Version 2.2



©DediProg Technology Co., Ltd. 2019 All rights reserved.



Table of Content:

| Ι. | Description | 2 |
|------|--|----|
| II. | Preparation before Setting | 2 |
| III. | From the Unique Key File | 6 |
| IV. | From the Serial Number | |
| V. | From the Serial Number for Multi-address | 15 |
| VI. | Revision History | 21 |

Important notice:

This document is provided as a guideline and must not be disclosed without consent of DediProg. However, no responsibility is assumed for errors that might appear.

DediProg reserves the right to make any changes to the product and/or the specification at any time without notice. No part of this document may be copied or reproduced in any form or by any means without prior written consent of DediProg.



I. Description

This application note illustrates the setting of Unique key in Dediware. There are three ways to program the key to the chips. Learn more about DediProg products and how to use them.

***** The Unique key only supported in the USB Mode (connect to PC) because the key is produced by the software (Dediware).

II. Preparation before Setting

The following figure shows the process of programming the Unique key:



Users can complete most of the settings in Engineering Mode. However, only Production Mode provides programming function.



Step 1: Select the chip and the programming file

First, start Dediware and select the chip's part number (Framed in blue) and load the programming file (Framed in green), and then you can find the **Unique** key icon in "Config" (Framed in red):



After setting the batch operation, click the "Unique key" icon (Framed in red):

| Config | | |
|--------|--|---------------------------|
| Batch | Batch Setting Batch Operation | Operation Selected |
| | Program chip Checksum verify Start Mode Start from Manual Mc | Erase chip Blank check |
| | | OK Cancel |

www.dediprog.com



Step 2: Unique key set up window

Click the **Unique** icon, the settings will show as below:

| Config | | × |
|----------|--|---|
| Batch | Unique Key & Serial Num | |
| | Enable program unique key to different chips | ^ |
| Unique | PartitionName: Flash V | |
| T | StartAddress: 0x 0 Length: 0x 0 | |
| | From Unique Key File | |
| | Sample key file: | |
| | ☐ Reuse the failed keys | |
| | | |
| | O From serial number | |
| | Byte Order | |
| | ☐ Big Endian(MS Byte first) | |
| | Enable roll serial number function | |
| | O From serial number for multi-address | |
| | Byte Order | |
| | ☐ Big Endian(MS Byte first) | |
| | Multi-Address | ~ |
| | OK Cancel | |



| Config | \times |
|---|----------|
| Batch Unique Key & Serial Num | |
| Byte Order Big Endian(MS Byte first) Image: Little Endian(LS Bytes first) step:0x 1 | ^ |
| Enable roll serial number function | - 1 |
| From serial number for multi-address Big Endian(MS Byte first) Little Endian(LS Bytes first) step:0x Multi-Address Address counter StartAddress: 0x Length: 0x StartAddress: 0 | ~ |
| OK Cancel | |
| | |

(Framed in red)

• Enable program unique key to different chips: The Unique key will be activated automatically in production mode.

(Framed in green)

- Partition Name: Assign the partition for programming.
- Start Address: Assign the start address for programming (Hex).
- Length: The length of Unique key.

(Framed in yellow)

- From Unique key file: Load the serial numbers file to program
 - Sample key file: Read the length of key after loading the file.
 - > Reuse the failed keys: Reuse the key that has failed.
- From serial number: Given random key from Dediware automatically
 - Byte Order: Select the order of the key numbers by Big Endian or Little Endian.
 - Step: The cumulative value of the serial number. The default is 1, ex:0000, 0001, 0002...etc.
 - Enable roll serial number function: If the number exceeds the setting range, then it will start from the first number again.
- From serial number for multi-address: Given random key from Dediware automatically
 - > Byte Order: Select the order of the key numbers by Big Endian or Little Endian.
 - Step: The cumulative value of the serial number. The default is 1, ex:0000, 0001, 0002...etc.
 - Multi-Address:

www.dediprog.com



- Address counter: Set up to 5 groups of Unique keys
- Start Address: Assign the start address for programming (Hex).
- Length: The length of Unique key
- Enable roll serial number function: If the number exceeds the setting range, then it will start from the first number again.
- From Others: This selection is only for specific customers. (We will not introduce this use in this note.)

III. From the Unique Key File

Step 1: Choose the File, the Partition, and the Address

After selecting "Enable program unique key to different chips", choose a **Sample key file**, the purpose is to determine the length of file:

| | Enable program unique key to different chips |
|---|--|
| | Partition Name: |
| | Start Address: 0x 0 Length: 0x 10 |
| | ● From Unique Key File Sample key file: C:\Users\DediProg\Desktop\資料\Application Report\Resources\unique key f ✓ Reuse the failed keys |
| | From serial number Byte Order |
| | Big Endian Ittle Endian Step:0x |
| 5 | Enable roll serial number function |
| | © From Others |

As the figure above, the Length of the key is shown by Dediware automatically. Choose the "Partition name" of the key that you want to program and set the "Start Address" (Hex).

"Reuse the failed keys" decides the location of the keys that have failed. If it is selected, then the failed key will be reused; if it is not selected, then the failed key will move to the "Failed" folder.

Click "OK" to save the settings.



Step 2: Save your project and run in Production Mode

Save the project (Framed in red) after step 1.

| | y Moue | | | | _ | - | |
|----------|-------------------------|--------------|-----------------|-------------|-------------------------|------------|----------|
| Select | PROJ Load Prj | Doad Load | Z Buffer | Ç Config | PROJ Save Prj | IC Info | |
| ReadID | keadIC | Slank | Erase | Program | Verify | Auto Batch | |
| ve Prj | | | | | | | ; |
| Location | DESKTOP | test project | | | | | 1 |
| | Name | | | | Туре | Size | |
| PC | test | project.dprj | | | File | 10240KB | |
| | | | | | | | |
| | | | | | | | |
| DESUTOR | 6 | | | | | | |
| DESKTOP | | | | | | | |
| DESK TOP | | | | | | | |
| DESK TOP | | | | | | | |
| DESK TOP | | | | | | | |
| DESK TOP | | | | | | | |
| DESK TOP | | | | | | | Ô. |
| DESK TOP | FileName: | tes | st_project.dprj | | | | <i>б</i> |



Please click **OK** after editing the password or the note; if not, please click "Skip".

| oject Password&Notepa | ad | | |
|---|--------------|---------|------|
| Password: | | | |
| Project notepad | | | |
| -build version: 3.7.14 -release version: Dediware -config version: 3.00 Project Info: -Type:SPI_NOR -PartNum: W25Q16CV -Manufact: Winbond | version: 3.7 | . 14. 1 | |
| | | ж | Skip |

After saving the project, please switch to Production Mode, select the project that you have just saved, and click "Run Prj" icon.

| SelectPrj 2 | RunPrj | StopPrj | | | |
|---------------------------------|-----------------|-----------------|-------------|-----------------|-----------|
| 3 ique Key Mo | de | | | | |
| PartitionName: StartAddress: | Flash 0x0 | | | | |
| Length: | 0x10 |] | | | |
| Reuse the | failed key file | | | | |
| KeyFolder: | C:\Users | \DediProg \Desl | ktop/資料\App | lication Report | Resources |
| | | | | ОК | Cancel |
| | | | | | |

The Key Length and the KeyFolder address must be the same as the Key file that you need to program, click "OK" to start programming.

| www.ded | iprog.com |
|---------|-----------|
|---------|-----------|



Step 3: Key Check in Engineering Mode

After programming, the icon will change to PASS as below.



Check the key that has just been programmed through "Read IC" in Engineering Mode.

| idvance Help | | |
|--|---|---|
| Engineering Mode | n de la contra de | |
| | Partition 1 Hash [1ype:Hash] | P |
| Select Load Prj Load | Partition 1 Partition 2 Partition 3 Partition 4 Partition 5 Buffer Cho | |
| 1 2 | Address +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F | |
| ReadID ReadIC Blank | 0x0000000000000000 FF FF FF FF FF FF FF FF | |
| | 0x00000000000010 FF | |
| | | _ |
| StartWode: By Project Site #1 | 0x0000000000000000 FF FF FF FF FF FF FF FF | ^ |
| F/W Ver: 2.1.79 Pass: 0 | 0x00000000000000 FF FF FF FF FF FF FF FF F | |
| S/N: PMU028181 Fail: 0 | 0x000000000000055 FF F | |
| BLink Start 🗘 N 0.0% | 0x00000000000000 FF FF FF FF FF FF FF FF F | |
| | 0x0000000000000070 FF | |
| | 0x000000000000000000000000000000000000 | |
| | 0x000000000000000 FF FF FF FF FF FF FF FF | |
| | 0x0000000000000 FF FF FF FF FF FF FF FF FF | |
| | 0x00000000000000 FF FF FF FF FF FF FF FF F | |
| | 0x0000000000000 FF FF FF FF FF FF FF FF FF | |
| | 0x000000000000 FF FF FF FF FF FF FF FF FF | |
| | 0x0000000000000 FF FF FF FF FF FF FF FF FF | |
| | | |
| | 0x0000000000000000000000000000000000000 | |
| | | |
| | | |
| | 0x00000000000130 FF | + |
| | Buffer Checksum 0x0000000 Chip Checksum 0x0000000 | |
| | Gata 0x 0000000 | |
| ChipInfo | | |
| Type: SPI_NOR ID: Manufact: Winbond ADP P/N1 | Save Memory From 0x 00000000 Length;0x 00200000 | |
| Size: 0x00200340 ADP P/N2 | | |
| Package: SOP8 208ml ADP P/N3- PartNum: W25Q16CV | IUG3: U VXV | |
| e | ProjectName:112233.dprj ProjectCheckSum:0xea3bb65b | |



As the figure above, it shows the key that has been programmed correctly and is identical to the original file (shown as below).

```
      Image: Second system
      01.bin

      Offset(h)
      00
      01
      02
      03
      04
      05
      06
      07
      08
      09
      0A
      0B
      0C
      0D
      0E
      0F

      000000000
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      11
      <
```

Remark:

1. Please arrange the order of the Key File names from the smallest to the largest. Ex:01, 02, 03...

2. Dediware will create three folders, including "Used", "Failed" and "uc-log".

"Used" is for the key that has passed.

"Failed" is for the key that has failed.

"uc-log" is for the log file.

3. Unique Key length must be in $0x00 \approx 0x100000$.



IV. From the Serial Number

Step 1: Choose the Partition and set up the Address and key length, and then set the Byte Order and the Steps.

Select "Enable program unique key to different chips"; set the starting address for programming serial number and the numbers' length, and then select "From serial number".

| nfig | × |
|---|---|
| Batch Unique Key & Serial Num | |
| Image: Start Address: 0x 50 Length: 0x 10 | |
| Semple key file: | I |
| From serial number Byte Order Big Endian Little Endian Step:0x Step:0x | |
| O From Others Project: Midea | |
| OK Cance | 2 |

The Byte Order is the direction of the serial number:

- "Big Endian" means the lowest byte list from the highest address bit;
- "Little Endian" means the highest byte list from the lowest address bit.
- "Step" represents the value between each byte (Hex).
- "Enable roll serial number function" means it will reuse the used keys when it is out of range. After finishing with the settings, please click "OK".



Step 2: Save your project and run in Production Mode

Remember to save the project (Framed in red) after step 1.

| Lingineering | g Mode | | | | - | 4 | |
|---------------|---------------|----------------|-----------------|---------|----------|-----------|-----------------------|
| Salart | PROJ | Losd | Buffer | (Q) | PROJ | | |
| | 2000 II) | 2.2 | . FF | | Save II) | | |
| ReadID | seadIC ReadIC | Slank - | Erase | Program | Verify | Auto Bate | h |
| ve Prj | | | | | | | |
| Location | DESKTOP | \test project\ | 1 | | | | ✓ 1 |
| | Name | | | | Гуре | 1 | Size |
| PC | test_ | project.dprj | | | File | 1024 | DKB |
| | | | | | | | 34 |
| DESKTOP | ŕ | | | | | | |
| DESKTOP | | | | | | | |
| DESKTOP | | | | | | | 1040 |
| DESKTOP | | | | | | | (00) |
| DESK TOP | | | | | | | (\V) |
| DESKTOP | | | | | | | ((W) |
| DESKTOP | FileName: | tes | st_project.dprj | | | | OK |



Please click **OK** after editing the password or the note; if not, please click "Skip".

| Project Password&Notepad | × |
|---|---|
| Password: Password confirm: Project potepad | |
| SoftWare Info: -build version: 3. 12. 73 -release version:Dediware Version: 3. 12. 73. 4 -config version: 3.00 Project Info: -Type:SPI_NOR -PartNum:MX25U25645G -Manufact:Macronix_MXIC File Info: -Chksum:0x7ec17e79 -Path: 128Mbit.bin Chip checksum area: -Flash, 8K bit Secured OTP, Operation Frequency: 25.0 MHz | ~ |
| ОК | |

After saving, please switch to Production Mode to select the project that you just saved, and then click "Run Prj".

| Production Mode Production Mode SelectPri Ru | nPrj StopPrj | | |
|---|---|-------------------|--------|
| Serial Number M PartitionName: StartAddress: Length: C Enable roll se | ode Flash 0x50 0x10 erial number function | Step: 0x1 | × |
| ByteOrder | O Little Edian | Range Radix | |
| Range Begin: | 0x000000000000000 | 00000000000001122 | |
| Range End: | 0x000000000000000 | 00000000000009999 | |
| | | ОК | Cancel |

There are three kinds of Range Radix: HEX, DEC and BCD. Select the one that you want and set the Range for the serial number. For example, as the figure above, the serial number that needs to be programmed is from

www.dediprog.com



Step 3: Key check in Engineering Mode

When programming succeeds, the icon will turn PASS (Shown as below figure).

| Engineers Mada | (a) Dready | ution Made | |
|---|--|---|---|
| | | | [F. |
| PROJ PRO PRO | ; 🐝 🗆 🖌 | · · · · · · · · · · · · · · · · · · · | Powered by |
| Select Load Prj Load Buffer Config Save I | rj IC Info Selecti | Prj RunPrj StopPrj | |
| - + | | | |
| | ↓ K | | |
| eadID ReadIC Blank Erase Program Verify | Auto Batch | | |
| | | | Log Window |
| ProgMasterU4 Site #1 Site #2 | Site #3 | Site #4 | 11:18:28:Programmer[01],chip[02] is blank |
| Vode: by Project Pass: 1 Pass: 1 Pass: 1 | Pass: 0 | Pass: 0 | 11:18:28:Programmer[01].chip[02] Erase |
| PMU028181 Fail: 0 PASS Fail: 0 | PASS Fail: 0 | Fait 0 | 11:18:28 Programmer[01] chip[01] is blank |
| K Stort 🖓 K 🗰 10028 | N 0.0% | N 0.0% | 11/18/28/Programmar(011 chip/011 Unique Key |
| | | | 11.18.28.0 |
| | | | 11.10.20.Programmer[01] chip[01] current S/N or Onique Key Istuduto. |
| | | | 11:18:28:Programmer[01],chip[02] Erase success |
| | | | U 11:18:28:Programmer[01],chip[02] Blank |
| | | | (i) 11:18:28:Programmer[01], chip[02] is blank |
| | | | 11:18:28:Programmer[01],chip[02] Unique Key |
| | | | |
| | | | U 11:18:28:Programmer[01] chip[02] Current S/N or Unique Key is:00010. |
| | | | 11:18:28:Programmer[01] chip[02] Current S/N or Unique Key is:00010. 11:18:29:Programmer[01],chip[01] Unique Key success |
| | | | 11:18:28:Programmer[01] chip[02] Current S/N or Unique Key is:00010. 11:18:29:Programmer[01].chip[01] Unique Key success 11:18:29:Programmer[01].chip[01] batch success |
| | | | (1) 11.18.28Programme(01] chip(02) Current \$\cong V\$\) or Unique Key is:00010 (1) 11.8.29Programme(01] chip(01) Unique Key success [⊗] 11.18.29Programme(01] chip(01) batch success takes 5.43s (1) 11.18.29Programme(01] chip(01) bitch success takes 5.43s |
| | | | (J) 1113/28/Porgamme(01]chp(01] Univer Key is 30000. (J) 1113/28/Porgamme(01]chp(01] Univer Key success [∞] 11.18/29/Porgamme(01]chp(01] Univer Key success [∞] 11.18/29/Porgamme(01]chp(02] Unique Key success [∞] 11.13/29/Porgamme(01]chp(02] Unique Key success |
| | | | U 1118:28 Programmer[01].thip[02] Current S/N or Unique Key iso0000. U 1118:29 Programmer[01].thip[01] Unique Key isoccess II18:29 Programmer[01].thip[02] Unique Key isoccess 1118:29 Programmer[01].thip[02] Unique Key isoccess II18:29 Programmer[01].thip[02] Unique Key isoccess 1118:29 Programmer[01].thip[02] Unique Key isoccess II18:29 Programmer[01].thip[02] Unique Key isoccess 1118:29 Programmer[01].thip[02] Unique Key isoccess |
| | | | 1118:28 Programmer[01] (http://2) Current S/N or Unique Key is 00010. [) 1118:29 Programmer[01] (http://1) Lands Assuccess takes 5.438 [) 1118:29 Programmer[01] (http://2) Unique Key isoccess [) 1118:29 Programmer[01] (http://2) batch success takes 5.439 [) 1118:29 Programmer[01] (http://2) batch success takes 5.439 [] Swreting [] Clearing |
| lafe | Statistics | Batch Canfin Setting | U 113228-rogamme(0](mp)(2) current 5/N or Ungue Key iso0010. () 113228-rogamme(0)(h)(h)(0)(1)(n)(u) Key success (*) 11329-Rogamme(0)(h)(0)(1) bath success takes 5.435 (*) 11329-Rogamme(0)(h)(0)(2) bath success takes 5.4295 (*) 11329-Rogamme(0)(h)(2) bath success takes 5.4295 (*) 11329-Rogamme(0)(h)(2) bath success takes 5.4295 (*) 11329-Rogamme(0)(h)(h)(2) bath success takes 5.4295 (*) 11329-Rogamme(0)(h)(h)(2) bath success takes 5.4295 (*) 11329-Rogamme(0)(h)(h)(2) bath success takes 5.4295 (*) 11329-Rogamme(0)(h)(h)(h)(h)(h)(h)(h)(h)(h)(h)(h)(h)(h) |
| 1010 501 (002 101 101 101 101 101 101 101 101 101 | Statistics | Batch Config Setting StarWode: Start from Hander | 0 1118289-regamme[0] (mp)2(2 Lorrent 5/N or Unique Key is 20000. 0 111829-regamme[0] (hp)(0)] black success takes 5.43s 0 111829-regamme[0] (hp)(0)] black success takes 5.42s Swetsg Clear tag Check Sum Els (hards) Op Els (hards) |
| Info Eff 40 15 15 SPL / UCR ACP PAIL: 57-127-007008-327/mil-01FE MCP PAIL: 57-127-007008-327/mil-01FE The second | Statistics Success: 2 | Batch Config Setting Surbook: Start from Hender Ersec tho | 0 1118/28-Pogramme(01[.chip01] dingle Key isc0000. 0 1118/29-Pogramme(01[.chip01] dingle Key success ** 1118/29-Pogramme(01[.chip02] batch success takes 5.43% (0) 1118/29-Pogramme(01[.chip02] batch success takes 5.42% ** 1118/29-Pogramme(01[.chip02] batch success takes 5.42% Section Clear tag Check Sum Fie Name Op Fie Name |
| Info Dr lef 40 15 5 5PT_MOR Dr lef 40 15 5 5PT_MOR AUP PAILS 5PT-127-00F008-307mH-0 JFE 5000003-00 AUP PAILS 5PT-127-00F008-307mH-0 JFE AUP PAILS 5PT-127-00F008-307mH-0 JFE | Statistics Success: 2 Foldure: 0 | Batch Config Setting Stateblock: Start from Handler Erase chip Bank check. | 0 1118:28 Programme[0] (http://j.current.St/Nort/Inage Kay is 200010. () 1118:29 Programme[0] (http://j.init/Unique Kay uscess 0 1118:29 Programme[0] (http://j.init/Unique Kay uscess () 1118:29 Programme[0] (http://j.init/Unique Kay uscess 0 1118:29 Programme[0] (http://j.init/Unique Kay uscess () () 1118:29 Programme[0] (http://j.init/Unique Kay uscess 0 1118:29 Programme[0] (http://j.init/Unique Kay uscess () () () 5 weit.og Clear.tog () () () () Check Sum Clear.tog Fle OredSum Fle Name () () 0 Option Fle OredSum Fle Name () () |

Next, check the key through "Read IC" in Engineering Mode. As the figure below, the window shows the key has been programmed correctly.

By the order of Little Endian:

| artition 1 Flash [Type:Fla | sh] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | × |
|--|--------|-----|----|-----|---------------|----------|-----|--------|--------|-----|------|-------|------|----|----|----|-----|------|------|----|-----|------|------|------|----|-----|------|------|-----|-----|------|------|---|
| Partition 1 O Pa | rtitio | n 2 | 0 | Par | tition Buf | 3 fer | ⊖ F | Partit | tion · | 4 (|) Pa | rtiti | on 5 | | | | | | | | | Chip | | | | | | | | | | | |
| Address | +0 | +1 | +2 | +3 | +4 | +5 | +6 | +7 | +8 | +9 | +A | +B | +C | +D | +E | +F | +0 | +1 | +2 | +3 | +4 | +5 | +6 | +7 | +8 | +9 | +A | +B | +C | +D | +E | +F | |
| 0x000000000000000000000000000000000000 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | |
| 0x000000000000000000000000000000000000 | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | |
| 0x0000000000000000020 | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | |
| 0x000000000000000000000000000000000000 | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | |
| 0x00000000000000040 | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | ۲ŀ | FF | FF | FF | FF | |
| 0x000000000000000000000000000000000000 | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | 22 | | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | |
| 0x00000000000000000 | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | -FF | - 77 | - 77 | - | -17 | - | - 11 | - 77 | - | -77 | - 11 | - 17 | -77 | -17 | - 66 | - 77 | |
| 0x000000000000000000000000000000000000 | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | |

By the order of Big Endian:

Partition 1 Flash [Type:Flash]

| | aao | | 0 | | Buff | fer | Ŭ. | unit | | | | | | | | | | | | | 3 | Chip | | | | | | | | | | |
|--|-----|----|----|----|------|-----|----|------|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|----|----|-----|----|----|
| Address | +0 | +1 | +2 | +3 | +4 | +5 | +6 | +7 | +8 | +9 | +A | +B | +C | +D | +E | +F | +0 | +1 | +2 | +3 | +4 | +5 | +6 | +7 | +8 | +9 | +A | +B | +C | +D | +E | +F |
| 0x000000000000000000000 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 |
| 0x0000000000000000 | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF |
| 0x00000000000000020 | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF |
| Dx00000000000000030 | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF |
| 0x00000000000000040 | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | HE. | FF | FF |
| x00000000000000050 | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 11 | 22 |
| 0x00000000000000000 | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF |
| 0x000000000000000000000000000000000000 | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF |

Remark:

1. Please make sure the "Start Address" is blank, since the serial number will overwrite the existing file.

2. Serial number length must be less than 0x10 (16 bytes).





V. From the Serial Number for Multi-address

Step 1: Choose the Partition

Select "Enable program unique key to different chips", and then select "From serial number for multi-address".

| Unique | Image: PartitionName: Flash | ^ |
|--------|--|---|
| | StartAddress: 0x DISABLE Length: 0x DISABLE | |
| NOR | O From Unique Key File Sample key file: | |
| | Reuse the failed keys | |
| | From serial number Byte Order Big Endian(MS Byte first) Ittle Endian(LS Bytes first) Step:0x | |
| | Enable roll serial number function | |
| | From serial number for multi-address Byte Order O Big Endian(MS Byte first) Multi-Address | |
| | OK Cancel | * |



| Config | × |
|---|----|
| Batch Unique Key & Serial Num | |
| Byte Oraer | ^ |
| ○ Big Endian(MS Byte first) | |
| Enable roll serial number function | |
| SPI NOR Order Byte Order | n |
| O Big Endian(MS Byte first) O Little Endian(LS Bytes first) step:0x | |
| Multi-Address | 4 |
| Address counter | |
| 1.Length: UX | |
| 2.StartAddress: 0x 2.Length: 0x | |
| 3.StartAddress: 0x 4 Leasth: 0x | |
| 4.StartAddress: 0x 4.Length: 0x | |
| 5.StartAddress: 0x 5.Length: 0x | |
| Enable roll serial number function | |
| | 1 |
| ○ From Others | |
| project Arrow_SN Customized DLL path: | |
| | ~ |
| OK Cance | el |

The Byte Order is the direction of the serial number:

- "Big Endian" means the lowest byte list from the highest address bit.
- "Little Endian" means the highest byte list from the lowest address bit.
- "Step" represents the value between each byte (Hex).
- "Multi-Address" means set the starting address for programming serial number and the numbers' length. (Set up to 5 groups of Unique keys.)
- "Enable roll serial number function" means it will reuse the used keys when it is out of range. After finishing with the settings, please click "OK".

Step 2: Save your project and run in Production Mode

Remember to save the project (Framed in red) after step 1.





| Location: | DESKTOP\test | project\ | | | ~ |
|-----------|--------------|----------|--------------|----|---------------|
| PC | Name | ct,dprj | Type File | 10 | Size 240KB |
| DESKTOP | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |



Please click **OK** after editing the password or the note; if not, please click "Skip".

| Project Password&Notepad | \times |
|---|----------|
| Password: Password confirm: Project notepad | |
| SoftWare Info: -build version: 3. 12. 73 -release version: Dediware Version: 3. 12. 73. 4 -config version: 3.00 Project Info: -Type:SPI_NOR -PartNum:MX25U25645G -Manufact:Macronix_MXIC File Info: -Chksum:0x7ec17e79 -Path: 128Mbit.bin Chip checksum area: -Flash, 8K bit Secured OTP, Oneration Errequency: 25.0 MHz | |
| OK | |

After saving, please switch to Production Mode to select the project that you just saved, and then click "Run Prj".



| Production Mode | | | | |
|---|---------------------|-------------|-------|---|
| $1 \square 2 $ | | | | |
| SelectPrj Run | Prj StopPrj | | | |
| Multi-Address Ser | rial Nymber Moo | de | | × |
| PartitionName: | Flash 3 | | | |
| Multi-Address | | | | |
| 1. StartAddress 1. Length: | : 0x1 0x2 | | | |
| 2. StartAddress | : 0x5 | | | |
| 2. Length: | 0x2 | | | |
| StartAddress Length: | : 0x10 0x2 | | |) |
| 4. StartAddress | : 0x15 | | | |
| 4. Length: | 0x2 | | | c |
| 5. StartAddress | : 0x20 | | | <u>c</u> |
| 5. Length: | UX2 | | | c |
| Enable roll ser | rial number functio | n Step: 0x1 | | |
| ByteOrder | Out a t | Range Radix | 0.000 | |
| Big Endian | Little Endian | • HEX ODEC | | nicode-16 |
| 1. Range Begin: | 0x1111 | | | = |
| 1. Range End: | 0x1112 | | | ¢ |
| | | | | s and a second se |
| 2. Range Begin: | 0x1113 | | | |
| 2. Range End: | 0x1114 | | | |
| 3. Range Begin: | 0x1115 | | | |
| 3. Range End: | 0x1116 | | | |
| 4. Range Begin: | 0x1117 | | | la |
| 4 Range End | 0x1118 | | | 1 |
| 4. Kange End: | | | | |
| 5. Range Begin: | 0x1119 | | | |
| 5. Range End: | 0x111A | | | |
| L | | | | (|
| | | | OK | Cancel |

There are three kinds of Range Radix: HEX, DEC, BCD and Unicode-16. Select the one that you need and set the Range for the serial number. Click "OK" to start programming.

Step 3: Key check in Engineering Mode

When programming succeeded, the icon will turn PASS (Shown as the below figure).



| Engineer Mode Bett Load Buffer Couling Bever hij Lindo Revall Revall Enwer Program Versty Auto Batch State Revall Elawa Enwer Program Versty Auto Batch State Revall Site #1 PASS Site #2 PASS Site #2 Pass: 1 Pass: 0 Pass: 0 Site #3 Shr PHAU202819 Fail: 0 PASS Site #2 PASS Site #3 | © Production Mode Educiting Runking Balance Runking Balance Runking Balance Runking Balance Runking Runking Runking | StopPy ite #4 ass: 0 ast: 0 COS | Log Window Image: State St |
|--|--|---|---|
| Beket Load Pri Load Pri Duffer Couling Bow Pri U ResultD Essettic Blank Enne Program Ventry Auto Betch. 601 PropOlasteridi Blank Enne Program Ventry Auto Betch. 601 PropOlasteridi Blank Enne Program Ventry Auto Betch. 601 PropOlasteridi Site #1 Pross 1 Pross 1 Pross 1 Rate 0 PASS Ventry Auto Betch. | Zelectrig Rudrig | Stopfrj | Interfect by Interfect by </th |
| Redic 2 <th></th> <th>ite #4 ass: 0 at 0 505</th> <th>Iog Widow ① 11:18:28 Programmer[01],chip[02] is blank ① 11:18:28 Programmer[01],chip[02] Esse ① 11:18:28 Programmer[01],chip[01] is blank ① 11:18:28 Programmer[01],chip[02] is blank ① 11:18:28 Programmer[01],chip[02] is blank ① 11:18:28 Programmer[01],chip[02] Esse success ① 11:18:28 Programmer[01],chip[02] is blank ① 11:18:28 Programmer[01],chip[02] is blank</th> | | ite #4 ass: 0 at 0 505 | Iog Widow ① 11:18:28 Programmer[01],chip[02] is blank ① 11:18:28 Programmer[01],chip[02] Esse ① 11:18:28 Programmer[01],chip[01] is blank ① 11:18:28 Programmer[01],chip[02] is blank ① 11:18:28 Programmer[01],chip[02] is blank ① 11:18:28 Programmer[01],chip[02] Esse success ① 11:18:28 Programmer[01],chip[02] is blank ① 11:18:28 Programmer[01],chip[02] is blank |
| D1 ProgMastel ¹⁴ OHded: By Project Ver: 2177 Fait: 0 Fait: 0 K K K K K K K K K K K K K | DDSF | ite #4 Jass: 0 ait: 0 20% | Log Wirdew ① 11.18.28.Programmer[01].chip[02] is blank. ① 11.18.28.Programmer[01].chip[02] crass success ① 11.18.28.Programmer[01].chip[02] crass success ① 11.18.28.Programmer[01].chip[02] is blank. ① 11.18.28.Programmer[01].chip[02] is blank. |
| Il Proglassett ⁴ Site #1 Pass 1 Pass 1 PASS 1 PASS 1 PASS 1 Past 0 Past 0 | | iite #4 Jass: 0 ait: 0 | (1) 11.82.8Programme(011.chip(02) is blank (1) 11.82.8Programme(011.chip(02) is blank (1) 11.82.8Programme(011.chip(02) is blank (1) 11.82.8Programme(011.chip(02) is blank (1) 11.82.8Programme(011.chip(02) Fase success (1) 11.82.8Programme(011.chip(02) Fase success (1) 11.82.8Programme(011.chip(02) Blank (1) 11.82.8Programme(011.chip(011.chip(02) Blank (1) 11.82.8Programme(011.chip(011 |
| Yet: 21.75 PAUSS 1 PASS 1 PASS Past 1 PASS Past 1 PASS PAUSS 1 PAUSS 1 PASS Past 0 PASS | | ass: 0 ait: 0 | I 11.8.2.8.Programmer[01].rhip[02] Fess I 11.8.2.8.Programmer[01].rhip[01] is blank I 11.8.2.8.Programmer[01].rhip[01] Unique Key I 11.8.2.8.Programmer[01].rhip[02] Erses success I 11.8.2.8.Programmer[01].rhip[02] Erses |
| | 007 N | 0.0% | (1) 11.8.2.8.Programme(011-rhip(01) is blank (1) 11.8.2.8.Programme(011,rhip(01) unique Key (1) 11.8.2.8.Programme(011,rhip(02) Erase success (1) 11.8.2.8.Programme(011,rhip(02) Erase success (1) 11.8.2.8.Programme(011,rhip(02) Blank (1) 11.8.2.8.Programme(011,rhip(02) Blank |
| | N | 004 | 11:18:28 Programmer[01], chip[01] Unique Kay 11:18:28 Programmer[01], chip[01] Current 5/N or Unique Kay is:0 11:18:28 Programmer[01], chip[02] Erses success 11:18:28 Programmer[01], chip[02] Blank 11:18:28 Programmer[01], chip[02] shank |
| | | | ① 11:18:28-Programmer[01] chip[01] Cument S/N or Unique Key is:0 ③ 11:18:28-Programmer[01],chip[02] Erste success ③ 11:18:28-Programmer[01],chip[02] Blank ④ 11:18:28-Programmer[01],chip[02] blank |
| | | | 11:18:28:Programmer[01].chip[02] Erase success 11:18:28:Programmer[01].chip[02] Blank 11:18:28:Programmer[01].chip[02] is blank |
| | | | 11:18:28:Programmer[01],chip[02] Blank 11:18:28:Programmer[01],chip[02] is blank |
| | | | 11:18:28:Programmer[01],chip[02] is blank |
| | | | G |
| | | | 11:18:28:Programmer[01] chin[02] Unique Key |
| | | | 11:18:28 Deserver mear(01) ship (02) Current 5 (b) as Unique Keylin |
| | | | 11.18.20.Programmer[01] chip[02] Current SNV or Orinque Key IS.0 |
| | | | C 11.18.29.Programmer[01], cmp[01] onique key success |
| | | | 11:18:29:Programmer[01], Enip[01] Batch success, takes 5:43s 41:18:29:Programmer[01], Enip[01] Batch success, takes 5:43s |
| | | | 11:18:29:Programmer[01],chip[02] Unique Key success |
| | | | TT.16.29.Programmer[OT];cnip[O2] batch success;takes 5.429s |
| | | | Save Log Clear Log |
| Info Statistics | Batch Con | fig Setting | Check Sum |
| SPI_NOR ID: ef 40 15 Success: | 2 StartMode: 1 | Start from Handler | Chip File CheckSum File Name |
| Tact: Winbond ADP P/N1: SP1-127-SOP008-207mil-01FE Failure: | 0 Erase chip Black check | | Option |
| ige: SOP8 208ml ADP P/N3: Total: | 2 | | 0x0 |



By the order of Little Endian:

| Partition 1 Flash [Type:Fla | sh] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | × |
|--|--------|-----|----|------|---------------|------------|-----|--------|-------|-----|------|--------|------|----|----|----|----|------|------|----|------|------|------|----|----|----|------|----|----|------|----|----|---|
| Partition 1 O Pa | rtitio | n 2 | 0 | Part | tition Buf | i 3 fer | ⊖ F | Partit | ion 4 | 4 (|) Pa | artiti | on 5 | | | | | | | | 3 | Chip | | | | | | | | | | | |
| Address | +0 | +1 | +2 | +3 | +4 | +5 | +6 | +7 | +8 | +9 | +A | +B | +C | +D | +E | +F | +0 | +1 | +2 | +3 | +4 | +5 | +6 | +7 | +8 | +9 | +A | +B | +C | +D | +E | +F | |
| 0x000000000000000000000000000000000000 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | |
| 0x000000000000000000000000000000000000 | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | |
| 0x0000000000000000020 | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | |
| 0x000000000000000000000000000000000000 | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | |
| 0x00000000000000040 | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | ۲ŀ | FF | FF | FF | FF | FF | FF | FF | FF | FF | |
| 0x0000000000000000050 | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | 22 | -11 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | |
| 0x000000000000000000000000000000000000 | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | | - 77 | - 77 | | - 11 | - | - 11 | - | - | | - 11 | | | - 11 | - | - | |
| 0x000000000000000000 | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | |

By the order of Big Endian:

| Partition 1 Flash [Type:Flas | sh] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | \times |
|--|--------|-----|----|------|---------------|------------|----|--------|--------|-----|------|-------|------|----|----|----|----|----|----|----|----|------|----|----|----|----|----|----|----|----|----|----|----------|
| Partition 1 O Pa | rtitia | n 2 | 0 | Part | tition Buf | i 3 fer | OF | Partit | tion · | 4 (|) Pa | rtiti | on 5 | | | | | | | | 3 | Chip | | | | | | | | | | | |
| Address | +0 | +1 | +2 | +3 | +4 | +5 | +6 | +7 | +8 | +9 | +A | +B | +C | +D | +E | +F | +0 | +1 | +2 | +3 | +4 | +5 | +6 | +7 | +8 | +9 | +A | +B | +C | +D | +E | +F | |
| 0x000000000000000000000000000000000000 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | |
| 0x000000000000000000000000000000000000 | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | |
| 0x000000000000000000000000000000000000 | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | |
| 0x000000000000000000000000000000000000 | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | |
| 0x00000000000000040 | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | ۲ŀ | FF | FF | FF | FF | ۲ŀ | FF | FF | FF | FF | |
| 0x00000000000000000 | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 11 | 22 | |
| 0x00000000000000000 | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | |
| 0x00000000000000000070 | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | |

Remark:

1. Please make sure the "Start Address" is blank, since the serial number will overwrite the existing file.

2. Serial number length must be less than 0x10 (16 bytes).



VI. Revision History

| Date | Version | Changes |
|------------|---------|--|
| MM/DD/YYYY | 1.0 | Initial release. |
| 01/30/2016 | 2.0 | Re-edit, and add more figure. |
| 08/11/2016 | 2.1 | Renew some images and add more remarks. |
| 06/12/2019 | 2.2 | Add "From serial number for multi-address" feature |

DediProg Technology Co., Ltd.

 Taiwan Headquarter
 TEL: 886-2-2790-7932
 FAX: 886-2-2790-7916

 4F., No.7, Ln. 143, Xinming Rd., Neihu Dist., Taipei City 114, Taiwan

China Office TEL: 86-21-5160-0157

Room 518, Building 66, Lane1333, Xinlong Road, Vanke Hongqiao CBD.Min Hang District, Shanghai, P.R.C. 201101

U. S. Office TEL: 1-909-274-8860

209 E Baseline RD, Suite E208 #8, Tempe, AZ, 85283, USA

Technical Support: support@dediprog.com Sales Support: sales@dediprog.com

Information furnished is believed to be accurate and reliable. However, DediProg assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties which may result from its use. Specifications mentioned in this publication are subject to change without notice.

This publication supersedes and replaces all information previously supplied.

All rights reserved Printed in Taiwan.