

# Project D.I.R.E.

Special Thanks to the following individuals for their dedication and hard work.  
Without them this project would not be possible.

**znxDomain -> firmware unlocker, tester, researcher**

**Lucky.peic -> firmware gui editor, tester, prototype,  
initial writeups, researcher**

**B47M4N -> firmware gui editor, tester, pcb and case  
creator, researcher, user guide writer**

**Bowman -> alt firmware, prototype tester**

**Mystic -> supplied test files, tester**

**Obiima -> original designer and firmware creator**

(Obiima was not part of the team who released this project, Only the Original author and creator of the private emulator  
never released.)

## Notice

This document is intended for the sole purpose of education and mass release.

Functional specifications may completely be distributed, in whole or in part, to everyone, without prior written approval from those who have released this.

The information contained in this document is provided AS IS.

While all reasonable efforts have been made to ensure the accuracy of the information contained in this document, WE assume NO responsibility and disclaim all liability for any errors or omissions that may be contained herein. As design changes are made in the software, firmware, hardware, interfaces, and communication protocols used, parts of this document may become obsolete until a subsequent revision to this document is released.

This project and the contents herein have been created FREELY and are not to be paid for. Individual pieces of this project can be purchased, but none of which create any financial gain, incentive, commercial advantage, or private financial gain for anyone involved.

WE ARE NOT LAWYERS. Use your own judgement with the information in this document and do not sell this product or any part of this project.

We release this to you freely and openly with all source files included.

In short: Enjoy the game! Don't be a prick and charge someone to use or build this free software.

---

## Contents

<b>TITLE PAGE .....</b>	<b>1</b>
<b>SPECIAL THANKS .....</b>	<b>1</b>
<b>1. PURPOSE AND SCOPE .....</b>	<b>4</b>
<b>2. ABBREVIATIONS .....</b>	<b>4</b>
<b>3. SUMMARY .....</b>	<b>5</b>
3.1. BOM .....	6
<b>4. BUILDING YOUR EMULATOR AND HARDWARE YOU NEED .....</b>	<b>6</b>
4.1. STM32 BLUEPILL .....	7
4.2. ST-LINK V2 .....	7
4.3. LCD WITH SD CARD READER .....	8
4.4. CUSTOM PCB .....	8
4.5. SOLDERING IRON .....	9
4.6. PUSH BUTTONS .....	9
4.7. FULLY ASSEMBLED PCB .....	10
<b>5. BUTTON USAGE .....</b>	<b>11</b>
5.1. EMULATOR SCREEN SLOTS .....	12
<b>6. SOFTWARE AND PROCESS .....</b>	<b>12</b>
6.1. ST-LINK .....	12
6.2. BANK EDITOR .....	14
<b>7. CASE .....</b>	<b>18</b>
<b>8. EXTRA .....</b>	<b>20</b>
<b>APPENDICES .....</b>	<b>21</b>
<b>APPENDIX A – SOFTWARE TOOLS .....</b>	<b>21</b>

## 1. Purpose and Scope

The purpose of this document is to functionally explain and use project D.I.R.E.

The document covers creating and usage of hardware, software, firmware, and 3D printed files.

## 2. Abbreviations

D.I.R.E.	Disney Infinity Reverse Engineered
STM32	STM32 Bluepill
BOM	Build of Materials

### 3. Summary

Project D.I.R.E. (Disney Infinity Reverse Engineered) is a portal emulator for Disney Infinity, Lego Dimensions, and Skylanders. This replaces the original portal in place of a custom created one to inject your own character dumps without the need for the original characters, vehicles, or playsets.

This emulator originally started several years ago after a talented coder named obiima did not release his project to the public. Obiima created and demoed a console portal emulator for 3 games: Disney Infinity, Lego Dimensions, and Skylanders. Obiima never released his project into the wild and the project he did release privately was closed firmware meaning locked to the stm32 Bluepill. This project picks up where he left off.

The Firmware allows this device to select between games by holding one of the buttons during boot. You have the option of using limited onboard memory or by using an SD card you can increase the character count in your base emulator. By adding your own character files known as bin files, you can inject them into the game without having to keep your original characters.

This emulator currently works on the Wii, Wii U, PS3, and PS4.

I CAN NOT confirm if this works on the PS5 or Nintendo Switch as I do not own the consoles to test with.

I have been told Skylanders plays on the Nintendo Switch but does NOT use the Portal of Power. I have also seen footage of Disney Infinity being played on a PS5.

XBOX 360 and XBOX One are Not Yet included in this release. There is a security chip required from an Activision portal to make this work and we have not yet discovered a simple method to allow these games to work on these specific consoles.

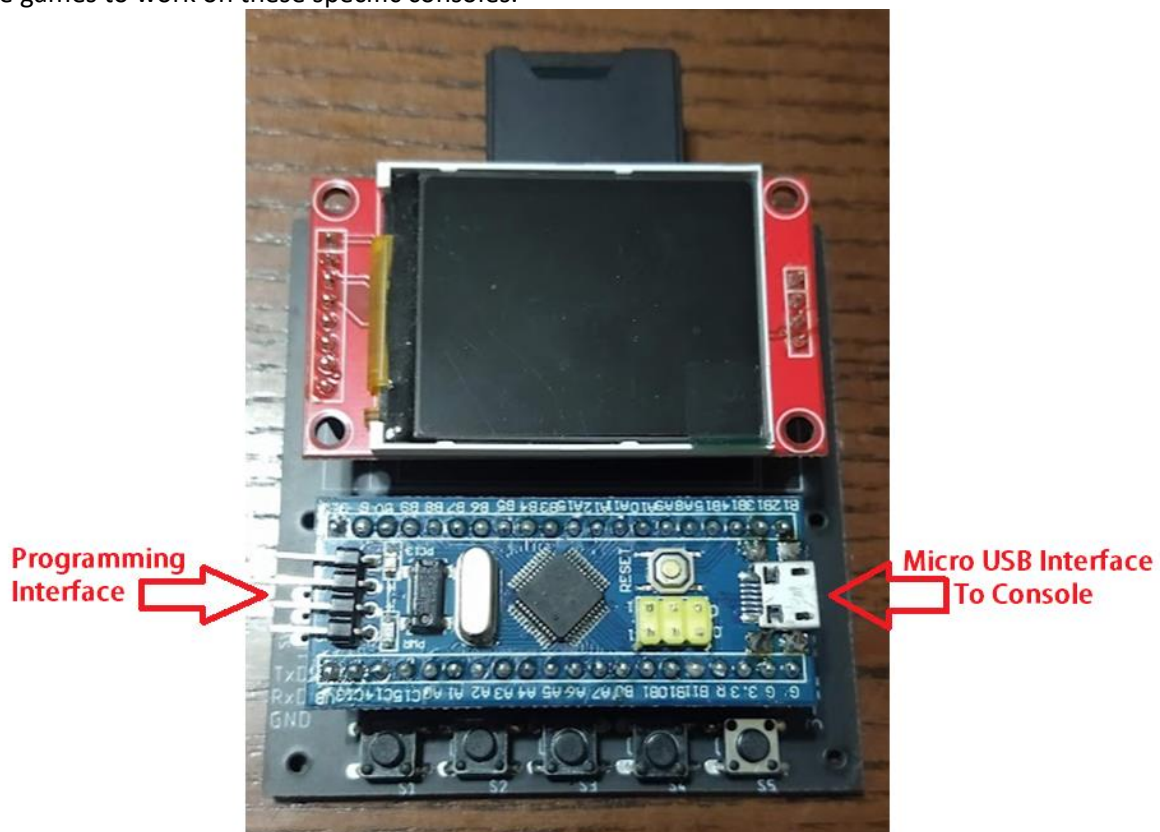


Figure 1: The Soldered Emulator

### 3.1. BOM

Item	Quantity
STM32 Bluepill *Some of the STM32 Bluepills come pre soldered with headers	1 *To fit in the provided 3D case files, please remove Boot 0/1 Headers
ST-Link V2 *Usually comes with 4 Female to Female jumpers for programming	1
Male Headers *Solder 40 to the STM32 Bluepill **Solder 12 to the LCD with SD Card Reader ***Solder the other end of the 52 to the custom PCB	52
LCD with Card Reader * Some of the LCDs come pre soldered with headers	1
Memory Card – 128Mb to 2Gb *Unless Following Section 8 to increase size limitation.	1
Custom PCB *All Files included in zip to get these printed – 2 sided board, all through hole, black color is nice and the silkscreen comes out perfectly readable	1
Tactile Push Buttons	5
Micro USB Cable	1
3D printed Case Bottom	*Optional, but nice to have
3D printed Case Top	*Optional Open finger or Closed top type (peg buttons included)
Screws to hold the case are #8-32 x 1/2in or 12mm length 4mm diameter	*4 – only required if using the 3D printed case

## 4. Building Your Emulator and Hardware You Need

## 4.1. STM32 Bluepill

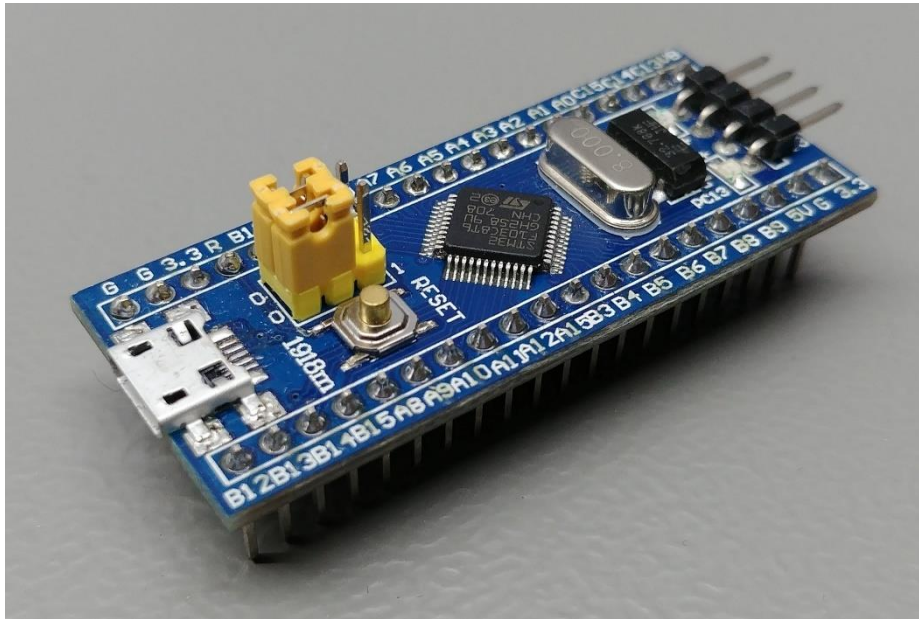


Figure 2: STM32 Bluepill – Brains of the emulator

NOTE: some come with headers soldered, some do not MAKE SURE headers are included

## 4.2. ST-Link V2



Figure 3: ST-LINK V2 - programming the Bluepill



### 4.3. LCD with SD Card Reader

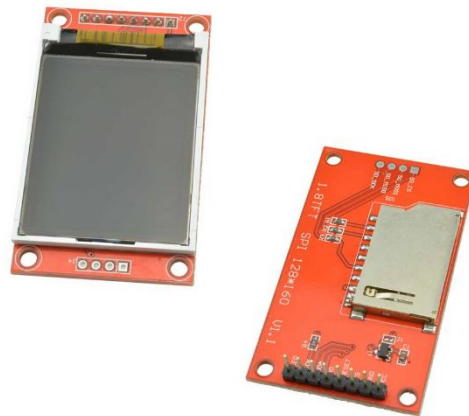


Figure 4: LCD with SD Card Reader

### 4.4. Custom PCB

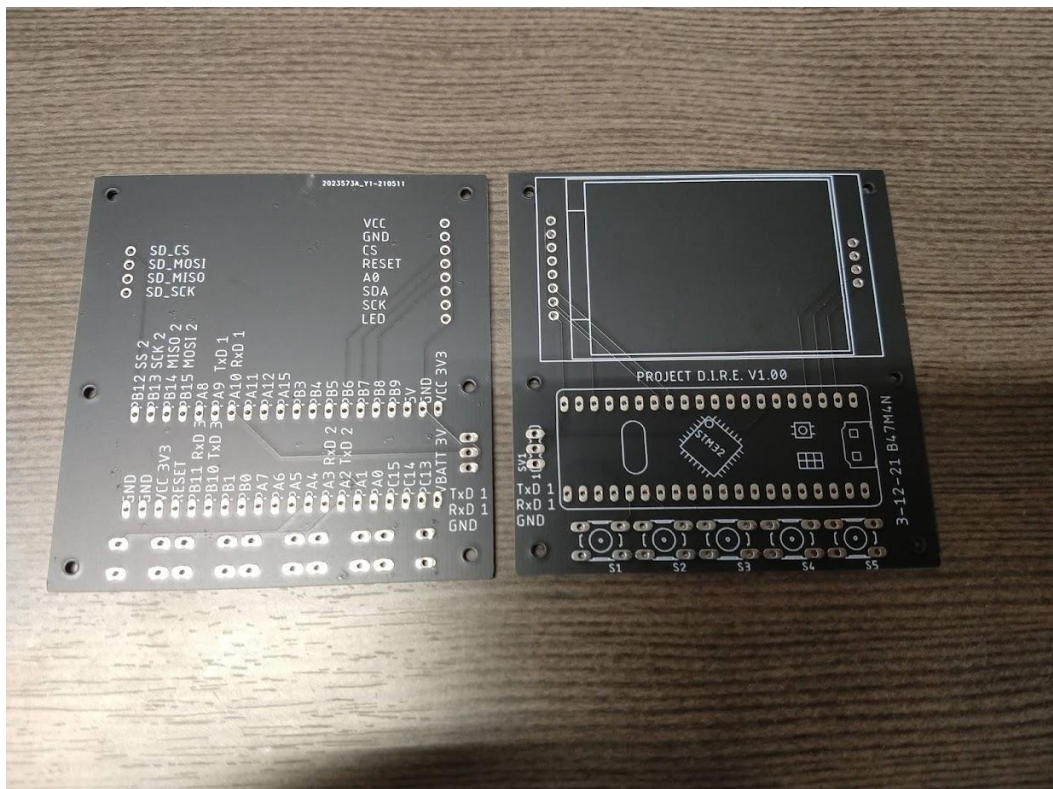


Figure 5: Custom Project D.I.R.E. PCB without need of breadboard

NOTE: Everything here is through hole soldering – If your LCD and STM32 don't come with headers presoldered you are looking at 124 Soldering joints, if they are soldered, only 72 soldering joints.



## 4.5. Soldering Iron



*Figure 6: Any Soldering iron will most likely work*

## 4.6. Push Buttons



*Figure 7: 5 push buttons one for each place on the PCB (6mm x 6mm x 5mm)*

## 4.7. Fully Assembled PCB



*Figure 8: Fully Assembled PCB with Boot 0/1 Headers Removed  
Note: To fit in the provided 3D case, please remove Boot 0/1 Headers*

## 5. Button Usage

The emulator makes use of the 5 buttons:

	Comments / Specifications
1. Left	Move Left within menu
2. Right	Move Right within menu
3. Select	Select place on portal [ - ] no choice or < - > choice
4. Clear	Clear selection
5. Settings	Changing the orientation and or type of LCD
Hold Left upon power on	Start in the Disney Infinity Portal Mode (Default mode)
Hold Right upon power on	Start in the LEGO Dimensions Portal mode
Hold Select upon power on	Start in the Skylanders Portal mode
Hold Left + Settings Button	Set Default Start up Disney Infinity Portal
Hold Right + Settings Button	Set Default Start up LEGO Dimensions Portal
Hold Select + Settings Button	Set Default Start up Skylanders Portal

Figure 9: Default Mode does not require holding a button down on start up



Figure 10: Usage - Elsa Selected from emulator, Level and experience Recorded



Figure 11: Usage In Game – Elsa Selected from emulator, Level and experience Recorded

## 5.1. Emulator Screen Slots

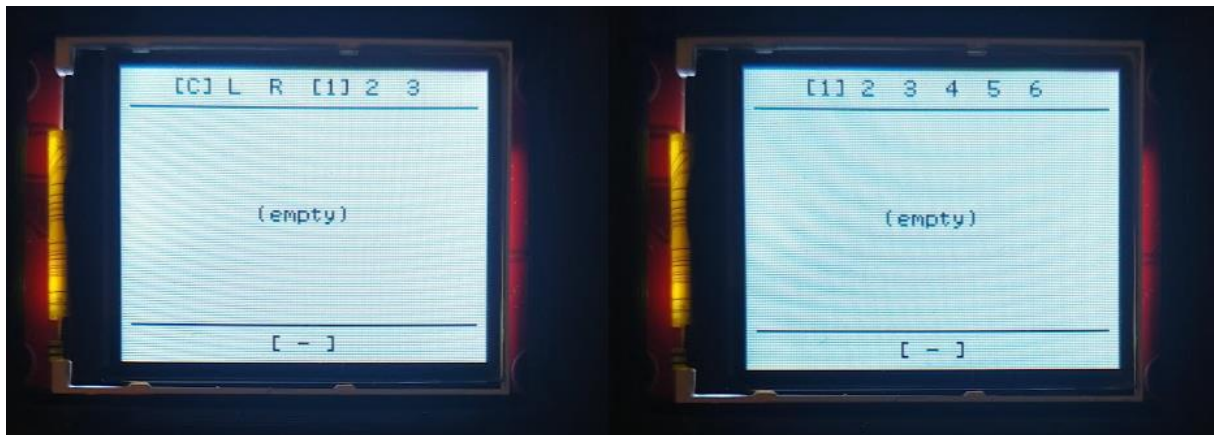


Figure 12: LEFT - Disney Infinity & Lego Dimensions, RIGHT – Skylanders

Function				
SLOT	Disney Infinity	Lego Dimensions		SLOT Skylanders
<b>C – Center C [1]</b>	Playset	Vehicle Placement for building or character		<b>[1]</b> Level or Character
<b>C – Center C [2]</b>	Hex disc Customization	Vehicle Placement for building or character		<b>[2]</b> Level or Character
<b>C – Center C [3]</b>	Hex disc Customization	Vehicle Placement for building or character		<b>[3]</b> Level or Character
<b>L – Left L [1]</b>	Character	Character or vehicle		<b>[4]</b> Level or Character
<b>L – Left L [2]</b>	Ability or Costume	Character or vehicle		<b>[5]</b> Level or Character
<b>L – Left L [3]</b>	Ability or Costume	Character or vehicle		<b>[6]</b> Level or Character
<b>R – Right R [1]</b>	Character	Character or vehicle		
<b>R – Right R [2]</b>	Ability or Costume	Character or vehicle		
<b>R – Right R [3]</b>	Ability or Costume	Character or vehicle		

Figure 13: Emulator Slot Usage

## 6. Software and Process

### 6.1. ST-LINK

Step 6.1 - Download and install the "STM32 ST-LINK Utility" software.

Step 6.2 - Download the "stlink\_flash\_firmware.bat" file.

Step 6.3 - Unplug the USB cable from the STM32 board.

Step 6.4 - Connect the ST-LINK adapter to the STM32 board (see Figure 14).

Step 6.5 - Connect the ST-LINK USB adapter to your PC.

Step 6.6 - Download and extract a compatible firmware (included in the zip).

Step 6.7 - Drag the firmware .bin file and drop it on the "stlink\_flash\_firmware.bat" (see Figure 15)



The firmware is loaded initially onto the Bluepill via the ST-Link USB dongle and the use of a batch file. To update or change your firmware on the STM32 simply connect your ST-Link USB to your computer and use the included 4 female wires to connect to your 4 pins on the STM32 following the diagram below.

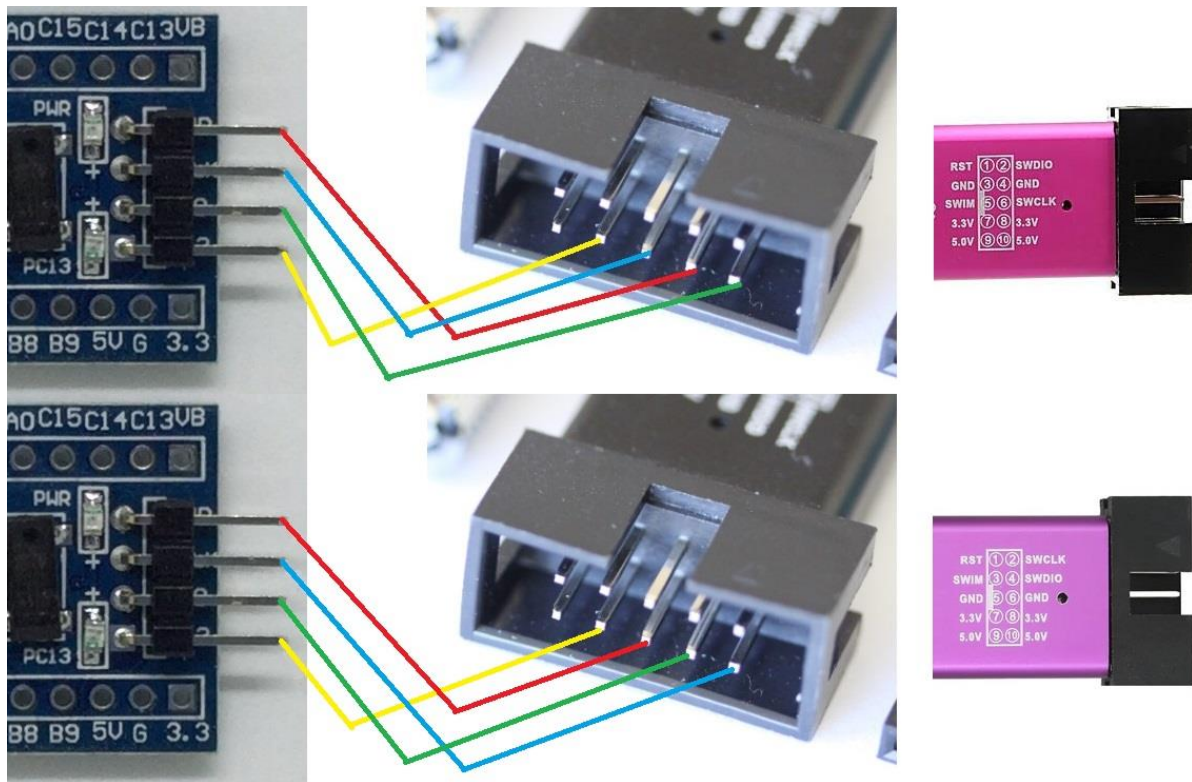
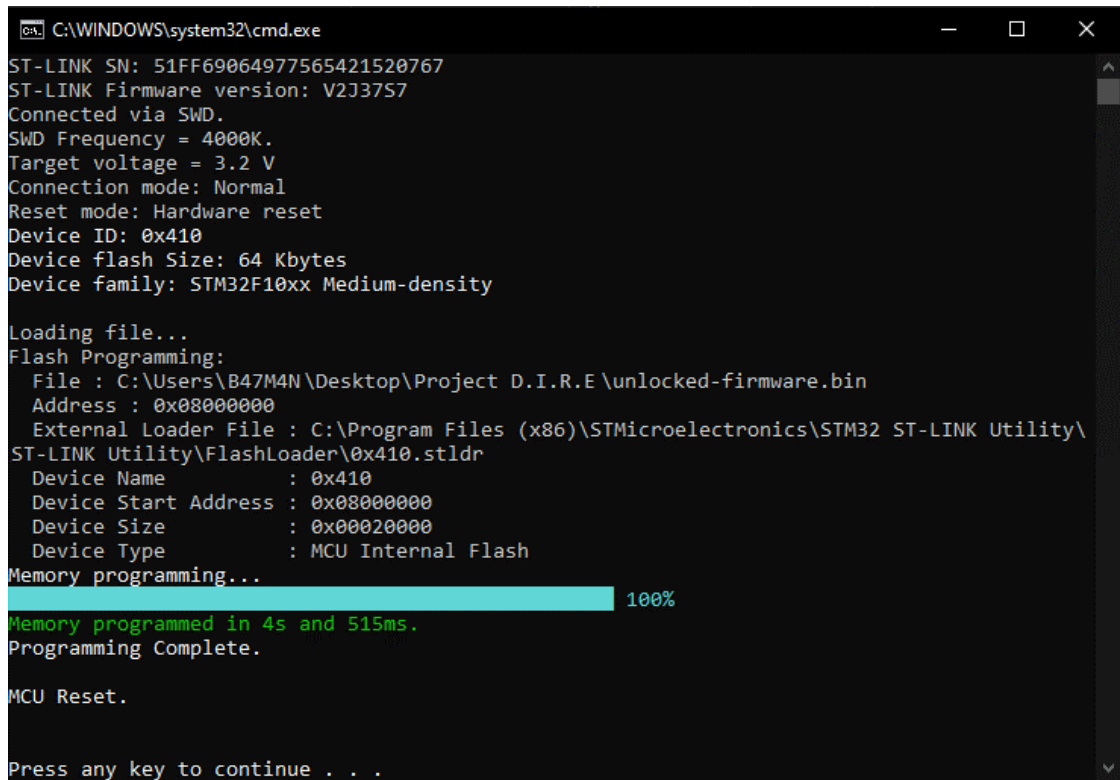


Figure 14: There are multiple Versions of the ST-Link V2



```

C:\WINDOWS\system32\cmd.exe
ST-LINK SN: 51FF69064977565421520767
ST-LINK Firmware version: V2J37S7
Connected via SWD.
SWD Frequency = 4000K.
Target voltage = 3.2 V
Connection mode: Normal
Reset mode: Hardware reset
Device ID: 0x410
Device flash Size: 64 Kbytes
Device family: STM32F10xx Medium-density

Loading file...
Flash Programming:
  File : C:\Users\B47M4N\Desktop\Project D.I.R.E\unlocked-firmware.bin
  Address : 0x08000000
  External Loader File : C:\Program Files (x86)\STMicroelectronics\STM32 ST-LINK Utility\
  ST-LINK Utility\FlashLoader\0x410.stldr
  Device Name : 0x410
  Device Start Address : 0x08000000
  Device Size : 0x00020000
  Device Type : MCU Internal Flash
Memory programming... 100%
Memory programmed in 4s and 515ms.
Programming Complete.

MCU Reset.

Press any key to continue . . .

```

Figure 15: ST Link

## 6.2. Bank Editor

The Bank Editor is the software setup allowing the usage of your own backup files bin files of characters or vehicles or play areas. To use any functionality of the Bank Editor you must plug in the micro USB cable into your computer and into your emulator. To use the SD card, plug in the SD Card before powering on your emulator via the USB cable

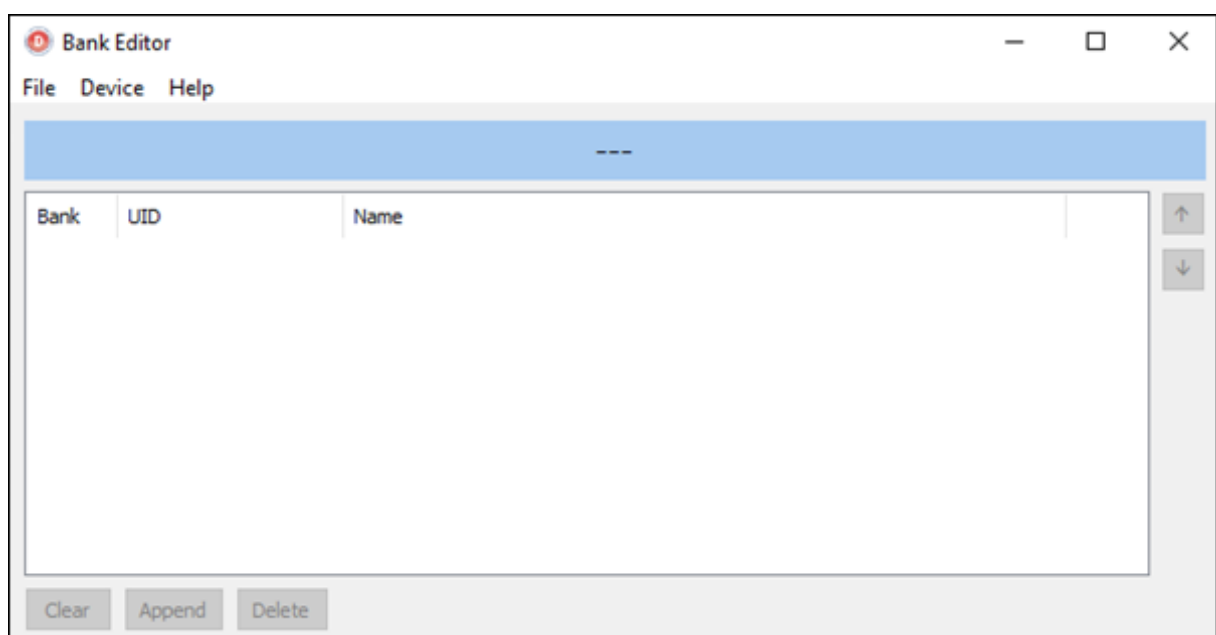


Figure 16: Bank Editor

To use the Bank Editor ensure your STM32 is plugged in via USB micro and click Device  
There are a few options: File, Device, and Help.

File opens your backed up bin files.

Device Loads the banks on the internal or external memory.

Device -> Format allows you to format internal or external memory.

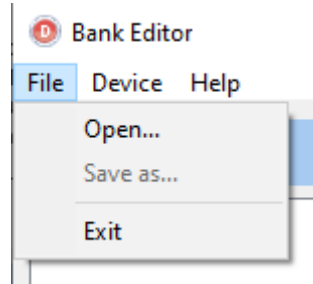


Figure 17: File Menu

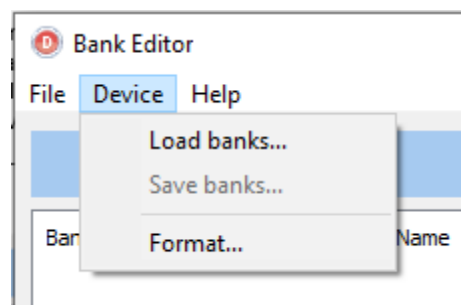


Figure 18: Device Menu

File opens your backed up bin files.

Device Loads the banks on the internal or external memory.

Device -> Format allows you to format internal or external memory.

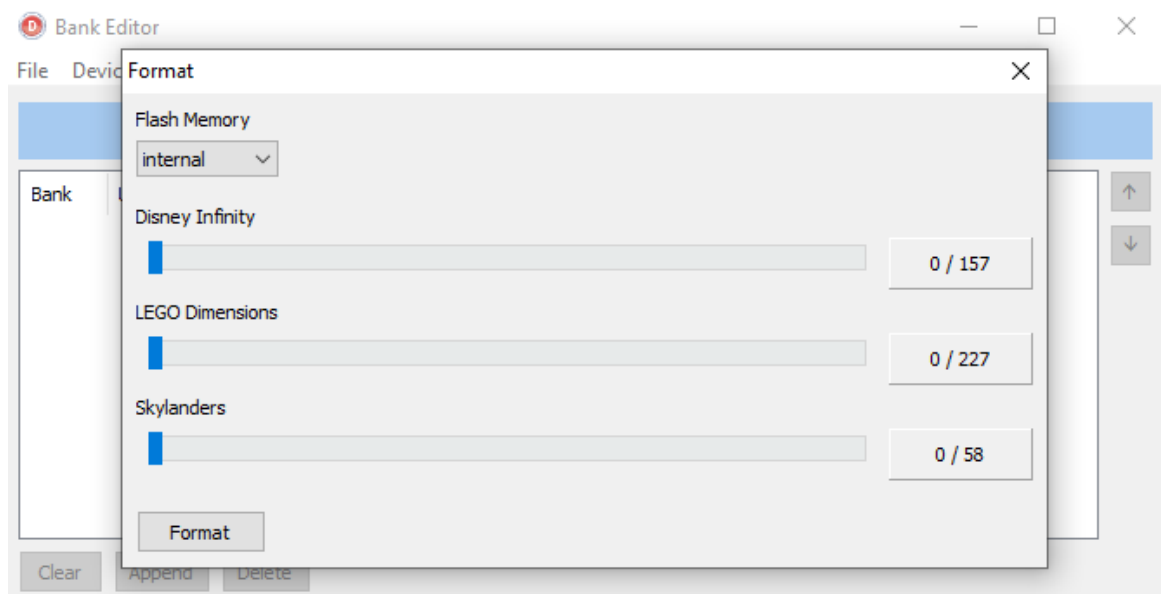


Figure 19: Internal Memory Format

Note: Internal memory slots are shared among all the banks



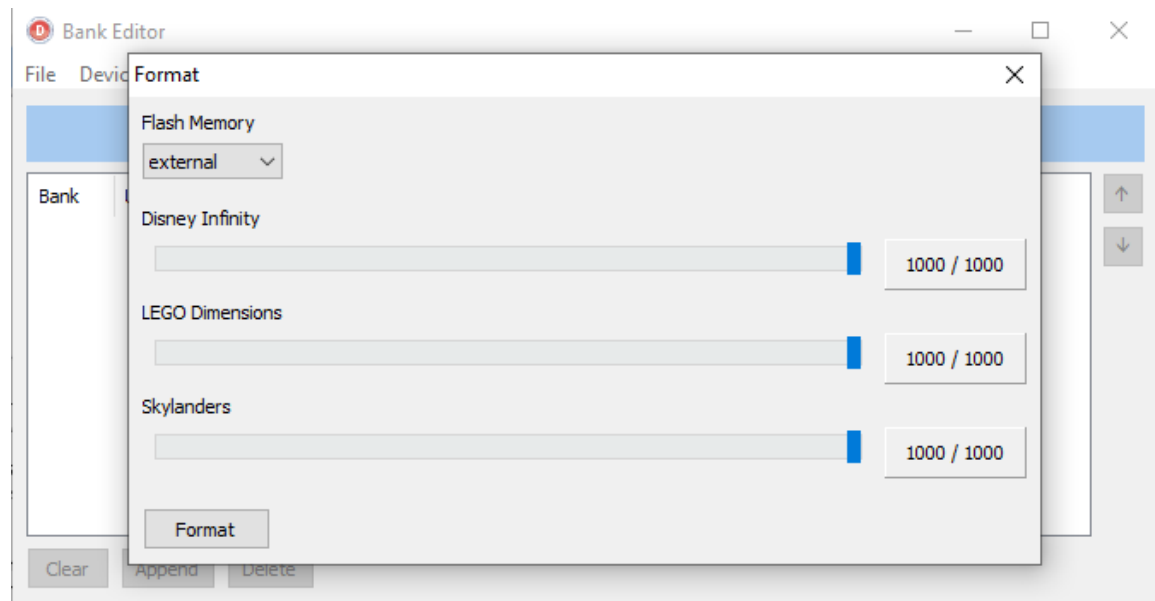


Figure 20: External Memory Format (SD Card)

Note: External memory slots are NOT shared among all the banks

After clicking on Format, you then click the Device button and load banks.  
Select your banks Disney Infinity, Lego Dimensions, or Skylanders then click OK.

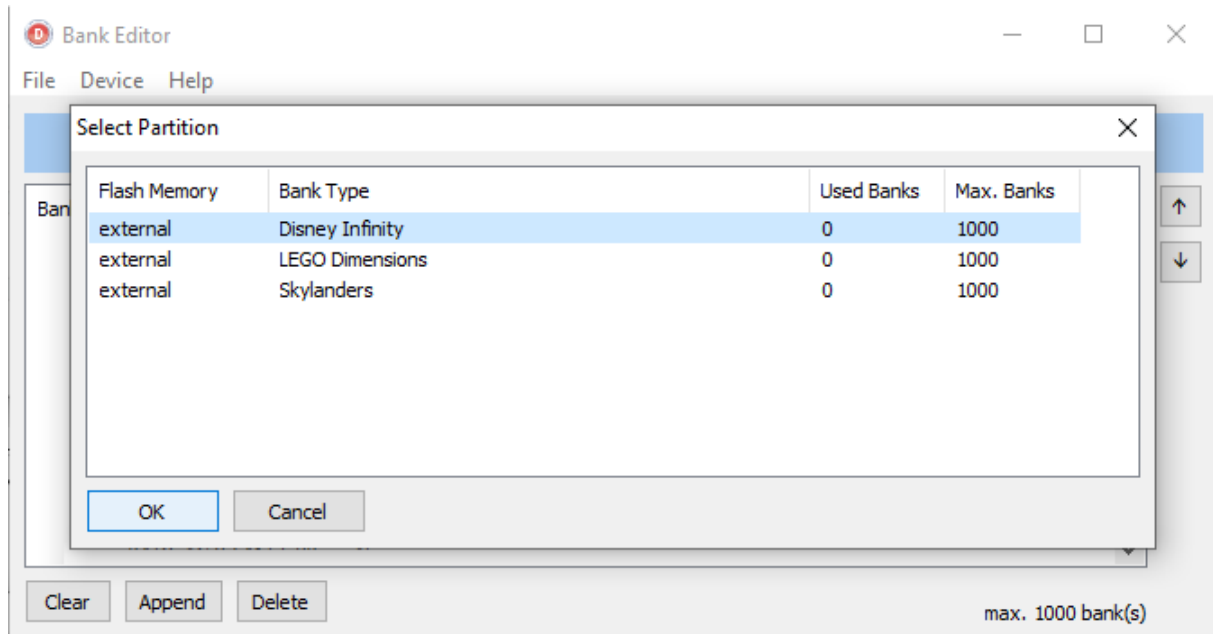


Figure 21: Select Your Bank, and click OK

Click Append  
Select your own bin file dumped files.  
Click Device and Save banks...

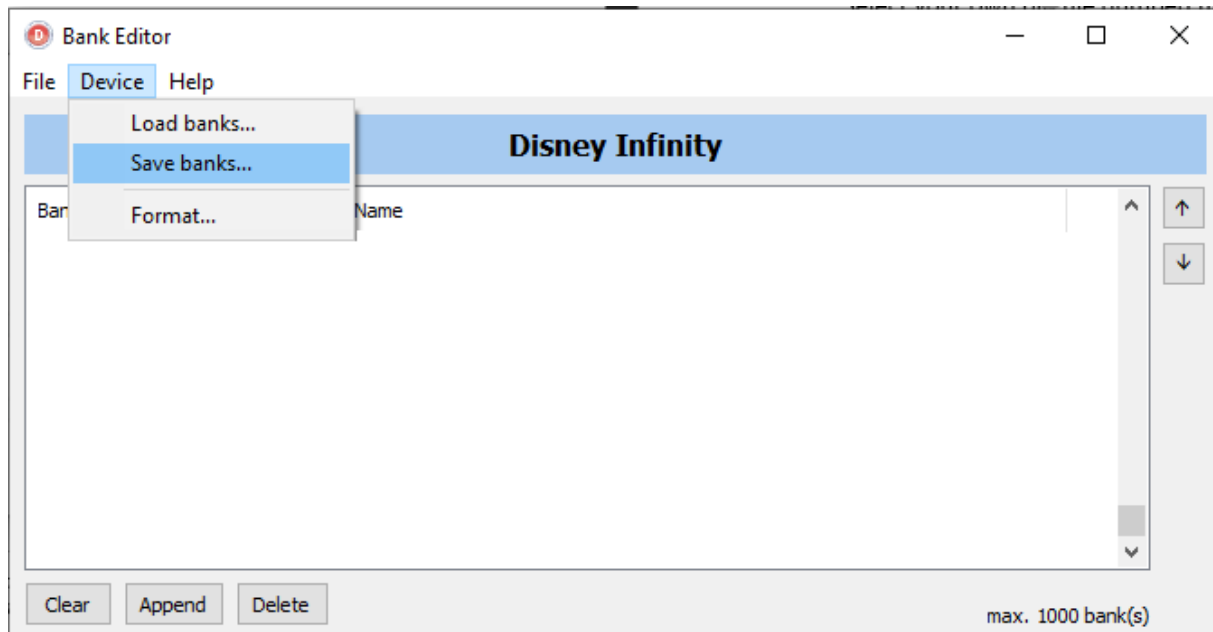
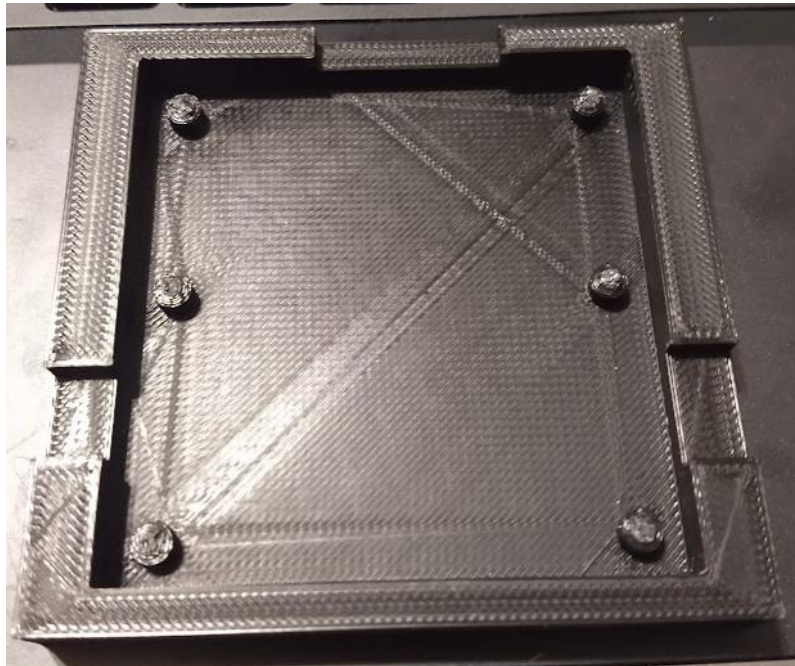


Figure 22: Save Banks

You're all set to play your game with your backed up characters.

## 7. Case

Many iterations were created, but only the two were good enough to present to you.



*Figure 23: The Base - Your board fits recessed inside this*



*Figure 24: The Open Frame Case*

*NOTE: 4 Screws to hold the case are #8-32 x 1/2in or 12mm length 4mm diameter*



*Figure 25: The Enclosed Top Button Case*

*NOTE: 4 Screws to hold the case are #8-32 x 1/2in or 12mm length 4mm diameter*

## 8. EXTRA

There is a software limitation on the Bank Editor, this will ONLY ALLOW 128Mb to 2Gb cards to be used. Most of us don't have those sizes of memory cards lying around.

To fix this problem, you can take a blank 128mb.raw image provided in the downloaded bundle and can use this in conjunction with etcher to write to any size sd card.

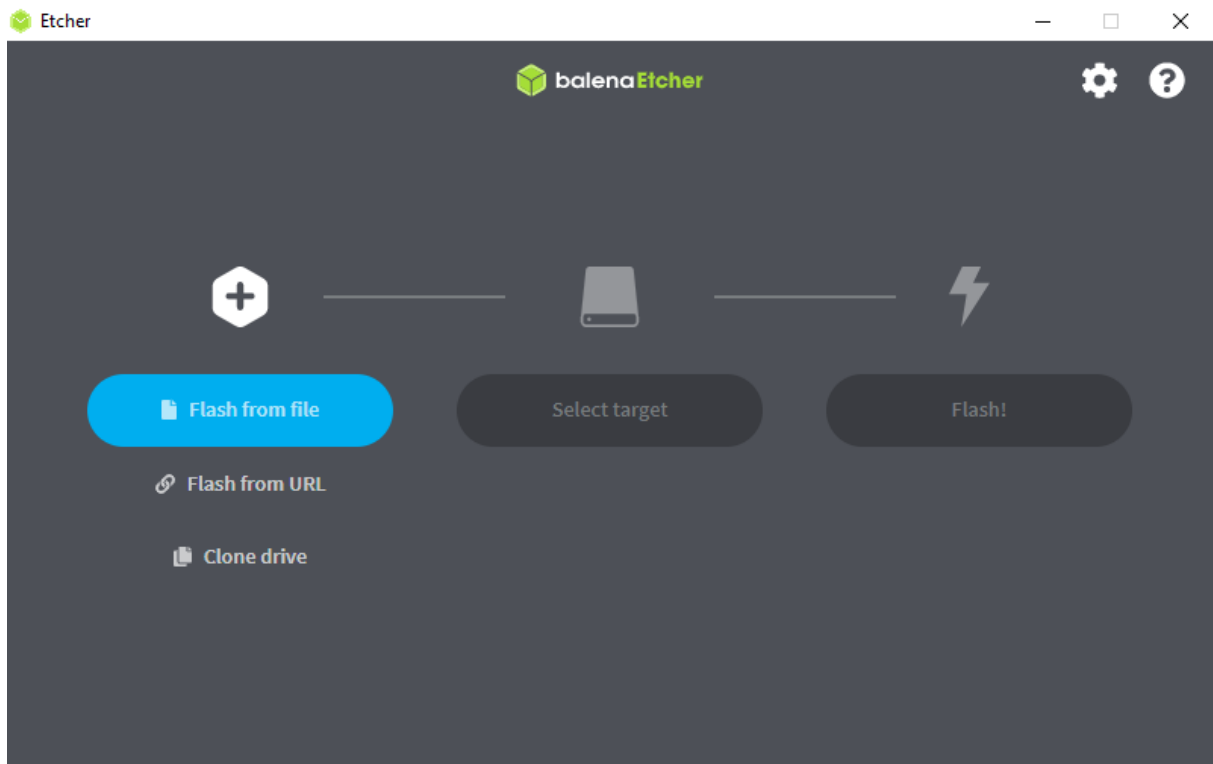


Figure 26: Etcher

Select Flash from File, Select your 128mb.raw file.

Select the SD card target

Click Flash!

Now insert your SD card into your emulator and continue onto section 6.2.

# Appendices

## Appendix A – Software Tools

The software tools used for Testing can be found in the following table:

General Software	
Tool	Comments
ST Link Utility	The Utility is used to upload firmware to the STM32
Bank Editor V6	As far as we know, Created by Obiima – Used to upload your bin file backups onto the SD card or internal memory.
Etcher v1.5.121	Used to write raw 128mb.raw to sd card
Firmware	
Tool	Comments
unlocked-firmware.bin	Originally Released by Obiima, unlock patched by znxDomain, gui edited by B47M4N and Lucky.peic
Ghidra	Used to Reverse Engineer the lockout
Case	
Tool	Comments
Solidworks	Created and edited by B47M4N
Ultimaker	Used to print the case parts
PCB	
Tool	Comments
Eagle PCB	Used to layout the PCB
JLPCB	Purchased usable boards
References	
Tool	Comments
<a href="https://nfc.toys/">https://nfc.toys/</a>	References to list how backups of your own toys are created using your own tools.