Nintendo Game And Watch Unlock

Max-Felix Müller

2021
December

Contents

List of Figures

1 The Nintendo Game and Watch

The Nintendo Game and Watch is a handheld gaming system from Nintendo, originally released some years ago. For the aniversary of Mario, a Nintendo character, they remade the console.

Figure 1: Nintendo Game and Watch Mario

The remade console runs three different games and a clock, all emulated. As it turns out, Nintendo uses an STM32H7 for this handheld, which is more or less hacker friendly.

The console further comprises a small display, some buttons, a speaker and a battery with charging circuit.

2 Homebrew on the Game and Watch

Homebrew loosely means applications, that are written at home. There are people that write custom games or other applications for systems like the Nintendo Wii, DS or in this case the Game and Watch.

For example, a port of DOOM already exists for the Game and Watch. Additionally, different emulators can then run on the Game and Watch hardware, e.g. a NES emulator.

3 Hardware Modifications

After opening the console with four "security" screws (the standard Nintendo triwing screws), the backplate easily pops up. There is one very small extra piece for the power button that can fall out, if not carefull.

The battery sits on top. The speaker is angled in such a way, that a slightly larger one can be used. Its port is on the side of the case, but by nearly laying the speaker down, it still can be rather big. The USB-C connector is only used for power.

Figure 2: Inside of the Game and Watch

Next to the speaker connections, a programming and debuggin port is exposed. It comprises five pins. The pins are connected to a pin header that was put on the lower left of the system.

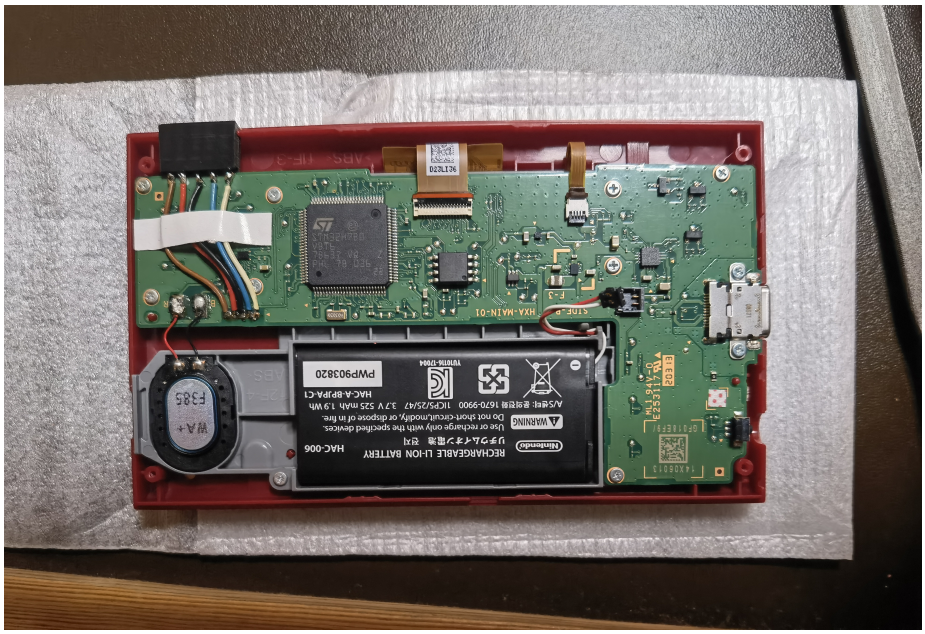

Figure 3: Debugging Connector

To make some space for the cabling, the standoffs from the backplate were removed. Those would provide additional support for the PCB, but is is screwed securely in place, unless the user presses wildly on the D-pad with excessive force.

Figure 4: Modified Backplate

By carefully cutting and filing away the plastic around the edge, the pin header can be fit in place very nicely. The height of the connector is the same as the border of the console, meaning it goes in seemless, if the cutting was done properly. In this project I took extra time to be careful here.

Figure 5: Closed System with Connector

4 Software Modifications

4.1 The System Lock

The STM32H7 is by default locked in such a way, that the flash (firmware) can not be read by a debug connection. This is a safety feature to keep the source code secret. Only the RAM can be read. The microcontroller can also be written to, in this mode.

A second stage of safety would have disabled all debug connections to the controller, rendering it useless for hackers.

There are some nice gentelmen on the internet who figured out how to get around this lock.

4.2 Memory Setup

The console has "basically" three sections of memory.

The flash of the microcontroller holds the firmware of the emulation.

An external 1Mb memory chip holds the three games codes in an encrypted format.

For operation, the game code is decrypted and stored in RAM on the microcontroller.

This might seem obsurd, but the controller does indeed feature only 128kb flash but 1,3Mb RAM.

4.3 Steps in Unlocking

When unlocking the microcontroller, all flash contents are erased. Thus the first step is to backup the original flash (and game code) contents. However, the flash can not be read in this locked state.

To read the flash anyways, a small piece of code is introduced to the microcontroller (remember, it can be written to). This code copies contents of the flash memory piece by piece into the RAM, allowing it to be read.

Since the game code from the external storage is already copied into RAM, it too can be read.

Afterwards, the lock can be removed, erasing the controller memory.

To restore the system, the previously read firmware can be re-written to the controller. This will restore the original functionality, the only difference being, that the flash is no longer read protected.

With the backup on hand, the system can be programmed, reprogrammed and restored as necessary.

4.4 Unlocking the System

To unlock the system, the battery has to be disconnected and the controller is powered via USB.

The debug port is connected to a programmer, best case a STLink, since it is a ST microcontroller. Only the Vcc pin is not connected!

Figure 6: Pinout of the Debug Port

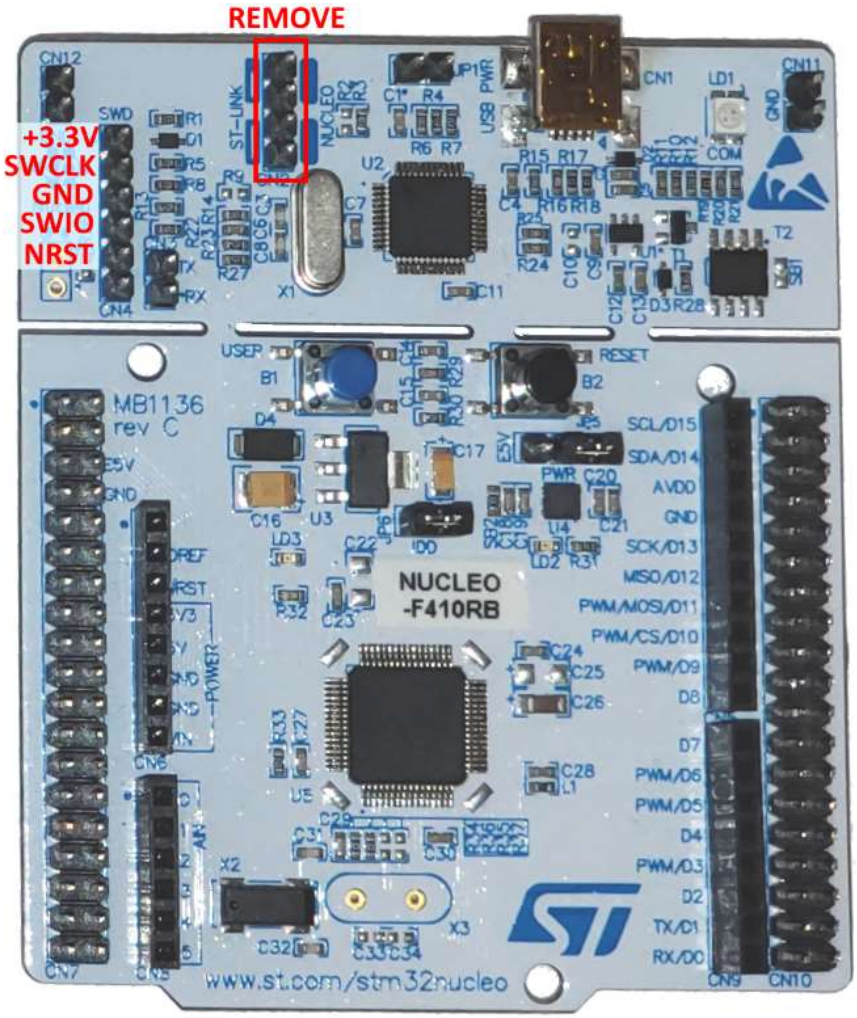

Figure 7: STLink on a Nucleo

Then it is simply a matter of following the instructions and executing the scripts in order.

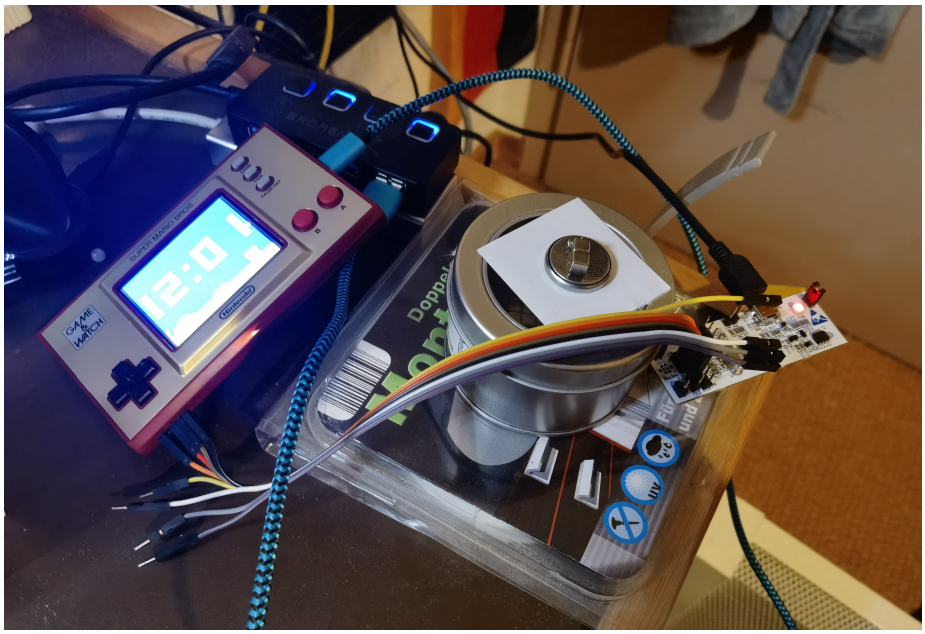

Figure 8: Programming

Now the system runs the original firmware, but in an unlocked state. From here, any program can be written to the controller.

5 Links

References:

Main video with instructions:
https://www.youtube.com/watch?v=-MzmoEFs0bQ

Video with pinout:
https://www.youtube.com/watch?v=Rsi8p5gbaps

Github with instructions and code:
https://github.com/ghidraninja/game-and-watch-backup

STLink Nucleo pinout:
https://visualgdb.com/tutorials/arduino/stm32/bluepill/