

Guide to Add New VCM ＆ EEPROM module Into DDK

Updated: 2017-08-27

Version: 0.1

Legal Notices and Disclaimers

**Disclaimers**

**INTEL CORPORATION MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. INTEL CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT. INTEL CORPORATION MAKES NO COMMITMENT TO UPDATE NOR TO KEEP CURRENT THE INFORMATION CONTAINED IN THIS DOCUMENT.**

**THIS SPECIFICATION IS COPYRIGHTED BY AND SHALL REMAIN THE PROPERTY OF INTEL CORPORATION. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED HEREIN.**

**INTEL DISCLAIMS ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF ANY PROPRIETARY RIGHTS, RELATING TO IMPLEMENTATION OF INFORMATION IN THIS SPECIFICATION. INTEL DOES NOT WARRANT OR REPRESENT THAT SUCH IMPLEMENTATIONS WILL NOT INFRINGE SUCH RIGHTS.**

**NO PART OF THIS DOCUMENT MAY BE COPIED OR REPRODUCED IN ANY FORM OR BY ANY MEANS WITHOUT PRIOR WRITTEN CONSENT OF INTEL CORPORATION.**

**INTEL CORPORATION RETAINS THE RIGHT TO MAKE CHANGES TO THESE SPECIFICATIONS AT ANY TIME, WITHOUT NOTICE.**

**Legal Notices**

Intel software products are copyrighted by and shall remain the property of Intel Corporation. Use, duplication or disclosure is subject to restrictions stated in Intel’s Software License Agreement, or in the case of software delivered to the government, in accordance with the software license agreement as defined in FAR 52.227-7013.

The Intel logo is a registered trademark of Intel Corporation.

Other brands and names are the property of their respective owners.

Revision History

| Revision | Date | Author | Reason for Changes |
| --- | --- | --- | --- |
| 0.1 | 10/19/2017 | Xiaochun Shi | Initial draft |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[VCM module (take DW9807 for example) 2](#_Toc491702267)

[In Vcm\_type.h 2](#_Toc491702268)

[In Vcm\_map.c 2](#_Toc491702269)

[EEPROM module (take M24C64 for example) 4](#_Toc491702270)

[In nvram\_type.h 4](#_Toc491702271)

[In nvram\_map.c 4](#_Toc491702272)

# VCM module (take DW9807 for example)

## In Vcm\_type.h

Add new module name define:

#define DW9807\_NAME "DW9807"

In enum VCM\_TYPE:

Replace VCM\_ex1 to a real module name like VCM\_DW9807;

## In Vcm\_map.c

1. #include "vcm\_dw9807.h"
2. Add callback functions in “Vcm[VCM\_NUM]”, replace below lines:

// below are function callback holders

{ VCM\_ex1, NONVCM\_NAME, 0, NULL, NULL, NULL, NULL, NULL, NULL, NULL }, { VCM\_ex2, NONVCM\_NAME, 0, NULL, NULL, NULL, NULL, NULL, NULL, NULL },

to

{ VCM\_DW9807, DW9807\_NAME, DW9807\_DEFAULT\_POS, DW9807\_Init, DW9807\_ResetPos, DW9807\_SetPos, DW9807\_GetPos, DW9807\_GetStatus, DW9807\_GetHPStatus, DW9807\_SetConfig },

{ VCM\_ex2, NONVCM\_NAME, 0, NULL, NULL, NULL, NULL, NULL, NULL, NULL },

1. In NTSTATUS ParseVcm(), add codes about dw9807:

NTSTATUS ParseVcm(

)

{

ANSI\_STRING ad5823;

......

ANSI\_STRING dw9807;

ANSI\_STRING name;

if (!pSsVcm)

{

return STATUS\_INVALID\_PARAMETER;

}

RtlInitAnsiString(&ad5823, AD5823\_NAME);

......

RtlInitAnsiString(&dw9807, DW9807\_NAME);

RtlInitAnsiString(&name, pVcmName);

if(RtlEqualString(&ad5823, &name, TRUE))

pSsVcm->VcmType = VCM\_AD5823;

......

else if (RtlEqualString(&dw9807, &name, TRUE))

pSsVcm->VcmType = VCM\_DW9807;

else

pSsVcm->VcmType = VCM\_NONE;

DoTraceMessage(FLAG\_LOG, "%s SensorCtx->VcmType:%d", DEVICE\_NAME, pSsVcm->VcmType);

return STATUS\_SUCCESS;

}

1. Add DW9807 source file and header file under CameraDDK\Sensor\Common\libiss\src\vcm

Vcm\_dw9807.c & vcm\_dw9807.h provided as template to implement VCM control functions

1. There are some functions might help you to write a new module, which is included in vcm.h.

# EEPROM module (take M24C64 for example)

## In nvram\_type.h

1. Add new module define in enum NVM\_TYPE, by replace “NVM\_ex1”.

// place holder, change to real name when needed

NVM\_M24C64,

## In nvram\_map.c

1. include header file of the module

#include "nvm\_M24C64S.h"

1. change NVM\_EX1 to a real eeprom type.

NVM\_TYPE NvmType[] =

{

NVM\_NONE,

NVM\_OTP,

NVM\_EEPROM,

......

// NVM module place holder

NVM\_M24C64,

NVM\_EX2,

NVM\_EX3,

NVM\_EX4,

NVM\_EX5,

};

1. In “Nvm[NVM\_NUM” add one line like below:

NVM\_FUNC Nvm[NVM\_NUM] =

{

......

// below are function callback holders, please add module above this line for inside.

{ NVM\_M24C64, 8192, 32, M24C64\_ADDR\_MIN, Cmd\_M24c64Write, Cmd\_M24c64Read, GetNvmData },

}

1. Add M24C64 source and header file under cameraDDK\Sensor\Common\libiss\src\nvm

nvm\_M24C64.c & nvm\_m24c64.h are provided as template to implement VCM control functions

1. There are some functions might help you to write a new module, which is included in nvram.h.