

**ITE SDK**

**NOR模組開發指南**

**V0.9**

**ITE TECH. INC.**

修訂記錄

修訂日期	修訂說明	頁次
2014/10/31	初建版本 V0.9	

目錄

<b>1.</b>	<b>前言</b> .....	<b>1</b>
1.1	編寫目的.....	1
1.2	適用範圍.....	1
1.3	適用人員.....	1
<b>2.</b>	<b>NOR模組介紹</b> .....	<b>2</b>
2.1	NOR支援列表 .....	2
2.2	如何支援新型號之NOR FLASH .....	3

## 1. 前言

### 1.1 編寫目的

介紹如何自行新增支援不同型號的NOR Flash。

### 1.2 適用範圍

ITE SOC可支援標準序列 (serial) SPI協議之NOR Flash，NOR Flash用來儲存程式碼和使用者資料。

### 1.3 適用人員

軟體應用程式，驅動程式開發者

## 2. NOR模組介紹

### 2.1 NOR支援列表

下面所列為目前ITE SOC可支援的NOR flash型號：

AMIC\_A25L032,  
AMIC\_A25LQ32A,  
ATMEL\_AT26DF161,  
ATMEL\_AT26D321,  
EON\_EN25P32,  
EON\_EN25B16,  
EON\_EN25B32,  
EON\_EN25B64,  
EON\_EN25F16,  
EON\_EN25F32,  
EON\_EN25Q16,  
EON\_EN25Q32A,  
EON\_EN25F80,  
EON\_EN25Q80A,  
EON\_EN25Q64,  
EON\_EN25QH32,  
ES\_ES25M16A,  
ESMT\_F25L16A,  
ESMT\_F25L32Q,  
ESMT\_F25L32PA,  
Micron\_N25Q032A,  
MX\_25L1605A,  
MX\_25L3205D,  
MX\_25L3235D,  
MX\_25L1635D,  
MX\_25L6445E,  
MX\_25L12835F,  
MX\_25L25635F,  
MX\_25L25735F,  
GD\_GD25Q64B,  
GD\_GD25Q32,  
GD\_GD25Q16,  
NUMON\_M25P20,  
NUMON\_M25P32,  
PMC\_PM25LQ032C,  
SPAN\_S25FL016A,  
SPAN\_S25FL032A,

```
SST_25VF016B,  
WIN_W25X16A,  
WIN_W25X32V,  
WIN_W25Q32BV,  
WIN_W25Q64BV,
```

## 2.2 如何支援新型號之NOR Flash

使用者如果要新增新型號之NOR flash，可以按照下面**紅色粗體字**相關步驟完成，相關程式碼請參考：  
sdk\driver\nor\nor.c

### 2.2.1 新增NOR ID

```
typedef enum  
{  
    AMIC_A25L032,  
    AMIC_A25LQ32A,  
    ATMEL_AT26DF161,  
    ATMEL_AT26D321,  
    EON_EN25P32,  
    EON_EN25B16,  
    EON_EN25B32,  
    EON_EN25B64,  
    EON_EN25F16,  
    EON_EN25F32,  
    EON_EN25Q16,  
    EON_EN25Q32A,  
    EON_EN25F80,  
    EON_EN25Q80A,  
    EON_EN25Q64,  
    EON_EN25QH32,  
    ES_ES25M16A,  
    ESMT_F25L16A,  
    ESMT_F25L32Q,  
    ESMT_F25L32PA,  
    Micron_N25Q032A,  
    MX_25L1605A,  
    MX_25L3205D,  
    MX_25L3235D,  
    MX_25L1635D,  
    MX_25L6445E,  
    MX_25L12835F,  
    MX_25L25635F,  
    MX_25L25735F,
```

```

GD_GD25Q64B,
GD_GD25Q32,
GD_GD25Q16,
NUMON_M25P20,
NUMON_M25P32,
PMC_PM25LQ032C,
SPAN_S25FL016A,
SPAN_S25FL032A,
SST_25VF016B,
WIN_W25X16A,
WIN_W25X32V,
WIN_W25Q32BV,
WIN_W25Q64BV,
NEW_NOR_ENUM_ID,
UNKNOWN_VENDOR = 0xFFFF
}NOR_VENDOR_ID;

```

## 2.2.2 編輯對應之Manufacturer ID及Deveice ID

```

NOR_VENDOR_CONTEXT nor_support_vendor[] = {
    {0x37, 0x3016, 0x15, "AMIC_A25L032", AMIC_A25L032},
    {0x37, 0x4016, 0x15, "AMIC_A25LQ32A", AMIC_A25LQ32A},
    {0x1F, 0x4600, 0xFF, "ATMEL_AT26DF161", ATMEL_AT26DF161},
    {0x1F, 0x4700, 0xFF, "ATMEL_AT26D321", ATMEL_AT26D321},
    {0x1C, 0x2016, 0x15, "EON_EN25P32", EON_EN25P32},
    {0x1C, 0x2015, 0x34, "EON_EN25B16", EON_EN25B16},
    {0x1C, 0x2016, 0x35, "EON_EN25B32", EON_EN25B32},
    {0x1C, 0x2017, 0x36, "EON_EN25B64", EON_EN25B64},
    {0x1C, 0x3115, 0x14, "EON_EN25F16", EON_EN25F16},
    {0x1C, 0x3116, 0x15, "EON_EN25F32", EON_EN25F32},
    {0x1C, 0x3015, 0x14, "EON_EN25Q16", EON_EN25Q16},
    {0x1C, 0x3016, 0x15, "EON_EN25Q32A", EON_EN25Q32A},
    {0x1C, 0x3114, 0x13, "EON_EN25F80", EON_EN25F80},
    {0x1C, 0x3014, 0x13, "EON_EN25Q80A", EON_EN25Q80A},
    {0x1C, 0x3017, 0x16, "EON_EN25Q64", EON_EN25Q64},
    {0x1C, 0x7016, 0x15, "EON_EN25QH32", EON_EN25QH32},
    {0x4A, 0x3215, 0x14, "ES_ES25M16A", ES_ES25M16A},
    {0x8C, 0x2015, 0x14, "ESMT_F25L16PA", ESMT_F25L16PA},
    {0x8C, 0x4116, 0x15, "ESMT_F25L32Q", ESMT_F25L32Q},
    {0x8C, 0x2016, 0x15, "ESMT_F25L32PA", ESMT_F25L32PA},
    {0x20, 0xBA16, 0xFF, "Micron_N25Q032A", Micron_N25Q032A},
    {0xC2, 0x2015, 0x14, "MX_25L1605A", MX_25L1605A},
    {0xC2, 0x2016, 0x15, "MX_25L3205D", MX_25L3205D},
    {0xC2, 0x5E16, 0x5E, "MX_25L3235D", MX_25L3235D},
    {0xC2, 0x2415, 0x24, "MX_25L1635D", MX_25L1635D},

```

```
{0xC2, 0x2017, 0x16, "MX_25L6445E", MX_25L6445E},
{0xC2, 0x2018, 0x17, "MX_25L12835F", MX_25L12835F},
{0xC2, 0x2019, 0x18, "MX_25L25635F", MX_25L25635F},
{0xC2, 0x2019, 0x18, "MX_25L25735F", MX_25L25735F},
{0xC8, 0x4017, 0x16, "GD_GD25Q64B", GD_GD25Q64B},
{0xC8, 0x4016, 0x15, "GD_GD25Q32", GD_GD25Q32},
{0xC8, 0x4015, 0x14, "GD_GD25Q16", GD_GD25Q16},
{0x20, 0x2012, 0xFF, "NUMON_M25P20", NUMON_M25P20},
{0x20, 0x2016, 0xFF, "NUMON_M25P32", NUMON_M25P32},
{0x7F, 0x9D46, 0x15, "PMC_PM25LQ032C", PMC_PM25LQ032C},
{0x01, 0x0214, 0xFF, "SPAN_S25FL016A", SPAN_S25FL016A},
{0x01, 0x0215, 0xFF, "SPAN_S25FL032A", SPAN_S25FL032A},
{0xBF, 0x2541, 0x41, "SST_25VF016B", SST_25VF016B},
{0xEF, 0x3015, 0x14, "WIN_W25X16A", WIN_W25X16A},
{0xEF, 0x3016, 0x15, "WIN_W25X32V", WIN_W25X32V},
{0xEF, 0x4016, 0x15, "WIN_W25Q32BV", WIN_W25Q32BV},
{0xEF, 0x4017, 0x16, "WIN_W25Q64BV", WIN_W25Q64BV}
{0xFF, 0xFFFF, 0xFF, "NEW_NOR_DISPLAY_NAME", NEW_NOR_ENUM_ID }
};
```

### 2.2.3 設定相關參數

最後根據要支援之NOR flash datasheet修改nor.c的norGetContext函式中page size、sector size、block size等相關參數

相關程式碼如下：

```
static void
norGetContext(NOR_OBJECT* norObject,
              NOR_VENDOR_CONTEXT* context)
{
    switch(context->vendorID) {
        case NUMON_M25P20:
            g_norAddrMap.bytesPerPage = 256;
            g_norAddrMap.pagesPerSector = 256;
            g_norAddrMap.bytesPerSector = 64*1024;
            g_norAddrMap.sectorsPerBlock = 1;
            g_norAddrMap.totalBlocks = 4;
            break;

        case ESMT_F25L16A:
            g_norAddrMap.bytesPerPage = 256;
            g_norAddrMap.pagesPerSector = 256;
            g_norAddrMap.bytesPerSector = 64*1024;
            g_norAddrMap.sectorsPerBlock = 1;
```



```

g_norAddrMap.totalBlocks      = 32;
break;

case EON_EN25P32:
case NUMON_M25P32:
    g_norAddrMap.bytesPerPage    = 256;
    g_norAddrMap.pagesPerSector  = 256;
    g_norAddrMap.bytesPerSector  = 64*1024;
    g_norAddrMap.sectorsPerBlock = 1;
    g_norAddrMap.totalBlocks     = 64;
    break;

case EON_EN25B16:
    g_norAddrMap.bytesPerPage    = 256;
    g_norAddrMap.pagesPerSector  = 256;
    g_norAddrMap.bytesPerSector  = 64*1024;
    g_norAddrMap.sectorsPerBlock = 1;
    g_norAddrMap.totalBlocks     = 32; //31x64k + 1x32k + 1x16k + 1x8k + 2x4k
    break;

case MX_25L1605A:
case MX_25L1635D:
case WIN_W25X16A:
case ATMEL_AT26DF161:
    case SPAN_S25FL016A:
        g_norAddrMap.bytesPerPage    = 256;
        g_norAddrMap.pagesPerSector  = 256;
        g_norAddrMap.bytesPerSector  = 64*1024;
        g_norAddrMap.sectorsPerBlock = 1;
        g_norAddrMap.totalBlocks     = 32;
        break;

case EON_EN25F16:
case EON_EN25Q16:
    g_norAddrMap.bytesPerPage    = 256;
    g_norAddrMap.pagesPerSector  = 256;
    g_norAddrMap.bytesPerSector  = 64*1024;
    g_norAddrMap.sectorsPerBlock = 1;
    g_norAddrMap.totalBlocks     = 32; //31x64k + 1x32k + 1x16k + 1x8k + 2x4k
    break;

case EON_EN25B32:
case EON_EN25F32:
case EON_EN25Q32A:
case EON_EN25QH32:

```

```

case Micron_N25Q032A:
case MX_25L3235D:
case MX_25L3205D:
case PMC_PM25LQ032C:
case SPAN_S25FL032A:
case WIN_W25X32V:
case WIN_W25Q32BV:
case ESMT_F25L32Q:
case ESMT_F25L32PA:
    g_norAddrMap.bytesPerPage      = 256;
    g_norAddrMap.pagesPerSector     = 256;
    g_norAddrMap.bytesPerSector     = 64*1024;
    g_norAddrMap.sectorsPerBlock    = 1;
    g_norAddrMap.totalBlocks        = 64;
    break;

case SST_25VF016B:
    g_norAddrMap.bytesPerPage      = 512;
    g_norAddrMap.pagesPerSector     = 128;
    g_norAddrMap.bytesPerSector     = 64*1024;
    g_norAddrMap.sectorsPerBlock    = 1;
    g_norAddrMap.totalBlocks        = 32;
    break;

case AMIC_A25L032:
case AMIC_A25LQ32A:
    g_norAddrMap.bytesPerPage      = 256;
    g_norAddrMap.pagesPerSector     = 256;
    g_norAddrMap.bytesPerSector     = 64*1024;
    g_norAddrMap.sectorsPerBlock    = 1;
    g_norAddrMap.totalBlocks        = 64;
    break;

case ATMEL_AT26D321:
    g_norAddrMap.bytesPerPage      = 256;
    g_norAddrMap.pagesPerSector     = 256;
    g_norAddrMap.bytesPerSector     = 64*1024;
    g_norAddrMap.sectorsPerBlock    = 1;
    g_norAddrMap.totalBlocks        = 64;
    break;

case ES_ES25M16A:
    g_norAddrMap.bytesPerPage      = 256;
    g_norAddrMap.pagesPerSector     = 256;
    g_norAddrMap.bytesPerSector     = 64*1024;

```

```

    g_norAddrMap.sectorsPerBlock    = 1;
    g_norAddrMap.totalBlocks        = 32;
    break;

case EON_EN25B64:
case EON_EN25Q64:
case WIN_W25Q64BV:
case MX_25L6445E:
    g_norAddrMap.bytesPerPage      = 256;
    g_norAddrMap.pagesPerSector    = 256;
    g_norAddrMap.bytesPerSector    = 64*1024;
    g_norAddrMap.sectorsPerBlock   = 1;
    g_norAddrMap.totalBlocks       = 128;
    break;

case MX_25L12835F:
    g_norAddrMap.bytesPerPage      = 256;
    g_norAddrMap.pagesPerSector    = 256;
    g_norAddrMap.bytesPerSector    = 64*1024;
    g_norAddrMap.sectorsPerBlock   = 1;
    g_norAddrMap.totalBlocks       = 256;
    break;

case MX_25L25735F:
case MX_25L25635F:
    g_norAddrMap.bytesPerPage      = 256;
    g_norAddrMap.pagesPerSector    = 256;
    g_norAddrMap.bytesPerSector    = 64*1024;
    g_norAddrMap.sectorsPerBlock   = 1;
    g_norAddrMap.totalBlocks       = 512;
    g_norAddrMap.use4BytesAddress  = true;
    break;

case EON_EN25F80:
case EON_EN25Q80A:
    g_norAddrMap.bytesPerPage      = 256;
    g_norAddrMap.pagesPerSector    = 256;
    g_norAddrMap.bytesPerSector    = 64*1024;
    g_norAddrMap.sectorsPerBlock   = 1;
    g_norAddrMap.totalBlocks       = 16;
    break;

case GD_GD25Q64B: // 8 MB
    g_norAddrMap.bytesPerPage      = 256;
    g_norAddrMap.pagesPerSector    = 256;

```

```

    g_norAddrMap.bytesPerSector    = 64 * 1024;
    g_norAddrMap.sectorsPerBlock   = 1;
    g_norAddrMap.totalBlocks       = 128;
    break;

case GD_GD25Q32: // 4 MB
    g_norAddrMap.bytesPerPage      = 256;
    g_norAddrMap.pagesPerSector    = 256;
    g_norAddrMap.bytesPerSector    = 64 * 1024;
    g_norAddrMap.sectorsPerBlock   = 1;
    g_norAddrMap.totalBlocks       = 64;
    break;

case GD_GD25Q16: // 2 MB
    g_norAddrMap.bytesPerPage      = 256;
    g_norAddrMap.pagesPerSector    = 256;
    g_norAddrMap.bytesPerSector    = 64 * 1024;
    g_norAddrMap.sectorsPerBlock   = 1;
    g_norAddrMap.totalBlocks       = 32;
    break;
case NEW_NOR_ENUM_ID:
    g_norAddrMap.bytesPerPage      = 256;
    g_norAddrMap.pagesPerSector    = 256;
    g_norAddrMap.bytesPerSector    = 64 * 1024;
    g_norAddrMap.sectorsPerBlock   = 1;
    g_norAddrMap.totalBlocks       = 64;
    break;

default:
    break;
}
}

```