Shall We Play a Game?

Hacking and weaponizing the NES Classic Mini

SHALL WE PLAY A GAME?

Who we are:

- Ross (@HypnInfosec) and Dale (@dale_nunns)
- like retro computing and consoles
- also like hacking/breaking/making/fixing stuff

SHALL WE PLAY A GAME?

Why this talk:

- combination of the stuff we like
- good way to explore embedded device hacking
- relatively affordable & easily accessible
- fun to play with ¯_(ツ)_/¯

SHALL WE PLAY A GAME?

About the NES mini

SHALL WE PLAY A GAME?

About the NES Classic mini:

- modern remake of the NES-001 gaming console

SHALL WE PLAY A GAME?

The NES Classic mini is <u>NOT</u>:

- able to play original cartridges
- hard to find (available at many online retailers)
- "just a Raspberry Pi" (eg: RetroPie)
- meant to be modified/customized

SHALL WE PLAY A GAME?

About the NES Classic mini:

- USB powered
- HDMI output
- various models

SHALL WE PLAY A GAME?

About the NES Classic mini:

- comes with 30 games (roms)
- has a UI for selecting games to play
- uses custom emulator "kachikachi" to play NES ROMs
- coded with SDL and other open source tools
- see https://www.nintendo.co.jp/support/oss/

SHALL WE PLAY A GAME?

SHALL WE PLAY A GAME?

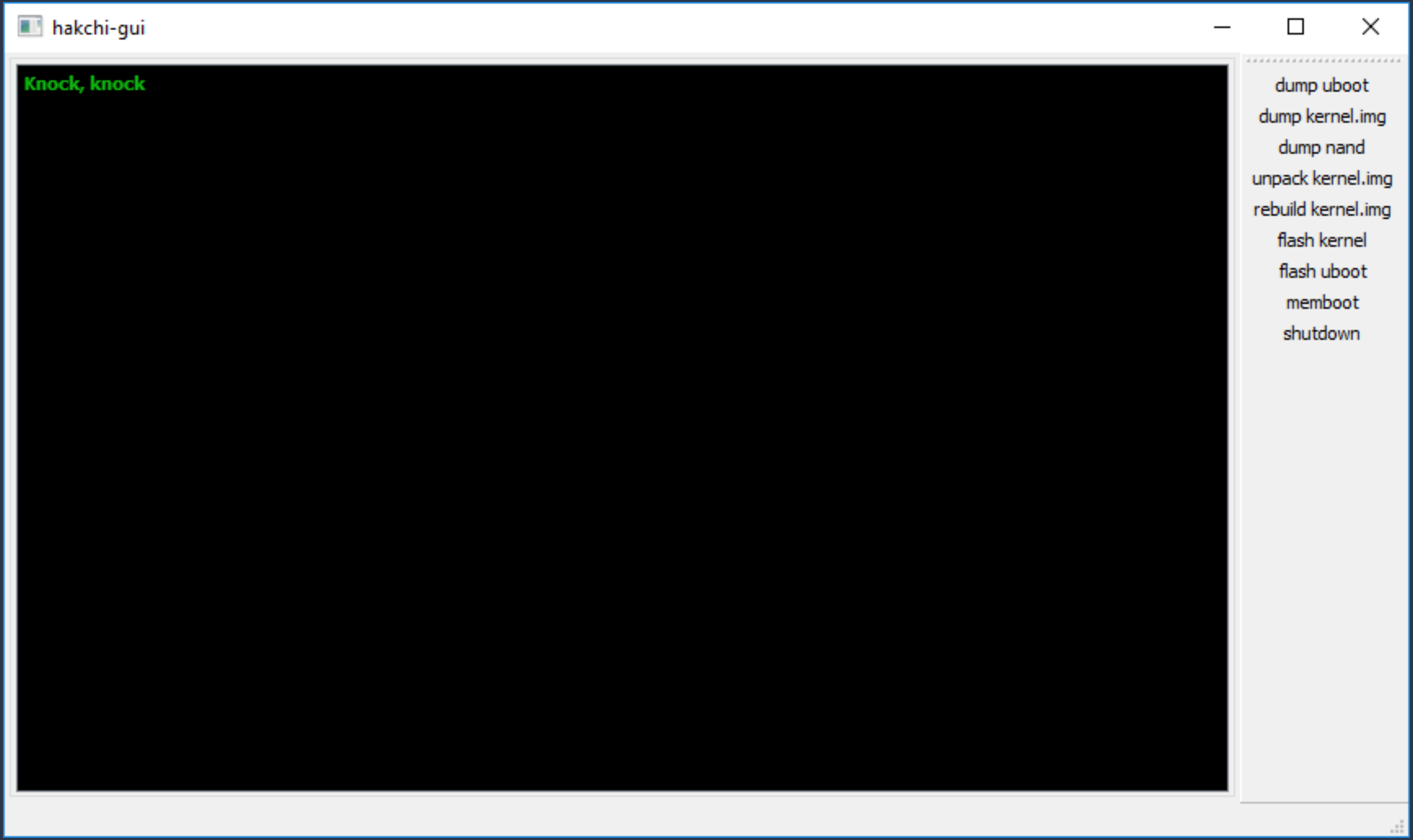
Software Hack

SHALL WE PLAY A GAME?

The "hakchi" mod/hack:

- initially written by "madmonkey"
- now on it's 3rd version: "Community Edition"

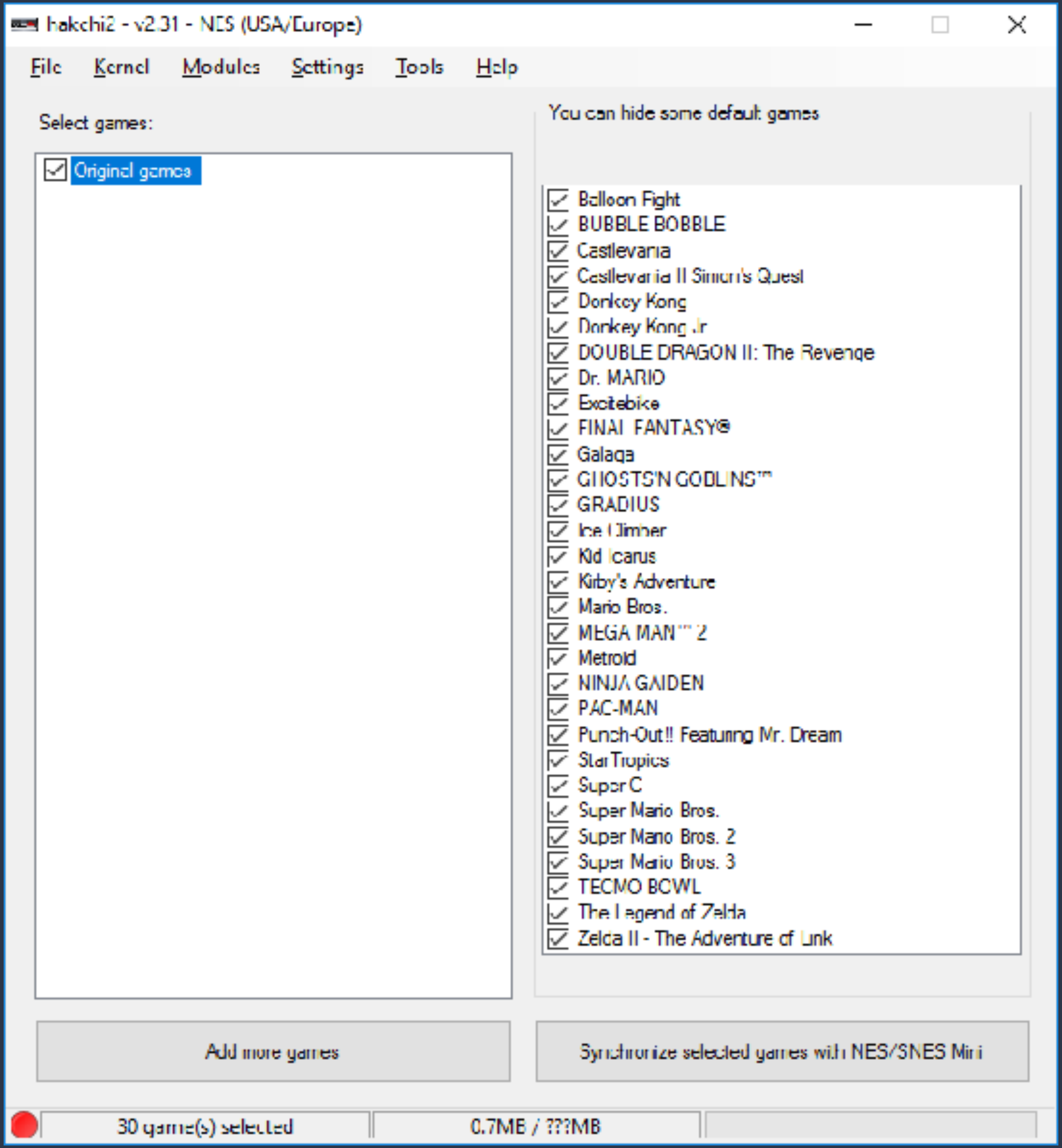
SHALL WE PLAY A GAME?



```
Knock, knock
```

hakchi by madmonkey

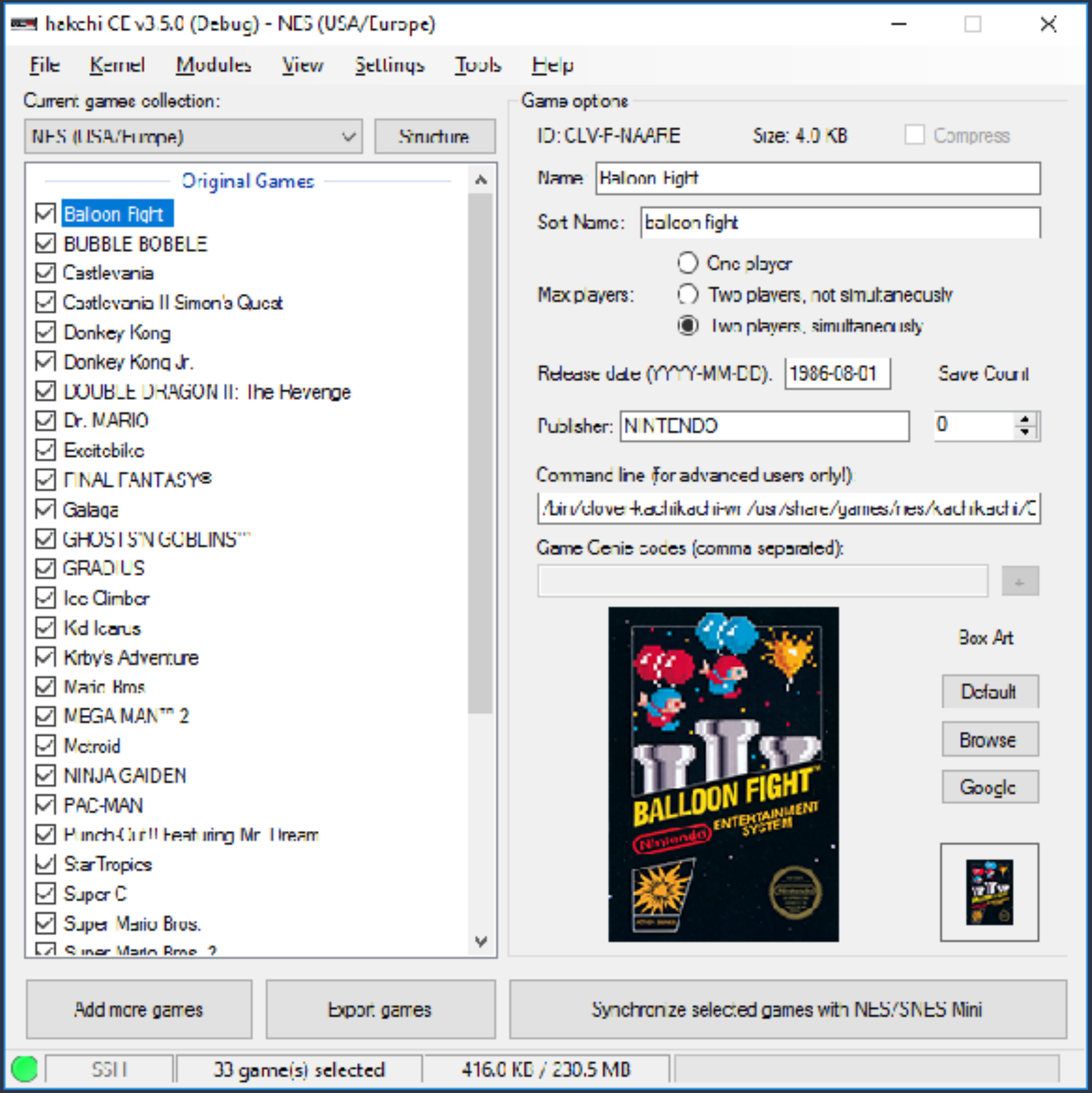
SHALL WE PLAY A GAME?


hakchi2 - v2.31 - NES (USA/Europe)
File
Kernel
Modules
Settings
Tools
Help
Select games:
Original games
You can hide some default games
Add more games
Synchronize selected games with NES/SNES Mini
30 game(s) selected
0.7MB / ???MB

- [✓] Balloon Fight
- [✓] BUBBLE BOBBLE
- [✓] Castlevania
- [✓] Castlevania II Simon's Quest
- [✓] Donkey Kong
- [✓] Donkey Kong Jr
- [✓] DOUBLE DRAGON II: The Revenge
- [✓] Dr. MARIO
- [✓] Excitebike
- [✓] FINAL FANTASY®
- [✓] Galaga
- [✓] GHOSTS'N GOBLINS™
- [✓] GRADIUS
- [✓] Ice Climber
- [✓] Kid Icarus
- [✓] Kirby's Adventure
- [✓] Mario Bros.
- [✓] MEGA MAN™ 2
- [✓] Metroid
- [✓] NINJA GAIDEN
- [✓] PAC-MAN
- [✓] Punch-Out!! Featuring Mr. Dream
- [✓] StarTropics
- [✓] Super C
- [✓] Super Mario Bros.
- [✓] Super Mario Bros. 2
- [✓] Super Mario Bros. 3
- [✓] TECMO BOWL
- [✓] The Legend of Zelda
- [✓] Zelda II - The Adventure of Link

hakchi2 by ClusterM

SHALL WE PLAY A GAME?



hakchi CE v3.5.0 (Debug) - NES (USA/Europe)

File Kernel Modules View Settings Tools Help

Current games collection: NES (USA/Europe) Structure

Original Games
- [✓] Baloon Fight
- [✓] BUBBLE BOBBLE
- [✓] Castlevania
- [✓] Castlevania II Simon's Quest
- [✓] Donkey Kong
- [✓] Donkey Kong Jr.
- [✓] DOUBLE DRAGON II: The Revenge
- [✓] Dr. MARIO
- [✓] Excitebike
- [✓] FINAL FANTASY®
- [✓] Galaga
- [✓] GHOSTS'N GOBLINS™
- [✓] GRADIUS
- [✓] Ice Climber
- [✓] Kid Icarus
- [✓] Kirby's Adventure
- [✓] Mario Bros.
- [✓] MEGA MAN™ 2
- [✓] Metroid
- [✓] NINJA GAIDEN
- [✓] PAC-MAN
- [✓] Punch-Out!! Featuring Mr. Dream
- [✓] StarTropics
- [✓] Super C
- [✓] Super Mario Bros.
- [✓] Super Mario Bros. 2

Game options

ID: CLV-P-NAAFE Size: 4.0 KB Compress

Name: Baloon Fight

Sort Name: balloon fight

Max players: One player / Two players, not simultaneously / [✓ Two players, simultaneously]

Release date (YYYY-MM-DD): 1986-08-01

Save Count: 0

Publisher: NINTENDO

Command line (for advanced users only!): /bin/clover-kachikachi-wr /usr/share/games/nes/kachikachi/C

Game Genie codes (comma separated):

Box Art: Default Browse Google

Add more games | Export games | Synchronize selected games with NES/SNES Mini

SSH | 33 game(s) selected | 416.0 KB / 230.5 MB

hakchi Community Edition

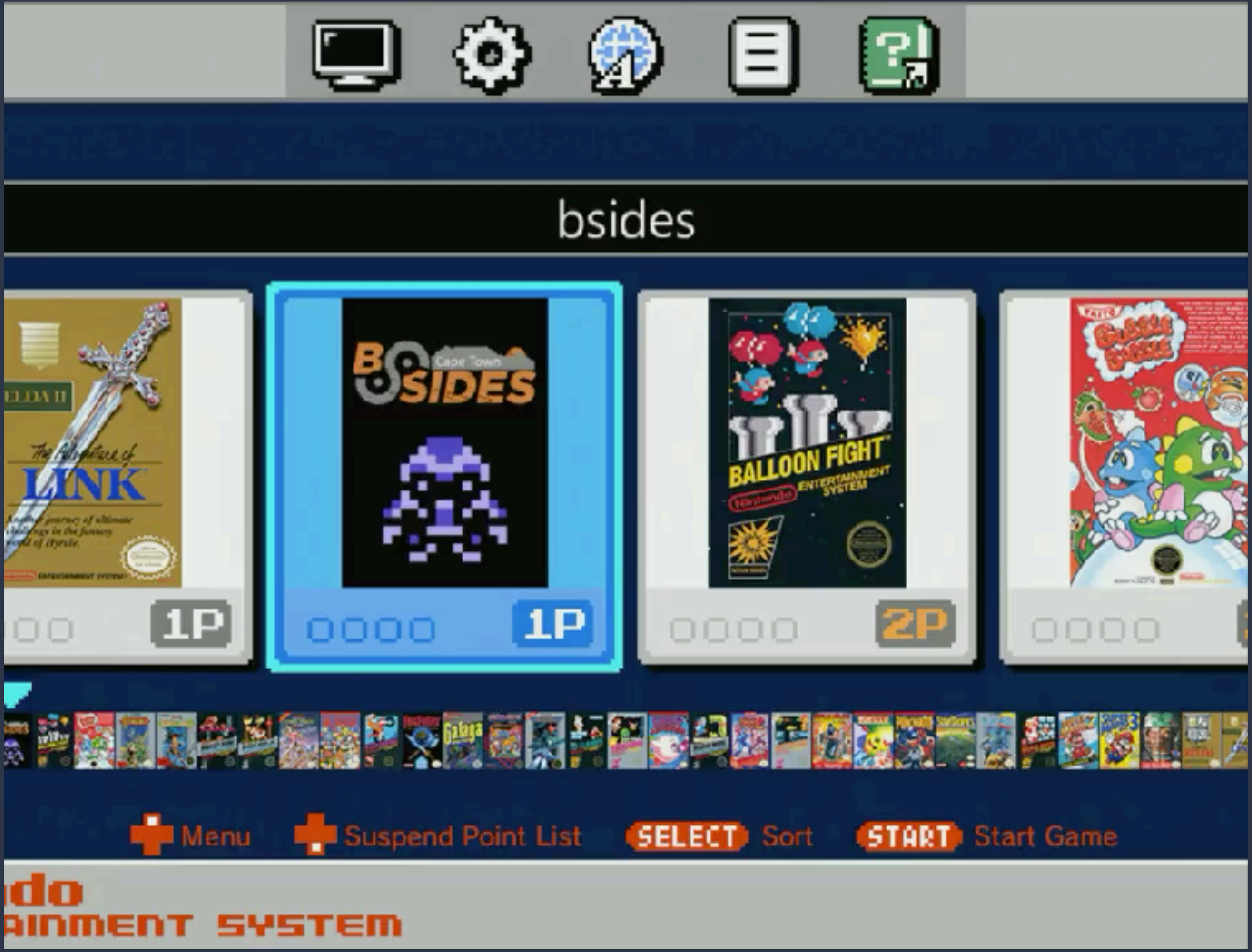
SHALL WE PLAY A GAME?

The "hakchi" mod/hack:

- changes the firmware
- changes startup scripts (eg: for mods)
- remaps paths (eg: symlinks ROM location)
- allows custom NES roms

SHALL WE PLAY A GAME?

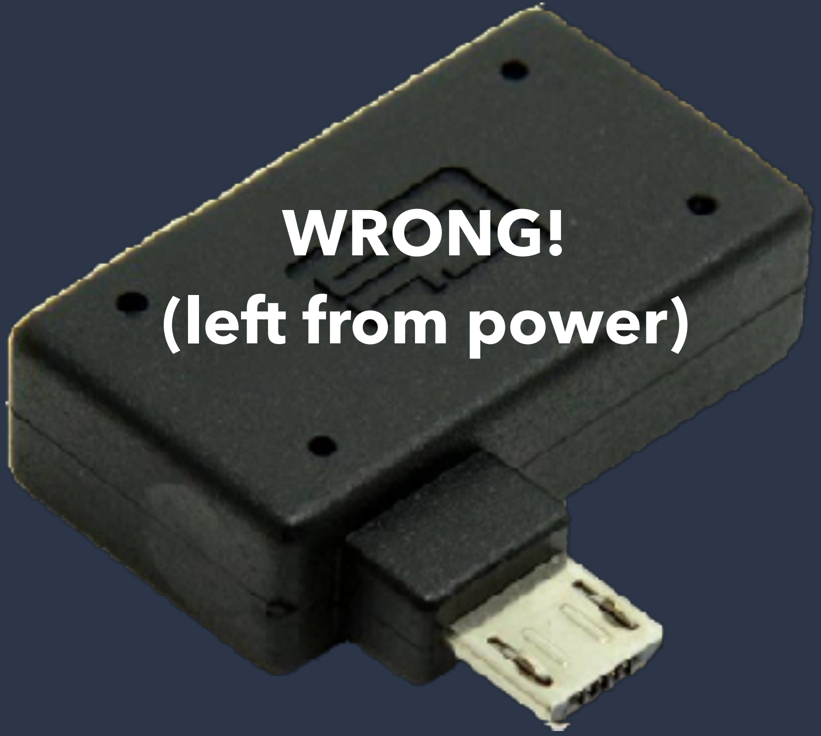
allows custom NES roms

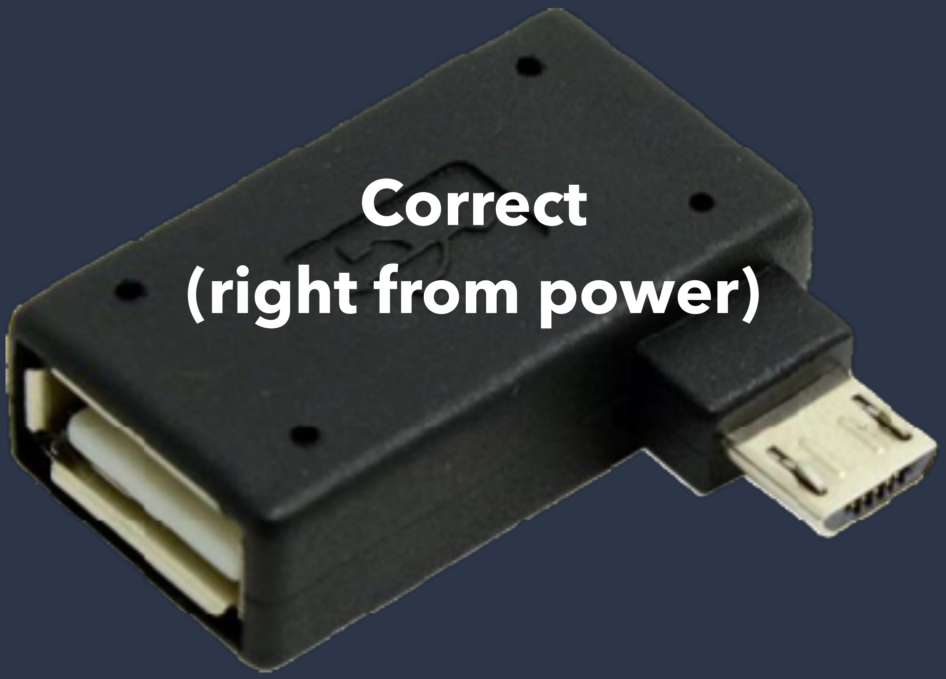


SHALL WE PLAY A GAME?

The "hakchi" mod/hack:

- adds some USB support via USB Host OTG adapters
- "90 degree" adapter seems to work well
- prevents Hakchi from communicating with your console





SHALL WE PLAY A GAME?

The "hakchi" mod/hack:

- USB storage - allowing more games (and emulators)
- formatting as NTFS seems to work best
- use https://github.com/swingflip/Hakchi-USB-tools
- not all flash drives work - but the "SanDisk Cruzer Blade" flash drives seem to
- mounts to "/media"

SHALL WE PLAY A GAME?

The "hakchi" mod/hack:

- enables ftp + ssh (login "root" with no password)
- host-networking on ip: 169.254.13.37 (ports 21 or 22)
  (when not using OTG USB!)

```
login as: root

hakchi v1.0.4-122 by madmonkey

Special thanks to all of the contributors including: potyl, mistydemeo,
skogaby, DanTheMan827, KMFDManic, Mugi, princess_daphie, xorloser, ghoost82,
thomas-alrek, ClusterM, zigg, thomascrha and anyone else I may have forgotten

root@madmonkey:~#
```

SHALL WE PLAY A GAME?

Root?

"Big F*ing deal, what can you do with it"

https://twitter.com/officialmcafee/status/1025009808968171521

So What Can We Do?

SHALL WE PLAY A GAME?

So what can we do?

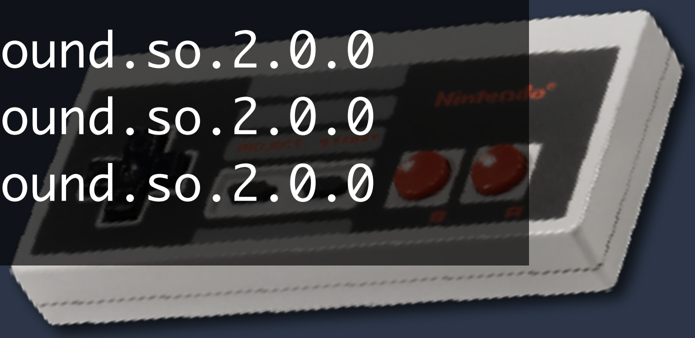
- very few commands/tools
- mostly just busybox widgets
- let's see how far bash + busybox can take us!

SHALL WE PLAY A GAME?

Getting access to process memory

- memory map of "kachikachi" (emulator)

```
root@madmonkey:~# pgrep kachikachi
1869

root@madmonkey:~# cat /proc/1869/maps
00010000-00073000 r-xp 00000000 fe:00 292        /usr/bin/kachikachi
00083000-0008d000 rw-p 00063000 fe:00 292        /usr/bin/kachikachi
0008d000-0009d000 rw-p 00000000 00:00 0
00ebc000-00f8f000 rw-p 00000000 00:00 0          [heap]
b1e40000-b2200000 rw-s 11420000 00:05 398        /dev/mali
b5717000-b5718000 ---p 00000000 00:00 0
[...]
b5718000-b5f17000 rwxp 00000000 00:00 0          [stack:1902]
b5f17000-b5fec000 r-xp 00000000 fe:00 365        /usr/lib/libasound.so.2.0.0
b5fec000-b5ffc000 ---p 000d5000 fe:00 365        /usr/lib/libasound.so.2.0.0
b5ffc000-b6000000 rw-p 000d5000 fe:00 365        /usr/lib/libasound.so.2.0.0
```



SHALL WE PLAY A GAME?

Dumping heap (memory)

```
#!/bin/bash

# Dumps the full memory heap to disk

PID=`pgrep kachikachi`

HEAP=`cat /proc/$PID/smaps | grep heap | awk -F"-" '{print $1}'`
END=`cat /proc/$PID/smaps | grep heap | awk -F"[- ]" '{print $2}'`
HEAP_DEC=`printf "%d" 0x$HEAP`
END_DEC=`printf "%d" 0x$END`

COUNT=$(($END_DEC - $HEAP_DEC))

if [[ "$#" -ne 1 ]]; then
     OUTPUT="/tmp/heap.bin"
else
     OUTPUT=$1
fi

dd if=/proc/$PID/mem bs=1 skip=$HEAP_DEC count=$COUNT 2>/dev/null > $OUTPUT

echo "Dumped heap to: $OUTPUT (`stat -c "%s" $OUTPUT`)"
```

SHALL WE PLAY A GAME?

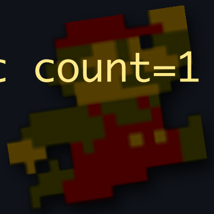
Reading bytes

```
#!/bin/bash

# Reads a byte from a given address

PID=`pgrep kachikachi`

if [ $PID ] then
     HEAP=`cat /proc/$PID/smaps | grep heap | awk -F"-" '{print $1}'`
     if [ $HEAP ] then
          BASE=`printf "%d" 0x$HEAP`
          OFFSET=`printf "%d" $1`
          ADDRESS=$(($BASE + $OFFSET))

          dd if=/proc/$PID/mem bs=1 skip=$ADDRESS count=1 2>/dev/null | \
            xxd -p | awk '{print $1}'
     else
          exit 1
     fi
else
     exit 1
fi
```

SHALL WE PLAY A GAME?

Writing bytes

```
#!/bin/bash

# Writes a byte to a given address

PID=`pgrep kachikachi`

if [ $PID ] then
     HEAP=`cat /proc/$PID/smaps | grep heap | awk -F"-" '{print $1}'`
     if [ $HEAP ] then
          BASE=`printf "%d" 0x$HEAP`
          OFFSET=`printf "%d" $1`
          ADDRESS=$(($BASE + $OFFSET))
          BYTE=`echo -ne "\x${2/0x/}"`

          echo -n $BYTE | dd of=/proc/$PID/mem bs=1 seek=$ADDRESS conv=notrunc count=1
     else
          exit 1
     fi
else
     exit 1
fi
```



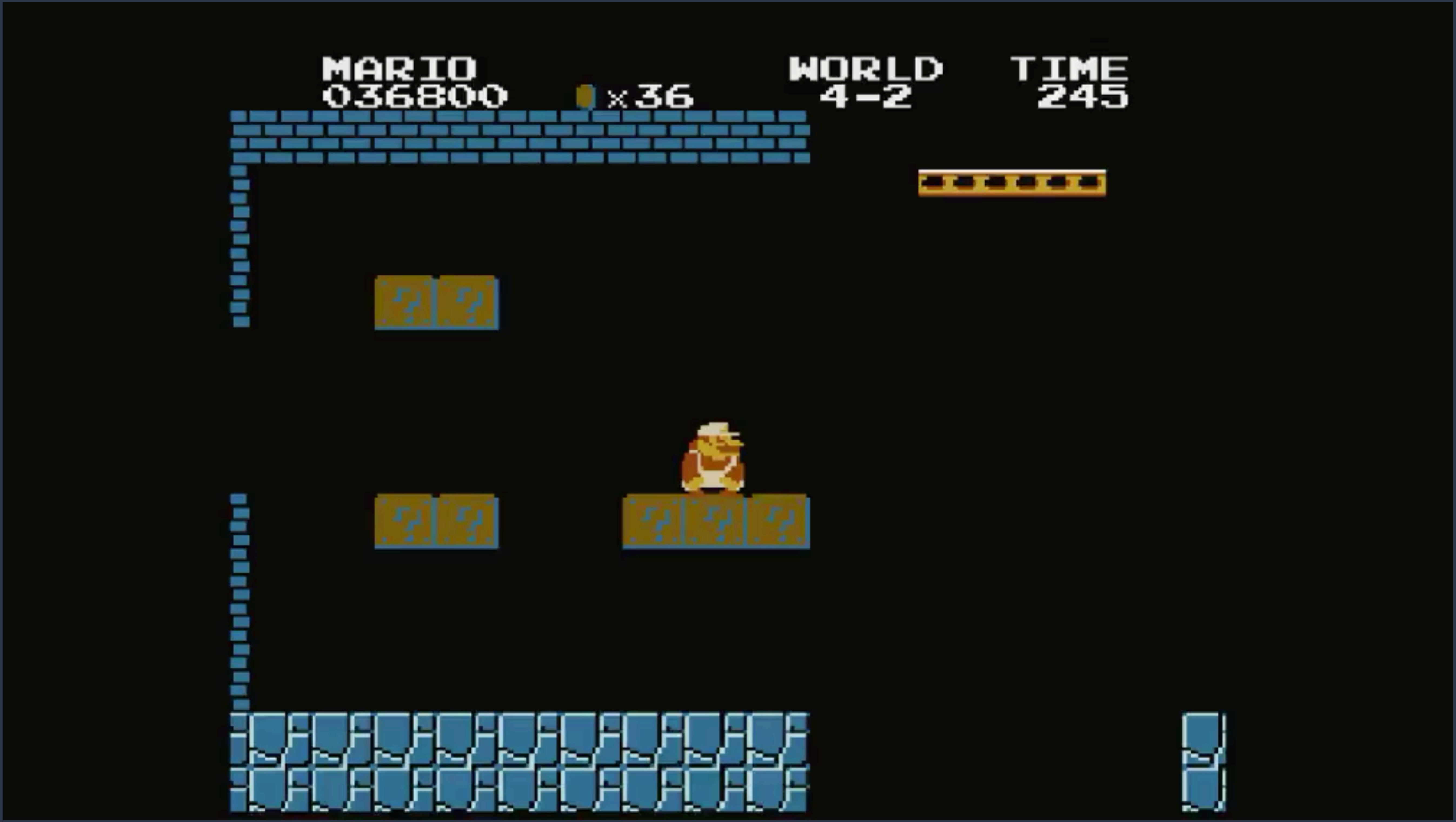
SHALL WE PLAY A GAME?

Using a mod...

- made by CompCom
- https://github.com/CompCom/OptionsMenu
- listens for key combo
- displays custom, configurable, menu

SHALL WE PLAY A GAME?

Extract game state from memory



SHALL WE PLAY A GAME?

Trigger events based on in-game activity

SHALL WE PLAY A GAME?

Manipulate memory / apply Game Genie codes

The Hardware

SHALL WE PLAY A GAME?

Hardware

- SoC: Allwinner R16 (4-core ARMv7)
- RAM: Ask NT5CC128M16IP-DI (256MB)
- NAND: Macronix MX30LF4G18AC-TI (512MB)
- PMU(Power Management Unit): X-POWER AXP223
- HDMI Transmitter: EPMI EP952

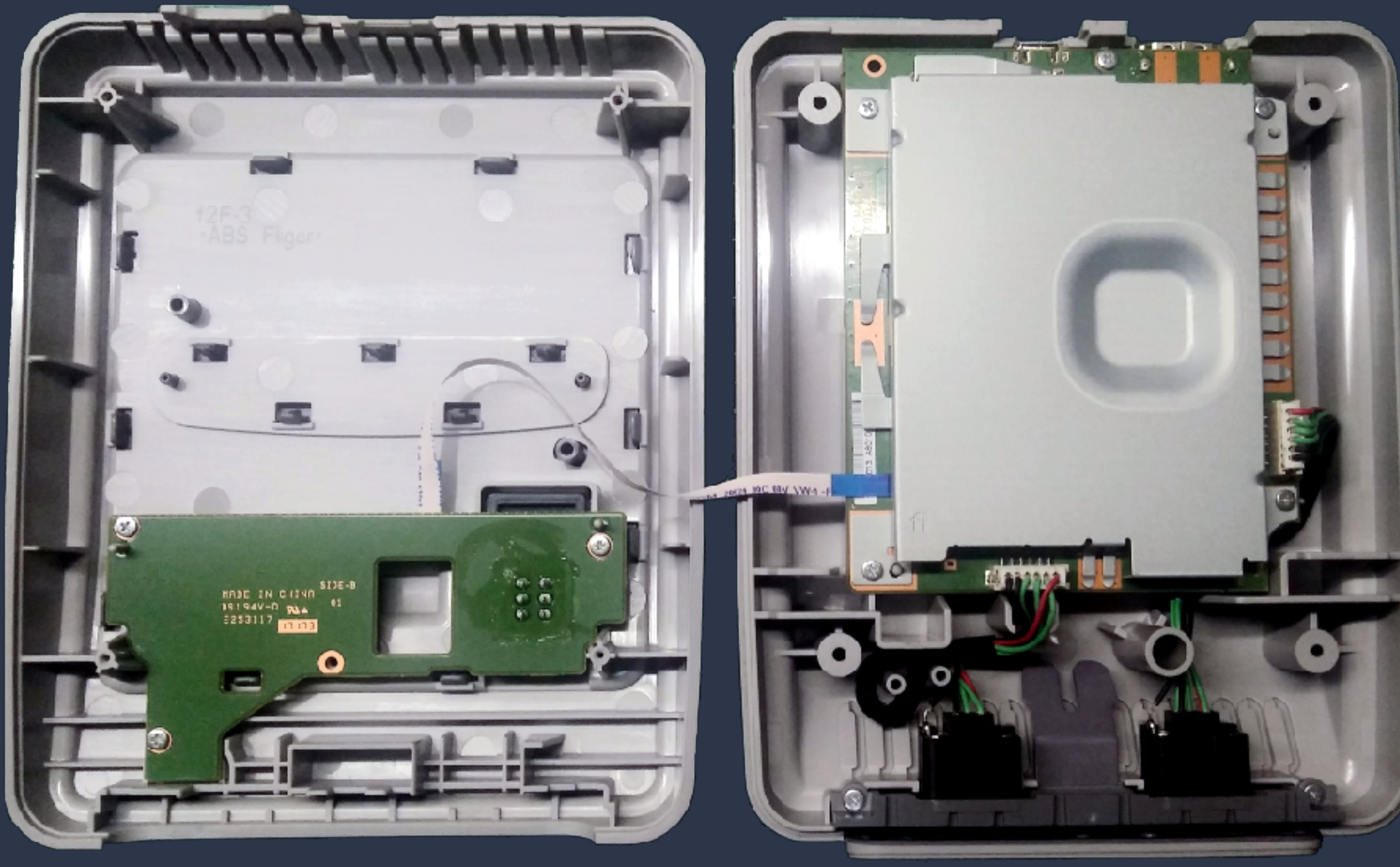
SHALL WE PLAY A GAME?

Hardware

- can put the console into FEL (recovery) mode (a type of recovery mode) and get low-level access
- micro-usb port normally used for power
- with correct drivers enabled in the kernel can use of RNDIS to establish a network connection with host PC
- usb port is OTG - client and usb host mode

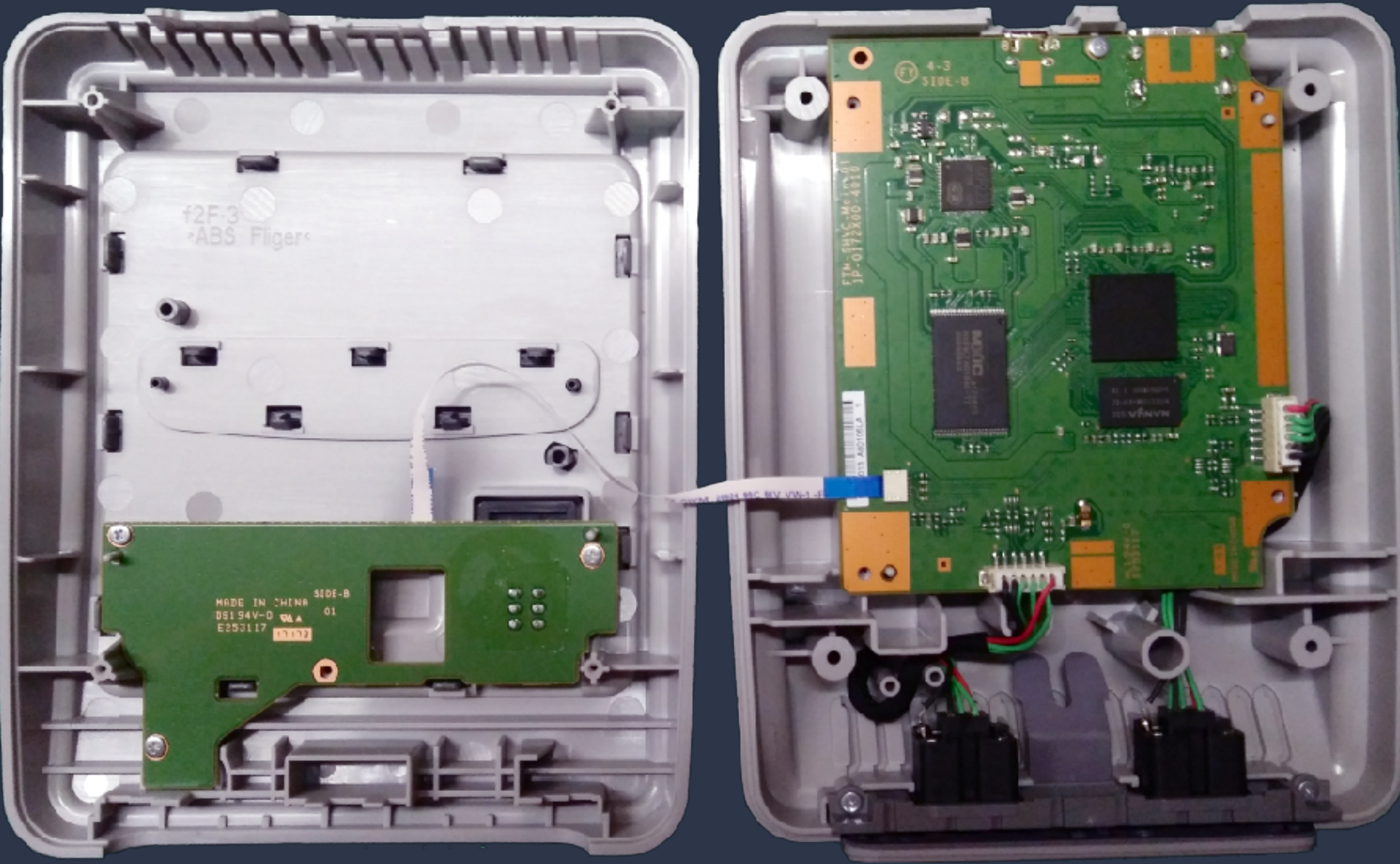
SHALL WE PLAY A GAME?

Internal Photos - SNES Classic



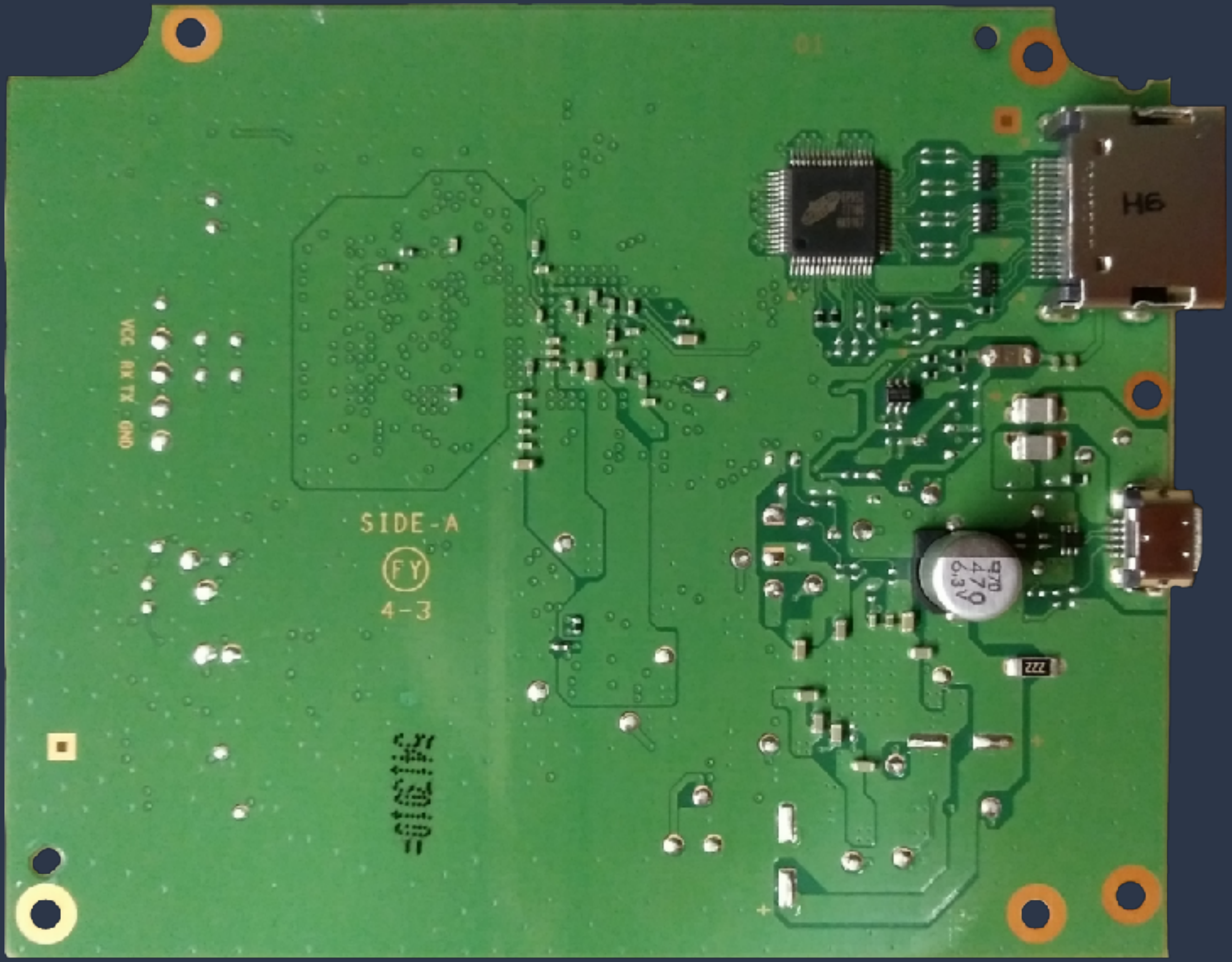
SHALL WE PLAY A GAME?

Internal Photos - SNES Classic



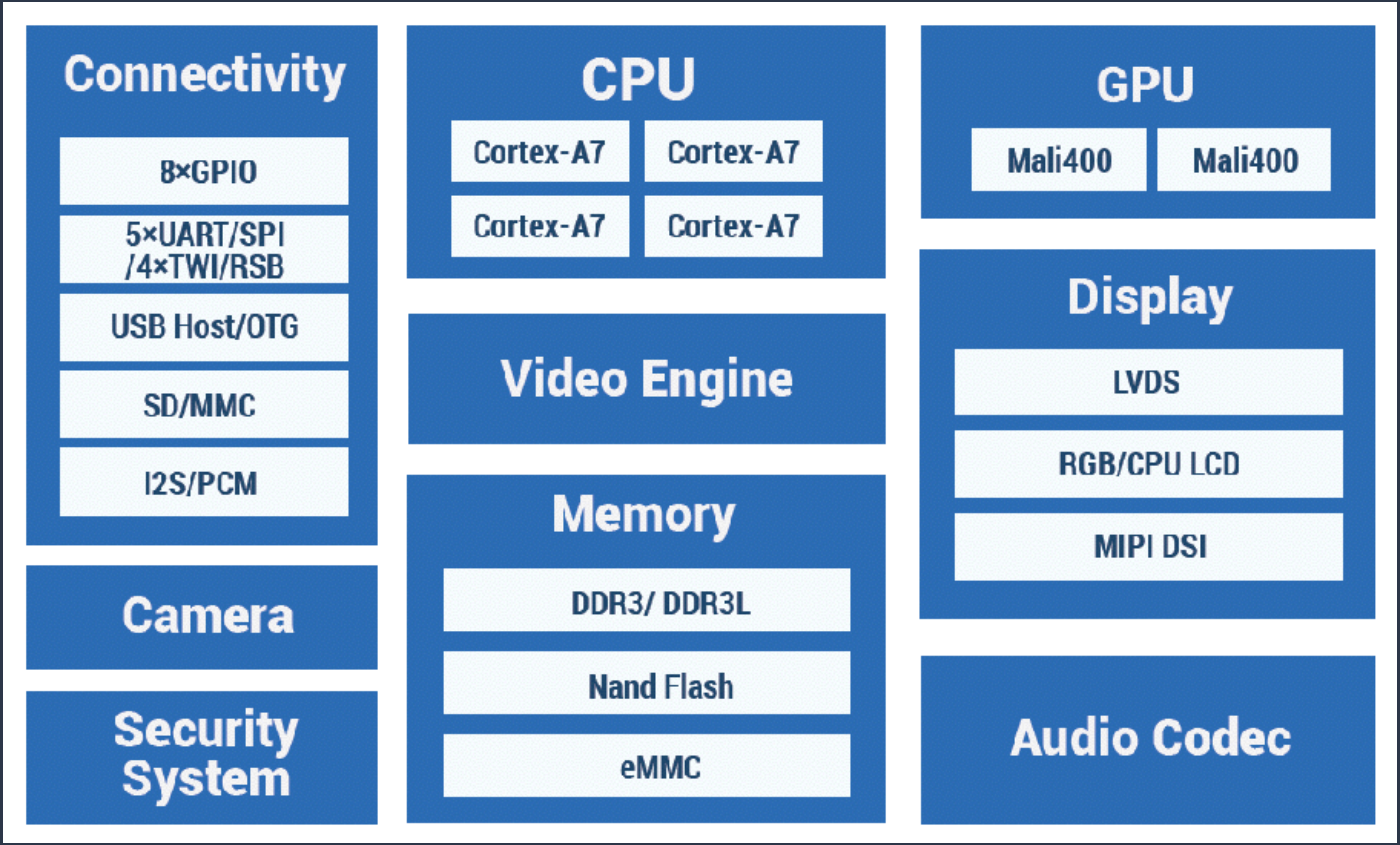
SHALL WE PLAY A GAME?

Internal Photos - SNES Classic



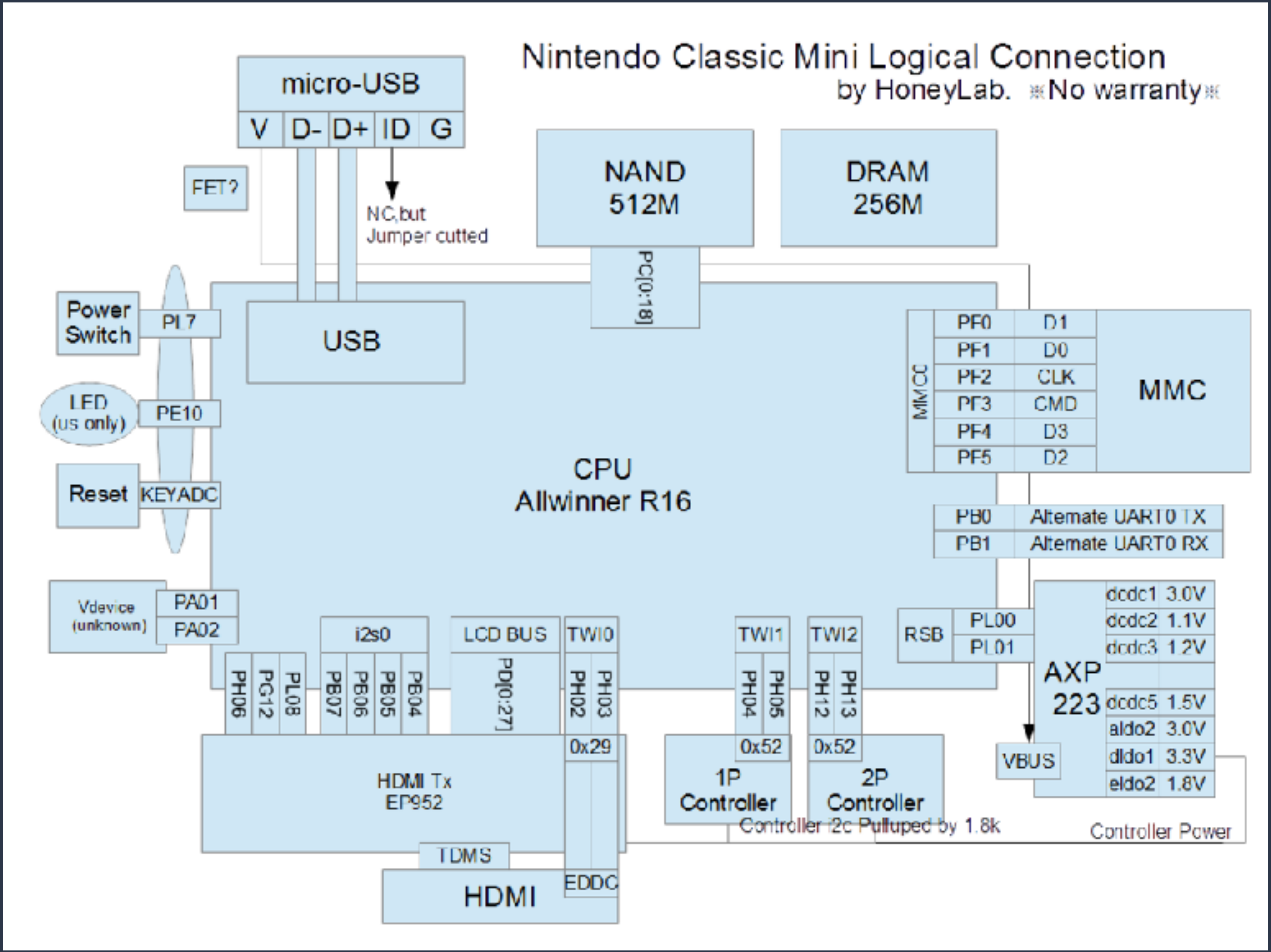
SHALL WE PLAY A GAME?

SoC Allwinner R16



SHALL WE PLAY A GAME?

Hardware connections



SHALL WE PLAY A GAME?

UART / Serial

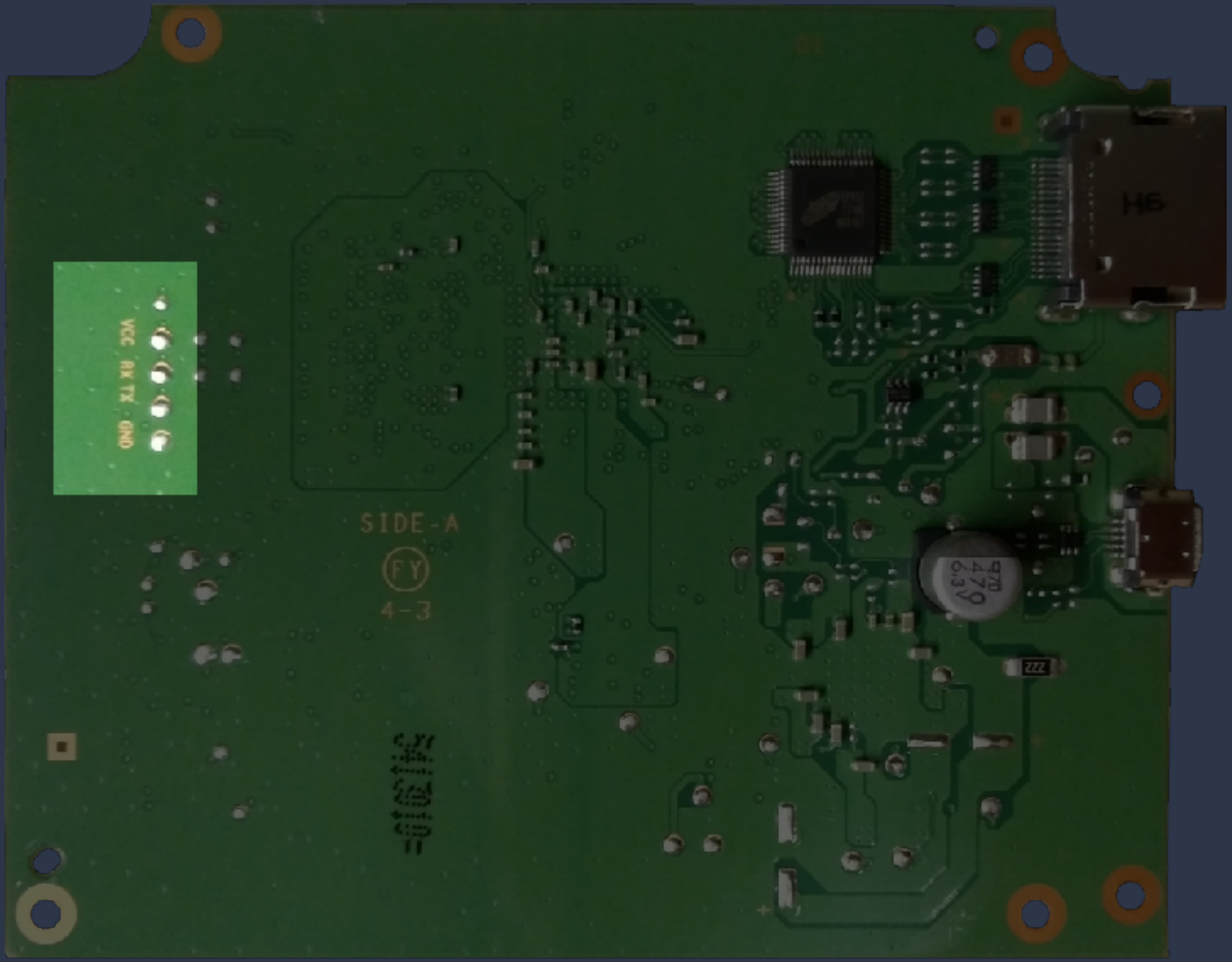
SHALL WE PLAY A GAME?

UART (Serial)

- hidden on the board
- gives access to uboot console and linux boot messages
- baudrate is 115200 and is 3.3v
- requires USB to Serial adapter (FTDI / CH430 etc)

SHALL WE PLAY A GAME?

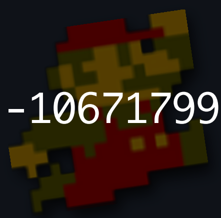
UART pins marked on back of board



SHALL WE PLAY A GAME?

UART (Serial) - boot messages:

```
U-Boot 2011.09-rc1 (May 11 2017 - 16:32:31) Allwinner Technology
[      0.220]version: 1.1.
[      0.223]uboot commit : eb2b275e0877fc98b3ba6d3b7bab225ebecb993a
ready

no battery, limit to dc
no key input
dram_para_set start
dram_para_set end

Using default environment

In:    Out:   Err:
Uncompressing Linux... done, booting the kernel.
ion: failed to create debug files.
<4sunxi_leds_fetch_sysconfig_para script_parser_fetch "leds_para" leds_used = -1067179912

[ audio ] err:try to get audio_pa_ctrl failed!

[I2S1]sunxi-i2s1 cannot find any using configuration for controllers, return directly!
[I2S]sndi2s1 cannot find any using configuration for controllers, return directly
ths_fetch_sysconfig_para: type err  device_used = 1.
fetch C0_LV_count from sysconfig failed
```



SHALL WE PLAY A GAME?

UART (Serial) - boot messages (hakchi):

```
hakchi init script version: 1.0.2

loading hakchi

waiting for usbhost ... 0
usbMode: device
overmounting /bin
overmounting /etc
overmounting /root
overmounting /etc/init.d/S92rndis on /etc/init.d/S90usb-gadget
menu code: 000

Welcome to CLOVER dp-sneseur-nerd release-v2.0.7-0-geb2b275BuildId final.cis.cis-d9-
clvrel0.20170511164229CEST

madmonkey login:
```

SHALL WE PLAY A GAME?

FEL Mode

SHALL WE PLAY A GAME?

FEL mode

- hold "reset" for 5 seconds while powering on
- a low-level USB programming and recovery mode in Boot ROM of Allwinner System-on-Chip
- FEL commands are chip, not device, specific
- read & write memory and execute code
- dump or write kernel and firmware

FEL mode - clean boot

```
[      0.213]

U-Boot 2011.09-rc1 (May 11 2017 - 16:32:31) Allwinner Technology

[      0.221]version: 1.1.0
[      0.223]uboot commit : eb2b275e0877fc98b3ba6d3b7bab225ebecb993a

ready
no battery, limit to dc
```

SHALL WE PLAY A GAME?

FEL mode - upload fes1.bin

```
sunxi-fel write 0x2000 fes1.bin
```

FEL mode - execute fes1.bin

```
sunxi-fel exec 0x2000
```

SHALL WE PLAY A GAME?

FEL mode - Initialising DRAM and switch to FES

```
fes commit : fc3061df4dbd4153819b2d2f141d82b88fea51cf

begin to init dram
DRAM DRIVE INFO: V1.7
DRAM Type =3 (2:DDR2,3:DDR3,6:LPDDR2,7:LPDDR3)
DRAM zq value: 00003bbbDRAM CLK =600 MHZ
ID CHECK VERSION: V0.1
using ic R16
USE PLL DDR1
USE PLL NORMAL
PLL FREQUENCE = 1200 MHZ
DRAM PLL DDR1 frequency extend open !
...
DRAM PHY INTERFACE PARA = 02040102
DRAM VTC is disable
DRAM dynamic DQS/DQ ODT is on
DRAM DQS gate is PD mode.
DRAM one rank training is on,the value is 91003587
DRAM work mode register value = 004318d4
DRAM SIZE =256 M
set one rank ODTMAP
DRAM simple test OK.
init dram ok
```

SHALL WE PLAY A GAME?

FES mode - upload uboot.bin

```
sunxi-fel write 0x47000000 uboot.bin
```

FES mode - command to run instead of kernel

```
sunxi_flash phy_read 47400000 30 20;efex_test
```

FES mode - kernel dump boot command

```
sunxi-fel write 0x470604ff kernel_dump_bootcmd
```

SHALL WE PLAY A GAME?

FES mode - uboot booting

```
[    194.359]

U-Boot 2011.09-rc1-00000-g1352b18-dirty (Jan 02 2017 - 10:46:29) Allwinner Technology

[    194.368]version: 1.1.0
[    194.371]uboot commit : fc3061df4dbd4153819b2d2f141d82b88fea51cf

ready

no battery, limit to dc
no key input
dram_para_set start
dram_para_set end
Using default environment

In:    Out:   Err:
Ph
```

SHALL WE PLAY A GAME?

Dump kernel to file

```
sunxi-fel read 0x47400030 0x600000 kernel.dump
```

(can then use additional tools to extract the "keyfile" from it to decrypt the NAND flash storage)

SHALL WE PLAY A GAME?

Weaponizing

SHALL WE PLAY A GAME?

Weaponizing:

- Connectivity
- Tools
- Backdooring

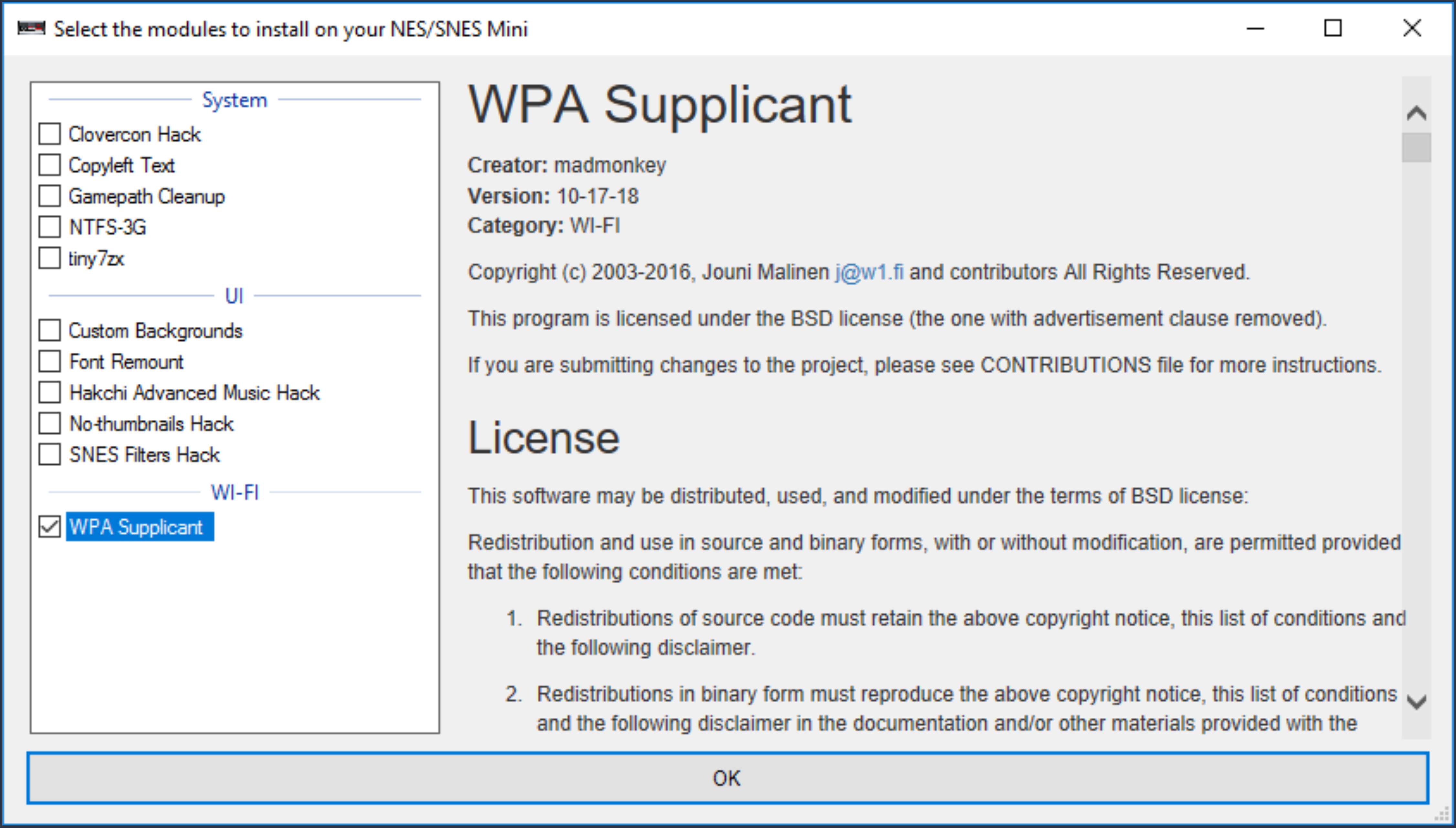
SHALL WE PLAY A GAME?

Adding WiFi:

- "WPA Supplication" (wifi) hmod
- correct OTG cable/adapter
- compatible network dongle (RTL8188 seems to work)
- SSH in, run "wifi-wpa-setup", restart with OTG & wifi plugged in, hope for the best
- no visual indicator it works :/

SHALL WE PLAY A GAME?

wifi-wpa-setup mod:


Select the modules to install on your NES/SNES Mini

System
- [ ] Clovercon Hack
- [ ] Copyleft Text
- [ ] Gamepath Cleanup
- [ ] NTFS-3G
- [ ] tiny7zx

UI
- [ ] Custom Backgrounds
- [ ] Font Remount
- [ ] Hakchi Advanced Music Hack
- [ ] No-thumbnails Hack
- [ ] SNES Filters Hack

WI-FI
- [✓] WPA Supplicant

WPA Supplicant

Creator: madmonkey
Version: 10-17-18
Category: WI-FI

Copyright (c) 2003-2016, Jouni Malinen j@w1.fi and contributors All Rights Reserved.

This program is licensed under the BSD license (the one with advertisement clause removed).

If you are submitting changes to the project, please see CONTRIBUTIONS file for more instructions.

License

This software may be distributed, used, and modified under the terms of BSD license:

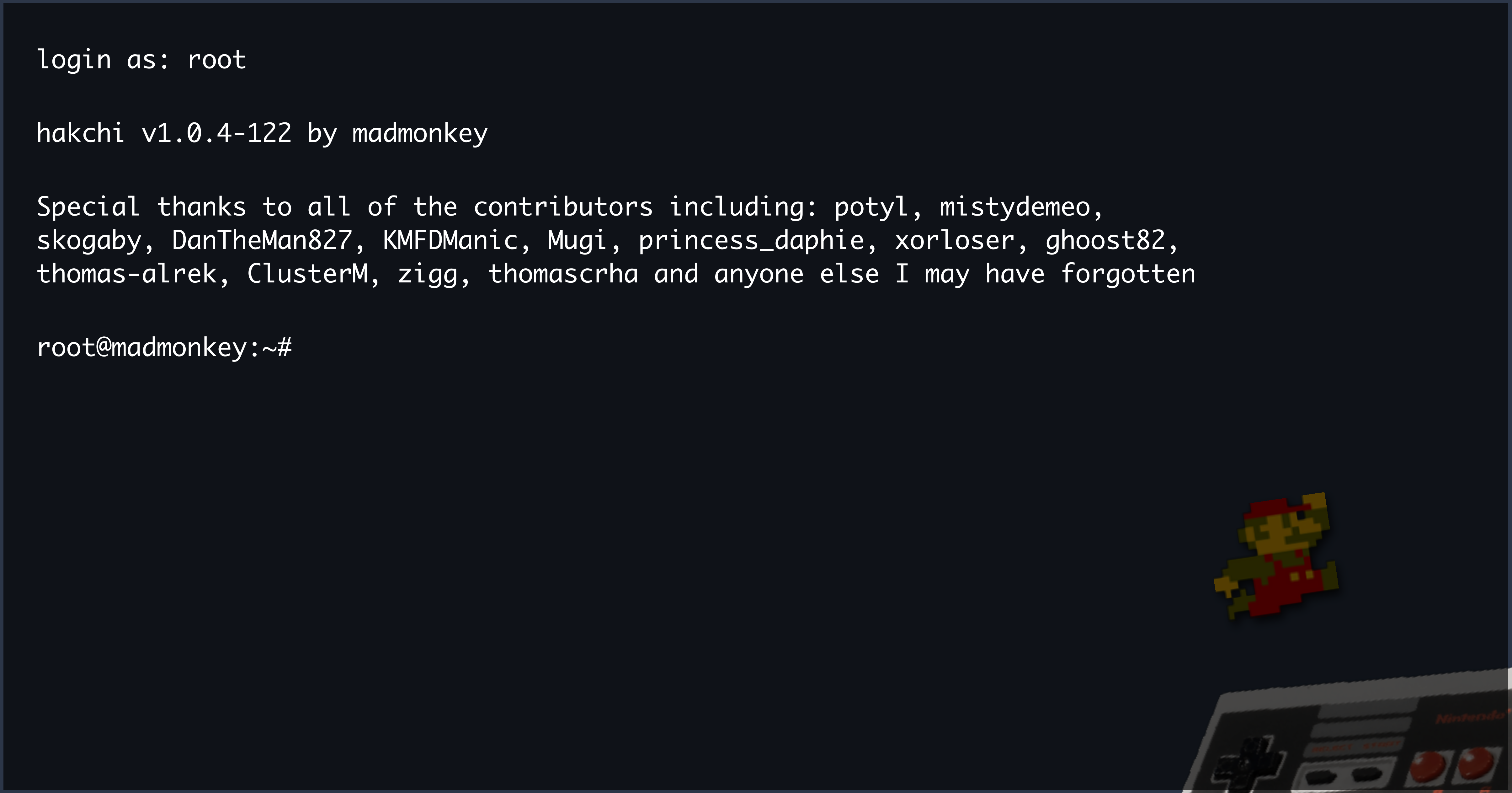
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the

SHALL WE PLAY A GAME?

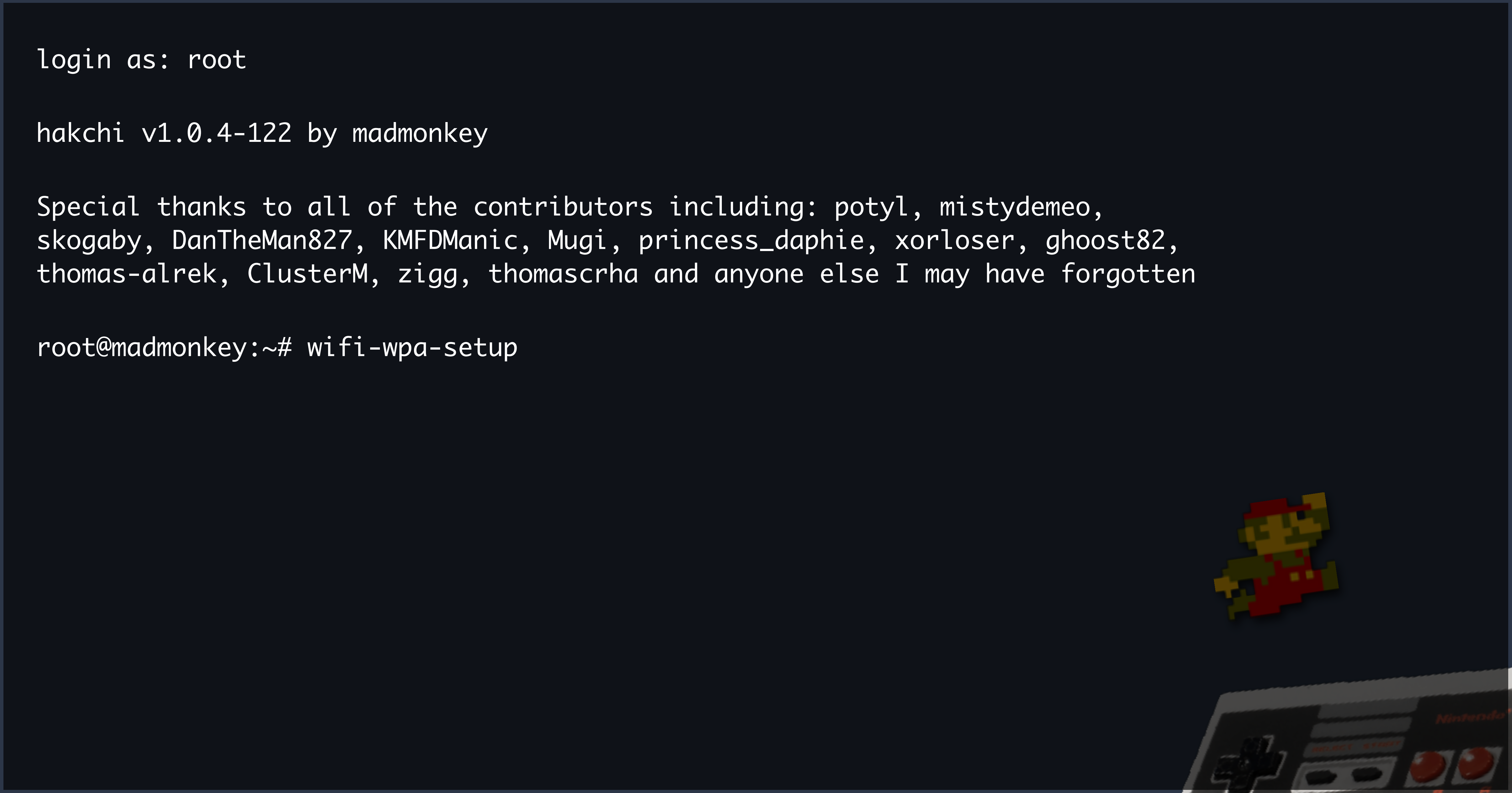
wifi-wpa-setup mod:



```
login as: root

hakchi v1.0.4-122 by madmonkey

Special thanks to all of the contributors including: potyl, mistydemeo,
skogaby, DanTheMan827, KMFDManic, Mugi, princess_daphie, xorloser, ghoost82,
thomas-alrek, ClusterM, zigg, thomascrha and anyone else I may have forgotten

root@madmonkey:~#
```

SHALL WE PLAY A GAME?

wifi-wpa-setup mod:



```
login as: root

hakchi v1.0.4-122 by madmonkey

Special thanks to all of the contributors including: potyl, mistydemeo,
skogaby, DanTheMan827, KMFDManic, Mugi, princess_daphie, xorloser, ghoost82,
thomas-alrek, ClusterM, zigg, thomascrha and anyone else I may have forgotten

root@madmonkey:~# wifi-wpa-setup
```

SHALL WE PLAY A GAME?

wifi-wpa-setup mod:

```
login as: root

hakchi v1.0.4-122 by madmonkey

Special thanks to all of the contributors including: potyl, mistydemeo,
skogaby, DanTheMan827, KMFDManic, Mugi, princess_daphie, xorloser, ghoost82,
thomas-alrek, ClusterM, zigg, thomascrha and anyone else I may have forgotten

root@madmonkey:~# wifi-wpa-setup

Enter your SSID (Your SSID must not include spaces):
D-Link

Enter your Wi-Fi password:
hunter2

Details entered. You can reset your console now. Make sure that only your wifi
adapter is connected on first boot. When your console boots, you need to turn
off and reconnect your OTG hub and USB drive (if applicable). Failure to do so
will mean you will have to restart the process.
```

SHALL WE PLAY A GAME?

Adding tools:

- via mods (eg: wireless tools, gdb)
- see HakchiResources.com "Experimental" mods
- or... add our own

SHALL WE PLAY A GAME?

Porting GoBuster:

(url/dns brute forcer for webapps)

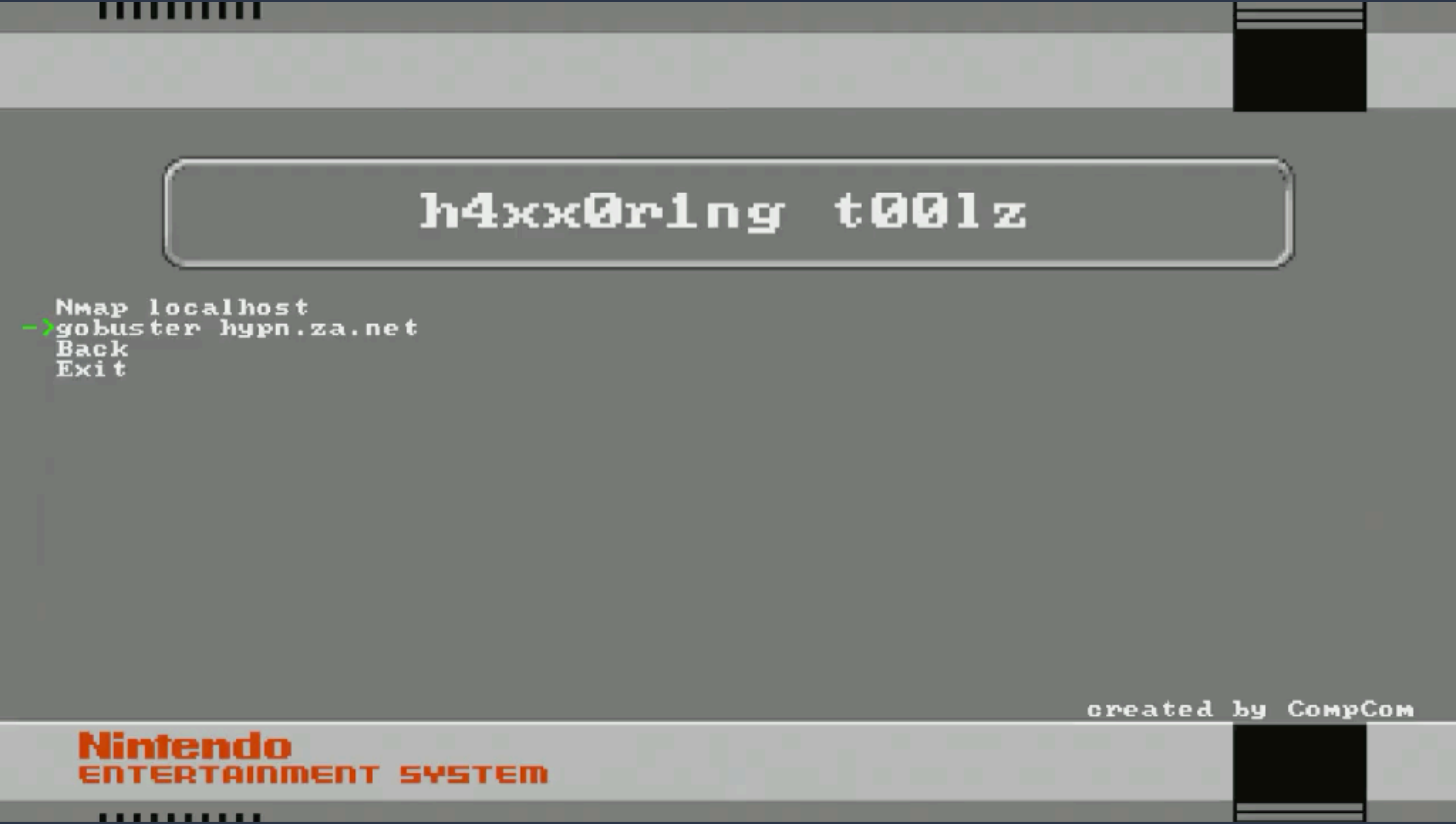
- git clone and compile with static flags:

```
CGO_ENABLED=0 GOOS=linux GOARM=7 GOARCH=arm go build -a \
-ldflags '-extldflags "-static"' .
```

- output of "file":

```
gobuster: ELF 32-bit LSB executable, ARM, EABI5 version 1 (SYSV),
statically linked, not stripped
```

SHALL WE PLAY A GAME?



h4xx0r1ng t00lz

```
  Nmap localhost
->gobuster hypn.za.net
  Back
  Exit
```

created by CompCom

SHALL WE PLAY A GAME?

Porting Nmap:

- Kali is available for "ARMHF"
- can be run on a Raspberry Pi
- comes bundled with tools :D
- output of "file":

```
/usr/bin/nmap: ELF 32-bit LSB pie executable ARM, EABI5 version 1 (SYSV),
dynamically linked, interpreter /lib/ld-linux.so.3, for GNU/Linux 3.2.0
```

SHALL WE PLAY A GAME?

Porting Nmap:

- output of "ldd":

```
libpcre.so.3 => /lib/arm-linux-gnueabi/libpcre.so.3 (0xb6c37000)
libssh2.so.1 => /lib/arm-linux-gnueabi/libssh2.so.1 (0xb6c02000)
libssl.so.1.1 => /lib/arm-linux-gnueabi/libssl.so.1.1 (0xb6ba1000)
libcrypto.so.1.1 => /lib/arm-linux-gnueabi/libcrypto.so.1.1 (0xb69cc000)
libz.so.1 => /lib/arm-linux-gnueabi/libz.so.1 (0xb69a2000)
liblua5.3.so.0 => /lib/arm-linux-gnueabi/liblua5.3.so.0 (0xb6964000)
liblinear.so.3 => /lib/arm-linux-gnueabi/liblinear.so.3 (0xb6944000)
libstdc++.so.6 => /lib/arm-linux-gnueabi/libstdc++.so.6 (0xb67fe000)
libm.so.6 => /lib/arm-linux-gnueabi/libm.so.6 (0xb673b000)
libgcc_s.so.1 => /lib/arm-linux-gnueabi/libgcc_s.so.1 (0xb670c000)
libc.so.6 => /lib/arm-linux-gnueabi/libc.so.6 (0xb65b9000)
/lib/ld-linux.so.3 (0xb6fc8000)
libpthread.so.0 => /lib/arm-linux-gnueabi/libpthread.so.0 (0xb658e000)
libgcrypt.so.20 => /lib/arm-linux-gnueabi/libgcrypt.so.20 (0xb64c1000)
libdl.so.2 => /lib/arm-linux-gnueabi/libdl.so.2 (0xb64ae000)
libblas.so.3 => /lib/arm-linux-gnueabi/libblas.so.3 (0xb6416000)
libgpg-error.so.0 => /lib/arm-linux-gnueabi/libgpg-error.so.0 (0xb63ec000)
libgfortran.so.5 => /lib/arm-linux-gnueabi/libgfortran.so.5 (0xb630b000)
```

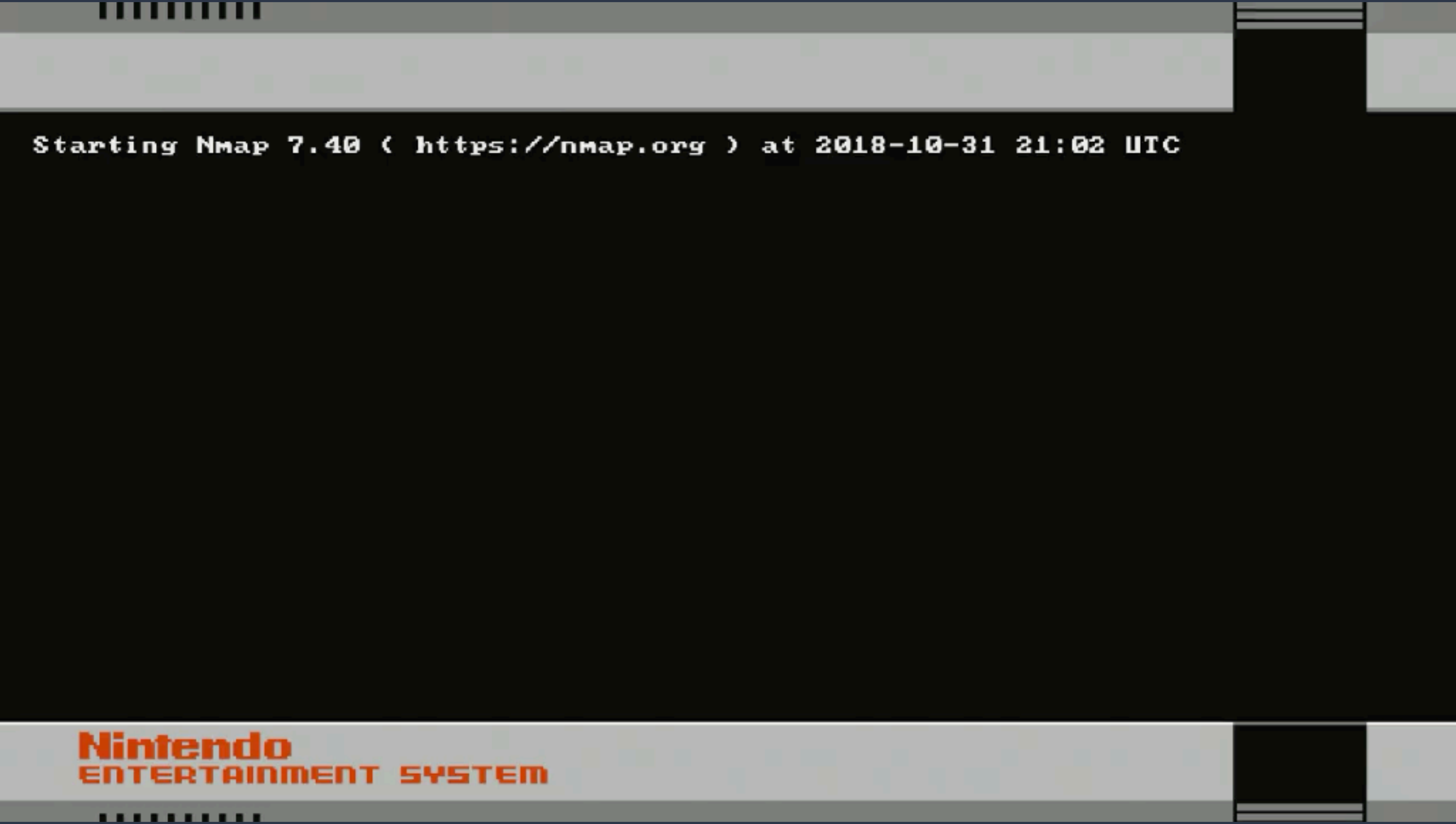
SHALL WE PLAY A GAME?

Porting Nmap:

- copy binary + files + dependencies to NES Classic
- use "LD_LIBRARY_PATH" to load dependencies when running

```
LD_LIBRARY_PATH=. ./nmap <target> <options>
```

SHALL WE PLAY A GAME?



```
Starting Nmap 7.40 ( https://nmap.org ) at 2018-10-31 21:02 UTC
```

SHALL WE PLAY A GAME?

Porting socat:

- In Ubuntu 16.04

```
sudo apt install gcc-4.8-arm-linux-gnueabihf

# download and extract socat .tar.get:
# http://www.dest-unreach.org/socat/download/socat-1.7.3.2.tar.gz

./configure LDFLAGS="-static"
sed -i 's/gcc/arm-linux-gnueabihf-gcc/g' Makefile
make
```

- Output of "file":

```
file ./socat
  => ELF 32-bit LSB executable, ARM, EABI5 version 1 (GNU/Linux),
statically linked, for GNU/Linux 3.2.0
```

SHALL WE PLAY A GAME?

Compiling sources:

- using a Raspberry Pi + Kali ARMHF ✓

```
gcc slideshow.c -o slideshow --static
  => ELF 32-bit LSB executable, ARM, EABI5 version 1 (...), statically linked, for GNU/Linux 3.2.0
```

- use GCC cross compilation ✓

```
arm-linux-gnueabihf-gcc slideshow.c -o slideshow --static
  => ELF 32-bit LSB executable, ARM, EABI5 version 1 (...), statically linked, for GNU/Linux 3.2.0
```

- using DockCross (dockcross-linux-armv7) ✗

```
./dockcross-linux-armv7 bash -c '$CC slideshow.c -o slideshow --static'
  => ELF 32-bit LSB executable, ARM, EABI5 version 1 (...), statically linked, for GNU/Linux 4.10.8
```

("for Linux 4.10.8", gives a "FATAL: kernel too old" error)

SHALL WE PLAY A GAME?

These Slides!

SHALL WE PLAY A GAME?

Interacting with the hardware:

- FrameBuffer is used for video: /dev/fb0
- have to pause UI/emulator to write to it

```
kill -STOP `pgrep kachikachi`
kill -STOP `pgrep ReedPlayer-Clover`
```

- controller events can be read from /dev/inputs/joystick
- events have a timestamp, value, type and number
- there's a "/dev/inputs/event4" for reset button event

SHALL WE PLAY A GAME?

Interacting with the hardware:

- "read_event" loop
- if left/right/A/B button
- next/previous slide
- check if video or image file exists
- output using "ffmpeg" or "decodepng" to frame buffer
- these slides are (hopefully) being played from our custom slideshow app \:D/

Other stuff

SHALL WE PLAY A GAME?

Misc stuff:

- hakchi .hmod (mods) are just tar files:

```
tar -czvf "../MyMod.hmod" *
```

- Can have multiple firmwares on a single device and swap between them (eg: NES on SNES)
- Active community on Reddit and Discord
- Wolf3D (and other games) ported across!
- Theres a "FBVNC" (FrameBuffer VNC) but frame rate is low with tearing

SHALL WE PLAY A GAME?

Misc stuff:

- "reset" button sends an event to "/dev/inputs/event4"
- input events have "time", "value", "type" and number"
- can capture + replay "reset" event:

```
cat /dev/inputs/event4 > reset.bin
cat reset.bin > /dev/inputs/event4
```

- was not able to automate/replay joystick inputs (was hoping to make a Mario bot)

SHALL WE PLAY A GAME?

RetroArch:

- RetroArch emulator adds supports from many more platforms:

  * snes9x2010 (Super Famicom/Super Nintendo)
  * gambatte_libretro (Game Boy, Game Boy Color)
  * mgba (Game Boy Advance)
  * genesis_plus_gx (Sega Master System, Genesis/Mega Drive, Game Gear)
  * stella (Atari 2600)
  * mednafen_pce_fast (PC Engine/Turbografx 16)
  * fb_alpha and fb_alpha_cps2 (various arcade machines)
  * picodrive (Sega Master System, Genesis/Mega Drive, Game Gear, Sega)
  * dosbox and more: http://buildbot.libretro.com/nightly/linux/armhf/

SHALL WE PLAY A GAME?

Stuff that sucked:

- Trying to get an OTG adapter (had to order online)
- getting USB and wifi working is mostly blind
- "just" use BuildRoot - https://hakchi.net/hakchi/sdk/
- "just" use GCC <= 4.9 (good luck if you need zlib)
- compiling stuff for SDL is a major pain
- relatively easy to crash to console

SHALL WE PLAY A GAME?

Links

- eBay OTG adapter
- Amazon OTG adapter
- Hakchi Community Edition
- HakchiResources (mods + Discord Channel)

SHALL WE PLAY A GAME?

Our notes + more info:

- https://www.xor.co.za/talks/shall_we_play/ (Dale)
- https://www.hypn.za.net/blog/?p=1272 (Ross)

Thanks!