

USB RUBBER DUCKY PROJECT

ABOUT

A decorative line graphic consisting of a horizontal line starting from a solid black circle, followed by a diagonal line segment extending downwards and to the right, and finally a horizontal line segment extending to the right edge of the page.

Transforming a standard ATTiny85 board into a USB Rubber Ducky offers a fascinating journey into the world of hardware hacking and cybersecurity tools. The USB Rubber Ducky, a device that emulates a keyboard (HID device) and carries out automated keystroke commands upon connection to a computer, is a potent tool for penetration testing. My project commenced with programming the ATTiny85 to mimic a keyboard, making use of libraries like V-USB for USB communication handling. Crafting scripts in Ducky Script, a straightforward scripting language tailored for this task, was essential to specifying the keystroke sequence for the device to execute. This endeavor not only enriched my grasp of microcontroller programming and USB protocols but also broadened my expertise in scripting and cybersecurity methodologies. It served as a hands-on approach to constructing tools that hold significant relevance in the realm of information security, laying a robust groundwork for delving into more advanced hacking devices and tactics.

WHY DIGISPARK

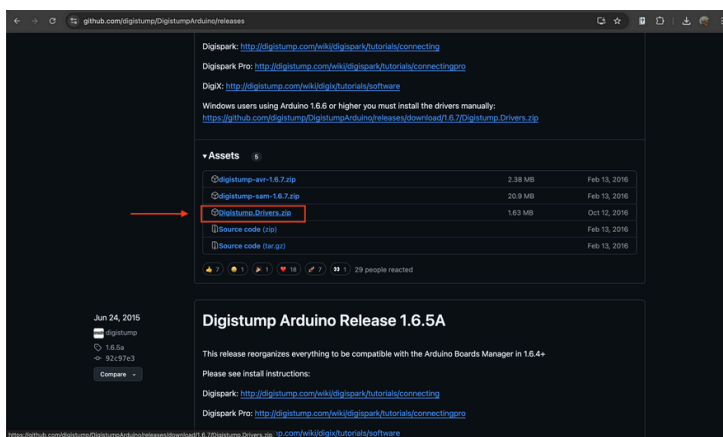
I selected this board due to its compact size, accessibility, and the inclusion of the HID feature.

Additionally, its low power consumption makes it ideal for portable projects. The robust community support and extensive documentation also played a significant role in my decision. With these resources at my disposal, troubleshooting and expanding my knowledge will be much easier. Whether I am working on a small robotics project or developing a custom peripheral, this board's versatility ensures it will meet my needs. One downside is that it lacks voltage regulation, so when the pins are shorted, it can ruin the USB port of your computer. One solution is insulation, but that doesn't completely eliminate the risk.



SET UP

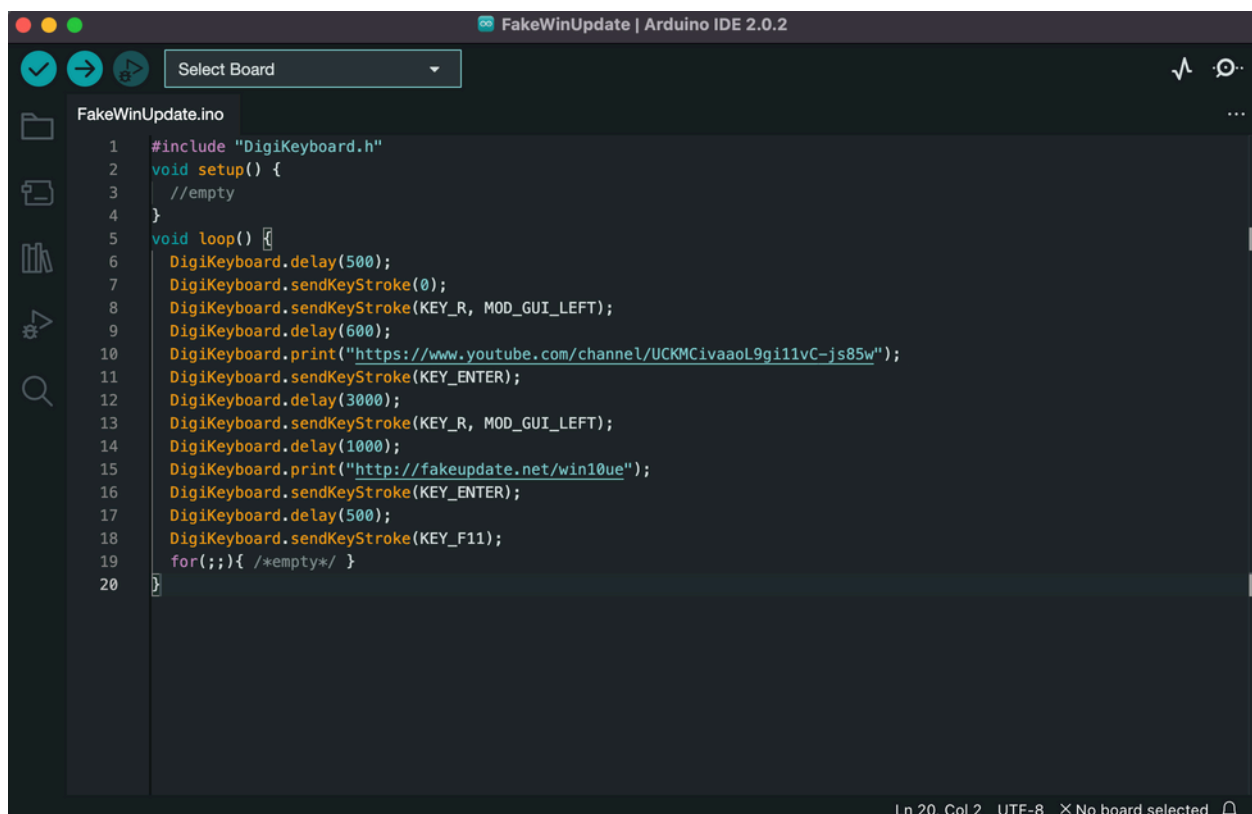
Initially, it is essential to download the Arduino IDE from the official website: <https://www.arduino.cc/en/software>. Subsequently, acquire the requisite drivers for the Digispark to enable its functionality as an HID/USB Rubber Ducky/Bad USB device. These drivers can be obtained from the DigiStump Drivers link: <https://github.com/digistump/DigistumpArduino/releases>.



Once you have downloaded and installed the Arduino IDE and the necessary drivers, the next step is to configure your development environment. Open the Arduino IDE and navigate to 'File' > 'Preferences'. In the 'Additional Board Manager URLs' field, paste the following URL: ``http://digistump.com/package_digistump_index.json``. This will allow the Arduino IDE to recognize the Digispark board.

Next, go to 'Tools' > 'Board' > 'Boards Manager' and search for 'Digistump AVR Boards'. Click 'Install' to add support for the Digispark boards to your Arduino IDE. Once installed, select 'Digispark (Default - 16.5mhz)' from the 'Tools' > 'Board' menu.

WINDOWS SCRIPT



```
1  #include "DigiKeyboard.h"
2  void setup() {
3      //empty
4  }
5  void loop() {
6      DigiKeyboard.delay(500);
7      DigiKeyboard.sendKeyStroke(0);
8      DigiKeyboard.sendKeyStroke(KEY_R, MOD_GUI_LEFT);
9      DigiKeyboard.delay(600);
10     DigiKeyboard.print("https://www.youtube.com/channel/UCKMCivaaol9gi11vC-js85w");
11     DigiKeyboard.sendKeyStroke(KEY_ENTER);
12     DigiKeyboard.delay(3000);
13     DigiKeyboard.sendKeyStroke(KEY_R, MOD_GUI_LEFT);
14     DigiKeyboard.delay(1000);
15     DigiKeyboard.print("http://fakeupdate.net/win10ue");
16     DigiKeyboard.sendKeyStroke(KEY_ENTER);
17     DigiKeyboard.delay(500);
18     DigiKeyboard.sendKeyStroke(KEY_F11);
19     for(;;){ /*empty*/ }
20 }
```

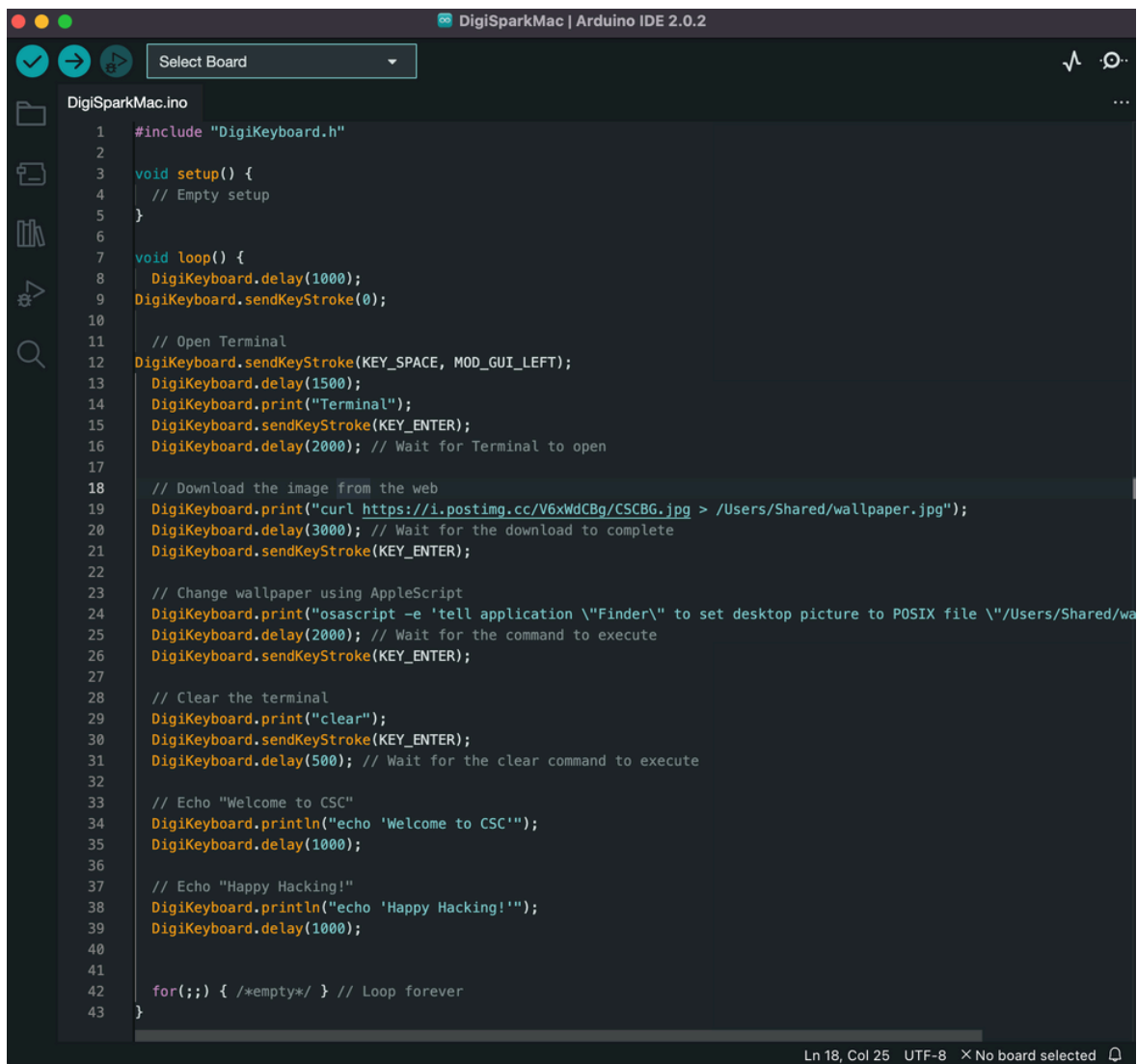
Ln 20, Col 2 UTF-8 × No board selected

EXPLANATION

This script is an example of how to use the DigiKeyboard library to automate keyboard actions on a computer. The script performs the following steps in an infinite loop:

1. It waits for 500 milliseconds.
2. It sends a key stroke with no key pressed, which essentially does nothing but ensures the keyboard is active.
3. It sends the key stroke combination for the Windows key (`GUI_LEFT`) and the 'R' key, which typically opens the Run dialog on Windows systems.
4. It waits for 600 milliseconds to allow the Run dialog to open.
5. It types in a YouTube channel URL and presses Enter, opening the URL in the default web browser.
6. It waits for 3000 milliseconds (3 seconds) to give the browser time to load the page.
7. It again sends the key stroke combination for the Windows key and 'R' to open the Run dialog.
8. It waits for 1000 milliseconds (1 second).
9. It types in a URL for a fake update website and presses Enter, opening this URL in the browser.
10. It waits for 500 milliseconds.
11. It sends the `F11` key stroke to put the browser in full-screen mode.
12. It enters an infinite loop, effectively stopping any further actions.

MAC SCRIPT



```
1 #include "DigiKeyboard.h"
2
3 void setup() {
4   // Empty setup
5 }
6
7 void loop() {
8   DigiKeyboard.delay(1000);
9   DigiKeyboard.sendKeyStroke(0);
10
11   // Open Terminal
12   DigiKeyboard.sendKeyStroke(KEY_SPACE, MOD_GUI_LEFT);
13   DigiKeyboard.delay(1500);
14   DigiKeyboard.print("Terminal");
15   DigiKeyboard.sendKeyStroke(KEY_ENTER);
16   DigiKeyboard.delay(2000); // Wait for Terminal to open
17
18   // Download the image from the web
19   DigiKeyboard.print("curl https://i.postimg.cc/V6xWdCBg/CSCBG.jpg > /Users/Shared/wallpaper.jpg");
20   DigiKeyboard.delay(3000); // Wait for the download to complete
21   DigiKeyboard.sendKeyStroke(KEY_ENTER);
22
23   // Change wallpaper using AppleScript
24   DigiKeyboard.print("osascript -e 'tell application \"Finder\" to set desktop picture to POSIX file \"/Users/Shared/wa");
25   DigiKeyboard.delay(2000); // Wait for the command to execute
26   DigiKeyboard.sendKeyStroke(KEY_ENTER);
27
28   // Clear the terminal
29   DigiKeyboard.print("clear");
30   DigiKeyboard.sendKeyStroke(KEY_ENTER);
31   DigiKeyboard.delay(500); // Wait for the clear command to execute
32
33   // Echo "Welcome to CSC"
34   DigiKeyboard.println("echo 'Welcome to CSC'");
35   DigiKeyboard.delay(1000);
36
37   // Echo "Happy Hacking!"
38   DigiKeyboard.println("echo 'Happy Hacking!'");
39   DigiKeyboard.delay(1000);
40
41
42   for(;;) { /*empty*/ } // Loop forever
43 }
```


EXPLANATION

The Mac script initiates by downloading a file, proceeding to update the desktop wallpaper with the newly downloaded image, and concluding by displaying specific text in the terminal. The following outlines the sequence of actions:

1. Introduce a 1000-millisecond delay.
2. Trigger a keystroke (0).
3. Access Terminal by simulating a keystroke for the left GUI key + space, followed by a 1500-millisecond pause, typing "Terminal," and executing by hitting Enter.
4. Retrieve an image from the web using the curl command.
5. Await the completion of the download process.
6. Modify the desktop wallpaper utilizing an AppleScript command to set the desktop picture.
7. Allow time for the command to finalize.
8. Clear the terminal content by inputting "clear" and pressing Enter.
9. Display "Welcome to CSC" and pause for 1000 milliseconds.
10. Present "Happy Hacking!" and pause for 1000 milliseconds.
11. Engage an infinite loop.

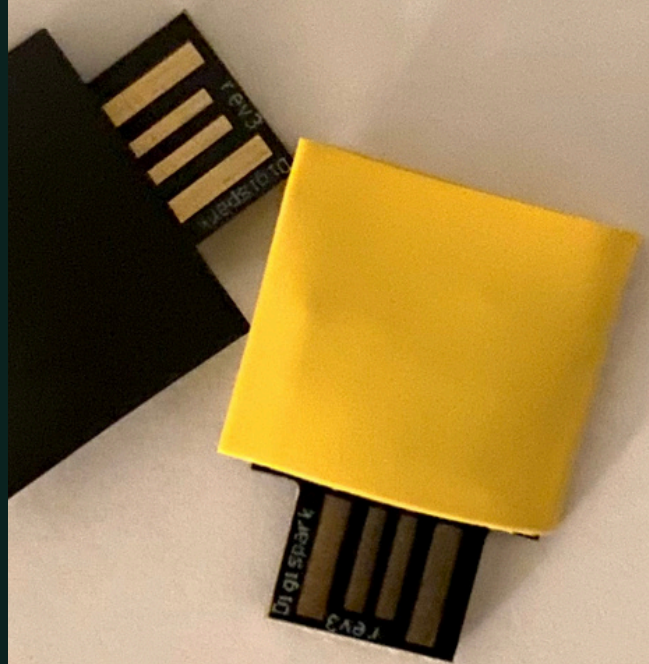
SEE HOW IT WORKS

[Windows Script Video](#)

[Mac Script Video](#)

[Windows Payload File](#)

[Mac Payload File](#)



REFERENCES

1. <https://github.com/byui-soc/bad-usb>
2. <https://github.com/CedArctic/DigiSpark-Scripts>
3. [Use USB Rubber Ducky Scripts & Payloads on an Inexpensive Digispark Board \[Tutorial\]](#)
4. [\\$1 BadUSB - DigiSpark Drive By HID Tutorial](#)
5. [Good Tutorial Video W/Troubleshooting](#)
6. [Run BadUSB Script on a \\$3 Digispark \(& how to change the keyboard layout\)](#)