



Read Chip Uuid SOP



© 2020 SigmaStar Technology Corp. All rights reserved.

SigmaStar Technology makes no representations or warranties including, for example but not limited to, warranties of merchantability, fitness for a particular purpose, non-infringement of any intellectual property right or the accuracy or completeness of this document, and reserves the right to make changes without further notice to any products herein to improve reliability, function or design. No responsibility is assumed by SigmaStar Technology arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights, nor the rights of others.

SigmaStar is a trademark of SigmaStar Technology Corp. Other trademarks or names herein are only for identification purposes only and owned by their respective owners.



REVISION HISTORY

Revision No.	Description	Date
0.1	<ul style="list-style-type: none">Initial release	02/20/2020
	<ul style="list-style-type: none">	
	<ul style="list-style-type: none">	



{SigmaStar Display}
{Smart Display}
{SSD201 + 1.0}

1. SDK MI_SYS_READUUID API 读取

读取 demo 如下：

```
#include <stdio.h>
#include "mi_sys.h"
#include "mi_common_datatype.h"

int main(void)
{
    MI_U64 u64Uuid;
    MI_S32 s32Ret = MI_ERR_SYS_FAILED;

    s32Ret = MI_SYS_ReadUuid (&u64Uuid);
    if(!s32Ret)
    {
        printf("uuid: %llx\n",u64Uuid);
    }

    return 0;
}
```

2. APP 直接通过 REG 读取

读取 Demo 如下:

```
#include <fcntl.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/mman.h>

#define DBG_INFO printf
#define DBG_ERR printf

#define BIT8      0x100

#define BANK_TO_ADDR32(b) (b<<9)
#define REG_ADDR(riu_base,bank,reg_offset) ((riu_base)+BANK_TO_ADDR32(bank)+(reg_offset*4))

typedef struct
{
    unsigned char *virt_addr;
    unsigned char *mmap_base;
    unsigned int mmap_length;
}MmapHandle;

static unsigned int const page_size_mask = 0xFFF;

MmapHandle* devMemMMap(unsigned int phys_addr, unsigned int length)
{
    int fd;
    unsigned int phys_offset;

    fd = open("/dev/mem", O_RDWR|O_SYNC);
    if (fd == -1)
    {
        DBG_ERR("open /dev/mem fail\n");
        return NULL;
    }
}
```

```
MmapHandle *handle = malloc(sizeof(MmapHandle));
phys_offset =(phys_addr & (page_size_mask));
phys_addr &= ~(page_size_mask);
handle->mmap_length = length + phys_offset;
handle->mmap_base = mmap(NULL, handle->mmap_length , PROT_READ|PROT_WRITE, MAP_SHARED, fd,
phys_addr);
handle->virt_addr = handle->mmap_base + phys_offset;
DBG_INFO("phys_addr: %#x\n", phys_addr);
DBG_INFO("virt_addr: %p\n", handle->virt_addr);
DBG_INFO("phys_offset: %#x\n", phys_offset);

if (handle->mmap_base == MAP_FAILED)
{
    DBG_ERR("mmap fail\n");
    close(fd);
    free(handle);
    return NULL;
}

close(fd);
return handle;
}

int devMemUmap(MmapHandle* handle)
{
    int ret = 0;

    ret = munmap(handle->mmap_base, handle->mmap_length);
    if(ret != 0)
    {
        printf("munmap fail\n");
        return ret;
    }
    free(handle);
    return ret;
}
```



```
int main()
{
    unsigned long long uuid;
    /* RIU mapping*/
    MmapHandle *riu_base = devMemMMap(0x1F000000, 0x2B0000);

    /*Configure PAD and Clock here*/

    //chg default dev2,3 pclk from d4 to gpio2
    *(unsigned short*)REG_ADDR(riu_base->virt_addr, 0x20, 0x03) &= ~BIT8;
    uuid = (unsigned long long )(*(unsigned short*)REG_ADDR(riu_base->virt_addr, 0x20, 0x16)) |
        ((unsigned long long )*(unsigned short*)REG_ADDR(riu_base->virt_addr, 0x20, 0x17)) << 16) |
        ((unsigned long long )*(unsigned short*)REG_ADDR(riu_base->virt_addr, 0x20, 0x18)) << 32);

    devMemUmap(riu_base);
    printf("uuid: %llx\n",uuid);

    return 0;
}
```

3. 临时通过 DEBUG 终端读取

读取方法如下：

1. 读取 bank20 offset30 的值，然后把 bit8 清 0

```
/ # /config/riu_r 20 3  
BANK:0x20 16bit-offset 0x03  
0x0000  
/ # /config/riu_w 20 3 0  
BANK:0x20 16bit-offset 0x03  
0x0000  
/ # █
```

2. 依次通过读取 bank20 offset 16,17,18 获取 uuid 的值
(16: uuid bit0~bit15,17: uuid bit16~bit31,18: uuid bit32~bit47)
如下面读出来 uuid 的值为 0x5b3aff222f5

```
/ #  
/ # /config/riu_r 20 16  
BANK:0x20 16bit-offset 0x16  
0x22F5  
/ # /config/riu_r 20 17  
BANK:0x20 16bit-offset 0x17  
0xAFF2  
/ # /config/riu_r 20 18  
BANK:0x20 16bit-offset 0x18  
0x05B3
```