Octoplus Pro Software

User Manual

Contents

Introduction

Octoplus Pro Software (same as Medusa Pro Software) is an application that works with the **Octoplus Pro** programmer on the **Windows** operating system.

Octoplus Pro Software provides a convenient interface for restoring bricked devices.

Octoplus Pro Software allows you to restore devices using **USB and eMMC** interfaces by directly connecting to the CPU or memory, as well as using original factory firmwares from the manufacturer for devices.

1. Software Description

The main program window looks like this.

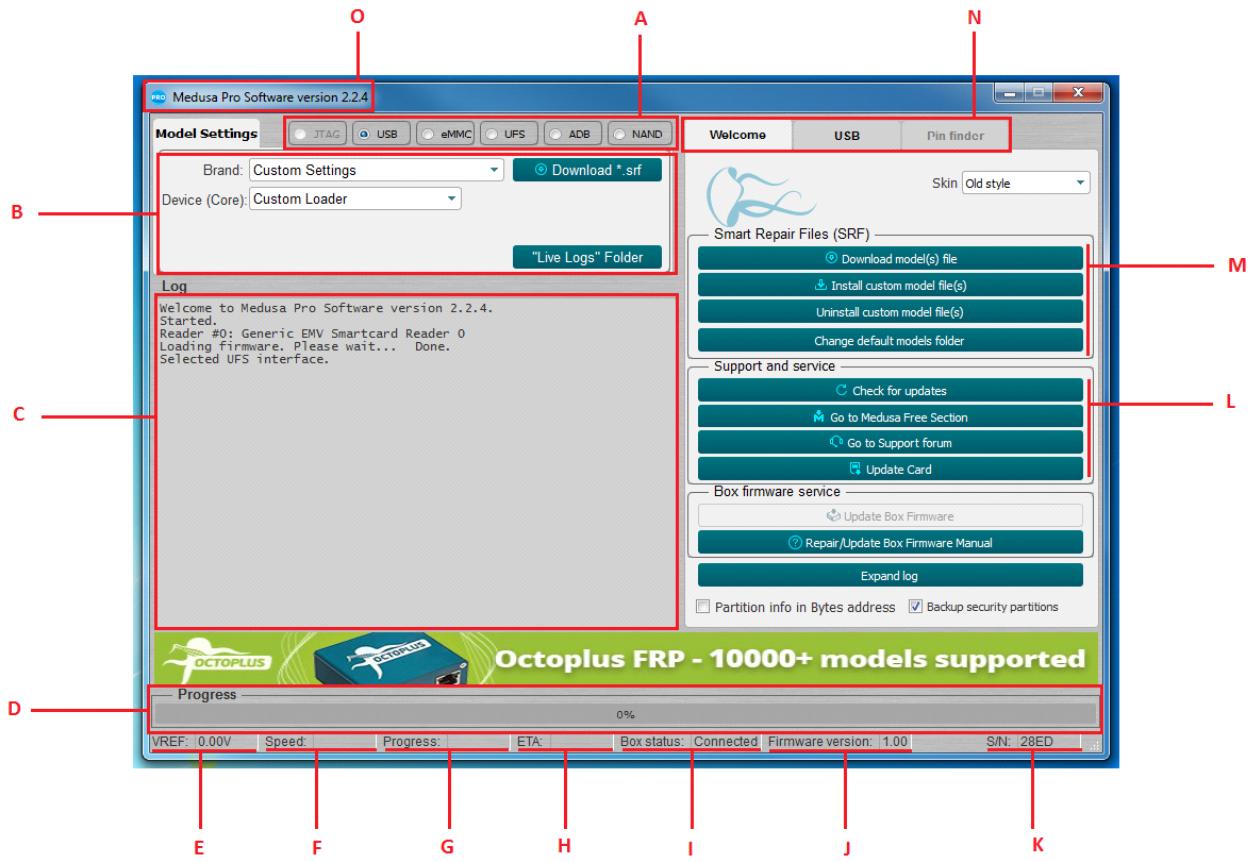


Fig. 1 Main window, **"Welcome"** Tab

A. Selection of the interface through which work with the connected device is performed;

B. Adjusting the interface according to the connected device;

C. Displaying device information and progress;

D. Progress of the running operation as a percentage;

E. Reference voltage;

F. Speed in kilobytes per second (KB/s), megabytes per second (MB/s), and gigabytes per second (GB/s);

G. Time elapsed since the operation started;

H. Approximate time remaining until completion of operation;

I. Box status: "Connected" and "Disconnected";

J. Current version of box firmware;

K. Box serial number;

L. Support and service;

M. SRF Manager;

N. Group of tabs to work with the box. The first **"Welcome"** tab is shown in Fig.1 and is designated for SRF control, software version and box firmware control. The second tab depends on the selected interface (Fig. 1A). The third tab Pin Finder is not in use.

O. Current software version

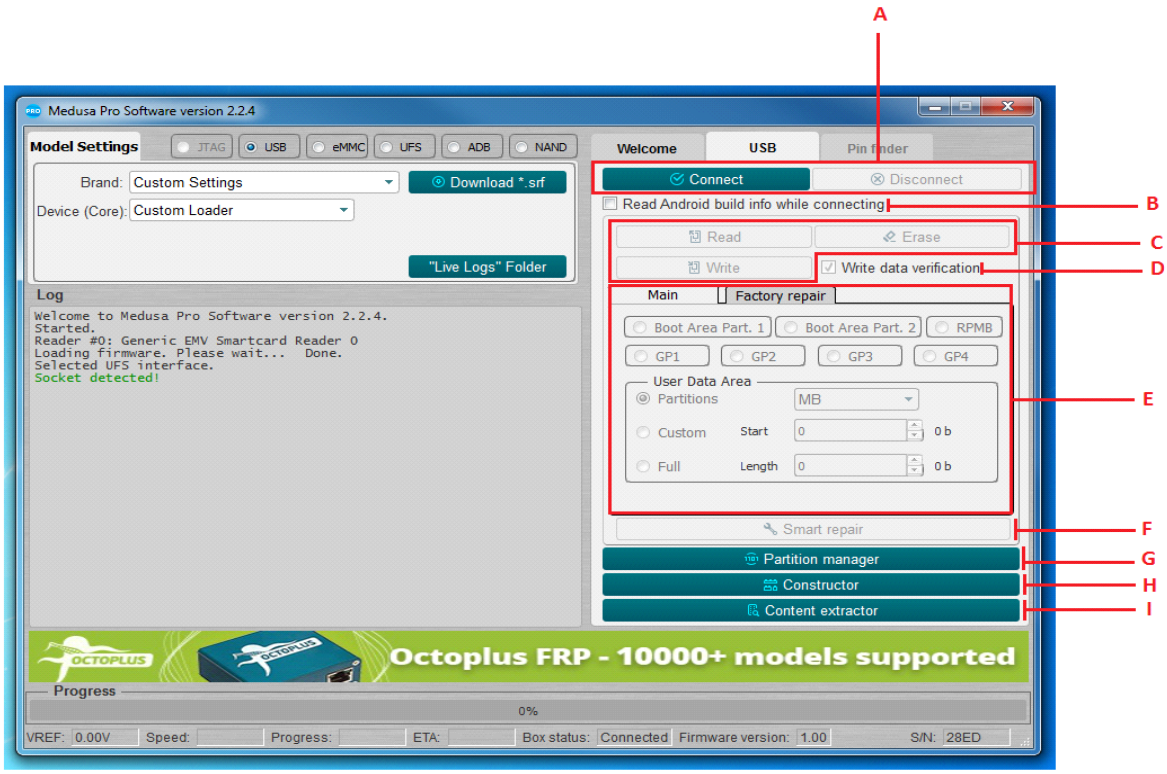


Fig. 2 Main Window, Interface Tab

A. Initialization of the connected device;

B. Finding and reading "Android Info" when initializing the device;

C. Standard read, write and erase features;

D. Checking recorded data;

E. Setting the parameters for reading, writing and erasing (address, size, individual partitions, the entire flash drive); it is possible to enter values in bytes, blocks, kilobytes and megabytes (bytes, blocks are entered in hexadecimal; kilobytes, megabytes in decimal form);

F. Recovering a flash drive using an SRF file;

G. Work with partitions;

H. Creating SRF files;

I. Parsing the flash drive content.

1.1. General Device Recovery Algorithm

In general, the device recovery process consists of several stages:

- It is necessary to physically connect the device to one of the interfaces Fig.1 (A);
- Select the desired interface;
- Configure the interface in the field shown in Fig.1 (B);
- Initialize the device by pressing the "Connect" button (Fig. 2 (A));
- The initialization results are displayed in the log (Fig. 1 (C)). In case of successful initialization, the log may contain certain device parameters, for example: device manufacturer, device model, serial number, media size, etc. If the device could not be initialized, information about the impossibility to initialize the device is displayed in the log;
- After successful initialization, you must select the method by which you plan to restore the device. For each individual device, the method may differ (factory firmware, previously saved device dumps, using original SRF files created by the Octoplus team for faster and easier device recovery).

2. Work with eMMC Flash Memory

Octoplus Pro works in accordance with the EMMS 5.1 specification (JESD84-B51) and are fully compatible with older versions of the specification.

Octoplus Pro Software allows you to work with flash media by choosing a data bus width of 1, 4 ~~or 8~~ bits.

Box	Data Bus Width, Bits
Octoplus Pro	1, 4

Table 1. Matching the eMMC bus width to the connected box

2.1 eMMC Initialization

Before starting eMMC initialization, you need to set basic connection parameters such as voltage (Voltage, default 1.8V), bus mode (Bus Mode, default 1 bit) and transmission frequency (Bus speed, default Auto). For most cases, the voltage and transmission frequency can be left untouched.

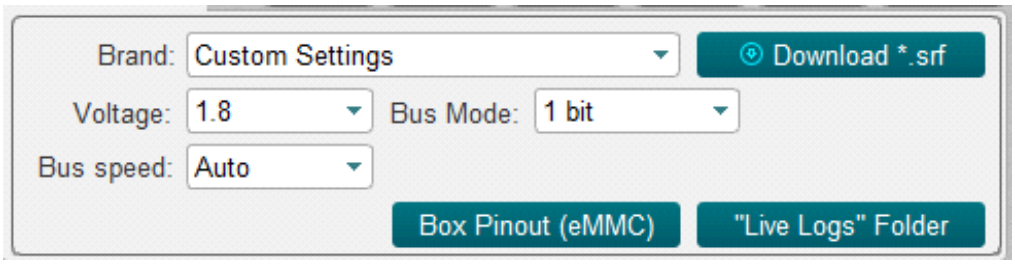


Fig. 3 Configuring Basic Parameters for Initializing of eMMC Flash Drive

By pressing the “Connect” button (Fig. 4), in case of successful initialization, information about the carrier is displayed in the log (example in Fig. 5).

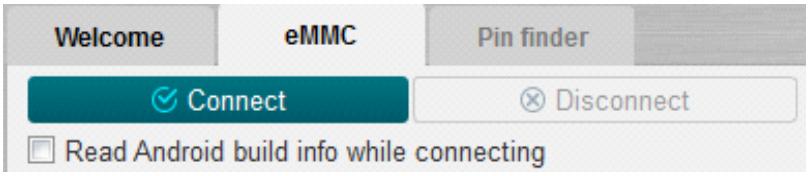


Fig. 4

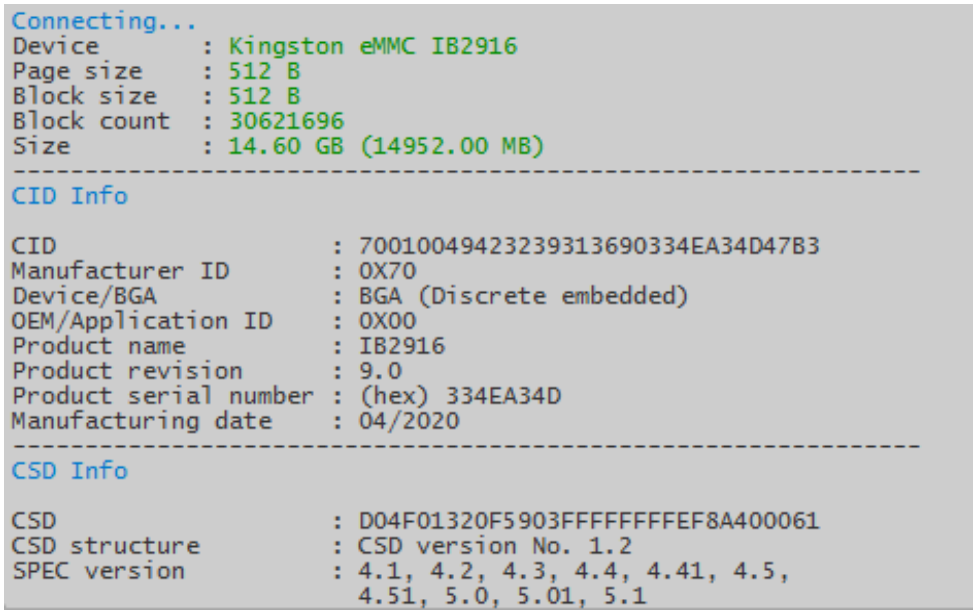


Fig. 5. Window with the Log of the Connected Flash Drive

From this moment on, the flash drive is considered initialized and you can work with it.

2.2. eMMC Standard Features (Main)

Standard read, write and erase features are available in the **"Main"** tab.

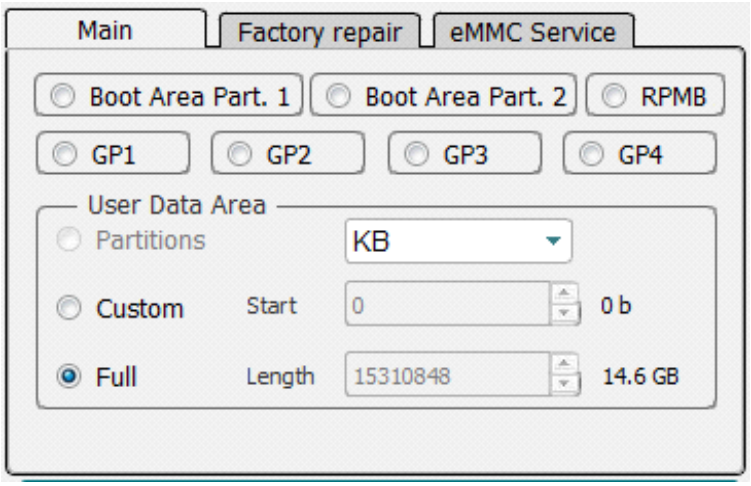


Fig.6

It is possible to select the memory area with which you plan to work in the upper part of the tab, if the selected area is not a zero size:

- Boot Area Part. 1;
- Boot Area Part. 2;
- RPMB;
- GP1 (General purpose 1);
- GP2 (General purpose 2);
- GP3 (General purpose 3);
- GP4 (General purpose 4);

2.2.1. eMMC Work with Partitions (Partitions)

If certain partitions were found on the flash drive during initialization, then to simplify working with them, you can select the necessary ones by first selecting the **“Partitions”** mode (Fig. 7) and pressing the **“Read”** button (Fig. 8), a window with partitions will open, Fig.9.

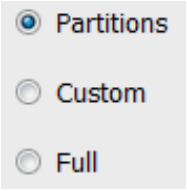


Fig. 7

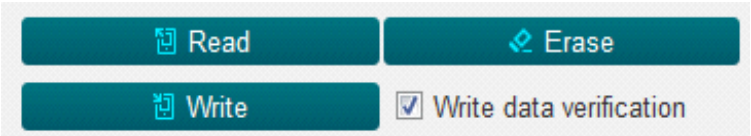
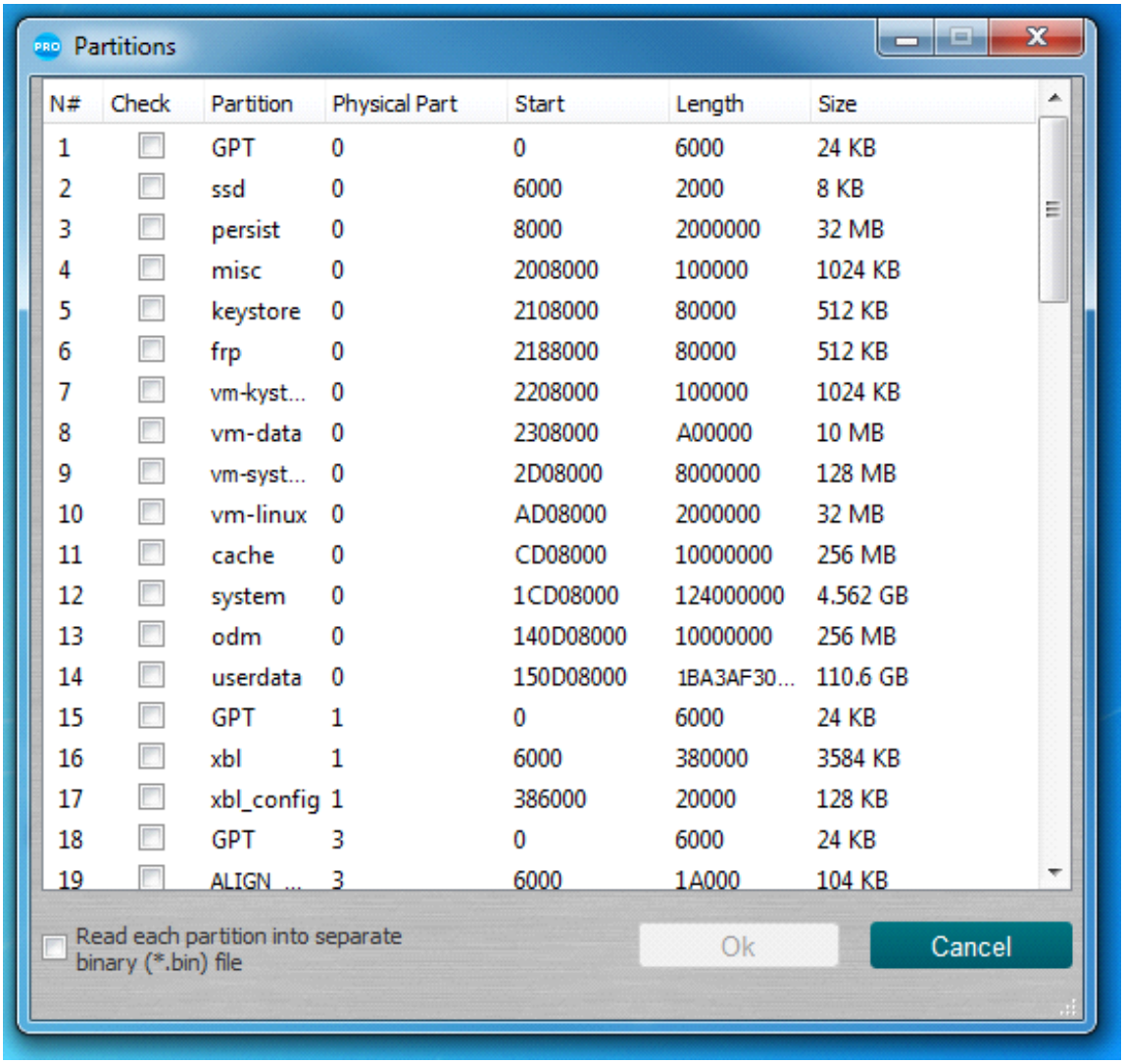


Fig. 8



N#	Check	Partition	Physical Part	Start	Length	Size
1		GPT	0	0	6000	24 KB
2		ssd	0	6000	2000	8 KB
3		persist	0	8000	2000000	32 MB
4		misc	0	2008000	100000	1024 KB
5		keystore	0	2108000	80000	512 KB
6		frp	0	2188000	80000	512 KB
7		vm-kyst...	0	2208000	100000	1024 KB
8		vm-data	0	2308000	A00000	10 MB
9		vm-syst...	0	2D08000	8000000	128 MB
10		vm-linux	0	AD08000	2000000	32 MB
11		cache	0	CD08000	10000000	256 MB
12		system	0	1CD08000	124000000	4.562 GB
13		odm	0	140D08000	10000000	256 MB
14		userdata	0	150D08000	1BA3AF30...	110.6 GB
15		GPT	1	0	6000	24 KB
16		xbl	1	6000	380000	3584 KB
17		xbl_config	1	386000	20000	128 KB
18		GPT	3	0	6000	24 KB
19		ALIGN ...	3	6000	1A000	104 KB

Fig. 9 Partitions Window

In this window, you need to select the partitions that you want to read. The specified partitions will be read into a file with the ***.mpt** extension. It is also possible to read the partitions into separate ***.bin** files; for this you need to check the **"Read each partition into separate binary (*.bin) file"** option.

To write partitions, you must select a file with the *.mpt extension, which was previously read, and click **"Write"** (Fig. 8).

2.2.2. eMMC Work with Arbitrary Addresses and Blocks (Custom)

When you need to write/read/delete data at a certain address and in a certain amount, switch to the Custom mode (Fig. 10), select from the list in which units the data will be entered:

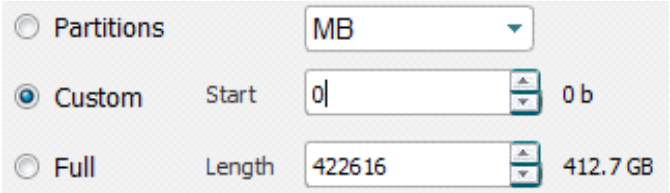


Fig. 10

- **Hex value** (in bytes, HEX);
- **Hex blocks** (in blocks, HEX);
- **KB** (in kilobytes, DEC);
- **MB** (in megabytes, DEC).

2.2.3. eMMC Work with the Full Capacity of the Flash Drive (Full)

If you need to write / read / delete information from the entire flash drive, you must switch to Full mode (Fig. 11)

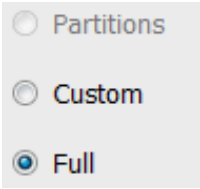


Fig. 11

Then perform the necessary operation (Fig. 8).

2.3. eMMC Work with Manufacturer's Firmware (Factory repair)

In this section (Fig. 12) you can restore the internal memory of the device with factory firmware from different manufacturers.

The complete recovery procedure comes down to selecting the required device by clicking the appropriate button on the tab and in the window that opens, select the firmware file with the required extension for this device and burn the selected firmware.

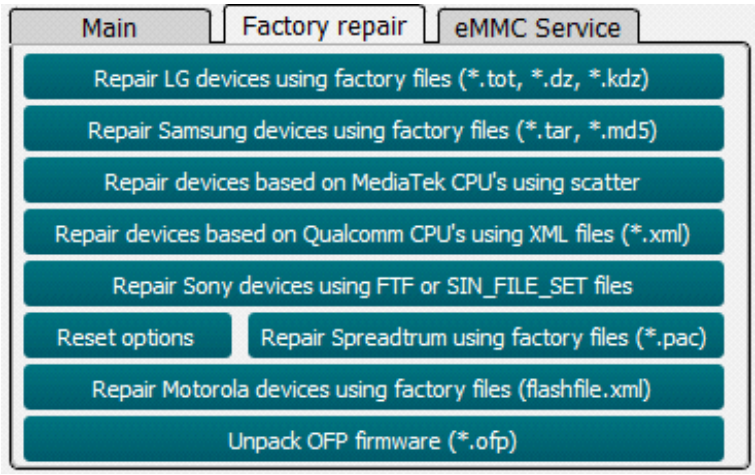


Fig. 12 Factory Repair Tab

2.4. eMMC Work with Service Features (eMMC Service)

This mode is used for working with internal eMMC registries (CID, CSD, EXT_CSD), partitioning a flash drive, switching flash drive operating modes, reading additional information, updating firmware.

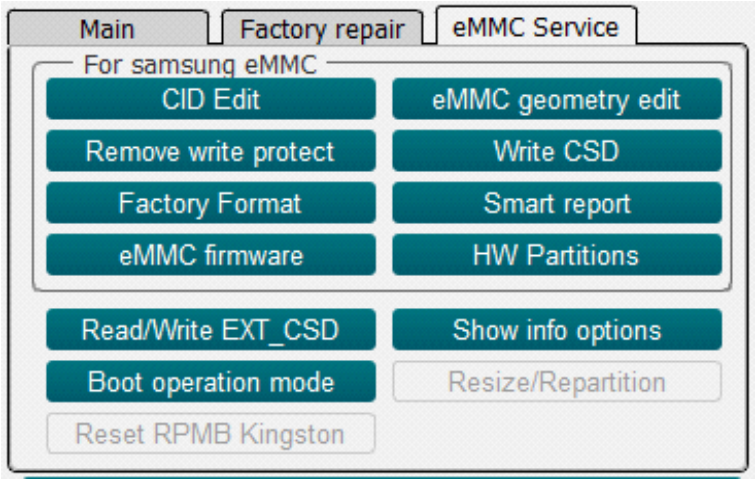


Fig. 13 eMMC Service Tab

“CID Edit” - Used for editing the CID register;

“Remove write protect” - Removes writing protection;

“Factory Format” - Completely overwrites a flash drive;

“eMMC firmware” - Firmware update for eMMC controller;

!!! Octoplus Pro Software is not responsible for a permanently damaged device during the update of the controller firmware. All operations to update the controller firmware are performed at the user's own risk.

“eMMC geometry edit” - Sizing Boot1, Boot2, RPMB;

“Write CSD” - Used for editing the CSD register;

“Smart report” - Reads information about the flash drive resource;

“HW Partitions” - Used for adjusting the size of GP1, GP2, GP3, GP4 and User area;

“Read/Write EXT_CSD” - Work with EXT_CSD;

“Boot operation mode” - Boot setup**;**

3. Work with USB

Octoplus Pro Software supports work via USB for devices with **Qualcomm** or **MediaTek (MTK)** CPUs.

To initialize a device, it must be in the EDL mode (Emergency Download Mode). It is possible to switch to EDL mode in different ways, the most effective way is to short certain points (**test points**) on the device board. You must partially disassemble the device to perform this procedure.

In some other cases, it is possible to put the device in EDL mode with a special command from the Android OS or other modes, such as **Recovery, Fastboot**, etc. After the device switches to EDL mode, it becomes available in the system as a COM port, through which the interaction takes place in EDL mode. Displaying devices in EDL mode connected via USB, Qualcomm (Fig.20) and MTK (Fig.21).

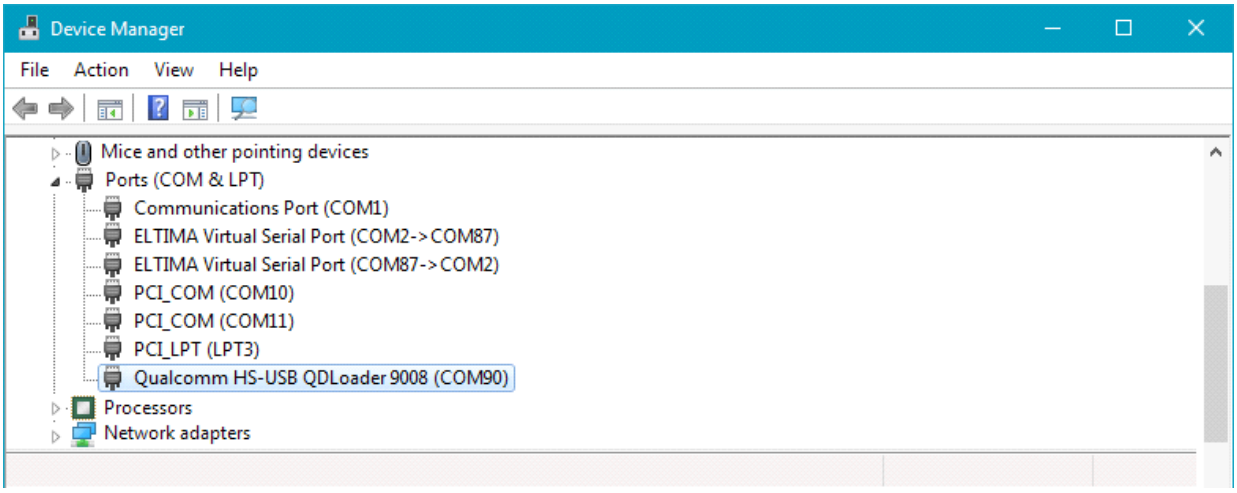


Fig. 20. Device with Qualcomm CPU in EDL mode, connected via USB

Fig. 21. Device with MTK CPU in EDL mode, connected via USB

3.1. Initialization of Qualcomm CPUs via USB

After making sure that the device is in EDL mode and defined in the system as **"Qualcomm HS-USB QDLoader 9008"** (Fig. 20), select from the list **"Device (Core)"** (Fig. 22) the CPU installed in the device and press **"Connect".**

If the name of the CPU in the device is unknown, you can use the feature of automatic detection of the CPU by selecting "**Auto Detect"** from the list **"Device (Core)"** and click **"Connect".**

If the initialization is successful, the log will display information about the device and from now on you can work with it using the standard read / write / erase functions from the **"Main"** tab (Fig. 23), and work with factory firmware in the **"Factory repair"** tab (Fig. 24).

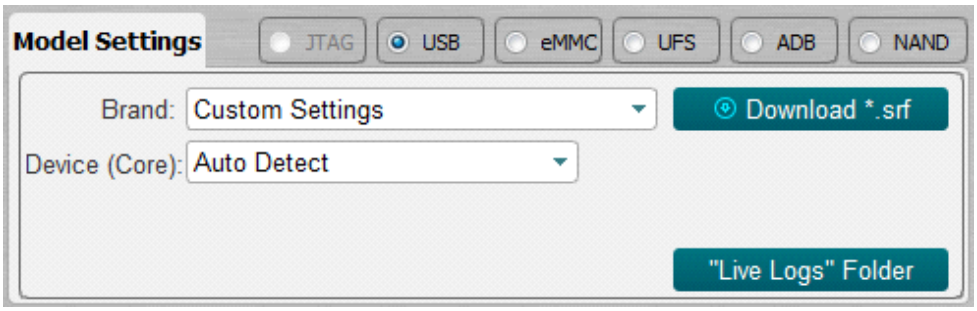


Fig.22

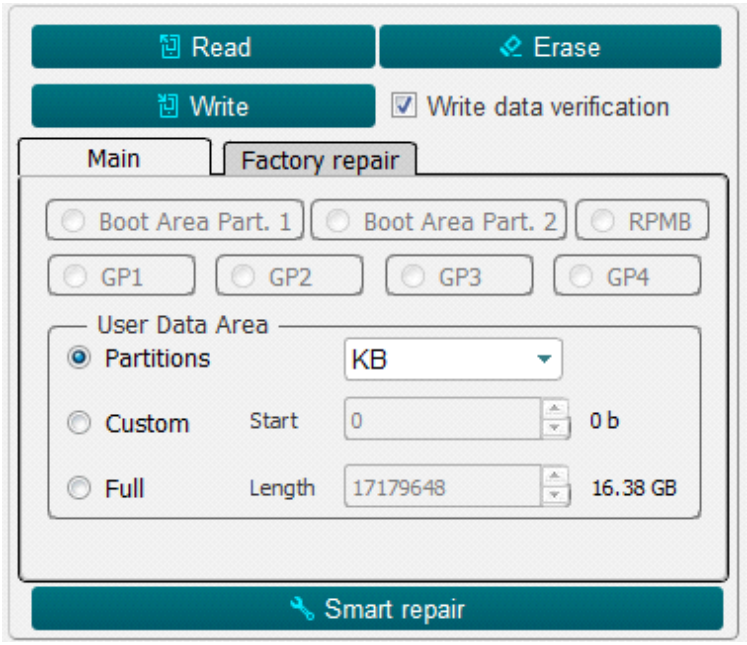


Fig. 23

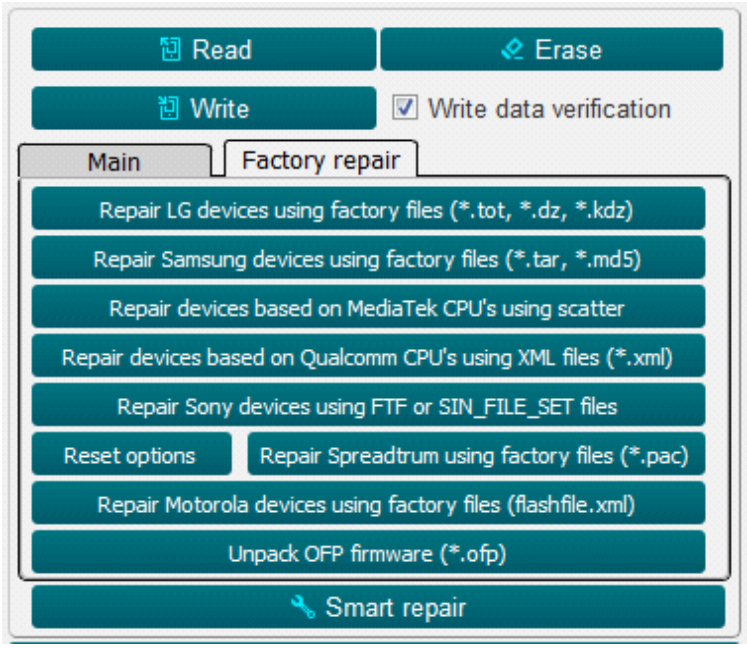


Fig. 24

3.2. Initialization of MediaTek (MTK) CPUs via USB

After making sure that the device is in EDL mode and defined in the system as **"MediaTek USB Port"** (Fig. 21), select from the list **"Device (Core)"** (Fig. 22) one of the two options - **"MTK Custom"** or **"MTK General"**.

The difference between these two options is that in **"MTK Custom"** to initialize the device you need to select 3 files: **"Download Agent (DA)", "Preloader" and "Authentication File" (AUTH file)** (Fig. 25).

For **"MTK General"** you only need to select one file: **"Preloader"** (Fig. 26) and click **"Connect".** If the initialization is successful, the log will display information about the device and from now on you can work with it using the standard read / write / erase features from the **"Main"** tab (Fig. 23), and work with factory firmware in the **"Factory repair"** tab (Fig. 24).

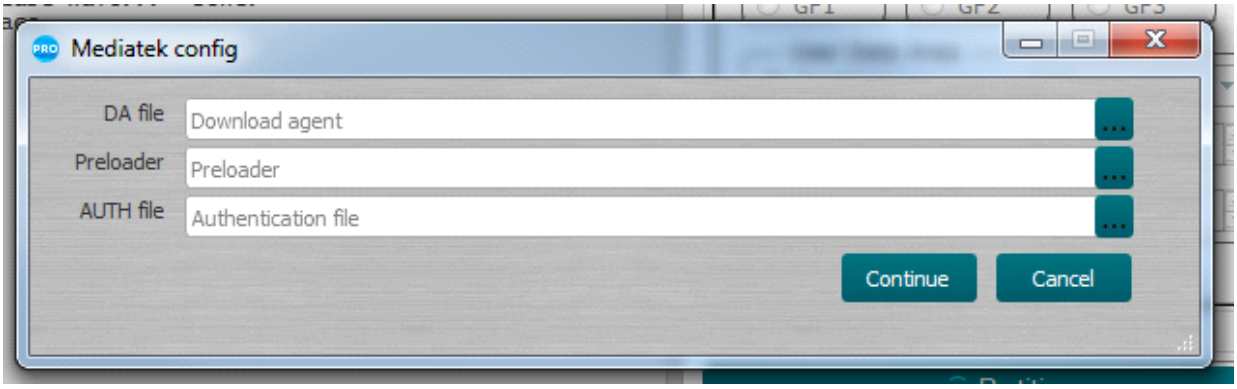


Fig. 25

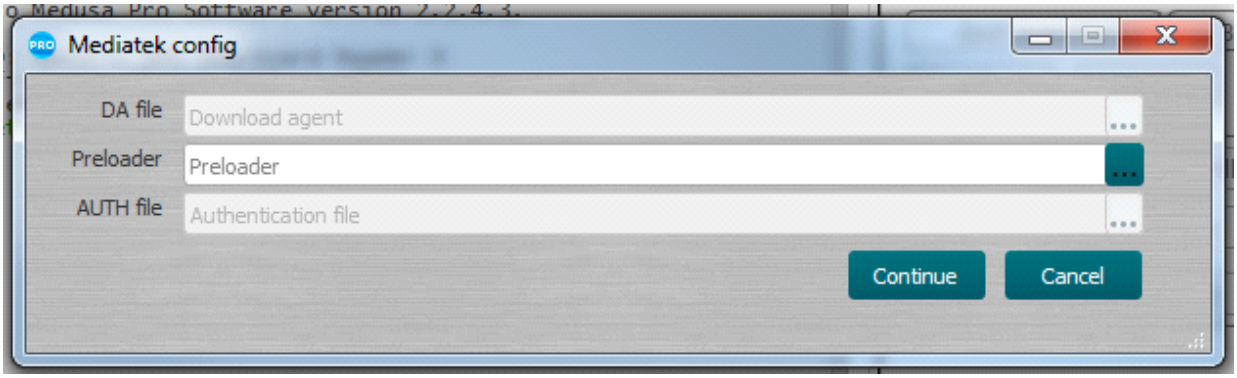


Fig. 26

4. Work with ADB (Android Debug Bridge)

The device must have ADB enabled. To enable ADB on your Android device, follow these steps:

- Go to **Settings** → **About phone** → **Software information**;
- Press **Build number** six times (until you see a message **You are now a developer**);
- Go to **Settings** menu and find new option **Developer options**;
- Make **USB Debugging** line switch active;
- Then you need to connect the device to PC and click **Connect**. If the initialization is successful, information about the device will be displayed in the log. Read-only is available in ADB mode.