# Medusa Pro Software

# User Manual

# Contents

# Introduction

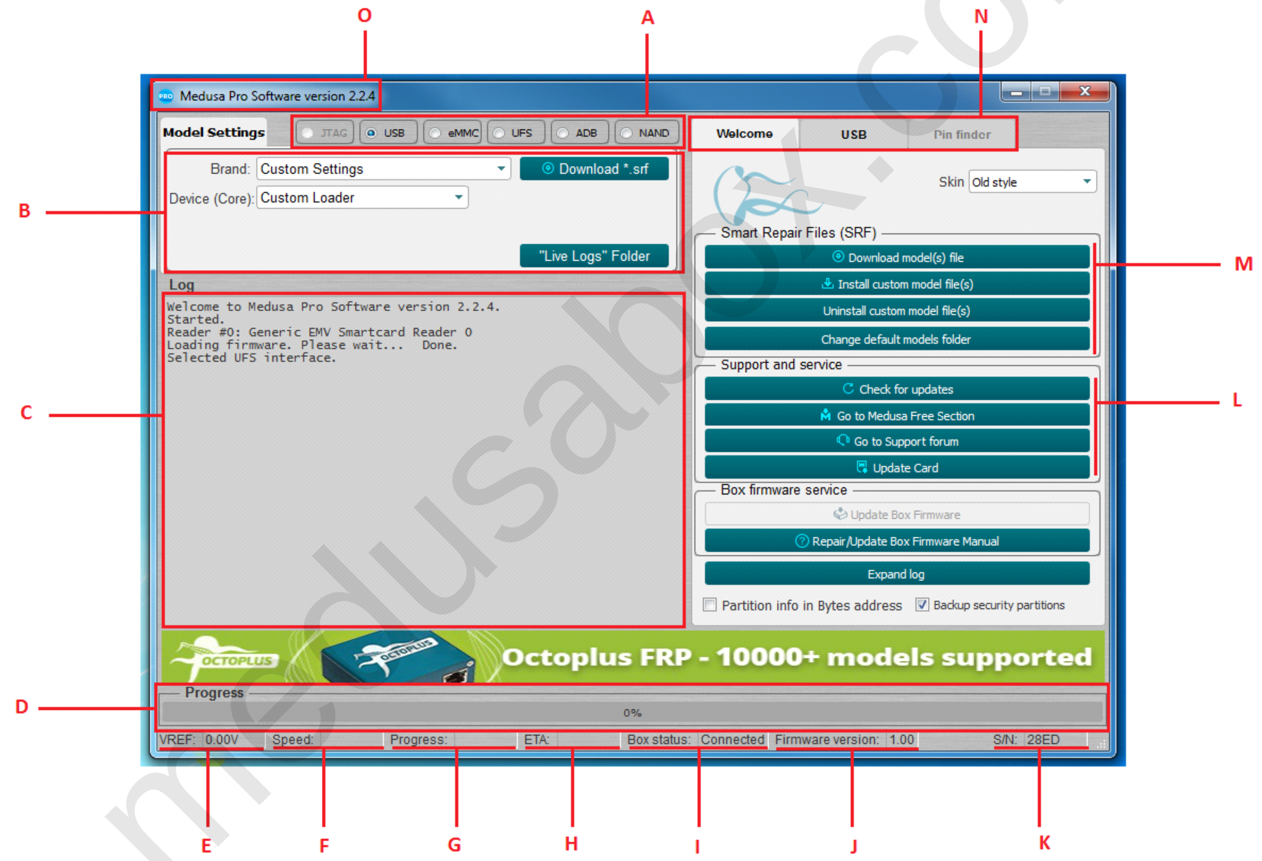
**Medusa Pro Software** - is an application that works with the **Medusa Pro** and **Medusa Pro II** programmers on the **Windows** operating system.

**Medusa Pro Software** - provides a convenient interface for restoring bricked devices.

**Medusa Pro Software** allows you to restore devices using **USB, eMMC, UFS, NAND** 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.



○ USB

Brand: Custom Settings

Device (Core): Custom Loader

Skin: Old style

```
Welcome to Medusa Pro Software version 2.2.4.
Started.
Reader #0: Generic EMV Smartcard Reader 0
Loading firmware. Please wait...   Done.
Selected UFS interface.
```

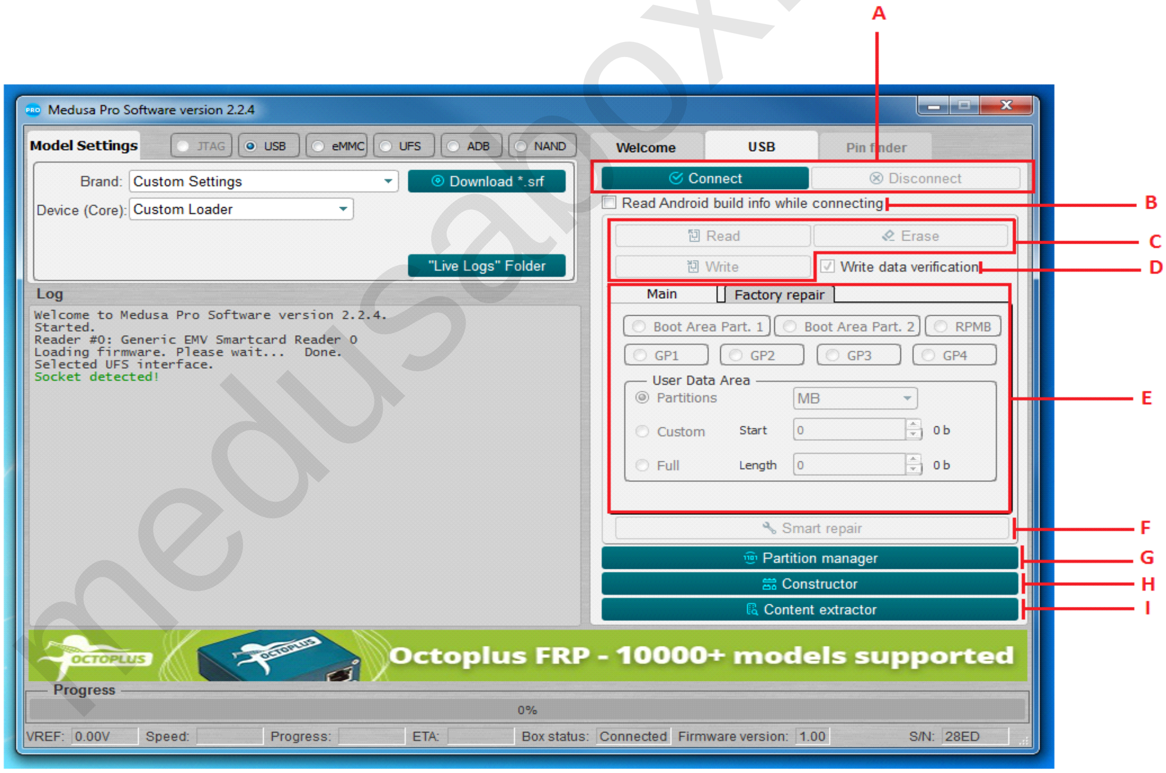
□ Partition info in Bytes address ☑ Backup security partitions

VREF: 0.00V | Speed: | Progress: | ETA: | Box status: Connected | Firmware version: 1.00 | S/N: 28ED

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

- Selection of the interface through which work with the connected device is performed;
- Adjusting the interface according to the connected device;
- Displaying device information and progress;
- Progress of the running operation as a percentage;
- Reference voltage;

- Speed in kilobytes per second (KB/s), megabytes per second (MB/s), and gigabytes per second (GB/s);
- Time elapsed since the operation started;
- Approximate time remaining until completion of operation;
- Box status: "Connected" and "Disconnected";
- Current version of box firmware;
- Box serial number;
- Support and service;
- SRF Manager;
- 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.
- Current software version



Brand: Custom Settings

Device (Core): Custom Loader

□ Read Android build info while connecting

☑ Write data verification

MB

Start 0 0 b

Length 0 0 b

Log

```
Welcome to Medusa Pro Software version 2.2.4.
Started.
Reader #0: Generic EMV Smartcard Reader 0
Loading firmware. Please wait...   Done.
Selected UFS interface.
Socket detected!
```

0%

VREF: 0.00V | Speed: | Progress: | ETA: | Box status: Connected | Firmware version: 1.00 | S/N: 28ED

Fig. 2 Main Window, Interface Tab

- Initialization of the connected device;
- Finding and reading "Android Info" when initializing the device;
- Standard read, write and erase features;
- Checking recorded data;

- 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);
- Recovering a flash drive using an SRF file;
- Work with partitions;
- Creating SRF files;
- 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 Medusa team for faster and easier device recovery).

# 2. Work with eMMC Flash Memory

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

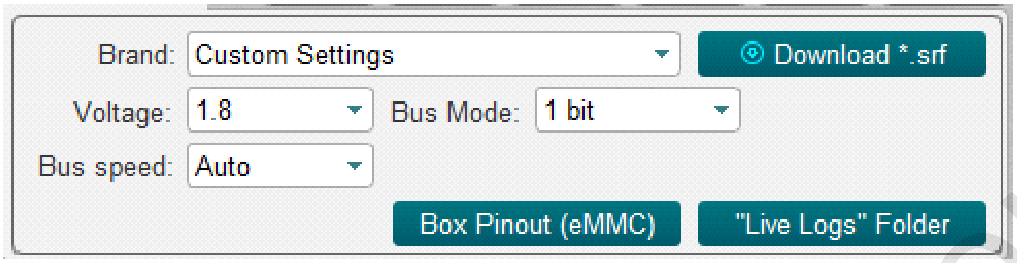
Medusa Pro Software allows you to work with flash media by choosing a data bus width of 1, 4 or 8 bits.

| Box | Data Bus Width, Bits |
|---|---|
| Medusa Pro | 1, 4 |
| Medusa Pro II | 1, 4, 8 |

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. If the flash drive is connected using the Medusa socket, then the bus width can be selected depending on which box is currently being used, according to Table 1.



Brand: Custom Settings

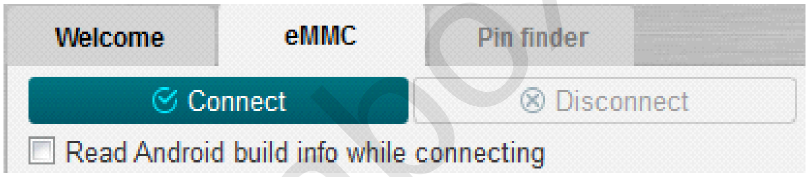
Voltage: 1.8

Bus Mode: 1 bit

Bus speed: Auto

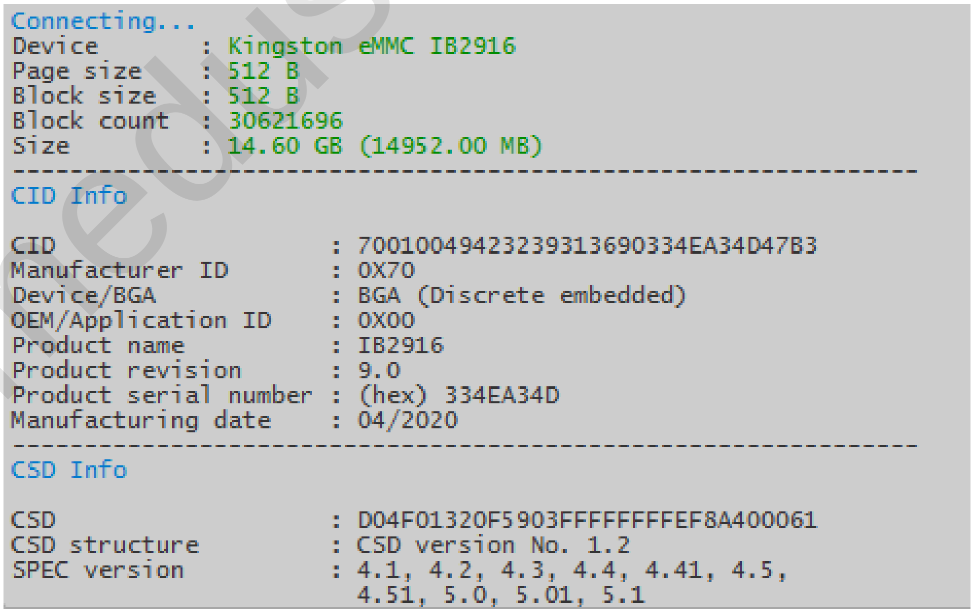
Fig. 3 Configuring Basic Parameters for Initializing of eMMC Flash Drive

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).



□ Read Android build info while connecting

Fig. 4



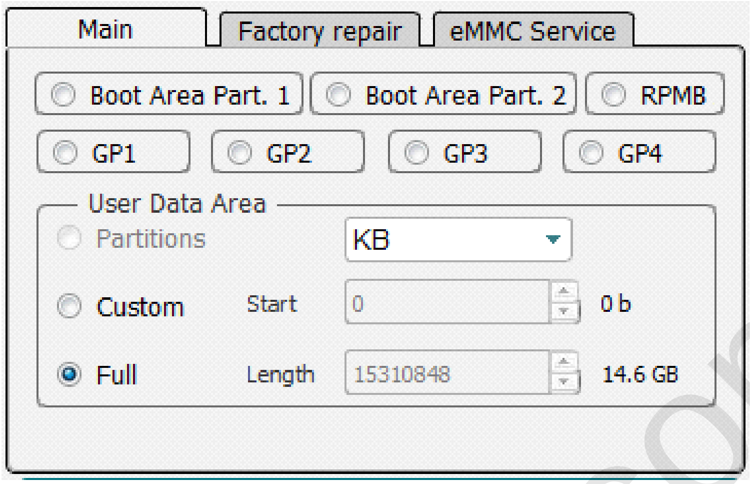
```
Connecting...
Device         : Kingston eMMC IB2916
Page size      : 512 B
Block size     : 512 B
Block count    : 30621696
Size           : 14.60 GB (14952.00 MB)
----------------------------------------------------------------
CID Info

CID                   : 70010049423239313690334EA34D47B3
Manufacturer ID       : 0X70
Device/BGA            : BGA (Discrete embedded)
OEM/Application ID    : 0X00
Product name          : IB2916
Product revision      : 9.0
Product serial number : (hex) 334EA34D
Manufacturing date    : 04/2020
----------------------------------------------------------------
CSD Info

CSD                   : D04F01320F5903FFFFFFFFEF8A400061
CSD structure         : CSD version No. 1.2
SPEC version          : 4.1, 4.2, 4.3, 4.4, 4.41, 4.5,
                        4.51, 5.0, 5.01, 5.1
```

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.



○ Partitions  KB

○ Custom  Start 0  0 b

● Full  Length 15310848  14.6 GB

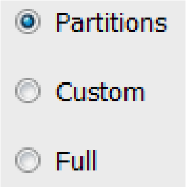
Fig.6

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.

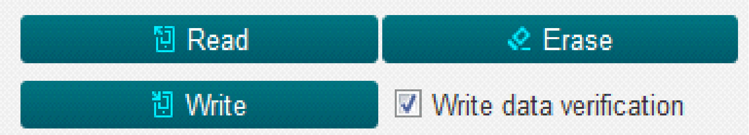


● Partitions

○ Custom

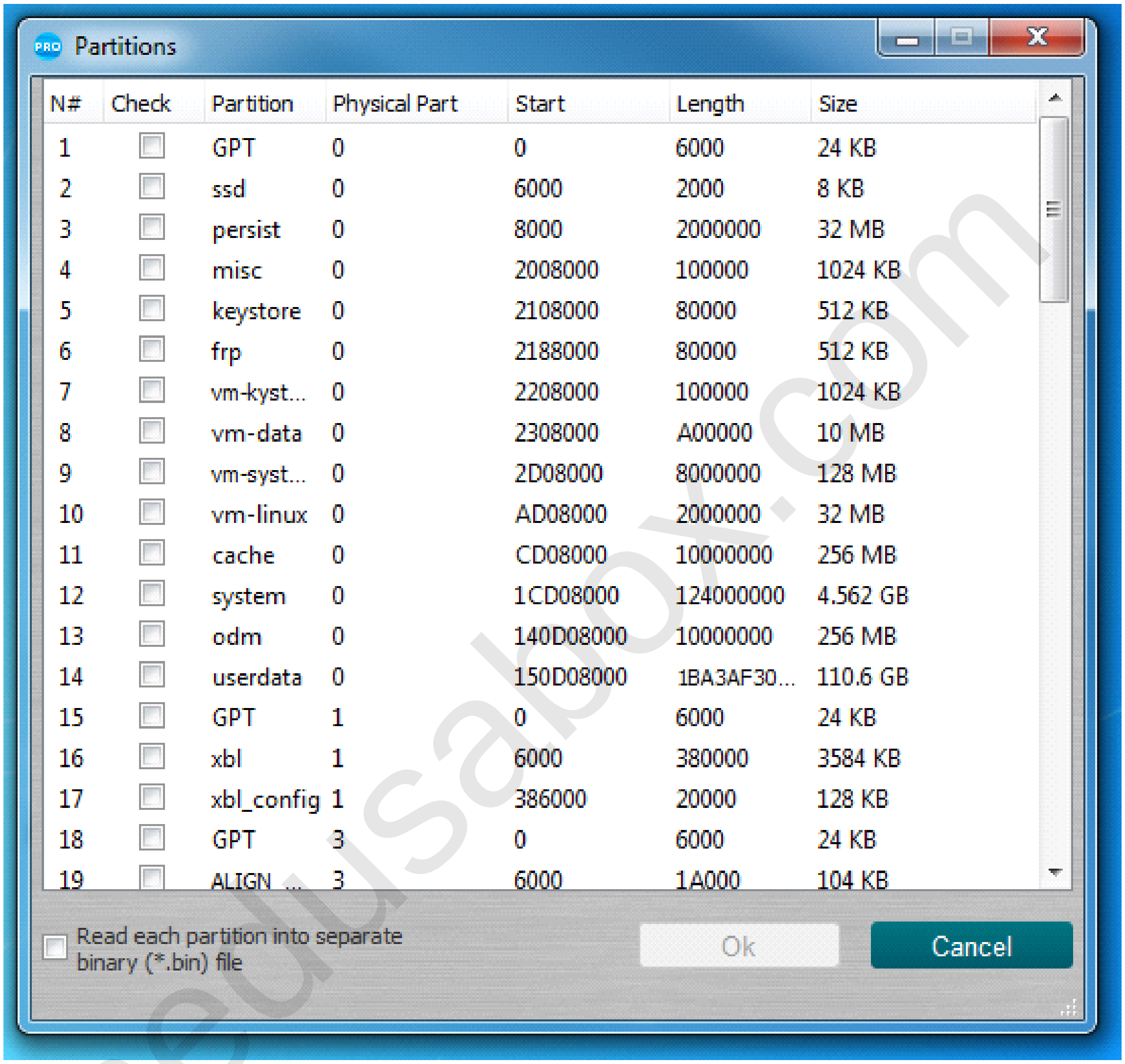
○ Full

Fig. 7



☑ Write data verification

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 |

☐ Read each partition into separate binary (*.bin) file

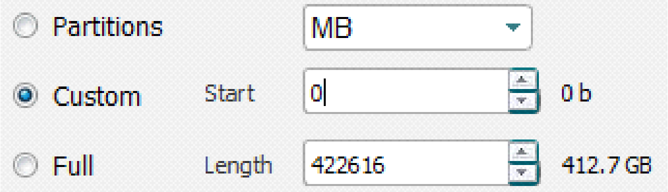
Fig. 9 Partitions Window

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 in separative 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:



○ Partitions MB

● Custom Start 0 0 b

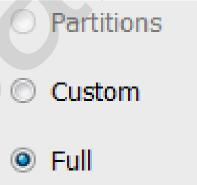
○ Full Length 422616 412.7 GB

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)



○ Partitions

○ Custom

● Full

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.

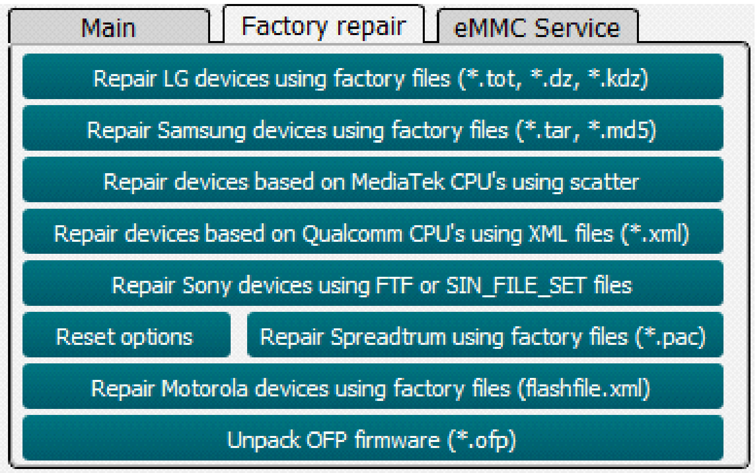

Main
Factory repair
eMMC Service
Repair LG devices using factory files (*.tot, *.dz, *.kdz)
Repair Samsung devices using factory files (*.tar, *.md5)
Repair devices based on MediaTek CPU's using scatter
Repair devices based on Qualcomm CPU's using XML files (*.xml)
Repair Sony devices using FTF or SIN_FILE_SET files
Reset options
Repair Spreadtrum using factory files (*.pac)
Repair Motorola devices using factory files (flashfile.xml)
Unpack OFP firmware (*.ofp)

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.

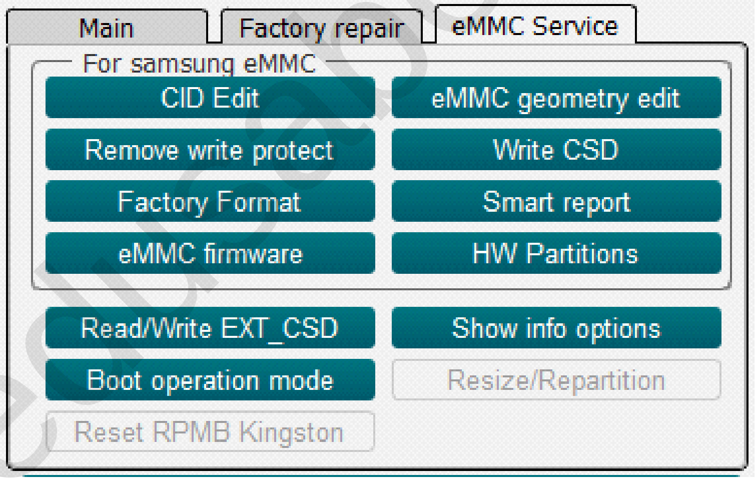


Fig. 13 eMMC Service Tab

**“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;

**!!! Medusa 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;

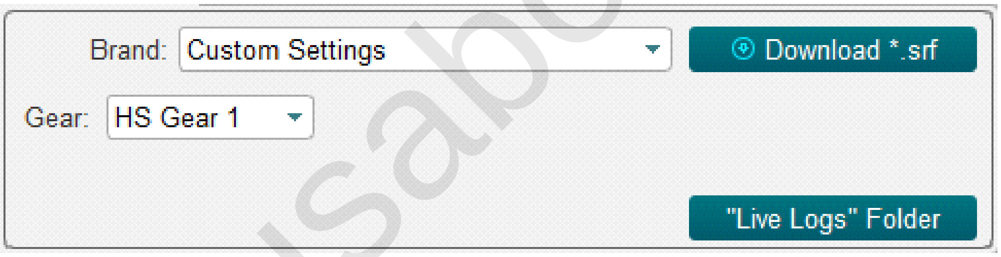
**“Boot operation mode”** - Boot setup**;**

## 3. Work with UFS Flash Media

Medusa Pro Software supports working with UFS-type flash media and **Medusa Pro II** box. **Medusa Pro does not support work with UFS!!!** UFS standards supported by Medusa Pro Software:

- UFS 2.0
- UFS 2.1
- UFS 3.0
- UFS 3.1.

Medusa Pro II works on one Lane and supports the following bus modes: LS PWM G1, LS PWM G2, LS PWM G3, LS PWM G4, HS G1. Working with UFS media basically coincides with the eMMC and the **UFS Service** tab (Fig. 15). The difference is in the initialization setting - in UFS this tab looks like in Fig. 14, where in the **"Gear"** field you can select in which of the modes the UFS interface will work (Table 2).



Brand: Custom Settings

Gear: HS Gear 1

Fig. 14 Setting basic parameters for initializing a UFS flash drive

| Gears | Min (Mbps) | Max (Mbps) |
|---|---|---|
| LS PWM-G1 | 3 (300KB/s) | 9 (900KB/s) |
| LS PWM-G2 | 6 (600KB/s) | 18 (1.8MB/s) |
| LS PWM-G3 | 12 (1.2MB/s) | 36 (3.6MB/s) |
| LS PWM-G4 | 24 (2.4MB/s) | 72 (7.2MB/s) |
| HS G1 | | 1248 (124.8MB/s) |

Table 2. Transfer rate matching from Gear

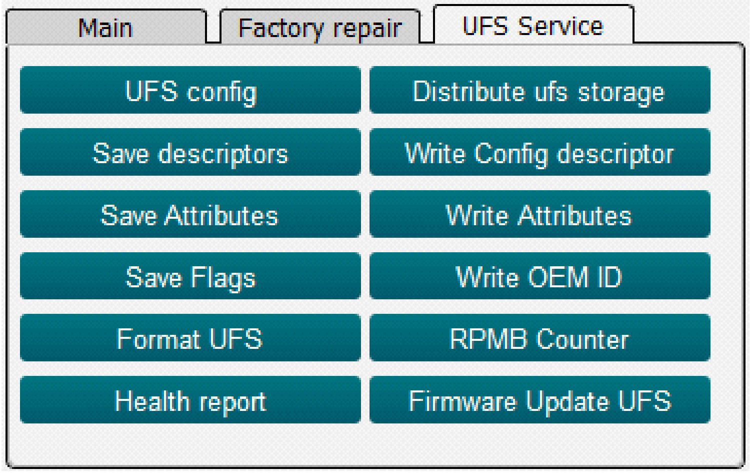


Fig. 15 UFS Service Tab

## 3.1. UFS Work with Service Features (UFS Service)

The "UFS config" button allows you to view all Descriptors, Flags and Attributes.

**"Save descriptors", "Save Attributes"** or **"Save Flags"** buttons are used to save Descriptors, Attributes and Flags.

**Write Config descriptor** - button is used for writing the configuration descriptor. According to the UFS standard, to configure flash media, you must write the Configuration Descriptor, which has been previously read. Since the Config Descriptor format depends on the version of the UFS specification, Medusa Pro Software provides automatic conversion of the Config Descriptor to the required version.

*For example: if you are trying to write Config Descriptor version 2.1 to a flash drive with version 3.1, the software will automatically convert the descriptor.*

**“Format UFS” -** Used for removing all LUs.

**“Distribute UFS storage”** - Used for partitioning of a flash drive into LUs.

**“Write Attribute”** - Used for writing attributes from a saved file.

**“Write OEM ID”** - Used for writing the OEM ID from a saved file.

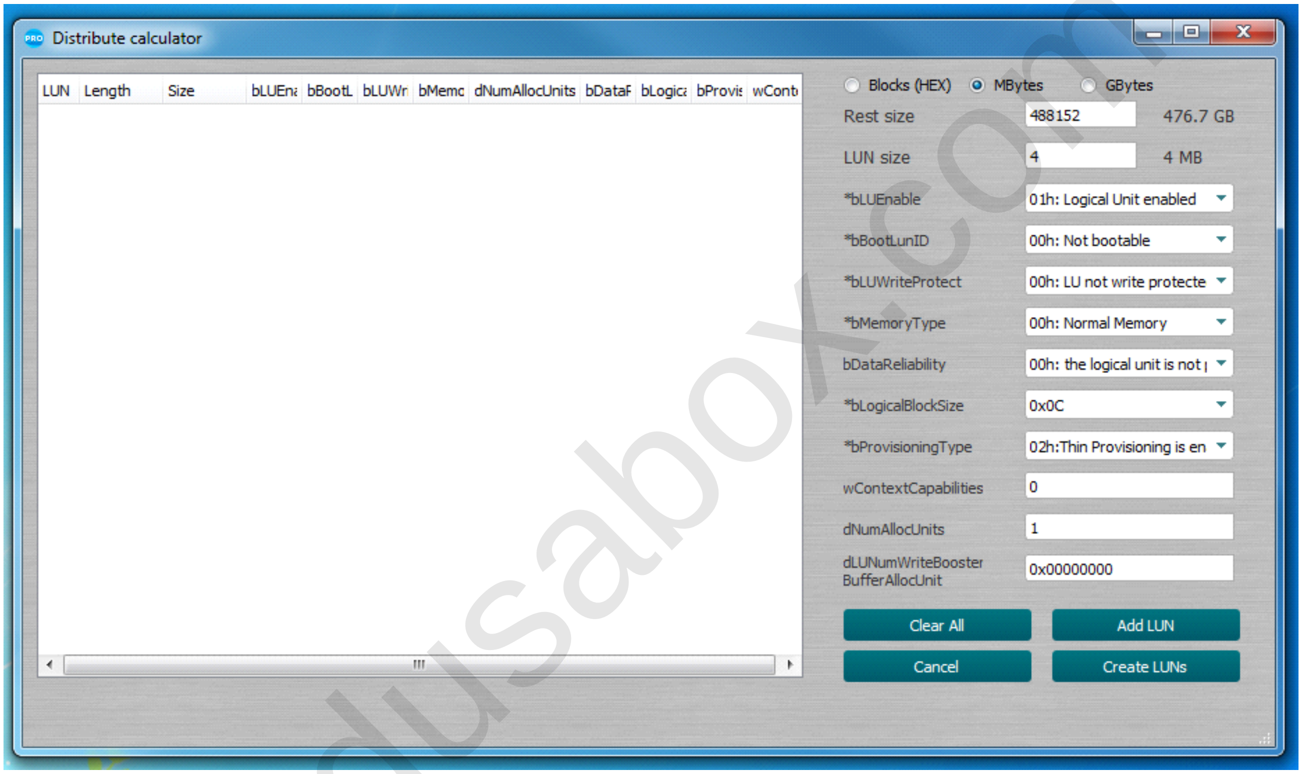
**“RPMB Counter”** - Used for reading the RPMB counter.

**“Health report”** - Reads information about the flash drive resource.

**“Firmware Update UFS”** - Used for updating UFS controller firmware.

## 3.1.1 Partitioning of UFS Media into LUs (Distribute UFS Storage)

To partition a flash drive into LU, you must click the **"Distribute UFS storage"** button; this will open the window shown in the Fig.16. LU sizes can be entered in blocks, megabytes and gigabytes, just switch the input mode Fig. 17(A). The LU size is entered in the **"LUN Size"** field, while the remaining volume will be displayed in the **"Rest size"** field (Fig. 18), pressing the input mode again transfers the residual size to the size field. 17(B). By clicking the **"Add LUN"** button, the section will appear in the list, thus creating a list of all LUs. To create these LUs on a flash drive, you must click the **"Create LUNs"** button. The log will show a message about the successful writing of the configuration. The software checks the number of created LUs, if the number exceeds the maximum possible, the **"Add LUN"** button becomes inactive.



Distribute calculator

| LUN | Length | Size | bLUEn: | bBootL | bLUWr | bMemc | dNumAllocUnits | bDataF | bLogic: | bProvi: | wConte |
|---|---|---|---|---|---|---|---|---|---|---|---|

○ Blocks (HEX) ● MBytes ○ GBytes

Rest size 488152 476.7 GB

LUN size 4 4 MB

*bLUEnable 01h: Logical Unit enabled

*bBootLunID 00h: Not bootable

*bLUWriteProtect 00h: LU not write protecte

*bMemoryType 00h: Normal Memory

bDataReliability 00h: the logical unit is not

*bLogicalBlockSize 0x0C

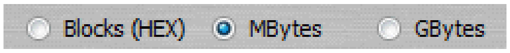
*bProvisioningType 02h:Thin Provisioning is en

wContextCapabilities 0

dNumAllocUnits 1

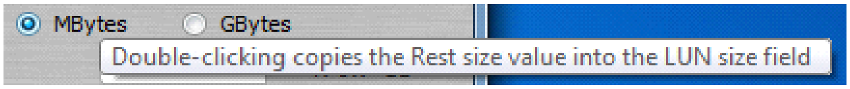
dLUNumWriteBoosterBufferAllocUnit 0x00000000

Fig. 16 Distribute UFS storage Window



○ Blocks (HEX) ● MBytes ○ GBytes

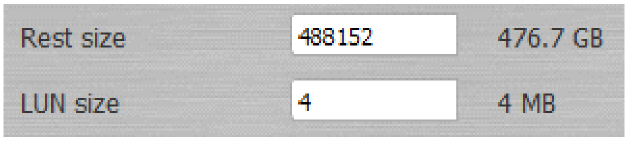
Fig. 17(A) LU size input mode



● MBytes ○ GBytes

Double-clicking copies the Rest size value into the LUN size field

Fig. 17(B) Double click on LU size input mode



Rest size 488152 476.7 GB

LUN size 4 4 MB

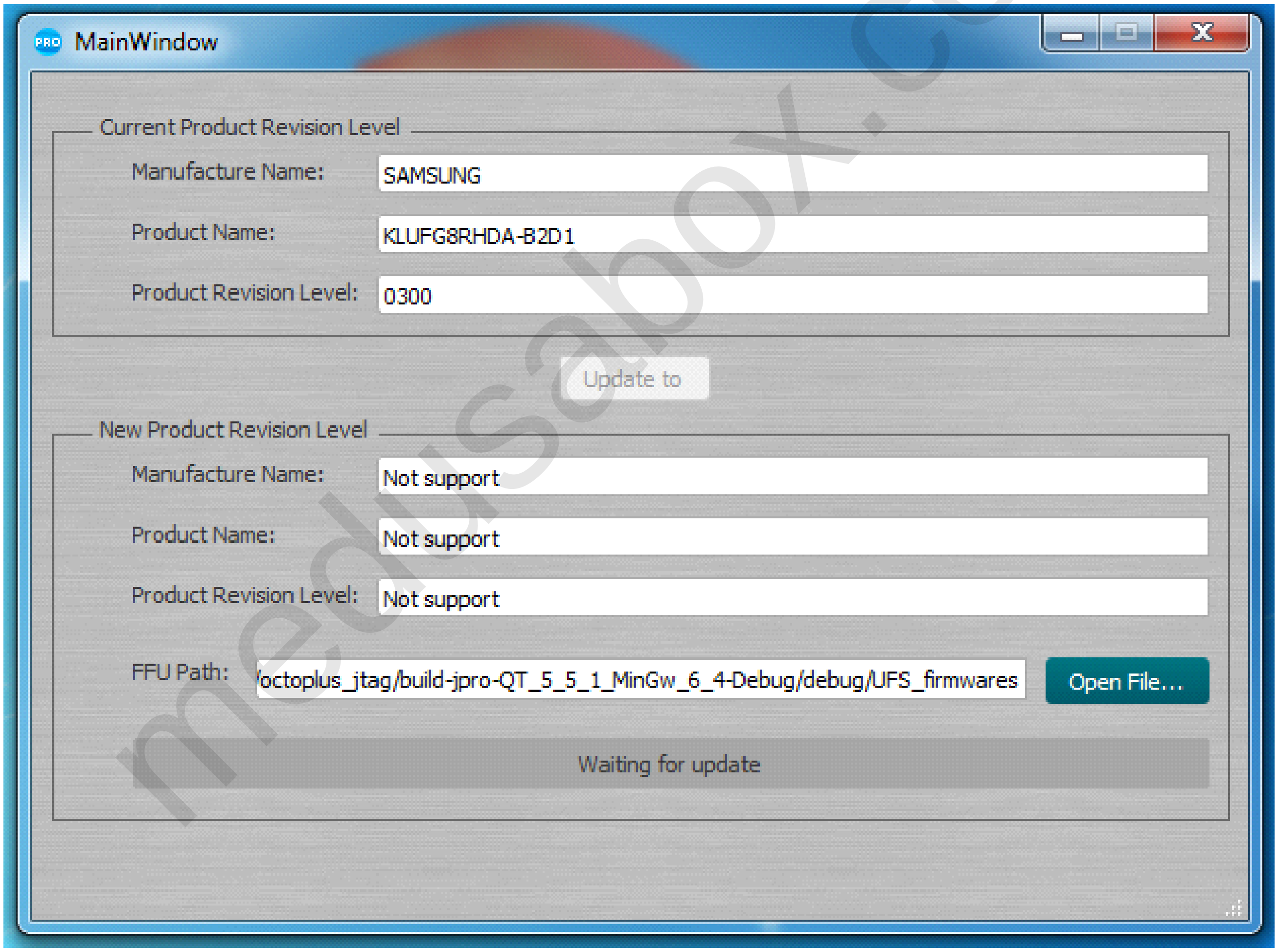
Fig. 18 Current LU remainder and size fields

## 3.1.2 UFS controller firmware update (Firmware update UFS)

If you need to update the firmware of the UFS controller, this can be done by clicking the **"Firmware Update UFS"** button (Fig.15), this will open the firmware recovery window (Fig.19). Manufacturer, name and current revision of the connected flash drive are indicated in the upper part of the window. The bottom part shows the manufacturer, name and revision of the firmware to which you plan to upgrade. If the firmware is found in the Medusa Pro Software database, all the fields at the bottom of the window will be filled in automatically and the **"Update to"** button will become active, otherwise the fields will be marked as **"Not supported".**

The user can update the UFS controller with his own firmware by clicking the **"Open file ..."** button and specifying the path to the custom firmware. If the firmware is parsed, the **"Update to"** button will become active.

**!!! Medusa 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.**



Current Product Revision Level

Manufacture Name: SAMSUNG

Product Name: KLUFG8RHDA-B2D1

Product Revision Level: 0300

New Product Revision Level

Manufacture Name: Not support

Product Name: Not support

Product Revision Level: Not support

FFU Path: /octoplus_jtag/build-jpro-QT_5_5_1_MinGw_6_4-Debug/debug/UFS_firmwares

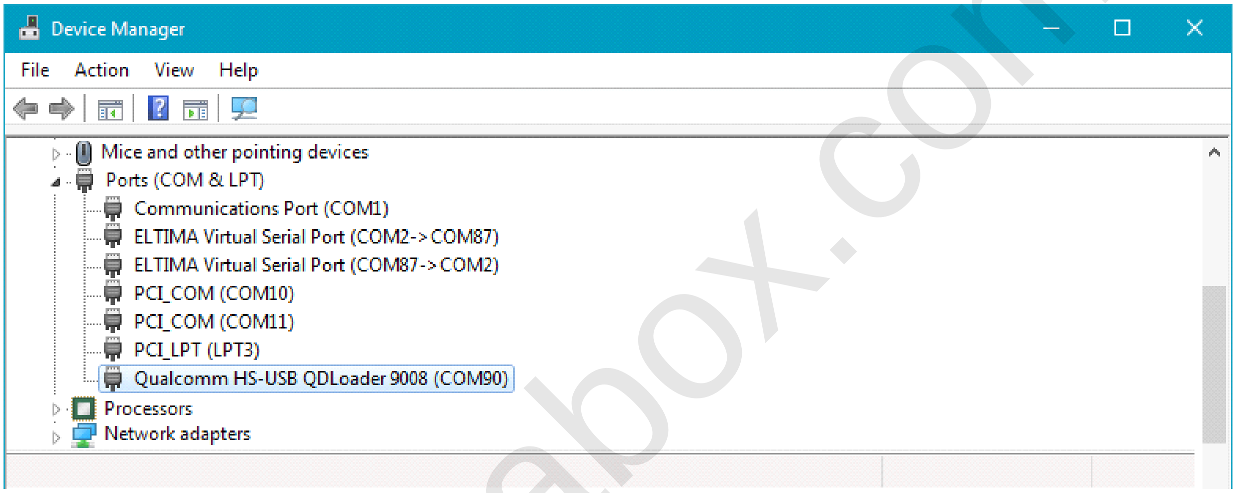
Fig.19 Firmware Update UFS recovery window

# 4. Work with USB

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

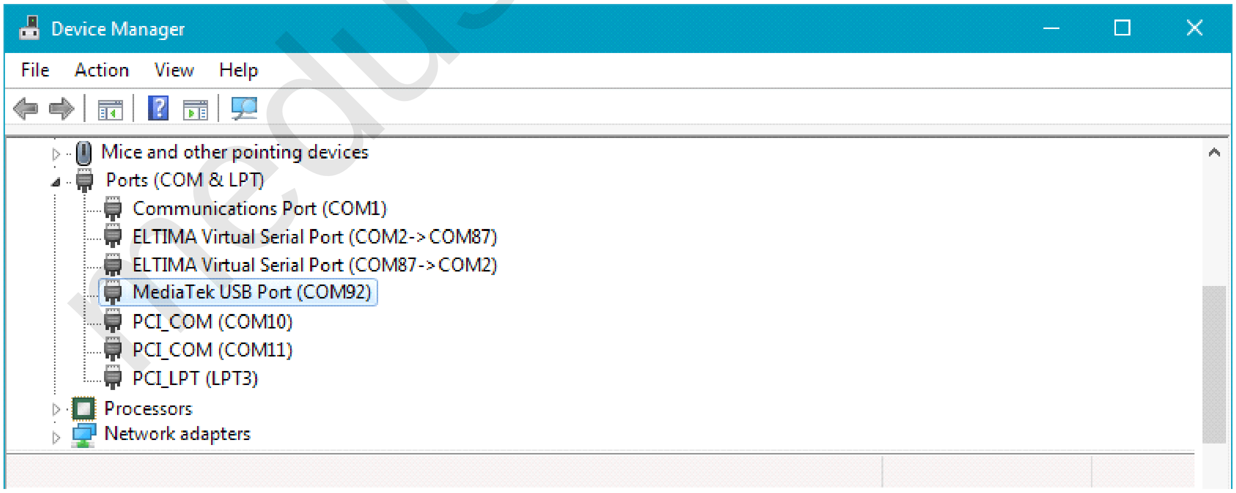
To initialize a device, it must be in the EDL mode (Emergency Download Mode). It is possible to switch to EDL mode in different ways, the most effective way is to short certain points (**test points**) on the device board. You must partially disassemble the device to perform this procedure.

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).



- Mice and other pointing devices
- Ports (COM & LPT)
  - Communications Port (COM1)
  - ELTIMA Virtual Serial Port (COM2->COM87)
  - ELTIMA Virtual Serial Port (COM87->COM2)
  - PCI_COM (COM10)
  - PCI_COM (COM11)
  - PCI_LPT (LPT3)
  - Qualcomm HS-USB QDLoader 9008 (COM90)
- Processors
- Network adapters

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



- Mice and other pointing devices
- Ports (COM & LPT)
  - Communications Port (COM1)
  - ELTIMA Virtual Serial Port (COM2->COM87)
  - ELTIMA Virtual Serial Port (COM87->COM2)
  - MediaTek USB Port (COM92)
  - PCI_COM (COM10)
  - PCI_COM (COM11)
  - PCI_LPT (LPT3)
- Processors
- Network adapters

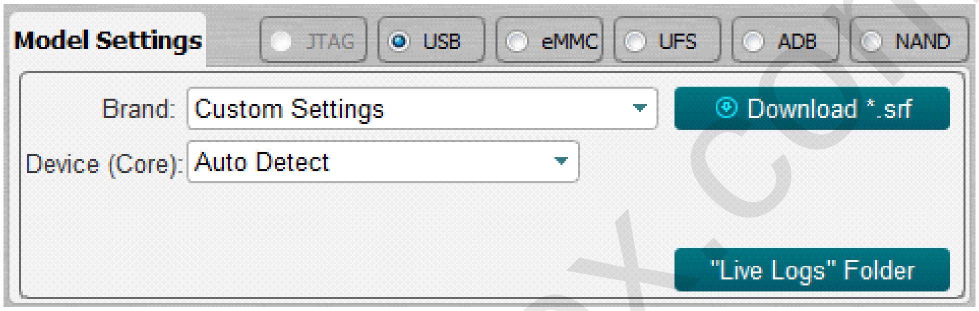
Fig. 21 Device with MTK CPU in EDL mode, connected via USB

## 4.1. 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).



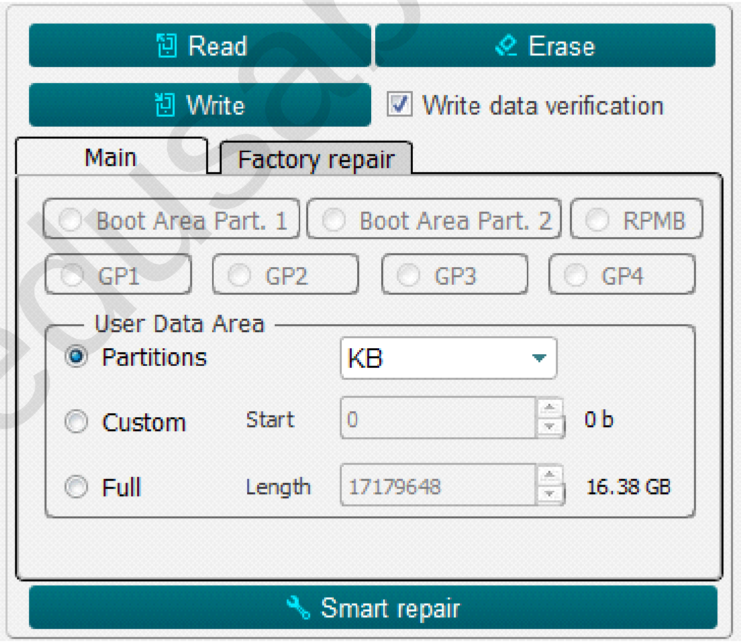
**Model Settings**

○ JTAG ● USB ○ eMMC ○ UFS ○ ADB ○ NAND

Brand: Custom Settings

Device (Core): Auto Detect

Fig.22



☑ Write data verification

○ Boot Area Part. 1 ○ Boot Area Part. 2 ○ RPMB

○ GP1 ○ GP2 ○ GP3 ○ GP4

User Data Area

● Partitions KB

○ Custom Start 0 0 b

○ Full Length 17179648 16.38 GB

Fig. 23

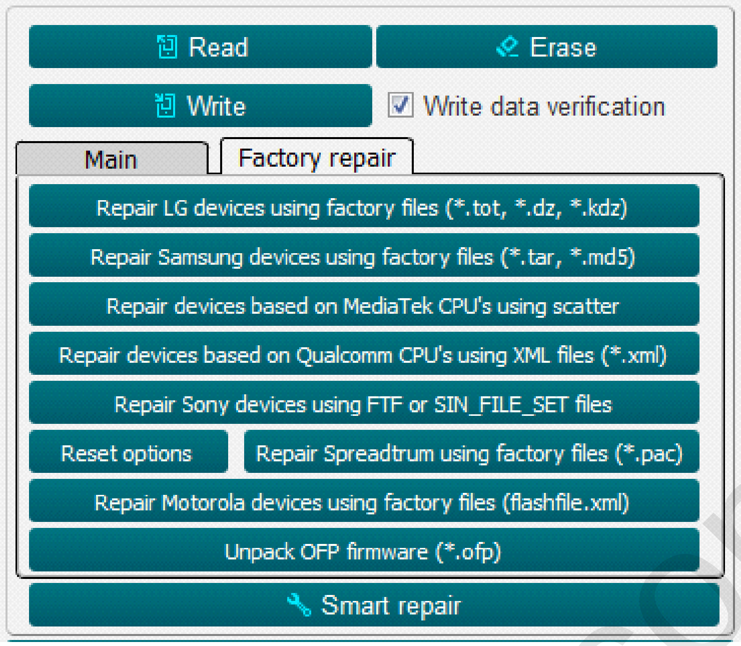

Read
Erase
Write
☑ Write data verification
Main
Factory repair
Repair LG devices using factory files (*.tot, *.dz, *.kdz)
Repair Samsung devices using factory files (*.tar, *.md5)
Repair devices based on MediaTek CPU's using scatter
Repair devices based on Qualcomm CPU's using XML files (*.xml)
Repair Sony devices using FTF or SIN_FILE_SET files
Reset options
Repair Spreadtrum using factory files (*.pac)
Repair Motorola devices using factory files (flashfile.xml)
Unpack OFP firmware (*.ofp)
Smart repair

Fig. 24

## 4.2. 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).

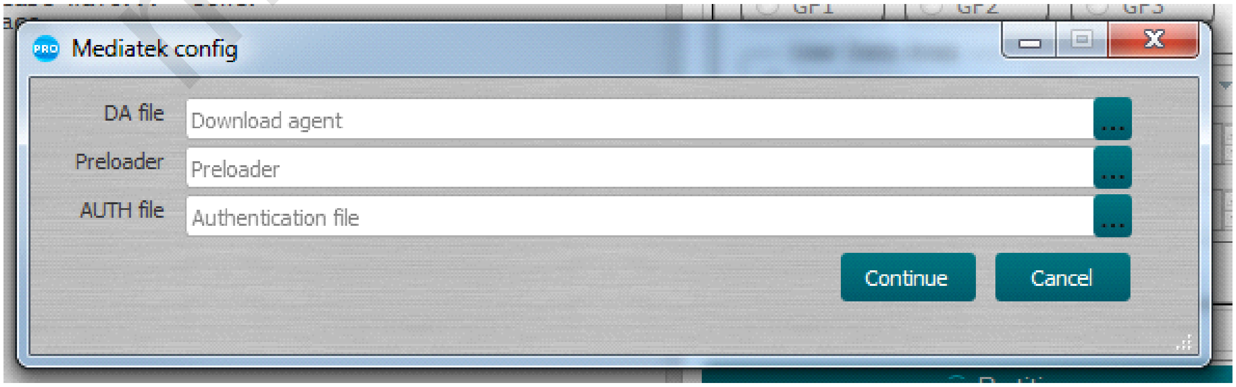
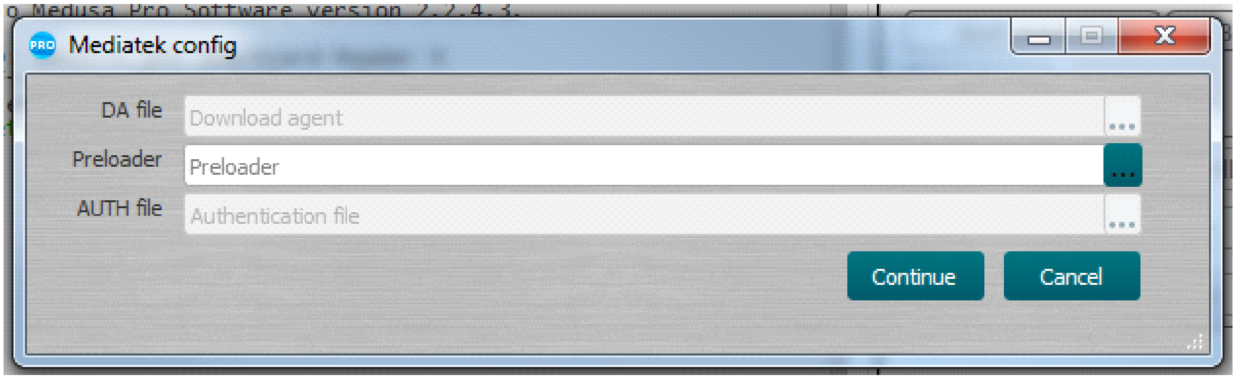


Fig. 25


Medusa Pro Software version 2.2.4.3.

Mediatek config

DA file: Download agent

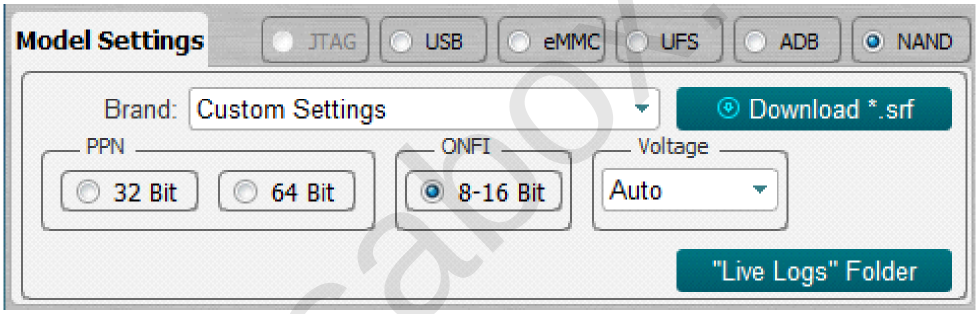
Preloader: Preloader

AUTH file: Authentication file

Fig. 26

## 5. Work with NAND

Medusa Pro Software supports JEDEC ONFI NAND flash drives with 8 and 16 bit widths and Apple PPN 32 / 64bit. Physically, the flash drive is connected via **Medusa** socket, or soldered to the **Pin Connector**. Depending on the type of flash drive, **PPN** or **ONFI** mode should be selected (Fig. 27)


Model Settings

○ JTAG ○ USB ○ eMMC ○ UFS ○ ADB ● NAND

Brand: Custom Settings

PPN: ○ 32 Bit ○ 64 Bit

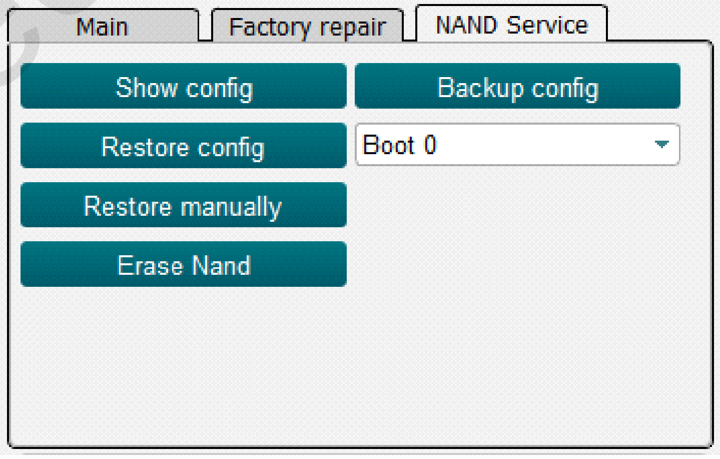
ONFI: ● 8-16 Bit

Voltage: Auto

Fig. 27

Only standard read / write / erase features are available in **ONFI** mode.

In **PPN** mode, in addition to standard features, additional features are in the NAND Service tab (Fig. 28).


Main | Factory repair | NAND Service

Boot 0

Fig. 28

- **Show config** - Highlights configuration information in the log;
- **Backup config** - Saves configurations to separate files;
- **Restore config** - Restores the configuration from the saved file in the selected **"Boot";**
- **Restore manually** - Provides access to configuration correction manually;
- **Erase NAND** - Overwrites the NAND drive.

## 6. 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.