March 1, 2019 at 17:48

Making ST-Link V2.1 from Chinese ST-Link V2

- From the sandbox
- Tutorial

Hello, Habr!

In this article I will tell you how to modify ST-Link V2 to ST-Link V2.1.

Perhaps for some it will not be news, but I did not find any special information on this topic in the internet.

Who cares - I ask for a cat.

Foreword

It just so happened that I'm tired of the extra wires.

After thinking a bit, I remembered that on Nucleo and Discovery boards - ST-Link combines SWD and VCP (Virtual Com Port).

The first thing that came to mind was to buy the cheapest of these boards, try to dump the firmware bypassing the protection and fill it in the programmer from China, or to breed a new board.
However, they prompted me a link to GitHub with an already extended bootloader, in the end it turned out what happened.

Getting to work

Modification can only be done on the software version for Windows, the cross-platform software version refuses to update the device!

There are several modification options, and some of them cannot be made if the chip is not suitable (there is not enough memory).

For example, the modification of STM32 + MSD + VCP can only be done if the chip is STM32F1xxCBxx, however, it has an analogue of STM32 + Audio, which will give STM32 + VCP (in principle, which is what we need).

You will need:

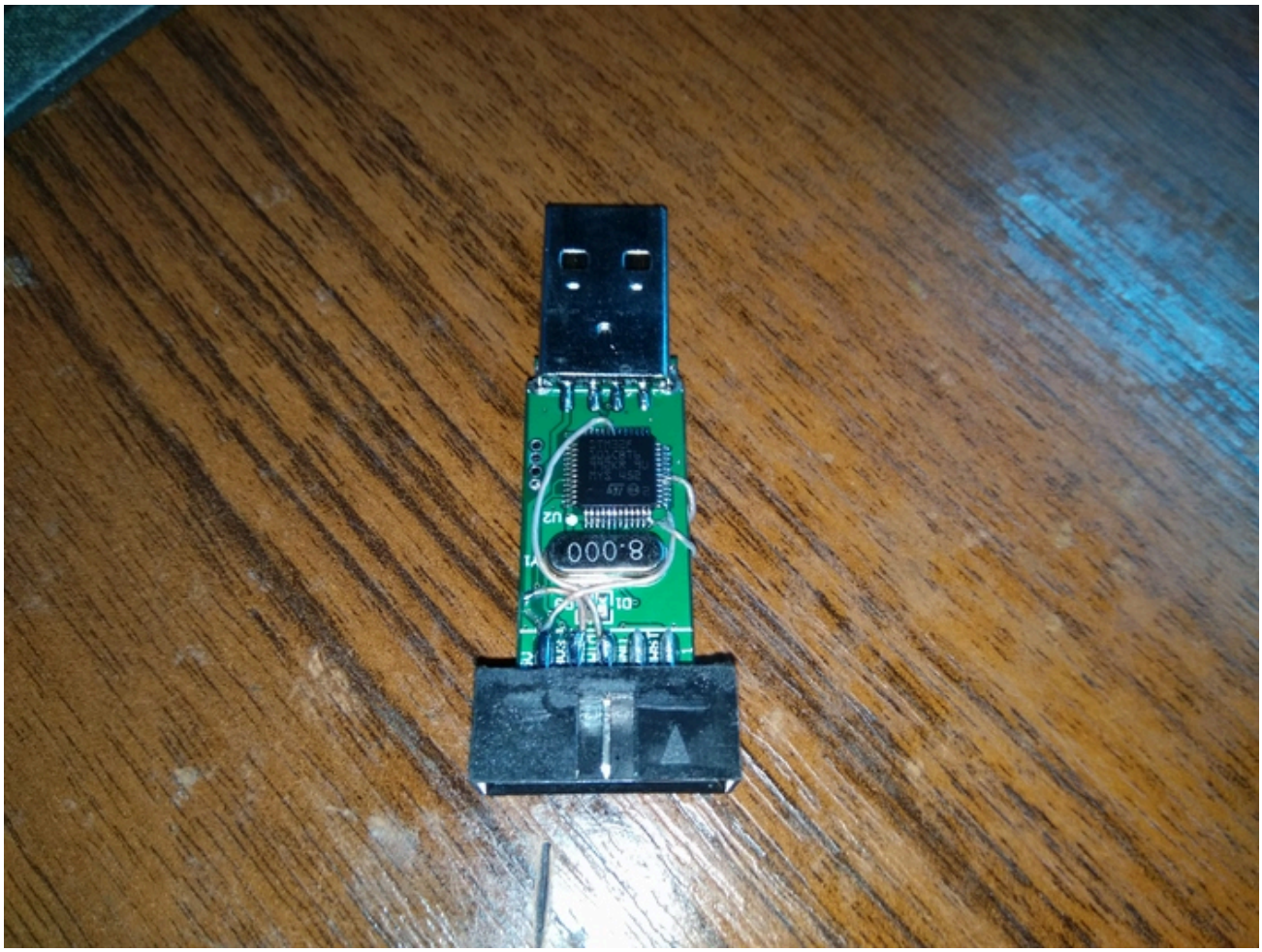
- Soldering iron;
- Multimeter with a proverb;
- PC with Windows (can get through Wine, have not tried);
- Archive with the necessary software and bootloader (PASS: QWK2tn + fM.EdjX6z).
- Chinese clone ST-Link V2;
- USB-UART adapter or a second ST-Link.

We reveal ...

Boards and chips are all different


WAVGAT
RST
SWIM
GND
3.3V
5.0V
SWCLK
SWDIO
GND
3.3V
5.0V
MINI ST-Link V2_A
8.000
SD8.00000


ST-LINK V2

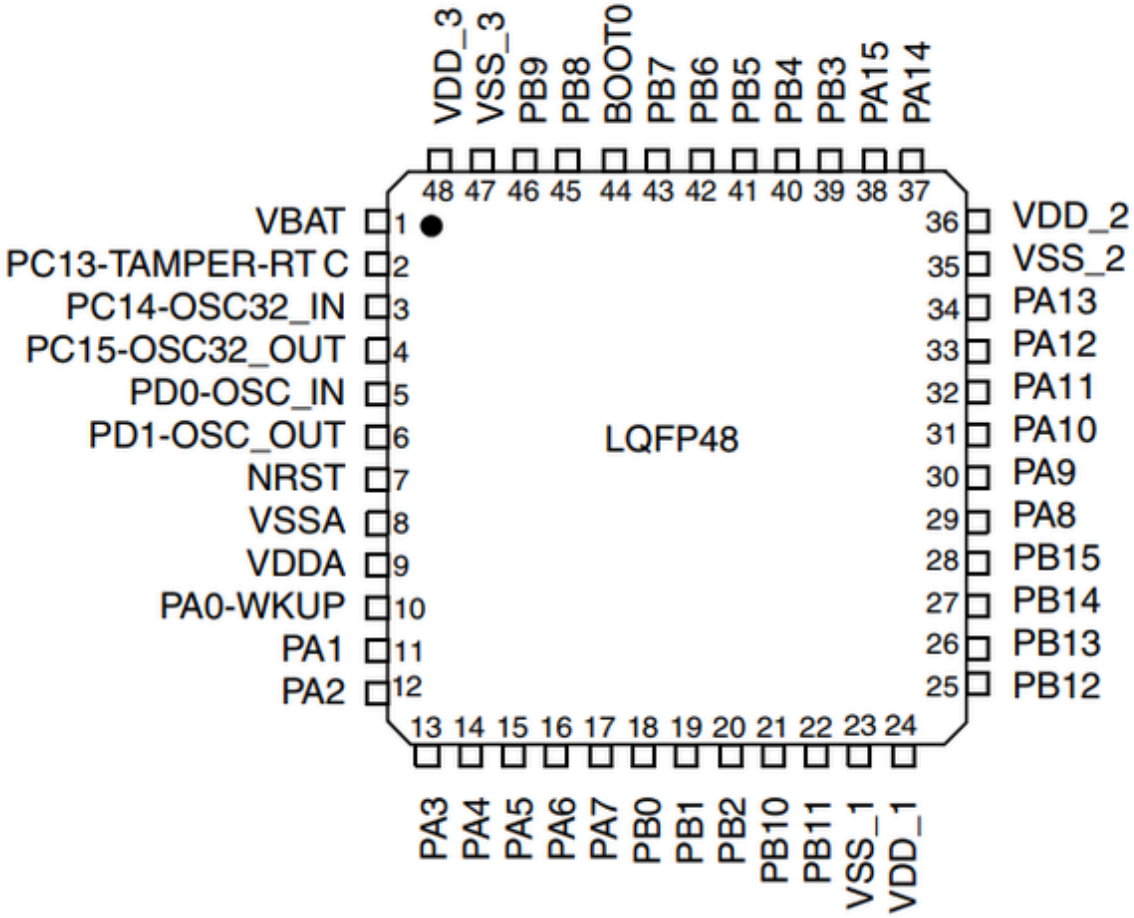

8.000

Firmware

There are two ways - USB-UART (a bit more complicated) or the second ST-Link.

USB UART

Figure 8. STM32F103xx performance line LQFP48 pinout

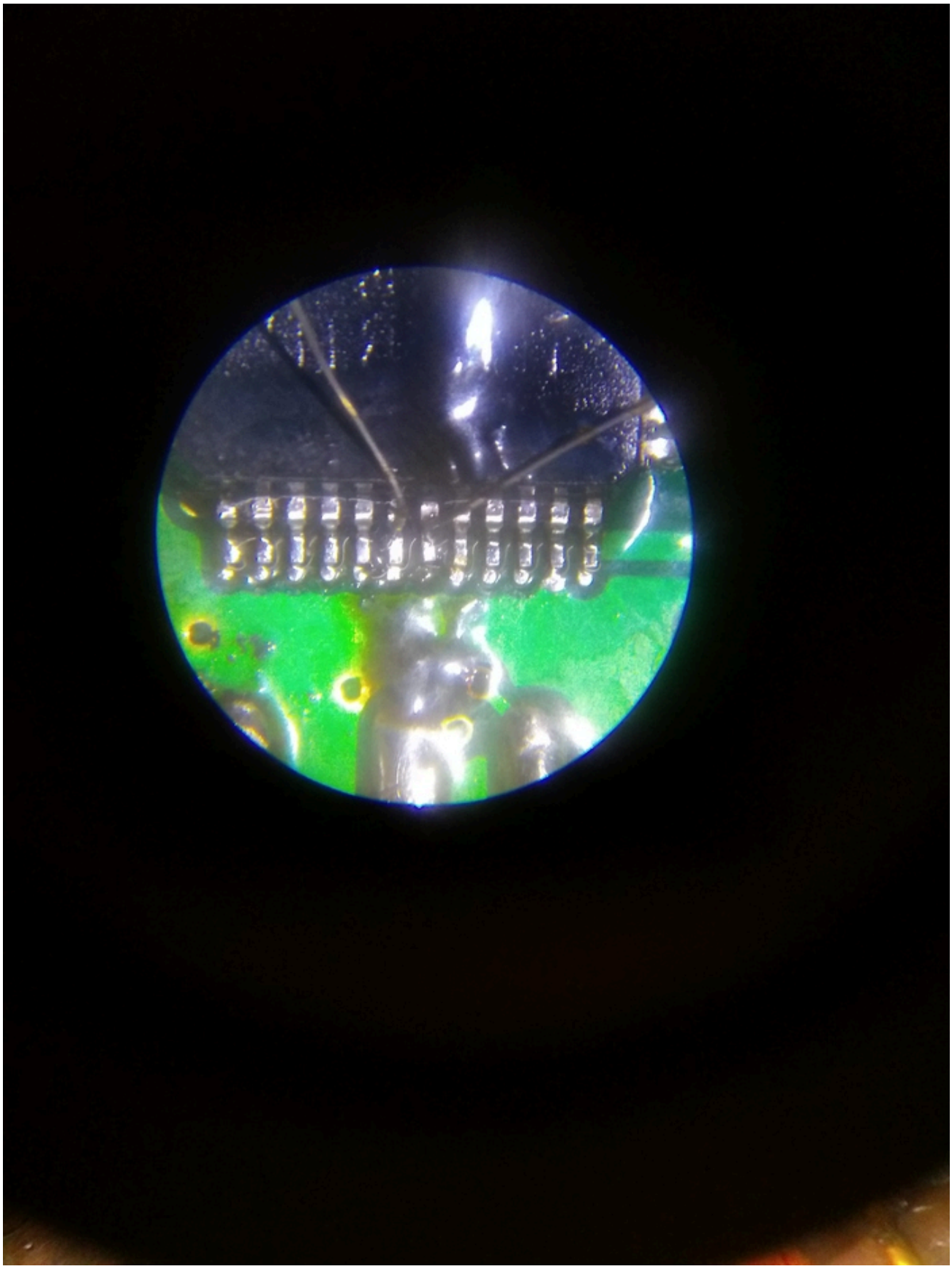


1) By the dial, we find the resistor that is connected to ***BOOT0***.
We make a jumper from the side of this resistor which is connected to ***BOOT0*** to ***3.3v***.

PA9 (TX) can be connected to the LED or the resistor next to it, so we call it.

Solder **UART** to ***PA9*** (***TX***) and ***PA10*** (***RX***).

I did it this way: We



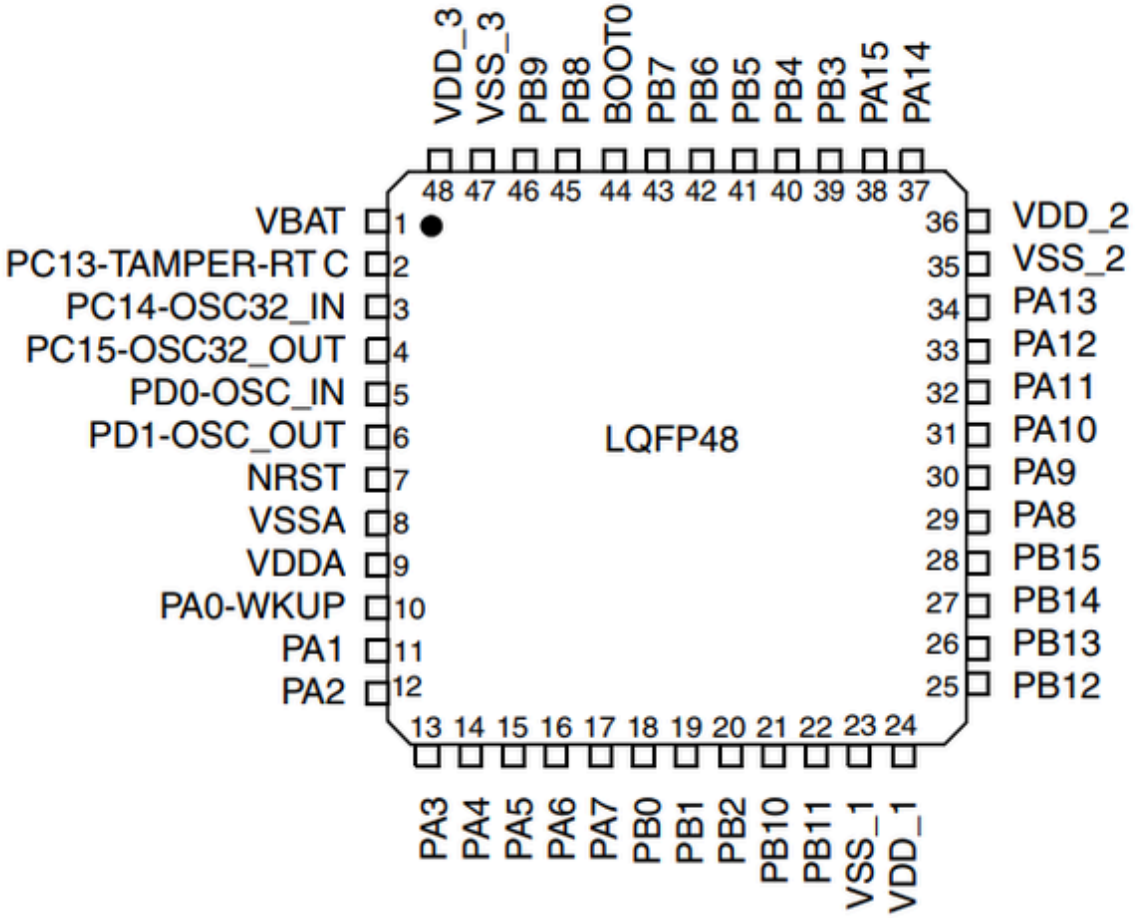
also solder food.

Flashing the Protected-2-1-Bootloader.bin bootloader using the STM32 Flash loader demonstrator .

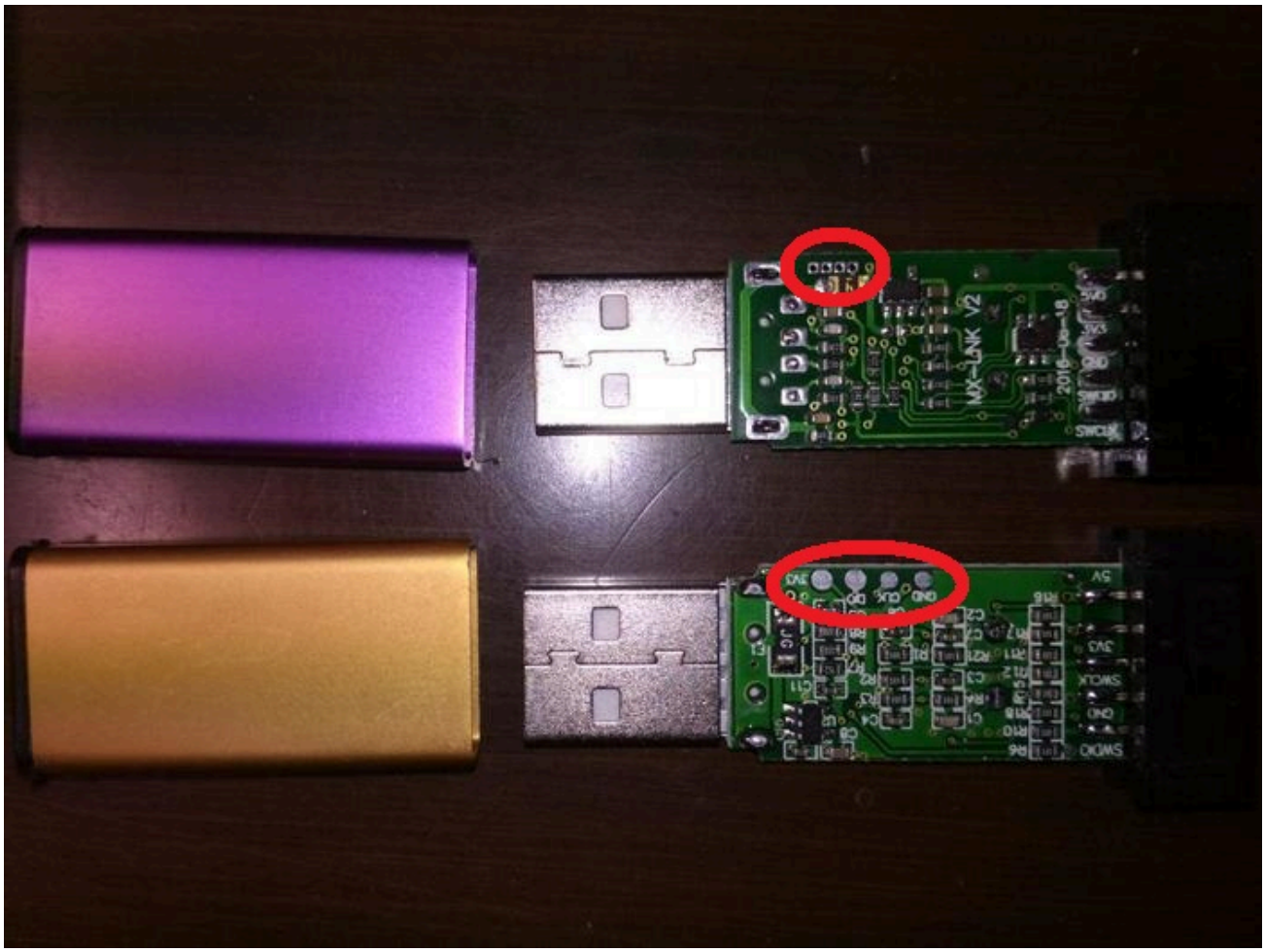
After flashing, solder the jumper, *PA9* and *PA10* (*PA10 is* left if we want to output *SWO*).

ST-Link

Figure 8. STM32F103xx performance line LQFP48 pinout



There are 4 pins on the boards, in some cases they are already marked, otherwise they are called up with respect to *PA13* (*SWDIO*) and *PA14* (*SWCLK*), soldered by the second ST-Link .



We also solder food.

We install STM32 ST-LINK Utility V4.3 from the archive, remove write protection and flash the Protected-2-1-Bootloader.bin bootloader.

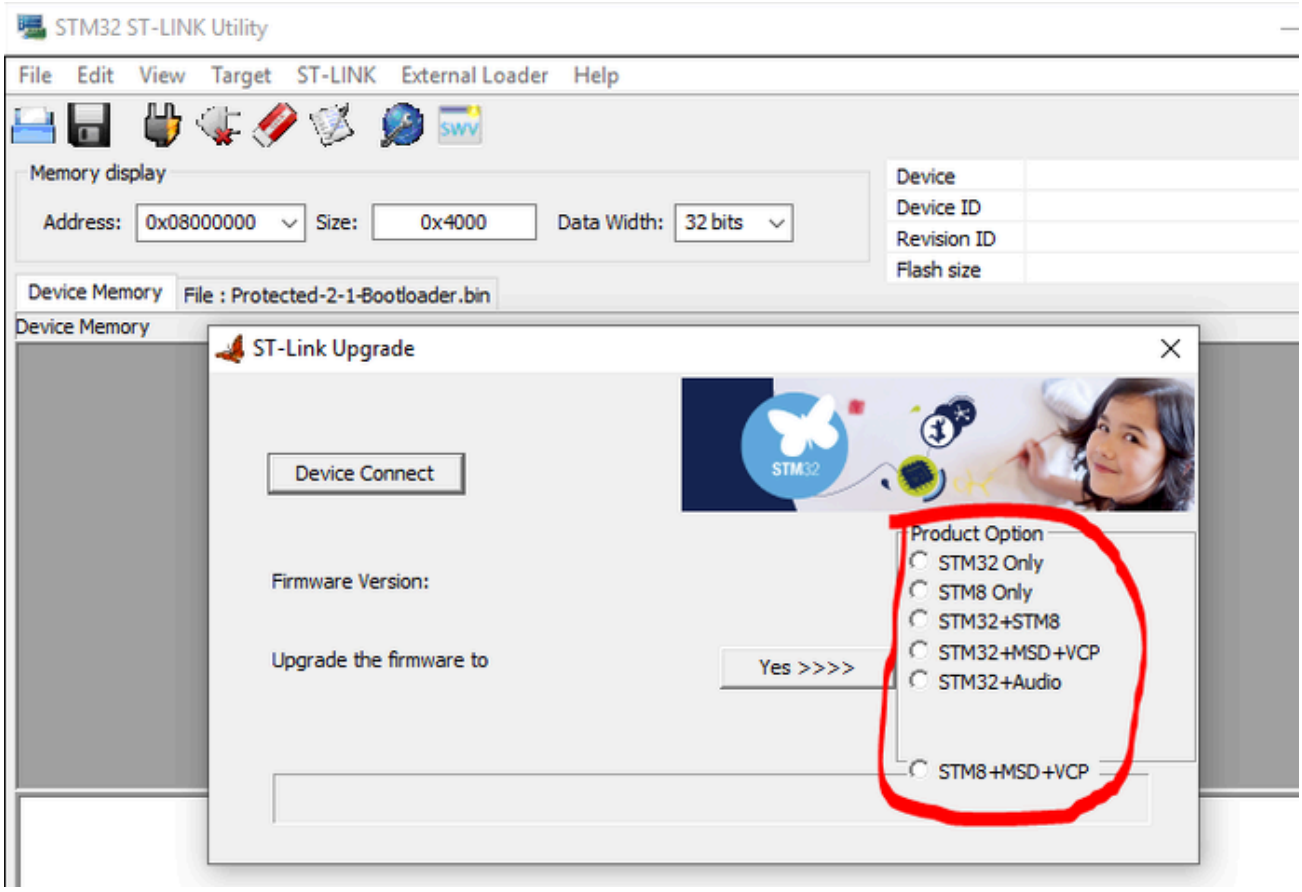
To remove protection in the **STM32 ST-LINK Utility** program, click *Target> Option Bytes* , switch *Read Out Protection* to *Disabled* and click *Apply* .

Update to ST-Link V2.1

After the firmware, we connect the stitched ST-Link (already almost V2.1) to the PC.

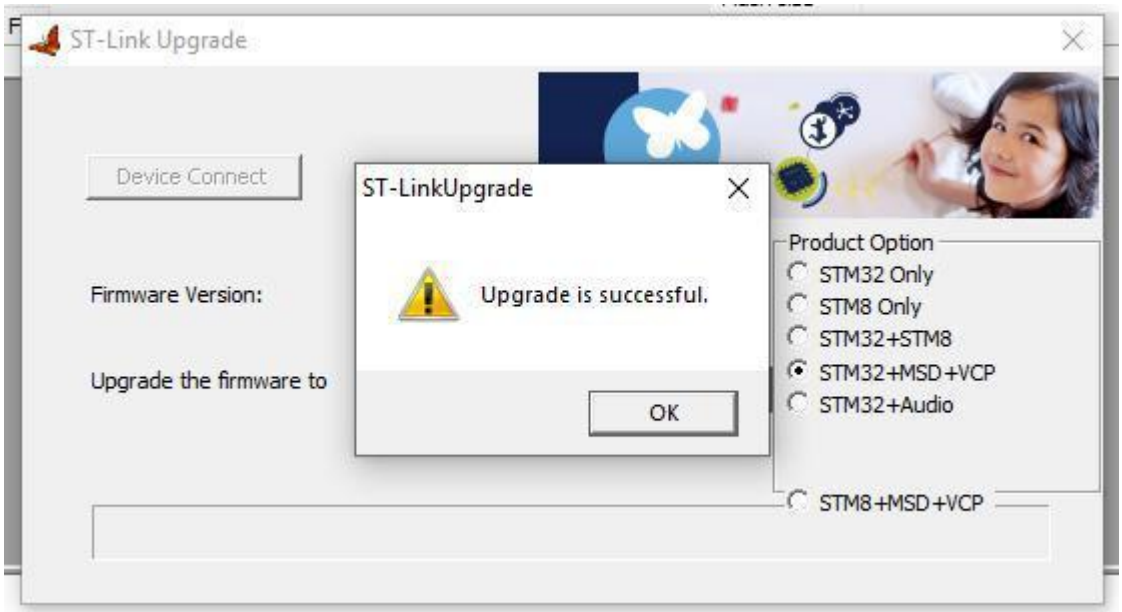
In the **STM32 ST-LINK Utility V4.3 program,** click *ST-LINK> Firmware update* .

Click *Device Connect* - we get a list of possible modifications:



Select the modification you need, in my case **STM32 + MSD + VCP** , click ***Yes >>>>*** .

We are waiting for the update to complete ...



Profit!

The final part

Since SWIM and RST do not work after such a modification, I cut them off.

I also cut off the duplicate 5V and 3.3V.

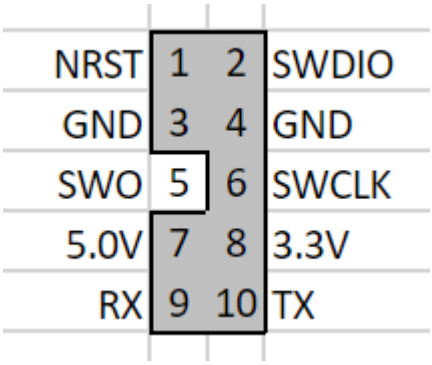
It turns out 4 free pins.

I solder them to the chip wiring:

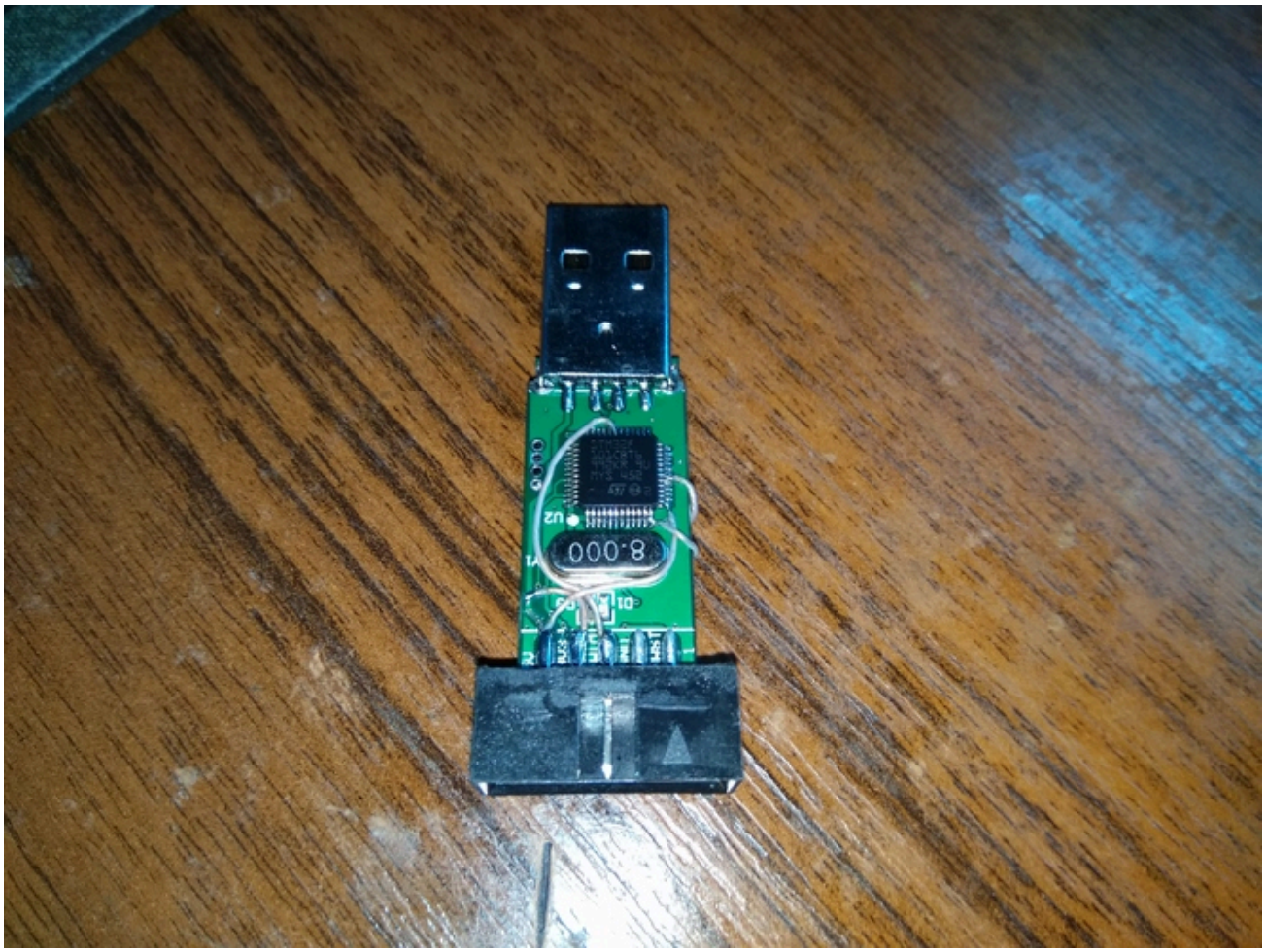
PA10 -> SWO
PB0 -> NRST
PA3 -> RX
PA2 -> TX I

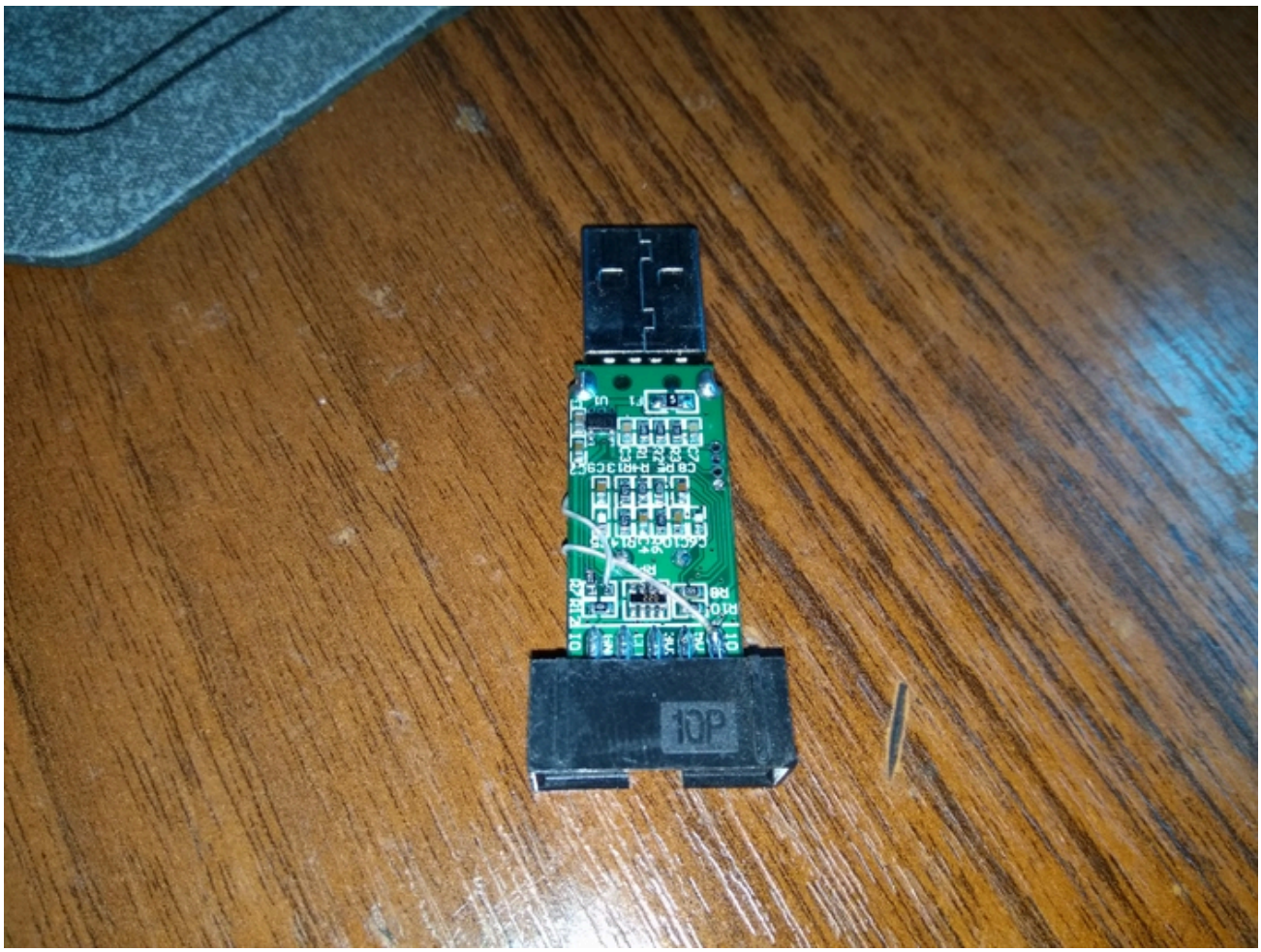
output everything to the main connector, to the remaining free pins.

The result was such a pinout:



My device after modification

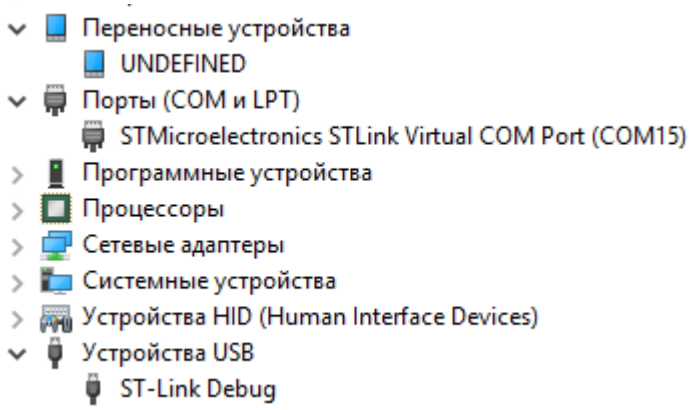

8.000

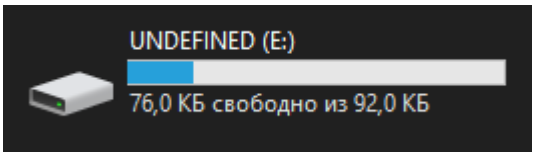

10P

I scribbled the markings on the case with a scalpel:

Do not forget to wash the board after soldering!

As a result, in a PC, the device is defined as follows:

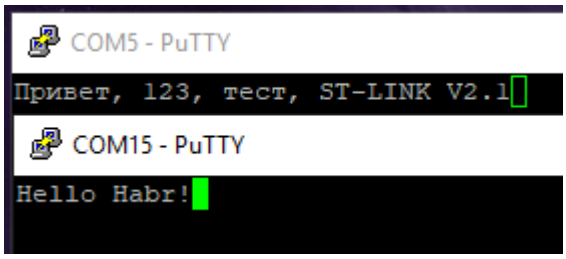




I have no idea what the volume of a virtual flash drive is (in this case, F103C8 was connected to ST-Link V2.1).

If you upload a firmware file to it, the programmer will flash the chip without programs.

Checking VCP:



Thank you for your attention!
When copying, please leave a link to the source.

With questions, please contact in the comments, as I can - I will help.

Tags:

stm32 st-link microcontrollers diy life hack

Also popular now:

Definition of dominant tones in the image

Writing a Simple Sublime Text 2 Plugin

Embed a second hard drive in Mac Mini 2011

And it's good for animals to make money

Testing ICQ Web Clients

How do pythonists read Haskell

Change time zones in Mac OS X Snow Leopard

Review Pocketbook A10. Fresh

Customer Response Solutions: extension dialing

MobiUs: HTML5 browser for iOS

---

sudonull@yahoo.com