Fobos SDR firmware update manual

As we know, the functionality of software-defined radios is determined by software as well as firmware. For different tasks different algorithms, softwares and *different firmwares* are required. This applies to any software-defined radio system, including those built on the Fobos SDR platform.

This guide will describe what firmwares exist for Fobos SDR, what tasks they are intended for and how to flash a firmware to the device. What is firmware, how it interacts with the host application software and what is application programming interface (API) library will not be discussed here. These are general questions to be answered in open sources.

Here are the features of the Fobos SDR platform:

- individual firmware is compatible with only corresponding API library, therefore the firmware+library term is used hereinafter;
- API libraries for the Fobos SDR devices are cross-platform, public and open source;
- firmwares are distributed as compiled binaries, source codes are not available for public.

Primary firmware+library

This tandem intended to communicate the Fobos SDR platform with a *general purpose* host application software. The main tasks to be performed:

- connected devices enumeration and selection;
- collecting and displaying information about the device;
- configuration (setting the frequency, sample rate, gain);
- clock source management;
- operation modes switching;
- waveform digitizing and data transferring to the host computer;
- GPO interface management.

Everything related to this API library kit (source codes, descriptions, guides, compiled binaries and utilities) could be found in the GitHub repository at the following link:

https://github.com/rigexpert/libfobos

Alternateive firmware+library

This tandem is used to communicate the Fobos SDR to the *specialized* frequency scanning, spectrum analysis and signal detection host software. Fast reconfiguration and frequency retuning as well as scanning capabilities have been added to the firmware. That is why this kit is called "Agile".

The basic functionality stays exactly the same as in the primary firmware+library kit, so this alternative kit could be perfectly used for regular fixed frequency signal acquisitions and waveform digitizing.

Everything related to this "Agile" API library kit (source codes, descriptions, guides, compiled binaries and utilities) could be found in the GitHub repository at the following link:

https://github.com/rigexpert/libfobos-sdr-agile

There are two firmware+library packages so far, there will be more in the future. The repositories are constantly updated and improved, new functions are added to the firmware and libraries, please stay tuned.

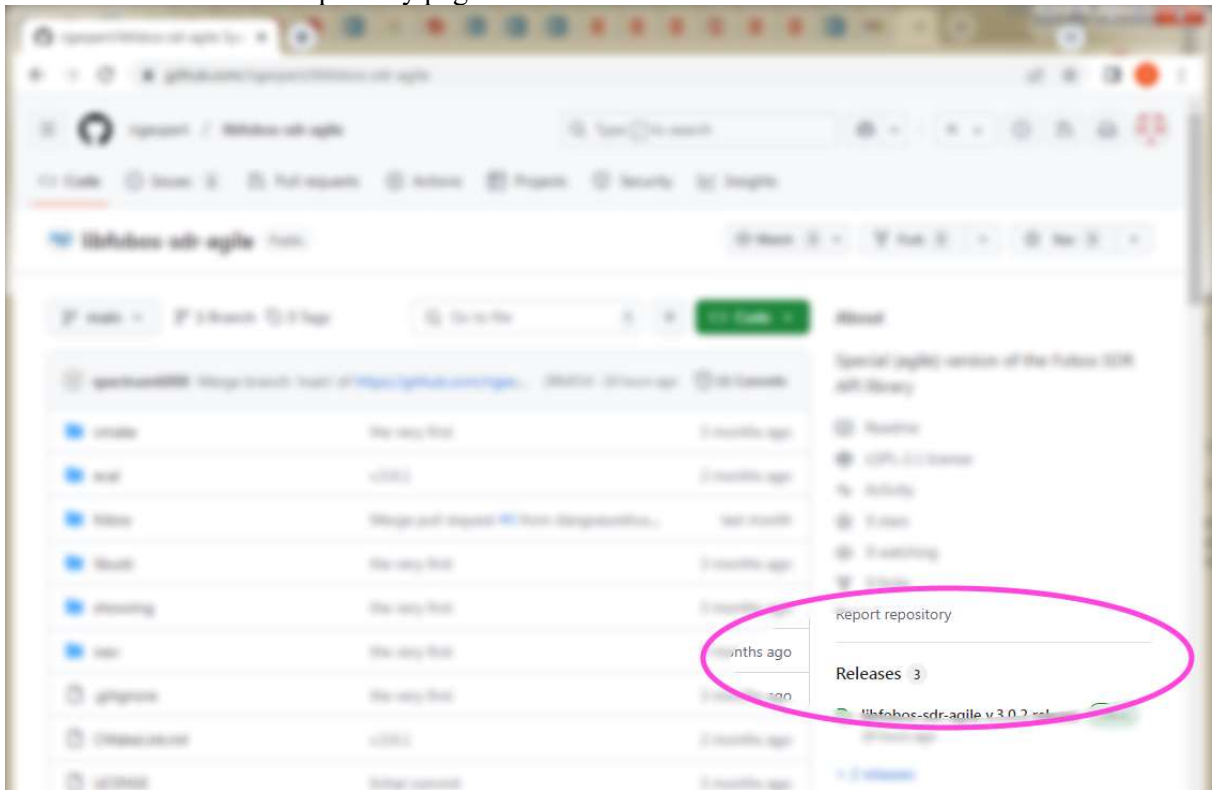There are two ways to update the firmware in Fobos SDR:
- bootloader mode;

- application (user) mode.

Firmware update in bootloader mode

To do this you need to have:
- drivers;
- the **fobos_load_firmware.exe** utility;
- a firmware binary file.

You can find all this in the mentioned earlier GitHub repositories. Follow the **Releases** section link located here on the repository page:

You also need to:
- make sure that the device you are flashing is in good condition;
- use a proper PC with the right USB connectors, working power and grounding;
- have a configured Windows 7, 10, 11 OS with the permitted devices and driver operations.

The firmware updating procedure in bootloader mode:

1. Download the latest release pack, unpack it into a convenient directory, make sure that it contains all necessary drivers, **fobos_load_firmware.exe** utility and firmware file(s).
2. Read the Readme files in the archives. There may be useful information relevant to a specific release.
3. Disassemble the metal case of the device to get access to the "USB Boot" and "Reset" buttons.

4. Connect Fobos SDR USB connector within a cable to the PC.
5. Press and hold the "USB Boot" button.
6. Press and release the "Reset" button.
7. Release the "USB Boot" button.

The "WestBridge" device or some other device will appear in the device manager:
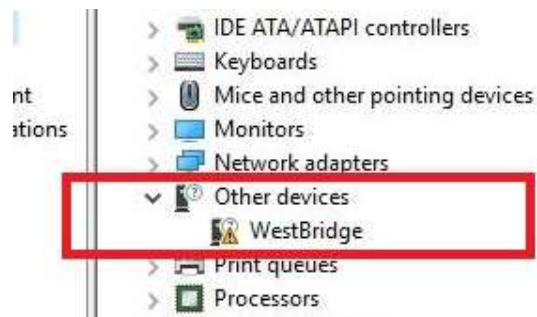
8. Install the drivers from the **drv** folder. Cypress FX3 USB BootLoader Device should appear:

9. Lounch the command prompt, navigate to the **fobos_load_firmware.exe** utility folder and run the command:

fobos_load_firmware.exe -f C:/full/path/to/fobos_usb_rx.img

where **C:/full/path/to/fobos_usb_rx.img** is example of the full path to the firmware file for. It will be different depending on what file is used, what its name is and where it is located.
If this path contains spaces, it should be enclosed in quotes:

fobos_load_firmware.exe -f "C:/full/path with spaces/fobos_usb_rx.img"
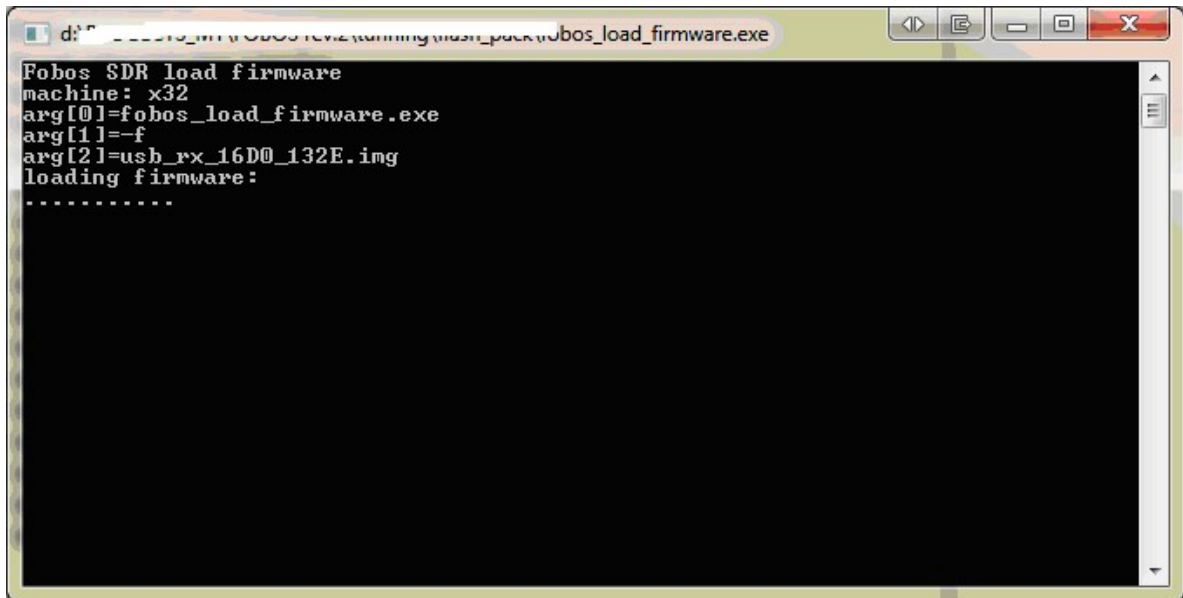
Please use pure ASCII quotes " ", don`t use these " " and these « ».

If the firmware file is located in the same directory as the utility, then you can do this:

fobos_load_firmware.exe -f fobos_usb_rx.img

where **fobos_usb_rx.img** is the firmware file name esample.

The utility informs about the firmware downloading progress using a row of dotteds. After the successful completion it should be Ok

10. Press and release the "Reset" button

The device will light up the LED bar and appear in the Device Manager as Fobos SDR. Than you can use it further with newly updated firmware.

Advantages of the bootloader method:
- the procedure and the **fobos_load_firmware.exe** utility are the same for all hardware versions;
- regardless of the version and status of the current firmware on the device;
- can be used to write firmware to a "clean", newly manufactured device (that is how it works at the production);
- can be used in cases of non-destructive failures and errors (this is how you can restore the device after a firmware failures).

Disadvantages of the bootloader method:
- you need to disassemble the case to get access to the "USB Boot" and "Reset" buttons;
- manipulations with the "USB Boot" and "Reset" buttons are required;
- the **fobos_load_firmware.exe** utility works only under Windows OS;
- you can only write the firmware to the device, you cannot read it;
- it is not possible to distinguish and select a device if several are connected and all of them are in bootloader mode.

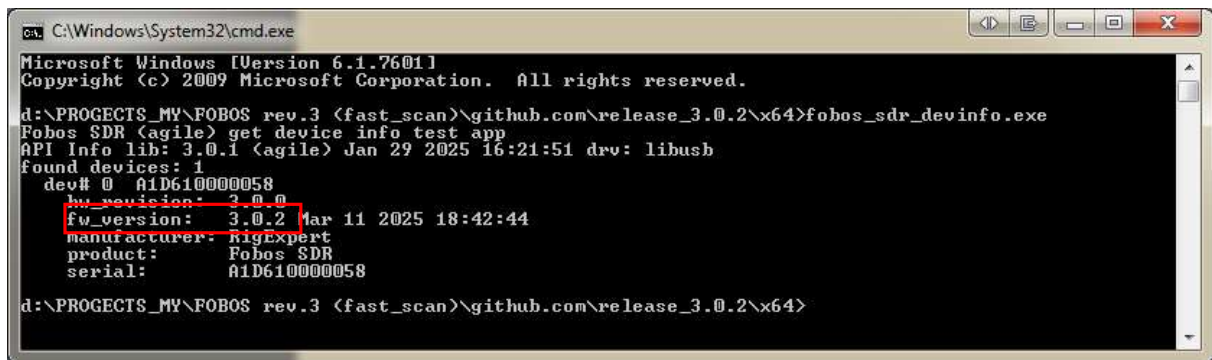# Firmware update in the application (user) mode

To do this, you need to have:
- firmware that supports this mode already loaded to the device;
- a utility from the compatible library kit;
- a firmware binary file.

The application (user) update mode is available for:
- primary firmware v.2.1.0 and higher;
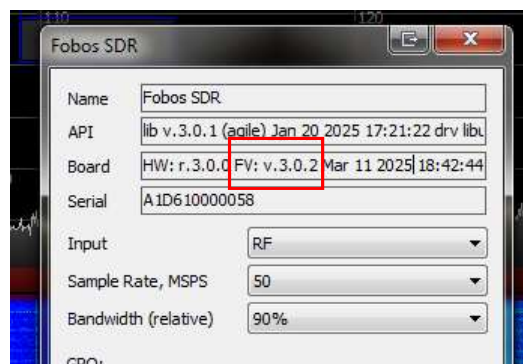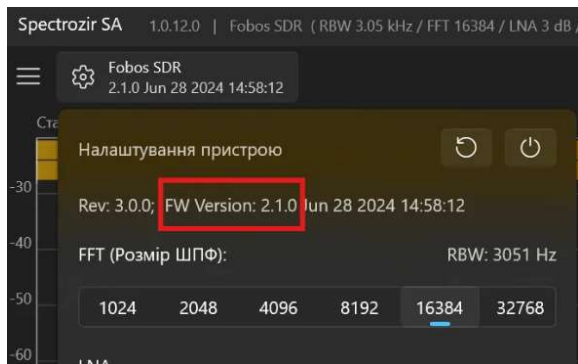- alternative firmware v.3.0.1 and higher.

You can check the firmware version using the **fobos_devinfo** and **fobos_sdr_devinfo** utilities (primary and alternative libraries, respectively) or in the application software dialog boxes.

If the firmware does not meet the requirements it should be updated using the bootloader method.

Utilities for working with a firmware:
- **fobos_fwloader** from the primary library kit;
- **fobos_sdr_fwloader** from the alternative library kit.

You can find the utilities and firmware files in the mentioned above repositories. Utilities and libraries are open source and cross-platform, so applications are very diverse:
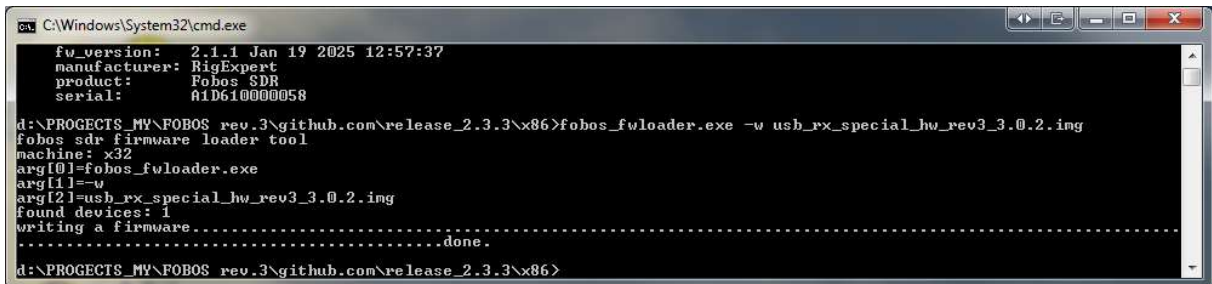- operate under various compatible operating environments;
- use and modify source codes within the license;
- use compiled binaries from release packs or build them yourself;
- use the **fobos_fwloader** and **fobos_sdr_fwloader** utilities to work with firmware from the command prompt or use the **fobos_**_write_firmware()** library functions in your own software.

The firmware updating procedure in application (user) mode,
the simplest case, primary firmware v.2.1.1 (library v.2.3.2):

1. Connect the device, install the drivers and library, make sure that the device and firmware are working normally.

2. Obtain the **fobos_fwloader** utility (download and unpack from the release package or build from sources using your favorite compiler under your favorite operating system).

3. Obtain the firmware file (unpack from the same release pack or from a different one) for example **usb_rx_special_hw_rev3_3.0.2.img** from the alternative kit for frequency scanning.

4. Copy all of them into the same directory, launch the command prompt and run the following command:
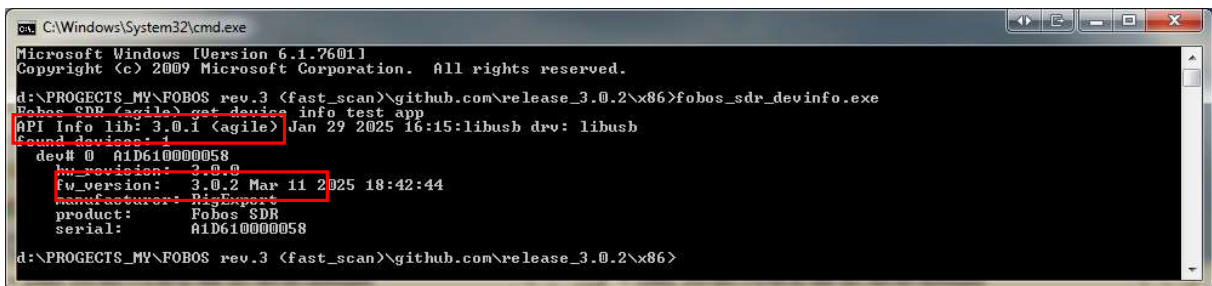
fobos_fwloader -w usb_rx_special_hw_rev3_3.0.2.img

5. Wait until the row of dots runs out and "done" message will appear:

6. Run the **fobos_sdr_devinfo** utility and ensure that the firmware was written properly:

After writing the alternative firmware, the device will work only with the alternative library and with software that supports the alternative firmware. To write newer alternative firmware versions or to return to the primary firmware you should use the **fobos_sdr_fwloader** utility from the alternative kit:

fobos_sdr_fwloader -w new_firmware_file.img

As you can see only command prompt and only one command are needed. No need to disassemble anything or press any buttons.

Advantages of the application (user) method of working with firmware:
- no need to disassemble the case and press the buttons;
- no physical access to the device is required at all;
- you can flash via a terminal or remote desktop;
- you can work with firmware in any compatible operating system;
- the process of writing a firmware file takes a little less time;
- you can both write and read firmware;

6

- instead of the mentioned utilities, you can use the library functions **fobos_**_write_firmware()** and **fobos_**_read_firmware()** to integrate the functionality of flashing the device into your own software.

Disadvantages of the application (user) method:
- you need to use utilities and library functions from the same kit as the firmware in the device;
- the device must have firmware that can flash itself, otherwise you need to disassemble the case to apply the bootloader mode;
- you need to use the bootloader method anyway to recover the firmware in case of incorrect actions, errors or failures.