

Hacking USB - USB Pico Ducky



Documentation & Guide

Taha Canturk

2023-04-18

Disclaimer

This hacking USB is intended only for educational purposes and should not be used for any unlawful activity. Any use of this USB for any purpose other than education is strictly prohibited. The creators of this USB are not responsible for any misuse of the USB or any damage that may occur from its use. The user accepts full responsibility for any and all consequences of the use of the USB.

This USB is provided "as is" without warranty of any kind, either expressed or implied, including but not limited to, implied warranties of merchantability and fitness for a particular purpose.

By using this USB, the user agrees to indemnify and hold harmless the creators of this USB and its affiliates from and against any claims, damages, losses, liabilities, costs, and expenses (including attorney's fees) arising out of or related to the use of the USB, including but not limited to any claims for libel, violation of privacy, infringement of copyright, trademark, or other intellectual property rights.

The user assumes all risk and responsibility for the use of the USB and agrees to comply with all applicable laws. The user further agrees to not use the USB for any illegal activities or use it to gain unauthorized access to any computer system, network, or other protected resource. The user also agrees not to use the USB in any manner that could damage, disable, overburden, or impair this USB or any related services or networks.

By using this USB, the user acknowledges that they have read and understood this disclaimer and agree to be bound by its terms and conditions. The user further agrees that this disclaimer shall be governed by and construed in accordance with the laws of the jurisdiction in which the user has obtained the USB. The user also agrees that any dispute arising out of or related to this disclaimer shall be subject to the exclusive jurisdiction of the courts located in the jurisdiction in which the user has obtained the USB.

You may not:

- (a) decompile, disassemble, reverse engineer or otherwise attempt to derive the source code of the Program;
- (b) distribute, rent, lease, lend, sublicense, or otherwise transfer the Program;
- (c) publish or otherwise disseminate any performance information or analysis (including, without limitation, benchmarks) relating to the Program;
- (d) use the Program to develop any software, application or other program that is, directly or indirectly, competitive with or in any way a substitute for the Program;
- (e) remove or alter any trademark, logo, copyright or other proprietary

notices, legends, symbols or labels in the Program or Documentation; or

(f) modify the Program

3. Ownership.

The Program and all rights therein, including all copyright, title and interest in and to the Program, shall remain the exclusive property of Taha Canturk. The terms of use does not transfer any ownership or intellectual property rights to You. All rights not specifically granted are reserved by Taha Canturk.

4. Disclaimer of Warranty.

THE PROGRAM IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND. TAHA CANTURK DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. FURTHERMORE, TAHA CANTURK DOES NOT WARRANT OR REPRESENT THAT THE PROGRAM WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE PROGRAM WILL BE UNINTERRUPTED OR ERROR FREE.

5. Limitation of Liability.

IN NO EVENT SHALL TAHA CANTURK BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL OR PUNITIVE DAMAGES, INCLUDING, BUT NOT LIMITED TO, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION OR OTHER PECUNIARY LOSS ARISING OUT OF THE USE OF OR INABILITY TO USE THE PROGRAM, EVEN IF TAHA CANTURK HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

6. Termination.

This user disclaimer (Terms of Use) terminates upon Your breach of any of the terms and conditions of this Disclaimer. In the event of termination, You shall destroy all copies of the Program.

7. General Provisions.

If any provision of this Terms of Use is invalid or unenforceable under applicable law, such provision or part shall be interpreted to give effect to the intent of the parties and the remainder of this Disclaimer shall remain in full force and effect. The failure by Taha Canturk to exercise or enforce any right or provision of this Terms of Use shall not constitute a waiver of such right or provision. This Terms of Use constitutes the entire agreement between You and Taha Canturk with respect to the Program and supersedes all prior or contemporaneous understandings, agreements and communications between You and Taha Canturk.

8. Severability.

If any provision of this terms of use is held to be invalid or unenforceable, such provision shall be struck out and the remaining provisions shall be enforced.

10. No Assignment.

The terms of use is personal to You and You may not assign or otherwise transfer Your rights hereunder.

By downloading, installing and/or using the Program, You acknowledge that You have read the terms of use (Disclaimer), understand it and agree to be bound by its terms and conditions. You further agree that it is the complete and exclusive statement of the agreement between You and Taha Canturk which supersedes any prior agreement, oral or written, and any other communications between You and Taha Canturk relating to the Program.

Contents

1	App	i
1.1	What is the USB Pico Ducky	i
1.2	payloads	i
1.3	Saving & Editing	i
1.4	Shortcuts	i
1.5	Cosmetics	i
1.5.1	Existing Themes	ii
1.5.2	Custom Themes	iii
2	The USB	iii
2.1	Uploading Payloads	iii
2.2	Execute	iv
2.2.1	Before Execution	iv
2.2.2	Run Duckyscript Program	iv
2.2.3	Run C/C++/Python Program	iv
3	Duckyscript Language	iv
3.1	Tutorial	iv
3.2	Keywords	iv
4	Setup	iv
4.1	Reset	iv
4.1.1	When to Reset USB	iv
4.1.2	How to Reset USB	iv
4.2	Hack Mode On	v
4.2.1	Duckyscript Language	v
4.3	Hack Mode Off	v
4.3.1	C/C++/MicroPython Language	v
4.3.2	Upload	vi

1 App

1.1 What is the USB Pico Ducky

The USB Pico Ducky is a Hacking USB. The physical device is a Raspberry Pi Pico. A Hacking USB is a device that fools a computer into thinking it is an external input device such as a keyboard, this means that the computer is fooled into thinking that the user is typing while the usb is the one inputting keystrokes. Ask yourself, what can I do using an input device (e.g. Keyboard)? Whatever you can do using an inputer peripheral, you can do using USB Pico Ducky. If you want to, as mentioned in section 4.3, you can turn the hacking mode off and program the Raspberry Pi Pico as a normal microcontroller.

This Application is an IDE (Integrated Development Environment) for Duckyscript. Duckyscript is a simple scripting language made by the company Hak5, originally known for the USB Rubber Ducky, which costs around \$130 CAD (plus tax & fees). Even though the USB Rubber Ducky is more than double the price, the USB Rubber Ducky doesn't come with an IDE while the USB Pico Ducky does.

This might seem too good to be true. but, there is one main difference, it's that only Duckyscript 1.0 is supported on a USB Pico Ducky while USB Rubber Ducky supports Duckyscript 3.0. This will not affect most users because Duckyscript 3.0 just provides extra keywords but the possibilities are just as endless without them.

1.2 payloads

Payloads are a file written in duckyscript. Payloads basically have instructions given to the USB so that the USB knows what keystrokes to input.

A hacking USB only supports one mode at a time, it can either execute a payload when connected to a device, or it can be in setup mode where you upload the payload (more details can be found in section 2.1),

1.3 Saving & Editing

1.4 Shortcuts

1.5 Cosmetics

The custom colors you can pick are for

1. **Background:** Background of the IDE
2. **Comment:** Colors of comment. Line after `REM`

[*REM*]

3. **Starting keywords:** keywords that generally start a line of duckyscript code

[*DELAY STRING PRINT IMPORT DEFAULT_DELAY DEFAULTDELAY LED*]

4. **Fkeys:** F-keys

[*F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12*]

5. **Shortcut keys:** starting keys for shortcuts

[*ALT CTRL CONTROL SHIFT SPACE ENTER BACKSPACE*
TAB CAPSLOCK ESC ESCAPE]

6. **Arrows:** Arrowkeys

[*UP UPARROW DOWN DOWNARROW LEFT LEFTARROW*
RIGHT RIGHTARROW]

7. **Windows:** Windows Button keywords

[*WINDOWS GUI*]

8. **Chars:** Colors of single characters

[*A B C D E F G H I J K L M N O P Q R S T U V W X Y Z*]

9. **Uncommon:** Uncommon keywords

[*APP MENU BREAK PAUSE DELETE END*
HOME INSERT NUMLOCK PAGEUP PAGEDOWN
PRINTSCREEN SCROLLLOCK]

10. **Numbers:** Base 10 numbers (numbers from 0-9)

[*0 1 2 3 4 5 6 7 8 9 10*]

11. **Text:** Non-keyword text


12. **Textbubble:** Background of text coding box (the background of the IDE code box)

13. **Background sidebar:** Background color of the file picking sidebar

14. **Color sidebar:** Color of the text of sidebar

1.5.1 Existing Themes

There are 2 already existing themes by default, Light and dark theme. You can access them by

View(left  button) → *Themes* → *White/Black*


OR

Or press **CTRL+ALT+B** for **DARK** Theme

and press **CTRL+ALT+W** for **LIGHT** Theme

1.5.2 Custom Themes

To select a custom theme, you can use the shortcut **CTRL+ALT+T** or you can press select it from the menu by hovering mouse on

View(left  button) → *Themes* → *Custom*

The custom colors are picked for the colors mentioned in section 1.5.

Here are the steps to make a **NEW** theme:

1. Click on **Input Theme Name** Button on the top middle (the button on the right) and type in the new name. Click **Ok**.
2. Click on the checkbox(es), select the ones you want to change to the same color, on the top menu, click the **RGB Color Picker** button on the middle. Now you can select the new color for the selected checkboxes.
3. Now click the **Save** Button on the left to save.

2 The USB

2.1 Uploading Payloads

First, write the duckyscript program, then save it as specified in section 1.3.

To upload payloads onto the USB Pico Ducky, click the upload button , which is right beside the Operating System Selector popup on the toolbar.

You can upload up to 2 files at a time. But only run one at a time.

In duckyscript, it is not possible to write a program that uses multiple files and this is also not necessary. You can add up to 2 payloads (e.g. `payload.dd`, `payload1.dd`) at a time. You can switch between the payloads using the white switch on the USB Pico Ducky

This is useful for when you need different payloads for different OSs. For example, you might want to have a Mac version and a Windows version of a password brute-forcing payload. This is useful for when you aren't sure about the OS of the target computer. **Not even USB Rubber Ducky comes with this feature**

The valid payload names are: `payload.dd`, `payload1.dd`

2.2 Execute

2.2.1 Before Execution

Before Execution, first select the target device's OS, then language (in order). Currently Mac supports less languages than Windows does, therefore make sure to select device's OS first

2.2.2 Run Duckyscript Program

To Execute a duckyscript program, insert the USB into a device while in attack mode. Turn on hack mode as specified in section

2.2.3 Run C/C++/Python Program

To run a C/C++/Python (or any supported language) program. Please refer to section 4.3 to setup the microcontroller.

Since the USB Pico Ducky is a Raspberry Pi Pico MicroController, you can run a C/C++/Python program as you wish as long as the hacking mode is turned off. Please note that the IDE doesn't support these languages. Therefore use a different IDE or text editor of your choosing. The IDE only supports duckyscript.

3 Duckyscript Language

3.1 Tutorial

3.2 Keywords


4 Setup

4.1 Reset

4.1.1 When to Reset USB

Resetting the USB turns hacking mode off. More information can be found in section 4.3. You can also reset back to the USB Pico Ducky (hacking mode) as specified in more detail in section 4.2.

4.1.2 How to Reset USB

You can reset the USB at any time, all it takes is the click of a simple button in the Duckyscript IDE (on the toolbar). The button is . This button

is at the most right on the toolbar. After clicking this button, a popup window will come up. Click the button on the middle (Select USB Pico Ducky Path), then select the path of your USB pico ducky. Now Another button will popup, if you entered the correct path, select the new button (Enable/Disable hacking mode) on the popup.

4.2 Hack Mode On

4.2.1 Duckyscript Language

4.3 Hack Mode Off

When the USB Pico Ducky hack mode is turned off, you can use the micro-controller using C/C++/MicroPython (resources in section 4.3.1).

4.3.1 C/C++/MicroPython Language

Here are some helpful links on setting up C/C++/Python for a Raspberry Pi Pico

Windows Tutorials:

1. [C Language](#)
2. [C++ Language](#)
3. [MicroPython Language](#)

Mac/Linux Tutorials:

1. C Language:
 - [Tutorial in Linux](#)
 - [Tutorial in Mac](#)
2. C++ Language:
 - [Tutorial in Linux](#)
 - [Tutorial in Mac](#)
3. MicroPython Language:
 - [Tutorial in Linux](#)
 - [Tutorial in Mac](#)

Other Helpful Sources:

- [MicroPython Tutorial](#)
- [Raspberry Pi Pico Documentation for C/C++](#)

4.3.2 Upload

The Duckyscript IDE doesn't have C/C++ support at this time. To use C/C++ on the USB Pico Ducky. First, reset the USB as mentioned in section 4.1.2, then use a different IDE/texteditor to write a C/C++ program. You can reserach for more information in section 4.3.1.