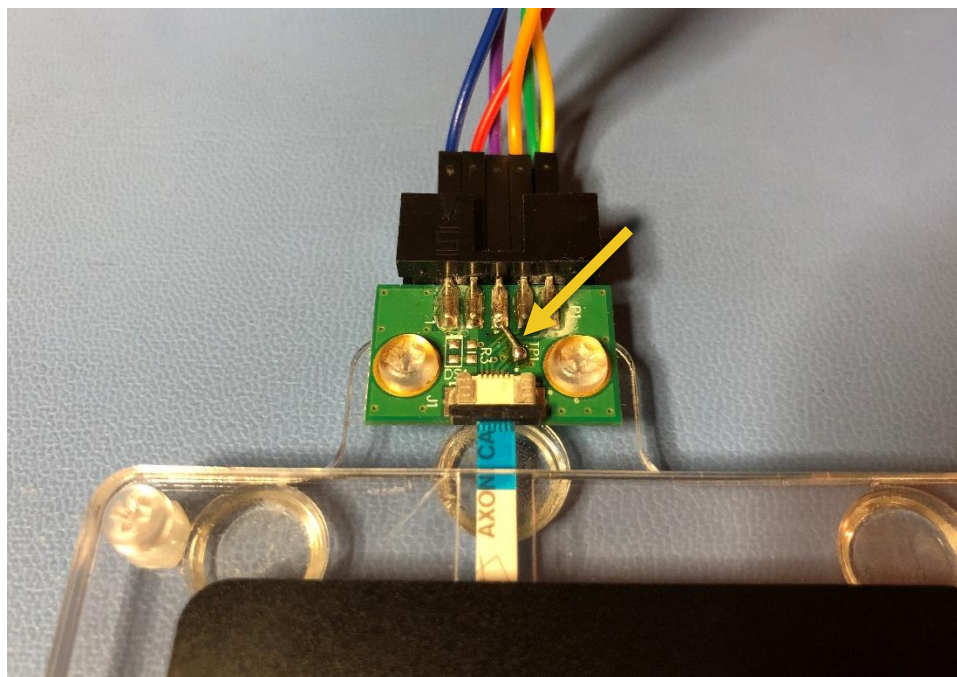
## Trackpad Usage

The newly enabled trackpad behaves like other multitouch trackpads found in various notebook computers. Tapping with one, two or three fingers represents a left-click, right-click or middle-click, respectively. Use two fingers to scroll, and double-tap to drag.

## Programming Firmware

The IQS550 Linux kernel driver can use the device's bootloader to program the device with firmware exported from Azoteq's IQS550-B000 GUI. The driver can even recover blank devices, as well as devices originally programmed with legacy A000 firmware.

To program the IQS550EV02 with new firmware, the module must first be modified by shorting pin 5 of the P1 connector to the nearby TP1 test point (reset) as shown in the following figure.





Please note the following precautions:

- Remove power from the Raspberry Pi before performing the modification.
- The IQS550EV02 must not be connected to the Azoteq CT210A configuration tool while the modification remains in place.

Use a sixth female-to-female jumper wire to connect pin 5 of the P1 connector to any available GPIO along the Raspberry Pi GPIO header (e.g. pin 11, i.e. GPIO17). Refer to Raspberry Pi's [official website](#) for schematics and pin number vs. GPIO number mapping.

Apply power to the Raspberry Pi, then wait until it boots into the desktop. Press CTRL+ALT+T to open a command-line terminal, then enter the following command to open the Raspberry Pi's boot configuration file.

```
$ sudo nano /boot/config.txt
```

Scroll down to the line where `dtoverlay=iqs550` was added, then add the `reset` parameter followed by the GPIO number as shown below. Note that the GPIO number is not necessarily equal to the pin number.

```
# Additional overlays and parameters are documented /boot/overlays/README
dtoverlay=iqs550,reset=17
```

Press CTRL+O followed by the enter key to save the changes. Press CTRL+X to exit the editor and return to the command prompt, then enter the following command to reboot the Raspberry Pi.

```
$ reboot
```

Once the Raspberry Pi reboots into the desktop, use a file transfer tool such as [WinSCP](#) to copy the new firmware image to the `/lib/firmware` directory of the Raspberry Pi. This operation requires root access through SSH; refer to any number of online resources for more information.

Press CTRL+ALT+T to open a command-line terminal, then enter the following command followed by the root password as shown below.

```
$ su
Password:
```

Write the name of the new firmware image (e.g. `IQS550_B000_Test_V1.0_BL.hex`) to the driver's firmware programming attribute as shown below.

```
# echo IQS550_B000_Test_V1.0_BL.hex > /sys/bus/i2c/devices/1-0074/fw_file
```



After several seconds, the command-line terminal returns to the command prompt and the updated IQS550EV02 is ready to use. The current firmware revision can be read at any time using the following command.

```
# cat /sys/bus/i2c/devices/1-0074/fw_info
40.15.2.2:1.0
```

The numbers to the left of the colon represent the product and project numbers, followed by the major and minor firmware revisions. The numbers to the right of the colon represent the exported major and minor revisions.

## Troubleshooting

In the event of unexpected behavior, enter the following command to display any errors reported by the IQS550 Linux kernel driver.

```
$ dmesg | grep 1-0074
[  9.092035] input: iqs550 as /devices/platform/soc/3f804000.i2c/i2c-1/1-0074/input/input3
```

The above example confirms the IQS550 was registered with the Linux kernel input subsystem as expected. On the other hand, the following example suggests the Raspberry Pi could not communicate with the IQS550.

```
$ dmesg | grep 1-0074
[  8.935479] iqs5xx 1-0074: Failed to read from address 0x0000: -121
[  8.952475] iqs5xx 1-0074: Failed to open bootloader: -121
[  8.952551] iqs5xx: probe of 1-0074 failed with error -121
```

In this case, check all connections between the Raspberry Pi and the IQS550EV02, then power-cycle the Raspberry Pi.

If the command returns nothing at all, it suggests the IQS550 Linux kernel driver was not loaded in the first place. In that case, check that the Raspberry Pi firmware is up-to-date and the boot configuration file was modified according to the previous instructions.

If the IQS550 was shown to be registered with the Linux kernel input subsystem but the trackpad still does not behave as expected, enter the following command to confirm the Multitouch X driver recognizes the IQS550.

```
$ grep iqs550 /var/log/Xorg.0.log | grep -m 1 multitouch
[  65.118] (II) Using input driver 'multitouch' for 'iqs550'
```



If the command returns nothing at all, it suggests the Multitouch X driver was not installed or configured properly. In that case, check that the Multitouch X driver was installed and its configuration file was created according to the previous instructions.

## Further Reading

The `/boot/overlays/README` file describes additional parameters that can be used to configure the IQS550 Linux kernel driver.

These parameters are useful in case the interrupt signal must be relocated to a different pin along the GPIO header of the Raspberry Pi, or the trackpad axes must be swapped or inverted to align with the display's orientation.

# LABUNDY.COM