

# Raspberry Pi As Robotic Arm Controller with Flick Hat

## [Preparing the Pi](#)

This article is based on the standard Raspbian Stretch and the first thing you will need to do is to prepare a micro SD card using NOOBS or by flashing an image directly using tools like Etcher from Resin.IO. It doesn't matter whether you get the Lite or the Full version of Raspbian but using the Full will provide a more intuitive platform which will be easier to use. There is plenty of documentation via the Raspberry Pi foundation's website to help you out with this particular step if you are not entirely familiar with it.

Unless you are using a keyboard and monitor connected the whole time to your Raspberry Pi we suggest enabling SSH access so that you can work from your main computer. We recommend using a keyboard, mouse and monitor at least for the following two commands which need to be run directly on the Pi.

### Parts for this project

- Raspberry Pi A+,B+ and above
- An SD card with your choice of OS (I am using Raspbian Jessie Full)
- Maplin Robotic Arm [Buy Here](#)
- Flick Board compatible with you Pi

This slideshow requires JavaScript.

Once the Raspbian has boot up you will need to open a terminal session (*Menu -> Accessories -> Terminal*) and run the following command:

Choose the interfacing options then SSH and enable the SSH Server. The SSH server is now running on your Raspberry Pi and you can use your favourite SSH client to connect to your Pi to go through the rest of the steps. Whether you are using Linux, Mac OS or Windows you can rely in some cases on readily available SSH clients or install a specialised program from the internet like Putty for Windows.

You should be able to connect to your Pi by simply using its mDNS name raspberrypi.local, alternatively you will need to find its IP address by running:

which you can run using a terminal window in a similar way you did to run the configuration step above. It will typically be in the form of 192.168.x.y.

Once the SSH server has been enabled and the IP address is known we can then go back to our main computer and establish a connection via our SSH client. Hurray! We are in. Next we need to go through the basic updates and installations that will prepare Raspbian to run all the software required for the Flick HAT and the robotic arm. Execute the following commands one at a time and follow the installation prompts.

Installing the above might require some time, as they say, get a cup of tea to let all the software install fully.

At this point we will need to install the Flick Board software by running the command below:

The installation is kicked off automatically. Among other things this will create a directory in your home folder where all the software from Flick's Github repository is stored. Feel free to explore the contents of the directory where you can find Python code examples and information about the Flick boards.

To finish the installation section we need the various programs and software which allow the Pi to communicate to the robotic arm. If you are reading an electronic version of this article we would suggest, if possible, to copy/paste each of block of code rather than typing out each line separately as this will reduce typing mistakes and some degree of inevitable frustration. By all means it is always good to try and understand what the commands do.

To install the PyUsb library then:

This library provides easy USB access from Python and is a [free contribution from Wander Lairson Costa](#) aka walac which has kindly developed and made this library available on Github.

The next library we will need to install allows the Pi to communicate to the Arm via USB. It has been [created by Matt Dyson](#) and once again kindly made available to everyone.

To install the RobotArmControl program execute the following commands:

With this we have concluded our installation part and we are ready to move to testing the whole setup. Time to have a play with our kit!

## [Testing your setup](#)



When this program is running you should be able to see inputs and values as you move your hand over the Flick board.

You will notice that more than one gesture will be triggered when you are hovering your hand over Flick. This is absolutely normal as for example a double tap is first of all a tap and it's also a touch in some cases. Train yourself with this demo and find out what's the best way to operate the board. When using your finger over the board for example try to keep your hand fairly vertical in relation to the surface of the board otherwise the rest of the hand will also be detected and lead to misreadings. The best way to get "clean" gestures is really to use your finger as if it was a pen or a magic wand if you prefer :)

With all of the above done the only thing left to do is to finally run the arm control demo which will be displayed in a very similar window. Run the following script which allows control of the Arm via the Flick board.

We need to use `sudo` with the `armcontrol` because of the `RobotArm` library which requires direct access to the serial

Follow the information provided on the screen to know which gesture moves which part of the robotic arm.

If you feel more adventurous and would like to understand more about what happens behind the curtains we invite you to check the source code of the flick-armcontrol to see how you can attach your own code to the various events provided by the [Flick library](#) which is a [fork of Pimoroni's Skywriter](#). Both libraries are available via Github and offer plenty of documentation and code examples.

## [FAQ](#)

### [The script complains it is not finding the robotic arm](#)

A part from obviously checking your batteries and the switch on the Robot Arm to be in the ON position we have found out that there are out there at least two batches of boards from the same manufacturer. The problem is to do with the USB identifier which is either 1267:0000 or 1267:0001. You can make sure of which is yours by running

You will get something similar to either of these lines in your output:

The *RobotArm.py* is preconfigured with the 0000 version. If your board is the other type then you will need to run:

and change the PRODUCT line to:

save the file and exit.

So that *testRobotArm.py* is working you will need to also change that same line in the original *armcontrol* folder.