

# USB Rubber Ducky Workshop

Penn State IEEE Student Chapter 2021  
Led by Will McGloughlin, Cole Baughman,  
Luke Miller, and Taylor Casavant

Go ahead and plug it in!



# USB Rubber Duckies

- Designed to mimic a keyboard/mouse to automate tasks on PCs
  - More commonly known as “BadUSBs”
  - Computers inherently and (often) unconditionally trust keyboard and mouse input via USB
  - If you can do something with just a keyboard, you can automate it with a USBRD
  - Devices like these are why you NEVER plug in a USB device without knowing what it does
    - But you guys wouldn't do that, right?
- Rubber Duckies in the wild
  - Most common commercially available USBRD from Hak5
  - Disguises itself as a flash drive
  - Uses custom scripting language
  - It's not cheap for what it is
- So what did we give you then?

USB RUBBER DUCKY DELUXE	HOTPLUG ATTACK COMBO KIT
\$49.99	\$219.99





# Digispark

- ATTiny85-powered Arduino development board
  - Small
  - Cheap
  - Has a male USB port on it
    - Makes USB HID device imitation trivially easy with the right libraries
  - Programmed similar to Arduino, with a few key differences
- So what?
  - We can take advantage of the USB HID device libraries to emulate the duties of a real USBRD
  - Harder to program, but...
    - Official Hak5 USB Rubber Ducky ---> \$50, and you have to learn a new language
    - Make your own BadUSB device ---> \$???, and you have to do that with little support
    - DIY USB Rubber Ducky w/ Digispark ---> \$5, and all you need is Arduino C



# Uses

Wide range of uses vary from funny to practical!

- Automatically and near-instantly mess with mouse settings
- Automatically navigate OS install dialogs
- Avoid going AFK with a mouse wiggle or keyboard input
- Execute repetitive shell commands
- Enter a difficult username/password
- Perform any basic, repetitive task



Program to your specific need. Very convenient because it is straightforward and self-contained.



# Scarier Applications

All of the following applications come from a single public source dealing specifically in scripts for Digispark. All of these include full scripts, most of which are for use on Windows machines. All of these, if used improperly and/or without permission, hold serious legal consequences.

- Fork bomb, both persistent and non-persistent
- Keylogger
- Reverse Shell
- Create a new user
- Arbitrary Powershell Execution
- DNS Poisoner

Penn State IEEE is not responsible for any damage done to any device using the Digispark Rubber Duckies we give you today.



# Ethics

When is it ok to use these scripts?

- Pentesting
  - Hacking a computer system with authorization from the owner
  - Simulates a real attack
  - Identifies vulnerabilities
  - Not uncommon to see BadUSBs in a pentester's arsenal
- On your own device
  - Make sure you know what your scripts do and how to reverse its effects
  - NEVER do it on devices that:
    - You do not own
    - You are not willing to risk
    - You are not sure you can fix if something goes wrong

All the scripts we show you will be completely harmless, and pose no risk to your devices.

# Programing the Digispark

- We will use Arduino IDE to program the Digispark
- It supports a dialect of C++, **Arduino C**
- In order to use the IDE we will need to add the Digispark in boards manager and install the driver for it (**More on that soon!**)
- Arduino C is a lower level programing language compared to the languages you may know like Python or Java
  - Python allows us to be more flexible, for example variable types can be determined at runtime by the data assigned to them
  - Arduino C code lacks this flexibility







# Setup and Loop

- When creating a program the Arduino IDE automatically creates two functions
- `setup()` is a place for you to initialize your variables. **Runs once.**
- `loop()` continuously runs on your Arduino. **Runs after setup, forever.**

```
void setup() {  
    // put your setup code here, to run once:  
  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
  
}
```




# Adding Libraries

- Libraries are a collection of non-volatile resources used by computer programs
- Existing Libraries for the Digispark will be included in our scripts
- They will be written in an include statement at the top of our program
- We will use functions from these libraries

```
#include "DigiKeyboard.h"
```

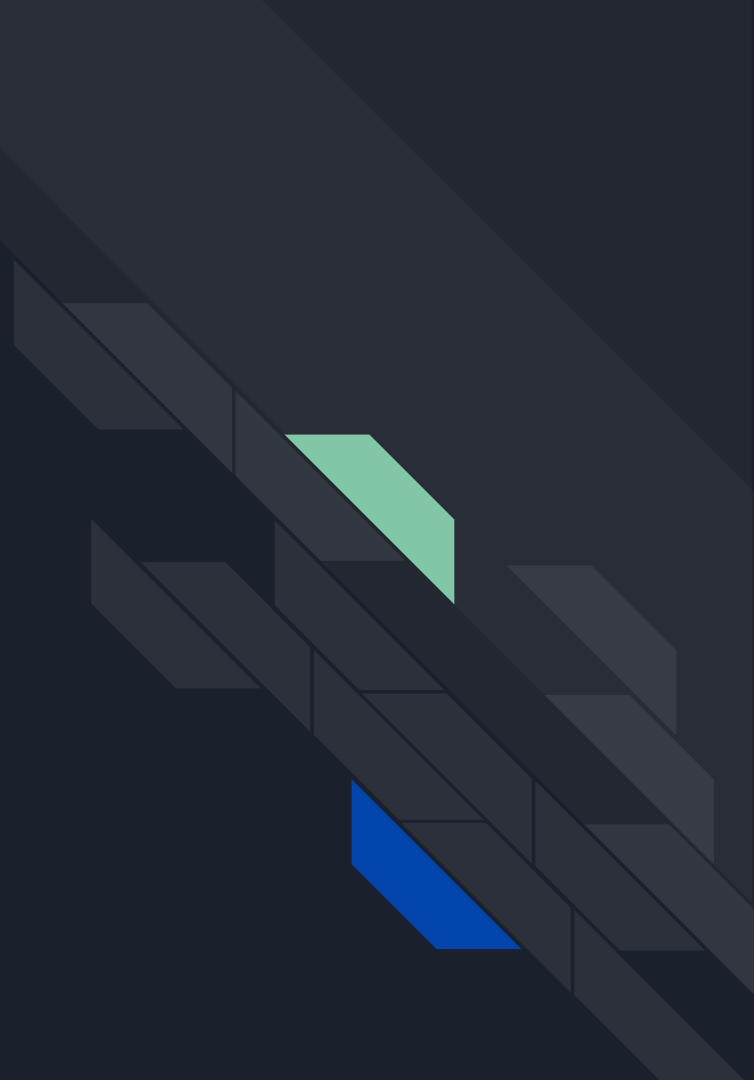
```
DigiKeyboard.sendKeyStroke(MOD_GUI_LEFT);  
DigiKeyboard.sendKeyStroke(KEY_ENTER);
```



# Installing Drivers and Adding Boards Manager

- Boards Manager URL: [http://digistump.com/package\\_digistump\\_index.json](http://digistump.com/package_digistump_index.json)
- Drivers URL: <https://github.com/digistump/DigistumpArduino/releases>

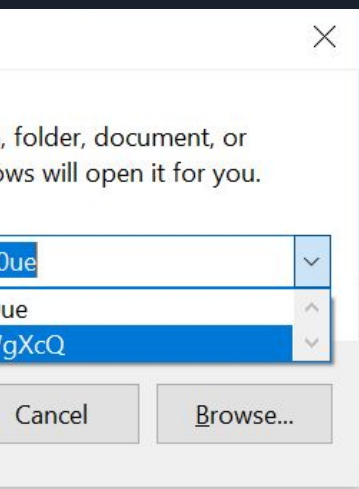
# Installation and Testing



# Example One: The Rickroll



Ric



```
2  #include "DigiKeyboard.h"
3  void setup() {
4      //empty
5  }
6  void loop() {
7      DigiKeyboard.delay(2000);
8      DigiKeyboard.sendKeyStroke(0);
9      DigiKeyboard.sendKeyStroke(KEY_R, MOD_GUI_LEFT);
10     DigiKeyboard.delay(600);
11     DigiKeyboard.print("https://youtu.be/dQw4w9WgXcQ?t=43s");
12     DigiKeyboard.sendKeyStroke(KEY_ENTER);
13     DigiKeyboard.delay(5000);
14     DigiKeyboard.sendKeyStroke(KEY_R, MOD_GUI_LEFT);
15     DigiKeyboard.delay(3000);
16     DigiKeyboard.print("http://fakeupdate.net/win10ue");
17     DigiKeyboard.sendKeyStroke(KEY_ENTER);
18     DigiKeyboard.delay(2000);
19     DigiKeyboard.sendKeyStroke(KEY_F11);
20     for(;;){ /*empty*/ }
21 }
```

## Example Two: The Mouse Wiggler



# Make something cool!

The Committee will be here to help!

