

The MediaTek logo is displayed in white, bold, uppercase letters within a white, parallelogram-shaped background element.

MEDIATEK

MT7981 Single Image SOP

2022/4/25

Document Revision History

Revision	Date	Author (Optional)	Description
0.1	2022-3-6	Micheal Su	Initial draft
1.0	2022-4-25	Jones Huang	Official release

Outline

- ❑ **Generate SPIM-NAND Single Image**
- ❑ **Generate eMMC Single Image**

Generate SPIM-NAND Single Image

Prepare File for SPIM-NAND Single Image

- **bl2.img**
 - Please refer to MT7981_Build_SOP_xxx.pdf application note.
- **fip.bin**
 - Please refer to MT7981_Build_SOP_xxx.pdf application note.
- **kernal_image**
 - Please refer to MT7981_Build_SOP_xxx.pdf application note.
- **mk_image.sh**
 - In ATF folder, i.e. `atf/tools/dev/single_img_wrapper/mk_image.sh`

*Note: You can also find the `mk_image.sh` in `atf` on MTK DCC center

How to Generate SPIM-NAND Single Image

- Put all those files under the same folder,
 - bl2.img
 - fip.bin
 - kernal_image, e.g. openwrt-mediatek-mt7981-mt7981-spim-nand-rfb-squashfs-factory.bin
 - mk_image.sh
- Run mk_image.sh
 - CMD:~/#> ./mk_image.sh -p <CHIP Name> -d <Flash Type> -b <bl2.img> -f <fip.bin> -k <Kernel image>
 - For example:
CMD:~/#> ./mk_image.sh -p mt7981abd -d spim-nand -b bl2.img -f fip.bin -k openwrt-mediatek-mt7981-mt7981-spim-nand-rfb-squashfs-factory.bin

Note: If you are using mt7981a/7981b/7981d then use **mt7981abd** as <CHIP Name>
- The single image “mt7981-spim-nand-XXXX-single-image.bin” generated in the same folder.

Customize partition config

- You can customize your own partition config in ./partitions
- Take spim-nand-default.yml for example:

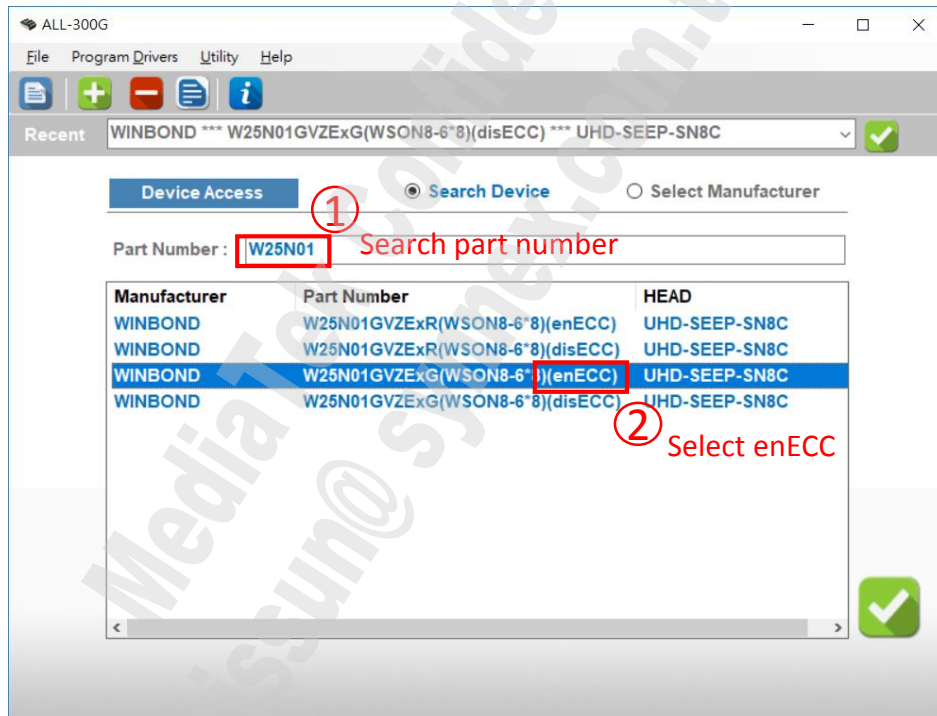
```
1 # NAND flash layout:
2 # 0x0~0x100000 : BL2, 1024K
3 # 0x100000~0x180000 : Uboot env, 512K
4 # 0x180000~0x380000 : RF, 2048K
5 # 0x380000~0x1080000: FIP, 13M
6 # 0x1080000~ : firmware
7
8 spim-nand:
9   bl2_start: 0x0
10  rf_start: 0x180000
11  fip_start: 0x380000
12  kernel_start: 0x1080000
```

You can modify partitions' offset here

How to Program SPIM-NAND Single Image by Programmer

The example of ALL-300G programmer

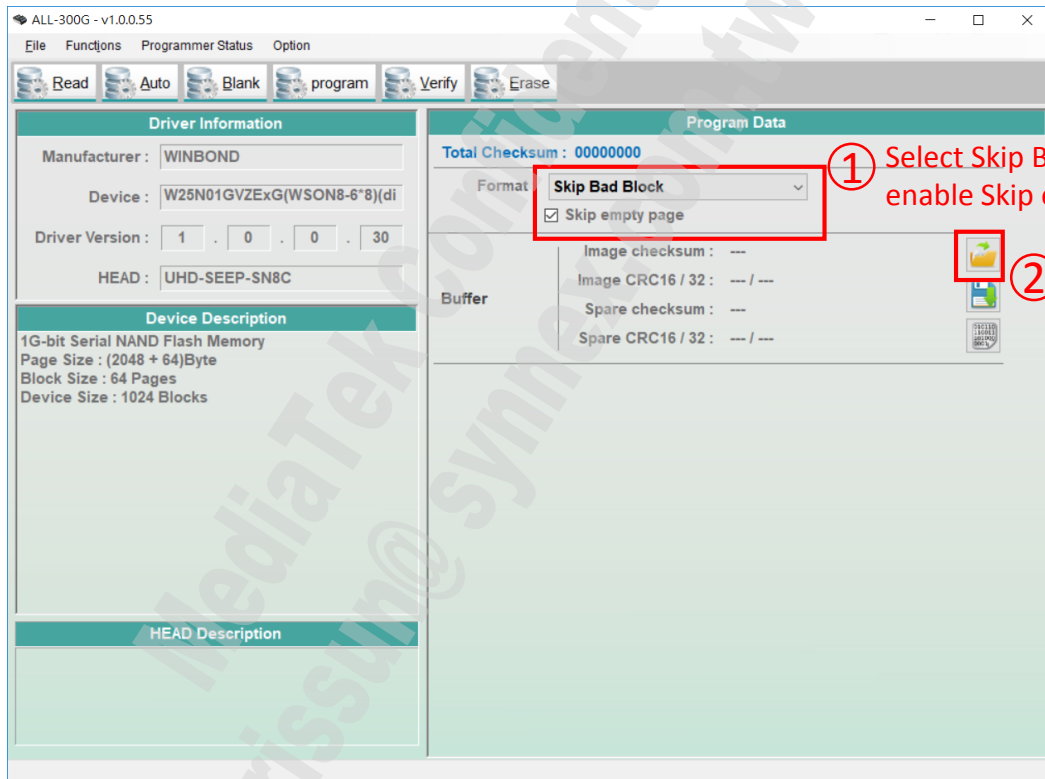
- Select SPI-NAND P/N (Winbond W25N01GVZEIG part as example)



How to Program SPIM-NAND Single Image by Programmer

The example of ALL-300G programmer

- Load image

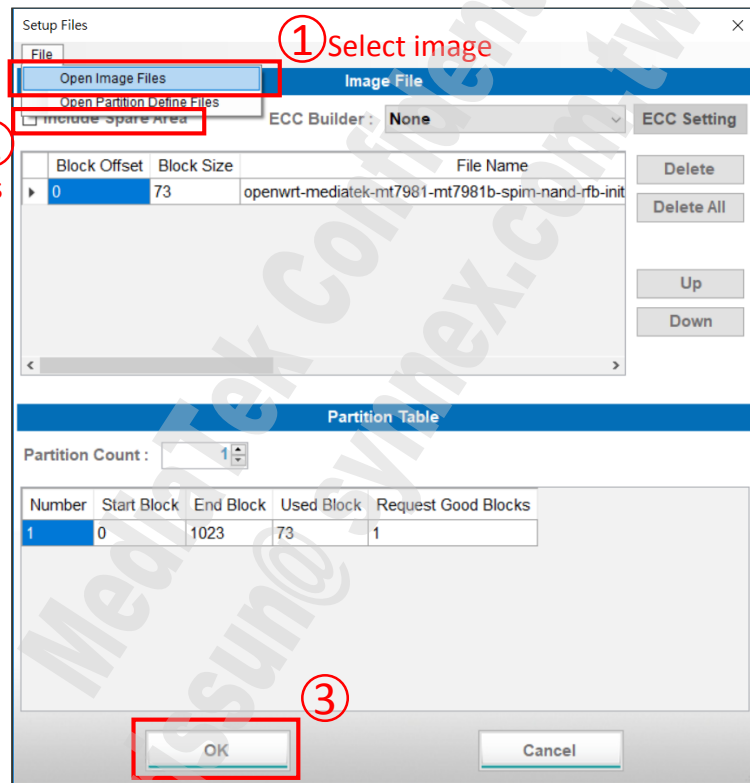


How to Program SPIM-NAND Single Image by Programmer

The example of ALL-300G programmer

- Load image

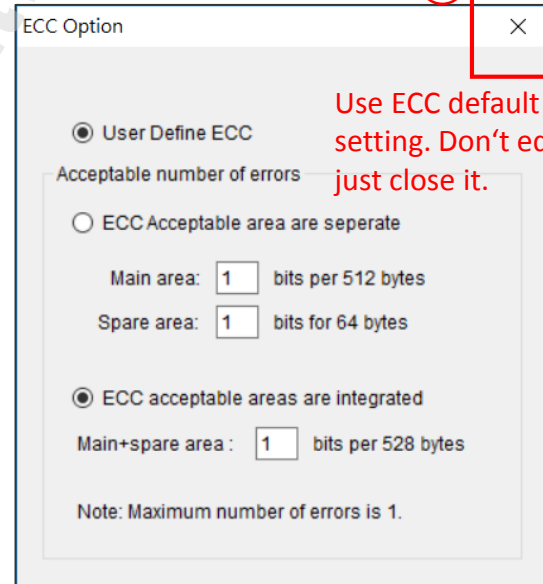
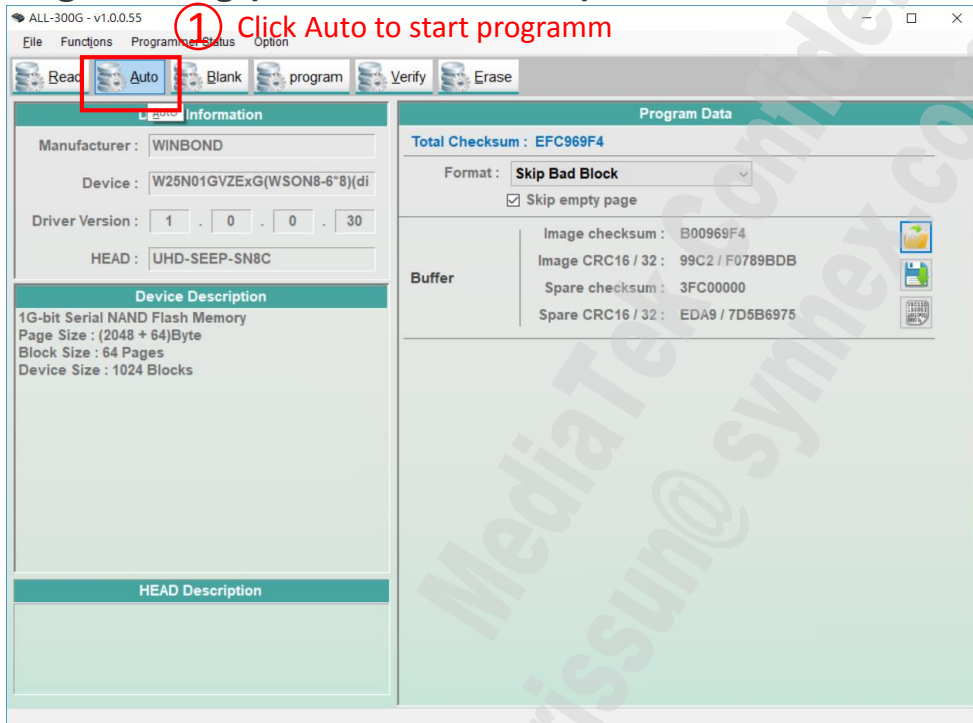
Uncheck Include Spare Ares



How to Program SPIM-NAND Single Image by Programmer

The example of ALL-300G programmer

- Programming (check contact first)



How to Program SPIM-NAND Single Image by Programmer

The example of ALL-300G programmer

- Programming (formal programming)

Auto

Manufacturer : WINBOND
Device : W25N01GVZExG(WSON8-6*8)(disECC)

Total Checksum : EFC969F4

Items	Programmer																Counter		
	Site#	Status	Socket #																Pass
<input checked="" type="checkbox"/> Short Test																			
<input checked="" type="checkbox"/> Contact	0																	0	0
<input checked="" type="checkbox"/> ID Check	1																	0	0
<input checked="" type="checkbox"/> Detect Blocks	2																	0	0
<input checked="" type="checkbox"/> Erase	3																	0	0
<input checked="" type="checkbox"/> Blank Check	4																	0	0
<input checked="" type="checkbox"/> Program/Verify	5																	0	0
	6																	0	0
	7																	0	0

① For MP programing, suggest enable all of options

Total : 0 0

Reset Counter

Run Close

Generate eMMC Single Image

eMMC Physical Partitions

- According to eMMC standard 5.1 section 6.2.1, eMMC devices have the following physical partitions
- In MTK's platforms, we use boot area partition 1 & user data area (UDA) only.
- BL2 is placed at boot partition 1, and the rest is at UDA.

6.2.1 General

The default area of the memory device consists of a User Data Area to store data, two possible boot area partitions for booting (see 6.3.2) and the Replay Protected Memory Block Area Partition (see 6.6.22) to manage data in an authenticated and replay protected manner. The memory configuration initially consists (before any partitioning operation) of the User Data Area and RPMB Area Partitions and Boot Area Partitions (whose dimensions and technology features are defined by the memory manufacturer).

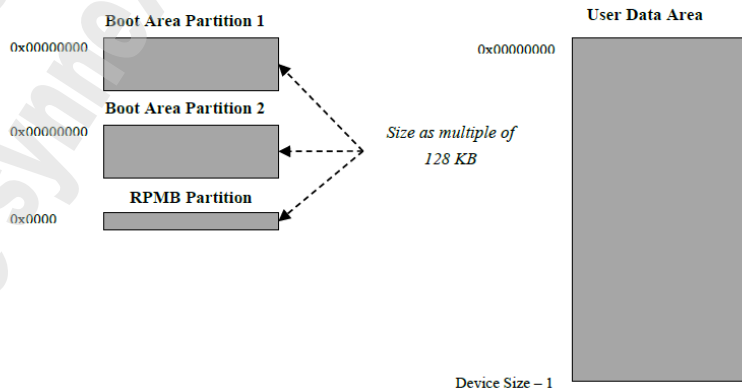
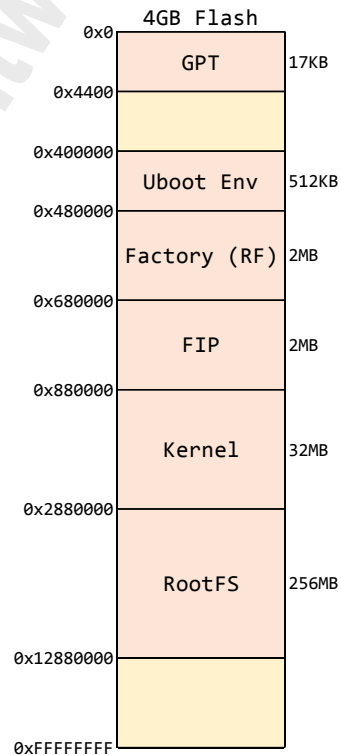


Figure 14 — eMMC memory organization at time zero

MT7981 eMMC Partition Layout

- **Boot partition 1: BL2**
- **UDA:**
The flash partition layout define at
`atf/tools/dev/gpt_editor/example/mt7981-emmc.json`



MT7981 eMMC Partition Layout

- For Programmer need to prepare bl2.img for boot partition 1 and mt7981-eMMC-single-image.bin for UDA.
- mt7981-eMMC-single-image.bin is includes,
 - GPT (GPT_EMMC)
 - FIP (fip.bin)
 - firmware (kernel image, openwrt-mediatek-mt7981-xxxxxx.bin)

Prepare File for eMMC Single Image

- **bl2.img**
 - Please refer to MT7981_Build_SOP_xxx.pdf application note.
- **fip.bin**
 - Please refer to MT7981_Build_SOP_xxx.pdf application note.
- **kernal_image**
 - Please refer to MT7981_Build_SOP_xxx.pdf application note.
- **GPT_EMMC**
 - `cd atf/tools/dev/gpt_editor`
 - `python mtk_gpt.py --i example/mt7981-emmc.json --o GPT_EMMC`
- **mk_image.sh**
 - In ATF folder, i.e. `atf/tools/dev/single_img_wrapper/mk_image.sh`

*Note

If you want to customize partition layout, please modify the setting in 2 files:

- `atf/tools/dev/gpt_editor/example/mt7981-emmc.json`
- `atf/tools/dev/single_img_wrapper/partitions/emmc-default.yml`

How to Generate eMMC Single Image

- Put all those files under the same folder,
 - GPT_EMMC
 - fip.bin
 - kernal_image, e.g. OF_openwrt-mediatek-mt7981-mt7981-emmc-rfb-squashfs-sysupgrade-xxxx.bin
 - mk_image.sh
- Run mk_image.sh
 - CMD:~/#> ./mk_image.sh -p <CHIP Name> -d <Flash Type> -g <GPT table> -f <fip.bin> -k <Kernel image>
 - For example:
CMD:~/#> ./mk_image.sh -p mt7981abd -d emmc -g GPT_EMMC -f fip.bin -k OF_openwrt-mediatek-mt7981-mt7981-emmc-rfb-squashfs-sysupgrade-xxxx.bin

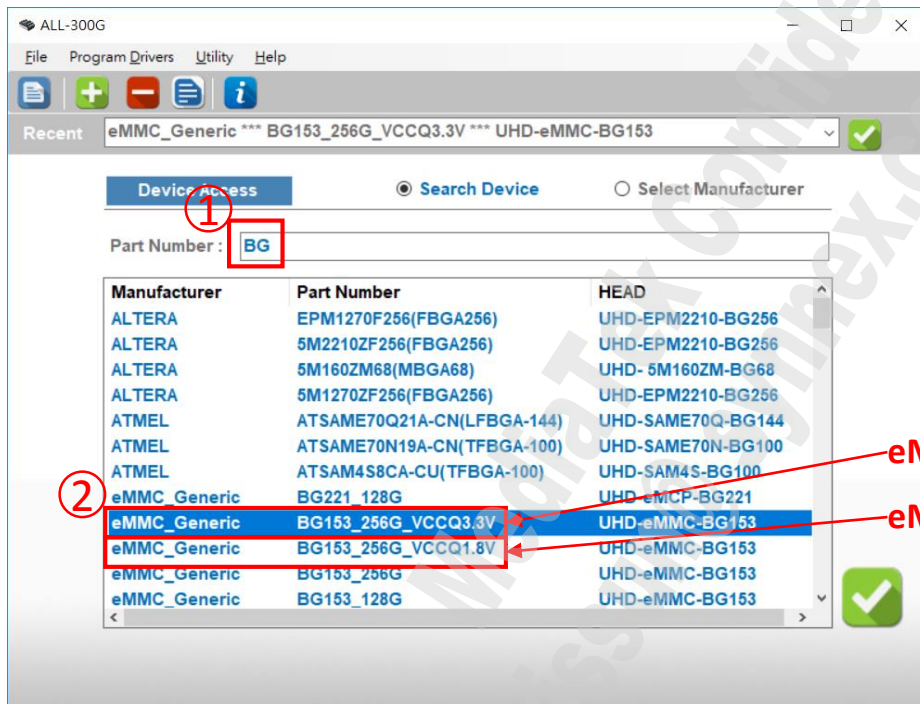
Note: If you are using mt7981a/7981b/7981d then use **mt7981abd** as <CHIP Name>

- The single image “mt7981-eMMC-single-image.bin” generated in the same folder.

How to Program Single Image by Programmer

The example of ALL-300G programmer

- Select eMMC flash type,



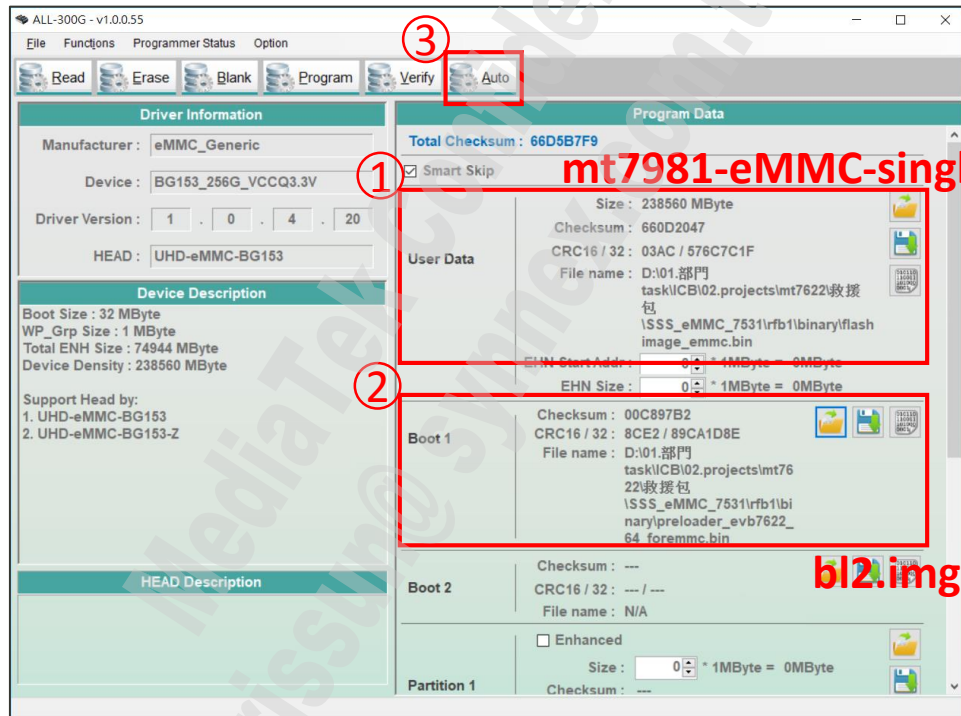
eMMC 4.5 → use BG153_256G_VCCQ3.3V

eMMC 5.1 → use BG153_256G_VCCQ1.8V

How to Program Single Image by Programmer

The example of ALL-300G programmer

- Load image,



How to Program Single Image by Programmer

The example of ALL-300G programmer

- Configure EXT_CSD register

Extended CSD register setting

☒ WR_REL_SET [167]

☒ Bit [0]: User Data ☒ Bit [1]: Partition 1 ☒ Bit [2]: Partition 2 ☒ Bit [3]: Partition 3 ☒ Bit [4]: Partition 4

☒ PARTITION_CONFIG [179] 0x 48

☐ BOOT_CONFIG_PROT [178] 0x 0

☒ BOOT_BUS_CONDITIONS [177] 0x 0

☐ BOOT_WP [173] 0x 0

☐ USER_WP [171] 0x 0

☐ FW_CONFIG [169] 0x 0

☐ BKOPS_EN [163] 0x 0

☒ RST_n_FUNCTION [162] 0x 1

☐ SEC_BAD_BLK_MGMNT [134] 0x 0

☐ EXT_PARTITIONS_ATTRIBUTE [53:52] 0x 0

PARTITION_SETTING_COMPLETED [155]

☒ Auto set PARTITION_SETTING_COMPLETED

☐ Manual set PARTITION_SETTING_COMPLETED = 0x 0

Production State Awareness (PSA)

PRE_LOADING_DATA_SIZE [25:22] 0x 0

PRODUCTION_STATE_AWARENESS [133] Disable

ext_csd[162] = 0x01
 ext_csd[167] = 0x1F
 ext_csd[177] = 0x00
 ext_csd[179] = 0x48

How to Program Single Image by Programmer

The example of ALL-300G programmer

- Start program,

Check

①

②

③

④

⑤

⑥

Auto

Manufacturer : eMMC_Generic

Device : BG153_256G_VCCQ3.3V

Total Checksum : 66D5B7F9

Items	Programmer																	Counter		
	Site#	Status	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Pass	Fail
<input checked="" type="checkbox"/> Short Test	0																		0	0
<input checked="" type="checkbox"/> Contact	1																		0	0
<input checked="" type="checkbox"/> ID Check	2																		0	0
<input checked="" type="checkbox"/> Erase	3																		0	0
<input type="checkbox"/> Blank	4																		0	0
<input checked="" type="checkbox"/> ExtCSD Setting	5																		0	0
<input checked="" type="checkbox"/> Program / Verify	6																		0	0
	7																		0	0

Total : 0 0

Reset Counter

Run Close

MediaTek Proprietary and Confidential

© 2021 MediaTek Inc. All rights reserved. The term “MediaTek” refers to MediaTek Inc. and/or its affiliates.

This document has been prepared solely for informational purposes. The content herein is made available to a restricted number of clients or partners, for internal use, pursuant to a license agreement or any other applicable agreement and subject to this notice. THIS DOCUMENT AND ANY ORAL INFORMATION PROVIDED BY MEDIATEK IN CONNECTION WITH THIS DOCUMENT (COLLECTIVELY THIS “DOCUMENT”), IF ANY, ARE PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE. MEDIATEK DOES NOT WARRANT OR MAKE ANY REPRESENTATIONS OR GUARANTEE REGARDING THE USE OR THE RESULT OF THE USE OF THIS DOCUMENT IN TERMS OF CORRECTNESS, ACCURACY, TIMELINESS, RELIABILITY, OR OTHERWISE. MEDIATEK SPECIFICALLY DISCLAIMS ALL WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTIES ARISING OUT OF COURSE OF PERFORMANCE, COURSE OF DEALING OR USAGE OF TRADE. This Document must be held in strict confidence and may not be communicated, reproduced, distributed or disclosed to any third party or to any other person, or being referred to publicly, in whole or in part at any time except with MediaTek’s prior written consent, which MediaTek reserves the right to deny for any reason. You agree to indemnify MediaTek for any loss or damages suffered by MediaTek for your unauthorized use or disclosure of this Document, in whole or in part. If you are not the intended recipient of this document, please delete and destroy all copies immediately.



MEDIATEK

everyday genius