GND_TO_VCC

Turn Your Raspberry Pi Zero into a USB Keyboard (HID)

Gnd_To_Vcc April 17, 2020 Uncategorized

In this project you'll learn how to turn a Raspberry Pi Zero board into a USB keyboard or HID (Human Interface Device). After following some simple steps, you can write a Python script to make your Pi act as a USB keyboard.

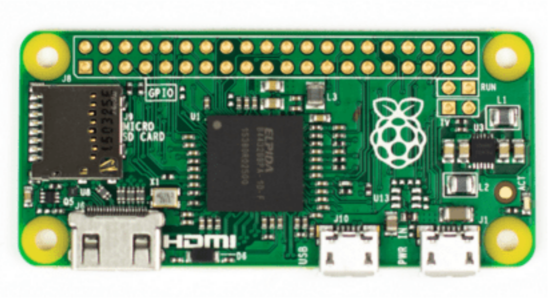
Prerequisites:

- You need a Raspberry Pi Zero board
- You should be familiar with the Raspberry Pi – **read Getting Started with Raspberry Pi**.
- You should have the Raspbian operating system installed in your Raspberry Pi – **read Installing Raspbian Lite, Enabling and Connecting with SSH**.

If you like home automation and you want to learn more about Node-RED, Raspberry Pi, ESP8266 and Arduino.

Parts Required

For this project you'll need a Raspberry Pi Zero board. **Important:** this tutorial doesn't work with a Raspberry Pi 3 board.



1. Enabling Modules and Drivers

These next steps to prepare the Pi Zero board are based on the instructions from **iSticktoit**. First, you need to run these three commands to enable the necessary modules and drivers:

```
pi@raspberrypi:~ $ echo "dtoverlay=dwc2" | sudo tee -a /boot/config.txt
pi@raspberrypi:~ $ echo "dwc2" | sudo tee -a /etc/modules
pi@raspberrypi:~ $ sudo echo "libcomposite" | sudo tee -a /etc/modules
```

2. Configuring the Gadget

Now, you have to define your Pi Zero (HID gadget) as a USB keyboard. The configuration is done via *ConfigFS*, a virtual file system located in */sys/*.

Creating the config script

The configuration is volatile, so it must run on each startup. Create a new file called *isticktoit_usb* in */usr/bin/* and make it executable:

```
pi@raspberrypi:~ $ sudo touch /usr/bin/isticktoit_usb
pi@raspberrypi:~ $ sudo chmod +x /usr/bin/isticktoit_usb
```

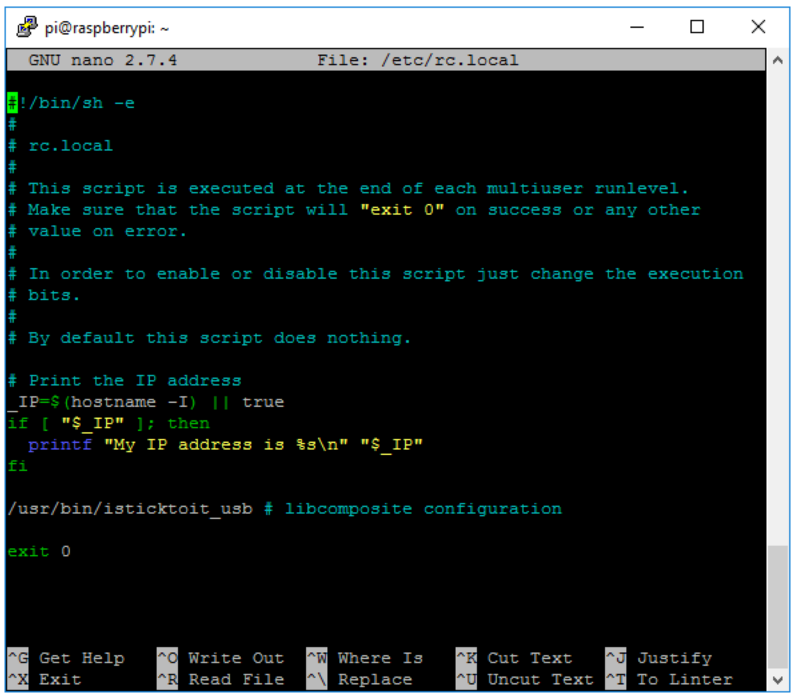
Then, you need to run this script automatically at startup. Open */etc/rc.local* with this command:

```
pi@raspberrypi:~ $ sudo nano /etc/rc.local
```

Add the following before the line containing ***exit 0***:

```
/usr/bin/isticktoit_usb # libcomposite configuration
```

Here's how your file should look like (to save the file, press Ctrl+X followed by Y and Enter):



```
pi@raspberrypi: ~
  GNU nano 2.7.4              File: /etc/rc.local

#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
  printf "My IP address is %s\n" "$_IP"
fi

/usr/bin/isticktoit_usb # libcomposite configuration

exit 0



^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Linter
```

3. Creating the gadget

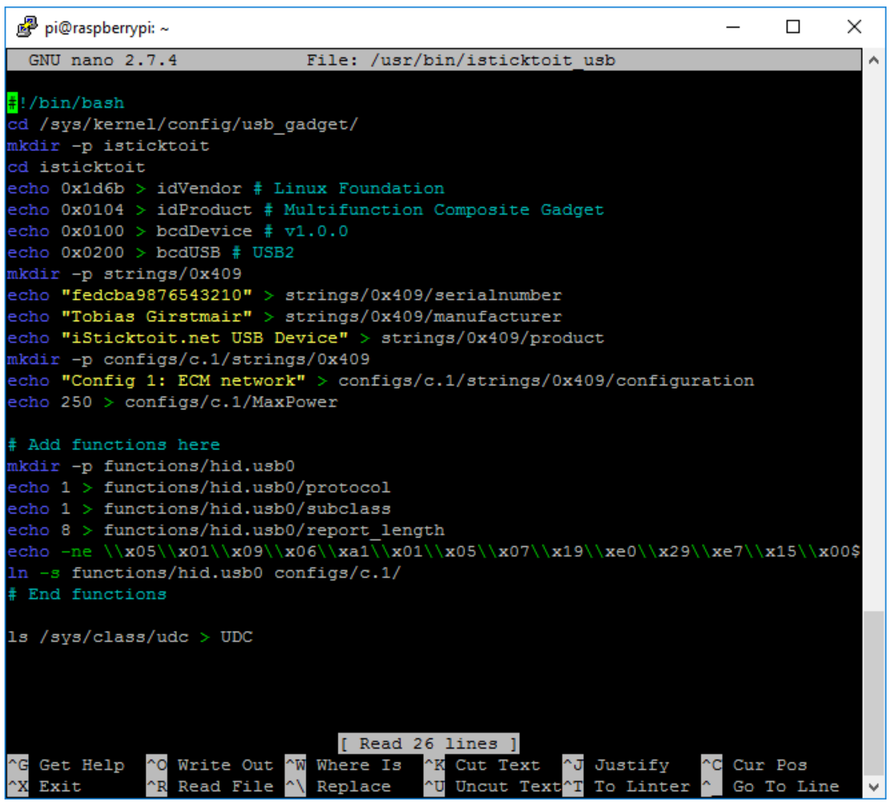
For this project, we will turn the Raspberry Pi into a USB keyboard, but you could make it work as a Serial adapter, Ethernet adapter, and Mass Storage. Open the file with:

```
pi@raspberrypi:~ $ sudo nano /usr/bin/isticktoit_usb
```

Leave the default values, but you could even change the serial number, manufacturer and product name to fit your specific needs.

```
#!/bin/bash
cd /sys/kernel/config/usb_gadget/
mkdir -p isticktoit
cd isticktoit
echo 0x1d6b > idVendor # Linux Foundation
echo 0x0104 > idProduct # Multifunction Composite Gadget
echo 0x0100 > bcdDevice # v1.0.0
echo 0x0200 > bcdUSB # USB2
mkdir -p strings/0x409
echo "fedcba9876543210" > strings/0x409/serialnumber
echo "Tobias Girstmair" > strings/0x409/manufacturer
echo "iSticktoit.net USB Device" > strings/0x409/product
mkdir -p configs/c.1/strings/0x409
echo "Config 1: ECM network" > configs/c.1/strings/0x409/configuration
echo 250 > configs/c.1/MaxPower

# Add functions here
mkdir -p functions/hid.usb0
echo 1 > functions/hid.usb0/protocol
echo 1 > functions/hid.usb0/subclass
echo 8 > functions/hid.usb0/report_length
echo -ne
\\x05\\x01\\x09\\x06\\xa1\\x01\\x05\\x07\\x19\\xe0\\x29\\xe7\\x15\\x00\\x25\\x01\\x75\\x01\\x95\\x08\\x81\\x02\\x95\\x01\\x75\\x08\\x81\\x03\\x95\\x05\\x75\\
> functions/hid.usb0/report_desc
ln -s functions/hid.usb0 configs/c.1/
# End functions

ls /sys/class/udc > UDC
```

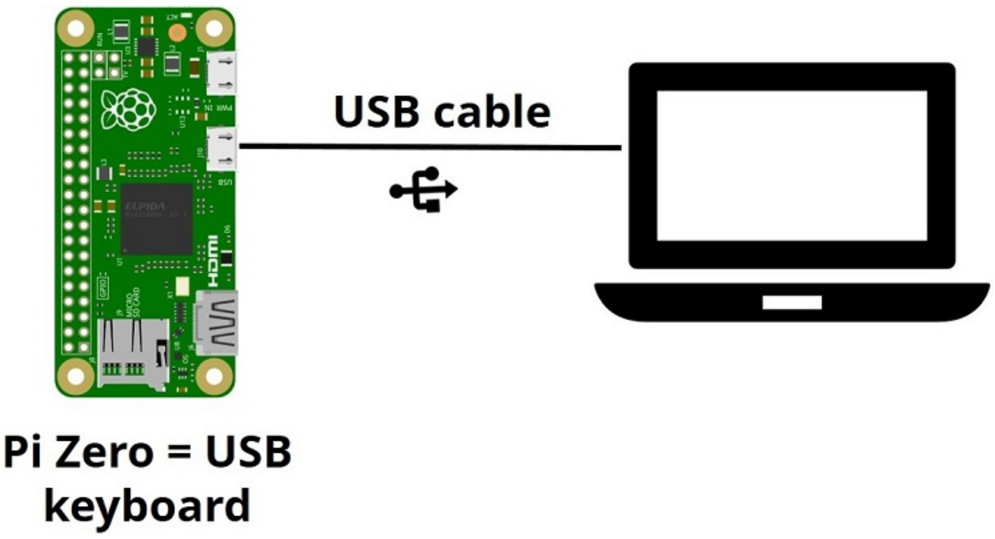
Here's how your file should look like in the end (to save the file, press Ctrl+X followed by Y and Enter):



```
#!/bin/bash
cd /sys/kernel/config/usb_gadget/
mkdir -p isticktoit
cd isticktoit
echo 0x1d6b > idVendor # Linux Foundation
echo 0x0104 > idProduct # Multifunction Composite Gadget
echo 0x0100 > bcdDevice # v1.0.0
echo 0x0200 > bcdUSB # USB2
mkdir -p strings/0x409
echo "fedcba9876543210" > strings/0x409/serialnumber
echo "Tobias Girstmair" > strings/0x409/manufacturer
echo "iSticktoit.net USB Device" > strings/0x409/product
mkdir -p configs/c.1/strings/0x409
echo "Config 1: ECM network" > configs/c.1/strings/0x409/configuration
echo 250 > configs/c.1/MaxPower

# Add functions here
mkdir -p functions/hid.usb0
echo 1 > functions/hid.usb0/protocol
echo 1 > functions/hid.usb0/subclass
echo 8 > functions/hid.usb0/report_length
echo -ne \\x05\\x01\\x09\\x06\\xa1\\x01\\x05\\x07\\x19\\xe0\\x29\\xe7\\x15\\x00$
ln -s functions/hid.usb0 configs/c.1/
# End functions

ls /sys/class/udc > UDC



                              [ Read 26 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Linter  ^_ Go To Line
```

4. Python Script

After preparing your Raspberry Pi Zero, connect it to a laptop or desktop computer through the micro USB port that is used for data and peripherals. That micro USB will both power the Pi Zero and act as a keyboard to the connected computer.



Establish an SSH connection with your Pi and use the next command to create a new Python script:

```
pi@raspberrypi:~ $ nano RPi_Keyboard_Example.py
```

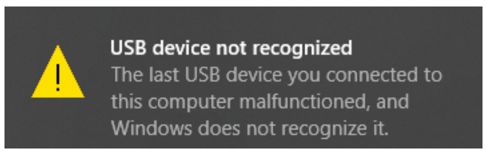
Copy and paste the next Python script to your Raspberry Pi.

```python
#!/usr/bin/env python3
NULL_CHAR = chr(0)

def write_report(report):
    with open('/dev/hidg0', 'rb+') as fd:
        fd.write(report.encode())

# Press a
write_report(NULL_CHAR*2+chr(4)+NULL_CHAR*5)
# Release keys
write_report(NULL_CHAR*8)
# Press SHIFT + a = A
write_report(chr(32)+NULL_CHAR+chr(4)+NULL_CHAR*5)

# Press b
write_report(NULL_CHAR*2+chr(5)+NULL_CHAR*5)
# Release keys
write_report(NULL_CHAR*8)
# Press SHIFT + b = B
write_report(chr(32)+NULL_CHAR+chr(5)+NULL_CHAR*5)

# Press SPACE key
write_report(NULL_CHAR*2+chr(44)+NULL_CHAR*5)

# Press c key
write_report(NULL_CHAR*2+chr(6)+NULL_CHAR*5)
# Press d key
write_report(NULL_CHAR*2+chr(7)+NULL_CHAR*5)

# Press RETURN/ENTER key
write_report(NULL_CHAR*2+chr(40)+NULL_CHAR*5)

# Press e key
write_report(NULL_CHAR*2+chr(8)+NULL_CHAR*5)
# Press f key
write_report(NULL_CHAR*2+chr(9)+NULL_CHAR*5)

# Release all keys
write_report(NULL_CHAR*8)
```

Demonstration

Let's test it, if you plug the Pi Zero to Computer #1, after a few seconds you'll see an alert message or sound that indicates that a keyboard was connected successfully.

Sometimes you might see this warning message saying "USB device not recognized". Throughout my tests, I found that you can ignore this warning message and your Pi Zero works as a keyboard without any additional configuration or drivers installation. So, you can continue and it will work just fine.

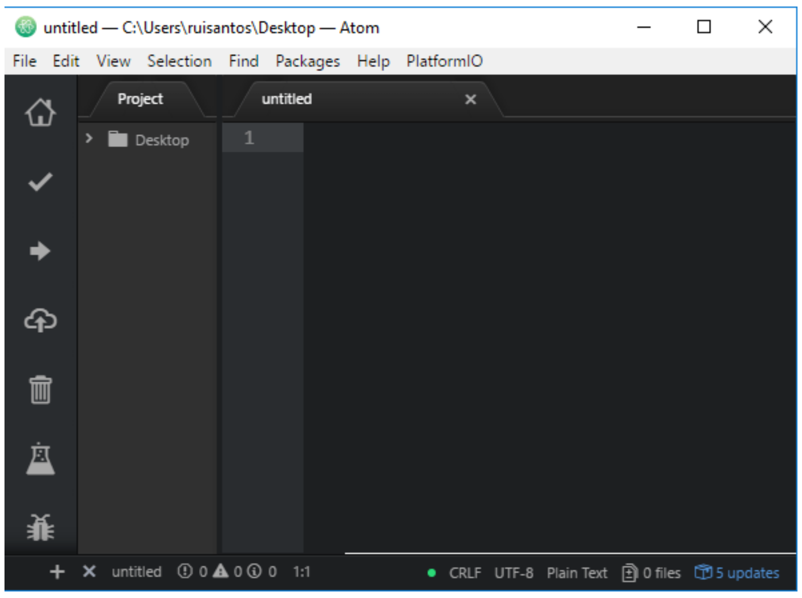


USB device not recognized

The last USB device you connected to this computer malfunctioned, and Windows does not recognize it.

Computer #1

Open any text editor program and leave your cursor in the new file:

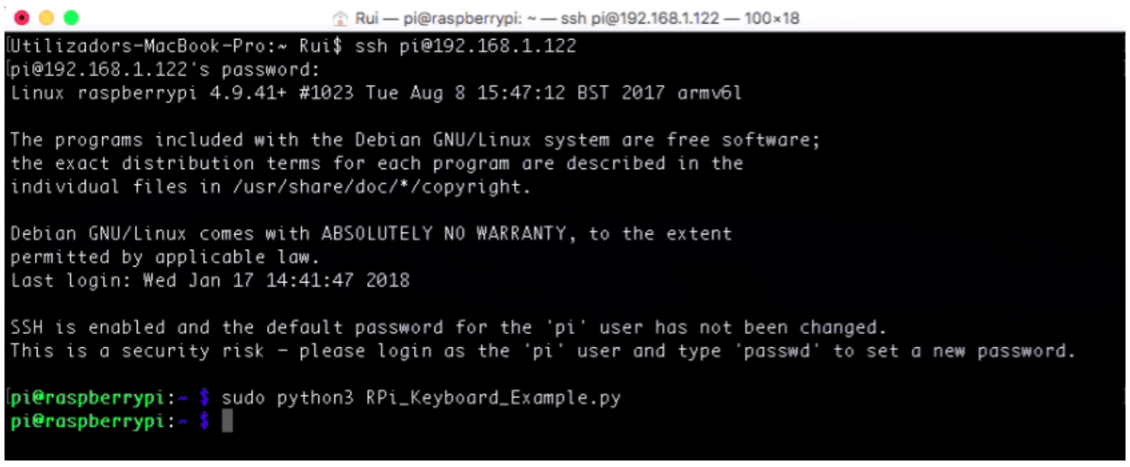


untitled — C:\Users\ruisantos\Desktop — Atom
File Edit View Selection Find Packages Help PlatformIO
Project
untitled
Desktop
1
untitled
0 0 0
1:1
CRLF UTF-8 Plain Text
0 files
5 updates

Computer #2

Establish an SSH connection with your Pi Zero and run the Python script created earlier:

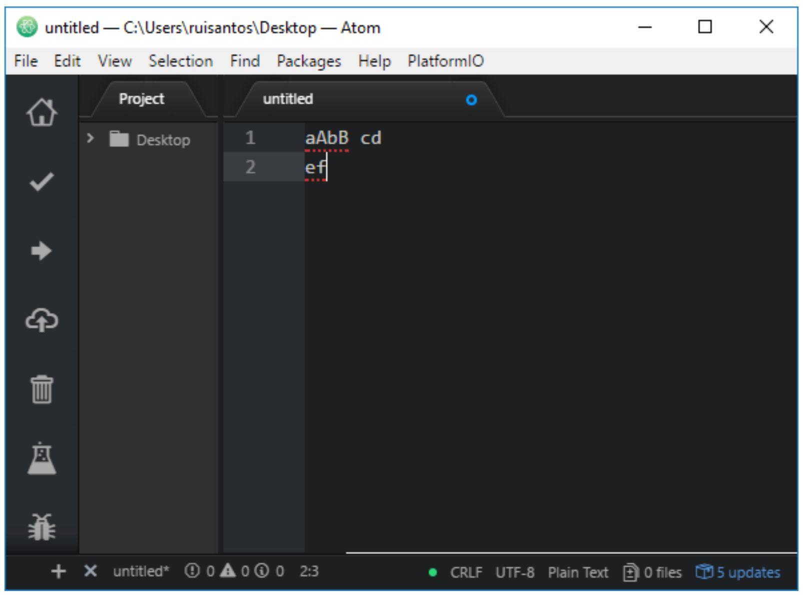
```
pi@raspberrypi:~ $ sudo python3 RPi_Keyboard_Example.py
```



```
Utilizadors-MacBook-Pro:~ Rui$ ssh pi@192.168.1.122
pi@192.168.1.122's password:
Linux raspberrypi 4.9.41+ #1023 Tue Aug 8 15:47:12 BST 2017 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Jan 17 14:41:47 2018

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.

pi@raspberrypi:~ $ sudo python3 RPi_Keyboard_Example.py
pi@raspberrypi:~ $ 
```

Result

The script will press these keys in that order: a – A – b – B – Space key – c – D – Enter/Return key – e – f.


untitled — C:\Users\ruisantos\Desktop — Atom
File Edit View Selection Find Packages Help PlatformIO
Project
untitled
Desktop
untitled* 0 0 0 2:3
CRLF UTF-8 Plain Text 0 files 5 updates

```
aAbB cd
ef
```

You can customize the Python script to act as a keyboard and press any other character sequence.

Note: the Pi Zero also acts as a keyboard when connected to a Mac or Linux machine without any additional changes.

Taking It Further

You can use Table 12: Keyboard/Keypad Page from this USB HID PDF to find the ID of each key that you would assign in the Python script.

Here's a section of Table 12. The Usage ID (Dec) column contains the number that you need to use in your Python script to refer to a key press:

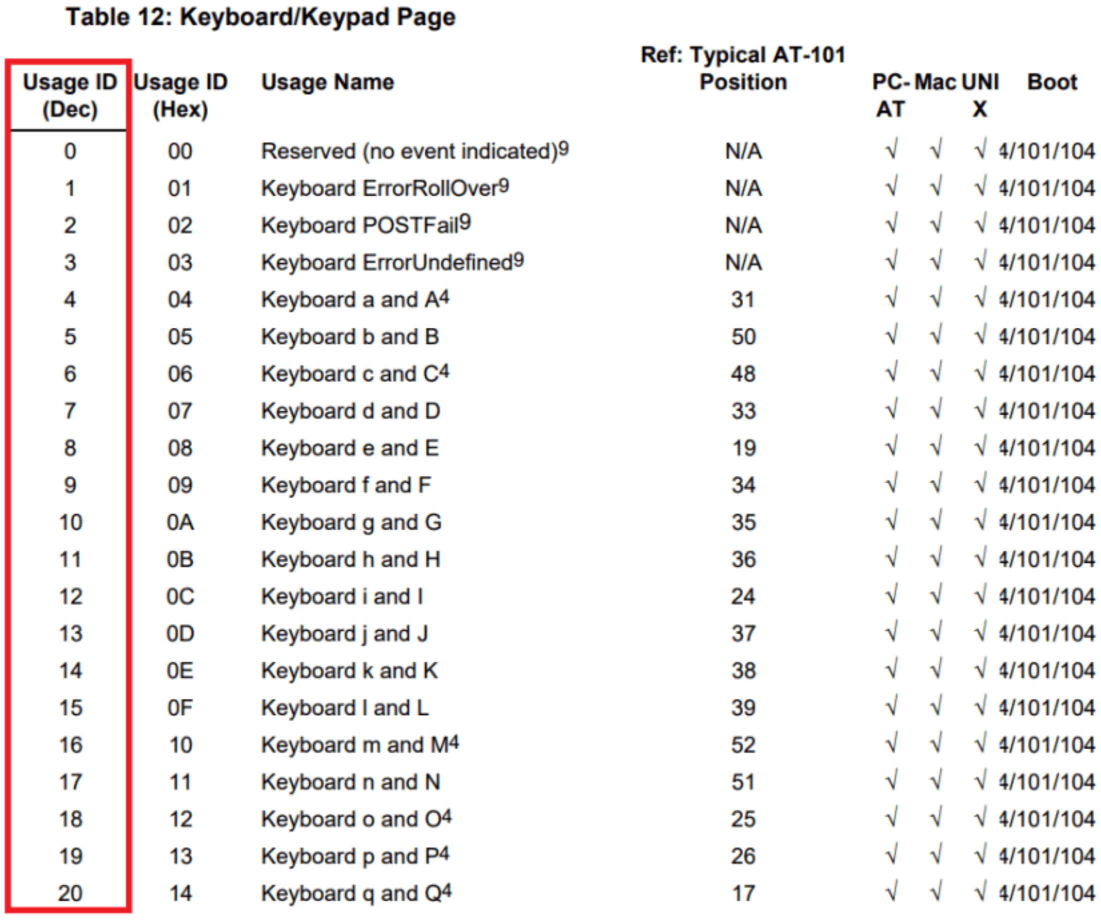


Table 12: Keyboard/Keypad Page

Usage ID (Dec)	Usage ID (Hex)	Usage Name	Ref: Typical AT-101 Position	PC-AT	Mac	UNIX	Boot
0	00	Reserved (no event indicated)[9]	N/A	√	√	√	4/101/104
1	01	Keyboard ErrorRollOver[9]	N/A	√	√	√	4/101/104
2	02	Keyboard POSTFail[9]	N/A	√	√	√	4/101/104
3	03	Keyboard ErrorUndefined[9]	N/A	√	√	√	4/101/104
4	04	Keyboard a and A[4]	31	√	√	√	4/101/104
5	05	Keyboard b and B	50	√	√	√	4/101/104
6	06	Keyboard c and C[4]	48	√	√	√	4/101/104
7	07	Keyboard d and D	33	√	√	√	4/101/104
8	08	Keyboard e and E	19	√	√	√	4/101/104
9	09	Keyboard f and F	34	√	√	√	4/101/104
10	0A	Keyboard g and G	35	√	√	√	4/101/104
11	0B	Keyboard h and H	36	√	√	√	4/101/104
12	0C	Keyboard i and I	24	√	√	√	4/101/104
13	0D	Keyboard j and J	37	√	√	√	4/101/104
14	0E	Keyboard k and K	38	√	√	√	4/101/104
15	0F	Keyboard l and L	39	√	√	√	4/101/104
16	10	Keyboard m and M[4]	52	√	√	√	4/101/104
17	11	Keyboard n and N	51	√	√	√	4/101/104
18	12	Keyboard o and O[4]	25	√	√	√	4/101/104
19	13	Keyboard p and P[4]	26	√	√	√	4/101/104
20	14	Keyboard q and Q[4]	17	√	√	√	4/101/104

For example, if you change the number highlighted in red, you can simulate a different key press:

```
write_report(NULL_CHAR*2+chr(4)+NULL_CHAR*5)
```

The number **4** correspondes to keyboard key **a**. You can find in the Usage ID (Dec) column the numbers for your desired key press sequence. If you use number **5** it corresponds to **b**, and so on…

Wrapping Up

You can use this method to make the Raspberry Pi Zero act as password filler or use it as a keystroke injection tool. That way you can easily create programs that type hundreds of keystrokes per minute.

You may also like some of our most popular Raspberry Pi projects:

- Email Alert System on Location Change with Raspberry Pi and GPS Module
- Node-RED with Raspberry Pi Camera (Take Photos)
- Video Streaming with Raspberry Pi Camera

Thanks for reading.

Gnd_To_Vcc April 17, 2020 Uncategorized

Published by Gnd_To_Vcc

Here to spread my knowledge . Knowledge should always be spread not stored. View more posts

4 thoughts on "Turn Your Raspberry Pi Zero into a USB Keyboard (HID)"

Pingback: 25+ Raspberry Pi Projects

Pingback: **25+ Raspberry Pi Projects – Gnd_To_Vcc**

Daniel Tompkins
March 1, 2021 at 12:44 am

Was trying this and I kept receiving `ls: write error: Device or resource busy` when running the isticktoit_usb script.
Turned out I had a faulty usb cable after a great deal of debugging and disappointment.

I had used 2 USB cables that were power only and would not work, then a third that “worked”— but ended up being faulty. The 4th finally REALLY worked. Use the `dmesg` command. It’s your friend.

★ Like

Reply

Steffen Lorenz
April 19, 2021 at 2:34 pm

How to press [COMMAND] + [SPACE] at the same time? btw [COMMAND] = [GUI LEFT]

★ Like

Reply

Leave a Reply