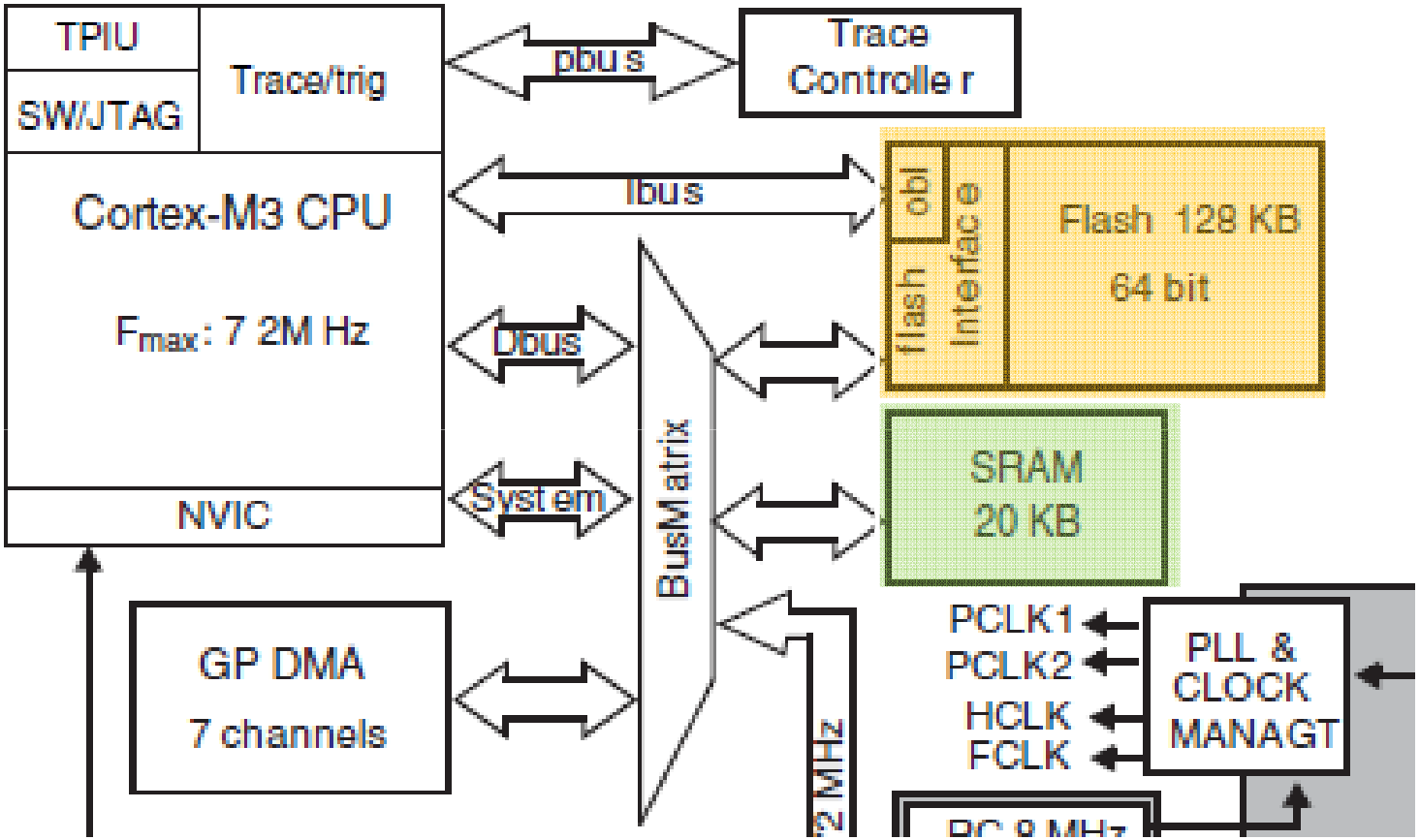
SIOC嵌入式軟體實驗
實驗五：Embedded Flash

WU-YANG
Technology Co., Ltd.
support.wuyang@gmail.com

實驗目的

- 瞭解STM32內部Flash記憶體與處理器界面的架構；實驗存取Flash的資料，並透過VCP傳送到超級終端機觀察結果

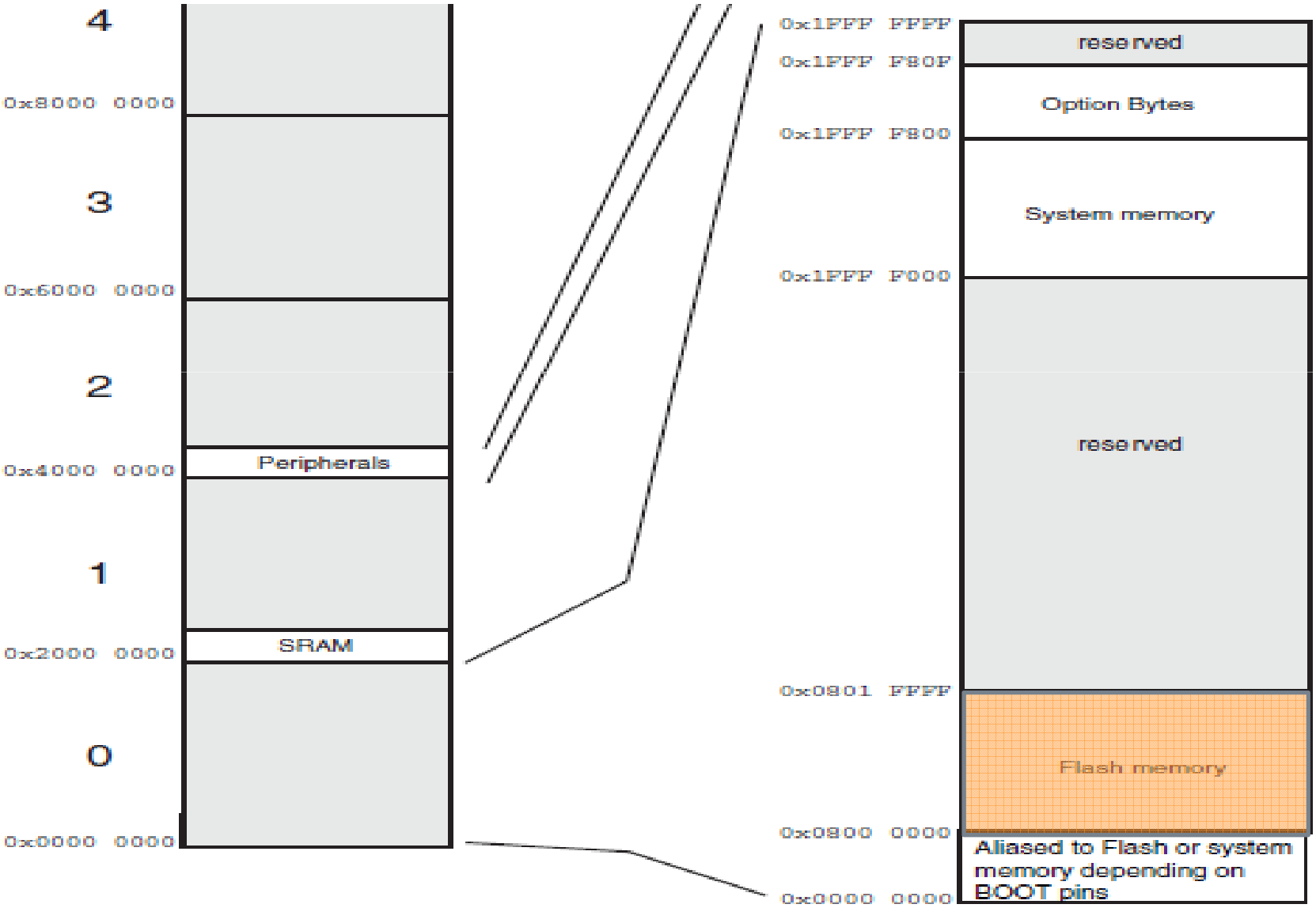
Flash界面架構


TPIU
SW/JTAG
Trace/trig
Cortex-M3 CPU
Fmax: 72M Hz
NVIC
pbus
Trace Controller
Ibus
Dbus
System
BusMatrix
obl
flash Interface
Flash 128 KB
64 bit
SRAM
20 KB
GP DMA
7 channels
PCLK1
PCLK2
HCLK
FCLK
PLL & CLOCK MANAGT

FLITF

- Features
  - Read interface with prefetch buffer (2x64-bit words)
  - Option byte Loader
  - Flash Program / Erase operation
  - Read / Write protection

Memory Mapping



Embedded Flash Memory

Table 3. Flash module organization (medium-density devices)

Block	Name	Base addresses	Size (bytes)
Main memory	Page 0	0x0800 0000 - 0x0800 03FF	1 Kbyte
	Page 1	0x0800 0400 - 0x0800 07FF	1 Kbyte
	Page 2	0x0800 0800 - 0x0800 0BFF	1 Kbyte
	Page 3	0x0800 0C00 - 0x0800 0FFF	1 Kbyte
	Page 4	0x0800 1000 - 0x0800 13FF	1 Kbyte
	. . .	. . .	. . .
	Page 127	0x0801 FC00 - 0x0801 FFFF	1 Kbyte
Information block	System memory	0x1FFF F000 - 0x1FFF F7FF	2 Kbytes
	Option Bytes	0x1FFF F800 - 0x1FFF F80F	16

Flash memory interface registers

Flash memory interface registers	FLASH_ACR	0x4002 2000 - 0x4002 2003	4
	FLASH_KEYR	0x4002 2004 - 0x4002 2007	4
	FLASH_OPTKEYR	0x4002 2008 - 0x4002 200B	4
	FLASH_SR	0x4002 200C - 0x4002 200F	4
	FLASH_CR	0x4002 2010 - 0x4002 2013	4
	FLASH_AR	0x4002 2014 - 0x4002 2017	4
	Reserved	0x4002 2018 - 0x4002 201B	4
	FLASH_OBR	0x4002 201C - 0x4002 201F	4
	FLASH_WRPR	0x4002 2020 - 0x4002 2023	4

Flash memory interface register

Flash control register (FLASH_CR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			EOPIE	Res.	ERRIE	OPTWRE	Res.	LOCK	STRT	OPTER	OPTPG	Res.	MER	PER	PG
			rw		rw	rw		rw	rw	rw	rw		rw	rw	rw

Bit 9 **OPTWRE: Option bytes write enable**
When set, the option bytes can be programmed. This bit is set on writing the correct key sequence to the FLASH_OPTKEYR register.
This bit can be reset by software
Bit 8 Reserved, must be kept cleared.
Bit 7 **LOCK: Lock**
Write to 1 only. When it is set, it indicates that the FPEC and FLASH_CR are locked. This bit is reset by hardware after detecting the unlock sequence.
In the event of unsuccessful unlock operation, this bit remains set until the next reset.

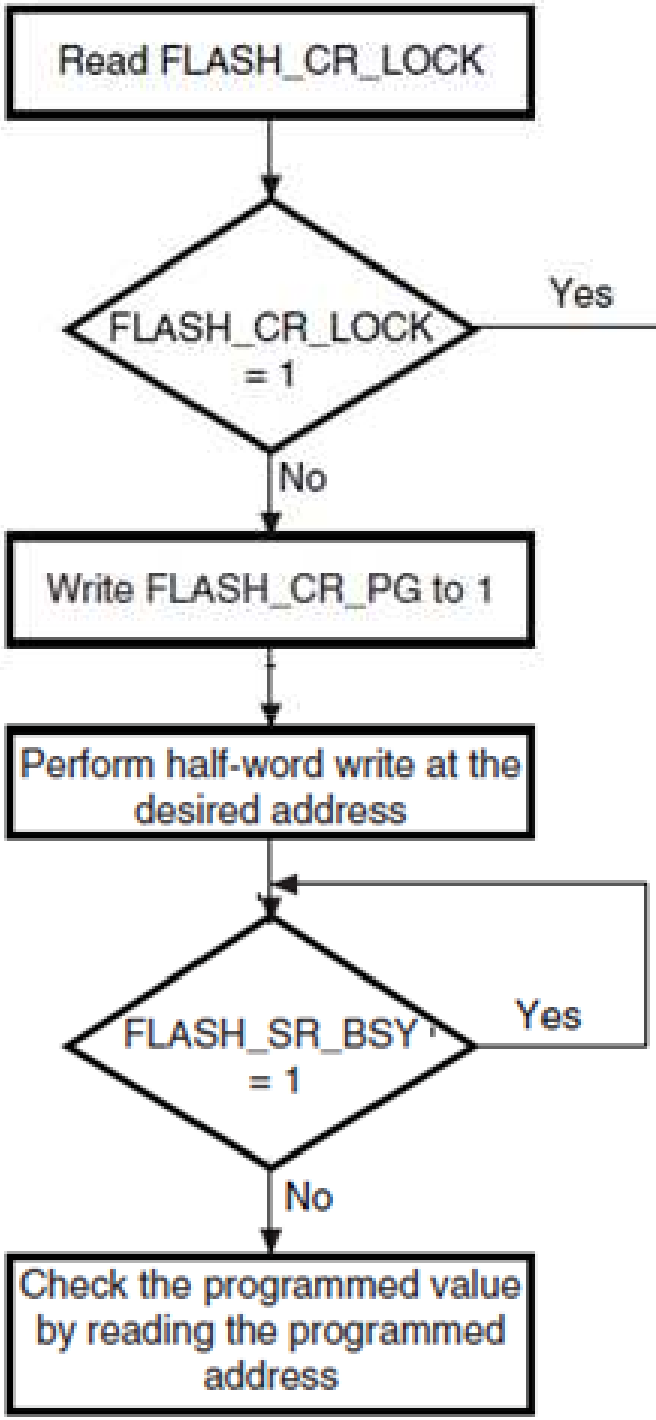
Bit 6 **STRT: Start**
This bit triggers an ERASE operation when set. This bit is set only by software and reset when the BSY bit is reset.
Bit 5 **OPTER: Option byte erase**
Option byte erase chosen.
Bit 4 **OPTPG: Option byte programming**
Option byte programming chosen.
Bit 3 Reserved, must be kept cleared.
Bit 2 **MER: Mass erase**
Erase of all user pages chosen.
Bit 1 **PER: Page erase**
Page Erase chosen.
Bit 0 **PG: Programming**
Flash programming chosen.

FLASH library function

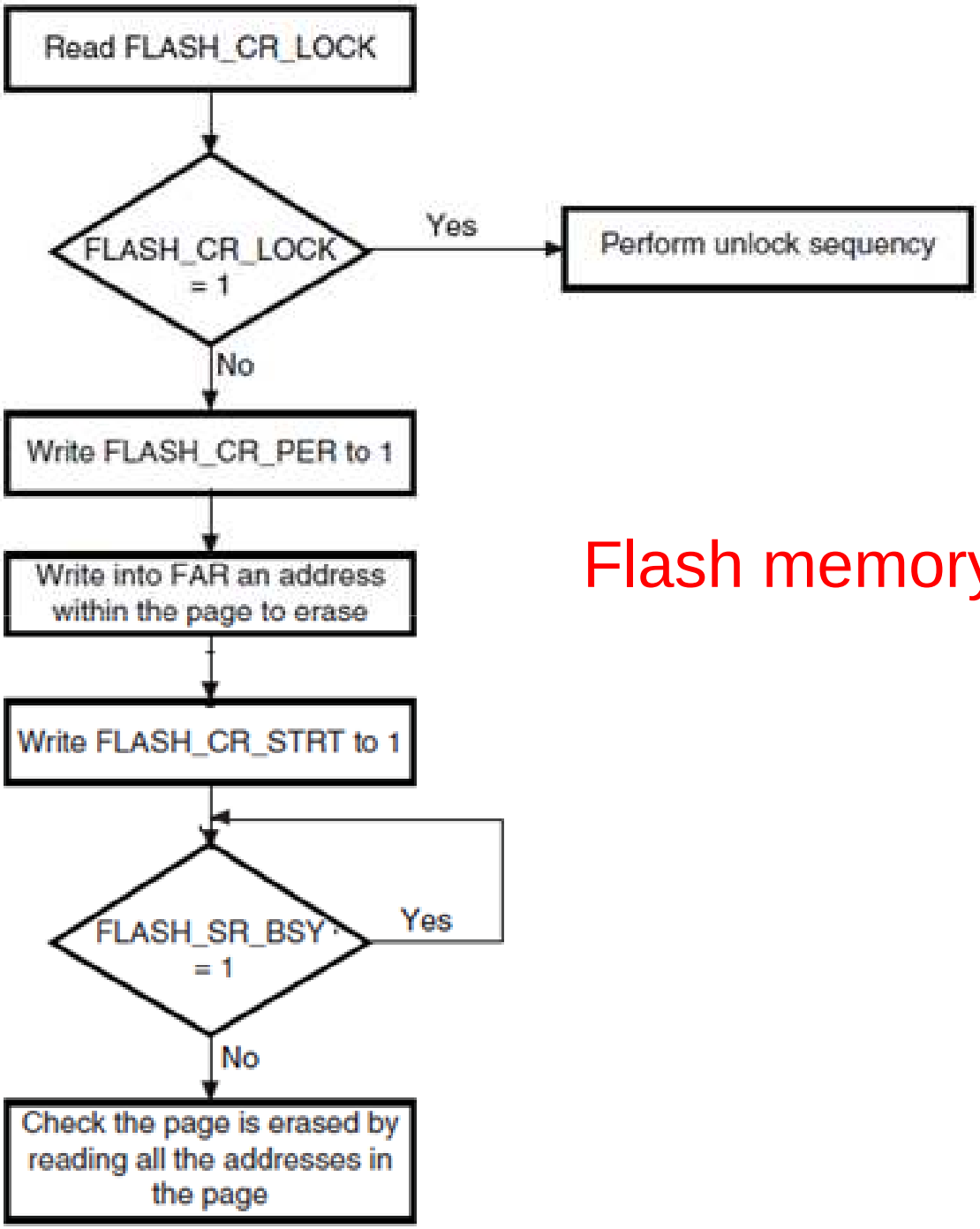
Function name	Description
FLASH_SetLatency	Sets the code latency value.
FLASH_HalfCycleAccessCmd	Enables or disables the Half cycle FLASH access.
FLASH_PrefetchBufferCmd	Enables or disables the Prefetch Buffer.
FLASH_Unlock	Unlocks the FLASH Program Erase Controller.
FLASH_Lock	Locks the Flash Program Erase Controller.
FLASH_ErasePage	Erases a specified FLASH page.
FLASH_EraseAllPages	Erases all FLASH pages.
FLASH_EraseOptionBytes	Erases the FLASH option bytes.
FLASH_ProgramWord	Programs a word at a specified address.
FLASH_ProgramHalfWord	Programs a half word at a specified address.
FLASH_ProgramOptionByteData	Programs a half word at a specified Option Byte Data address.
FLASH_EnableWriteProtection	Write protects the desired pages

FLASH library function(conti.)

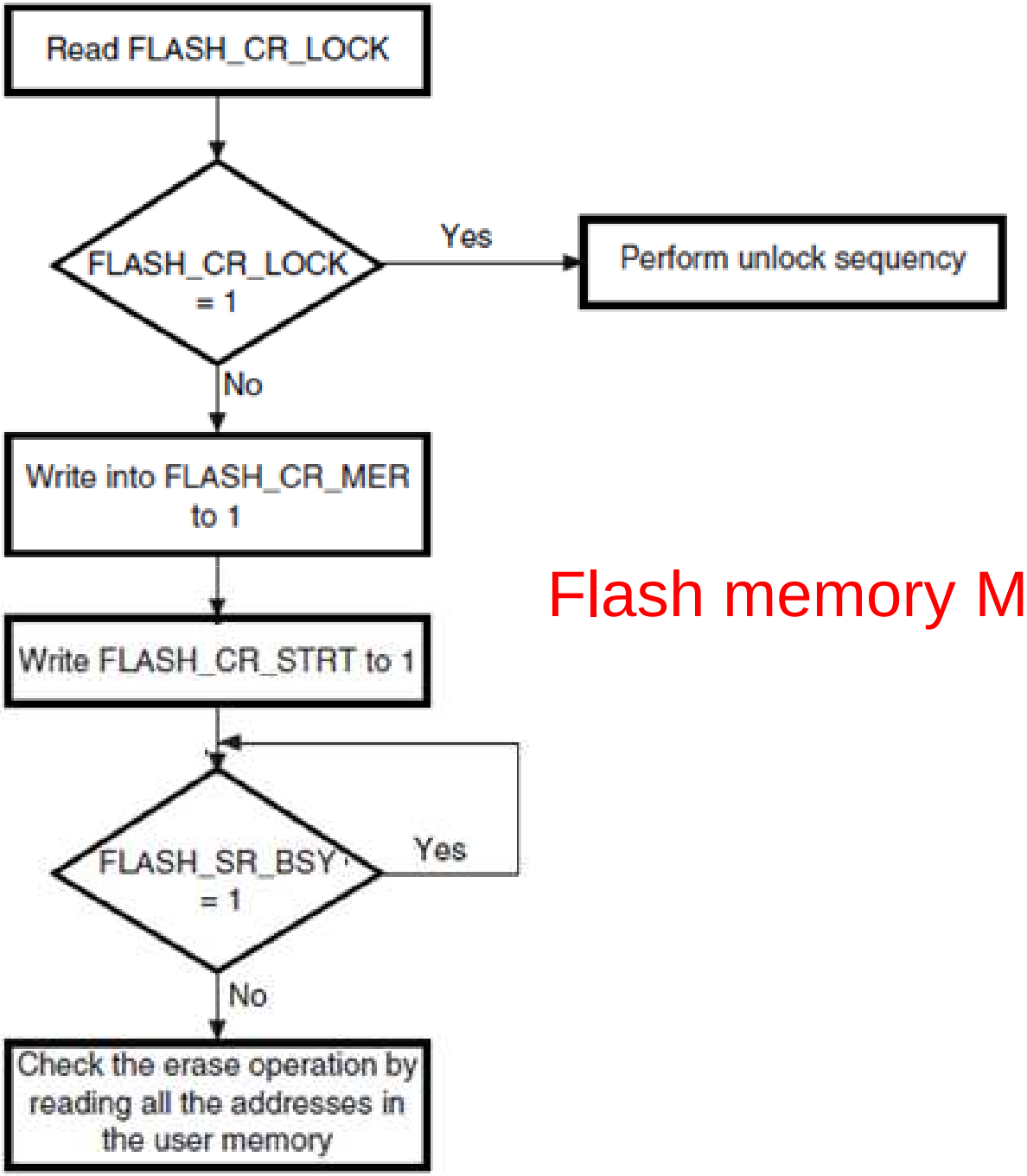
Function name	Description
FLASH_ReadOutProtection	Enables or disables the read out protection.
FLASH_UserOptionByteConfig	Programs the FLASH User Option Byte: IWDG_SW / RST_STOP / RST_STDBY.
FLASH_GetUserOptionByte	Returns the FLASH User Option Bytes values.
FLASH_GetWriteProtectionOptionByte	Returns the FLASH Write Protection Option Bytes Register value.
FLASH_GetReadOutProtectionStatus	Checks whether the FLASH Read Out Protection Status is set or not.
FLASH_GetPrefetchBufferStatus	Checks whether the FLASH Prefetch Buffer status is set or not.
FLASH_ITConfig	Enables or disables the specified FLASH interrupts.
FLASH_GetFlagStatus	Checks whether the specified FLASH flag is set or not.
FLASH_ClearFlag	Clears the FLASH pending flags.
FLASH_GetStatus	Returns the FLASH Status.
FLASH_WaitForLastOperation	Waits for a Flash operation to complete or a TIMEOUT to occur.



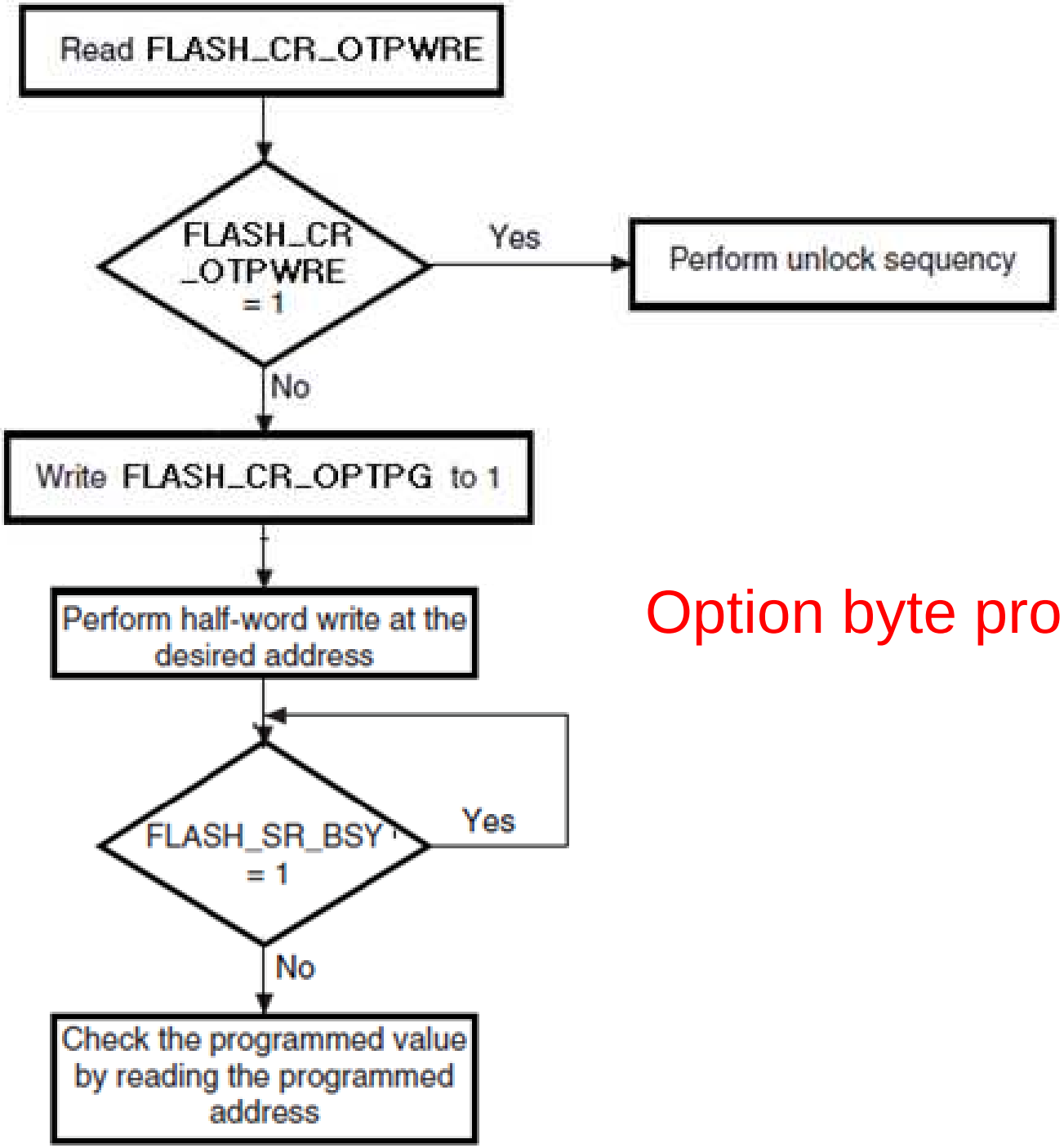
Main Flash memory programming



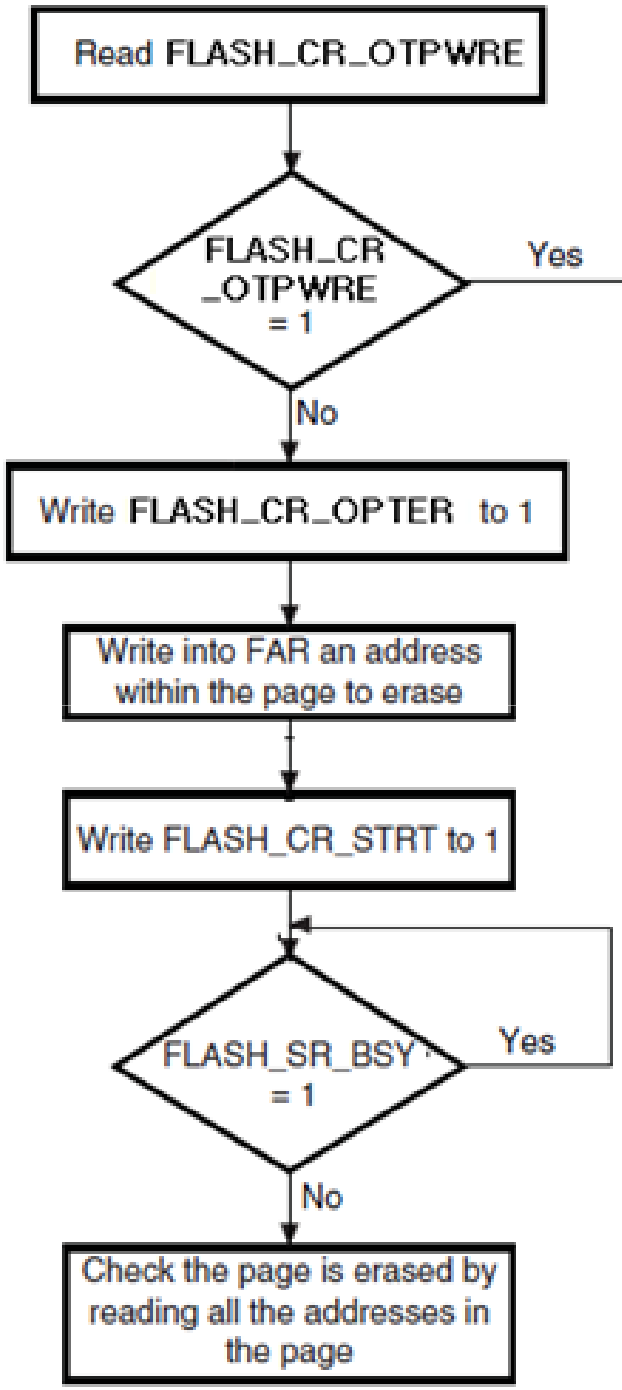
Flash memory Page Erase



Flash memory Mass Erase



Option byte programming



Option byte Erase

Development Flow

Embedded Software Side

- Programming
  - Bootup STM32F10x
    - NVIC Configure
  - Flash memory R/W operation
    - FLASH Unlock
    - FLASH ClearFlag
    - FLASH ErasePage
    - FLASH ProgramWord

Bootup STM32F10x

```
int main(void)
{
  u32 *temp;
  Set_System();

  /*pause*/
  getchar();
  printf("start \r\n");

  FLASHStatus =
FLASH_COMPLETE;
  MemoryProgramStatus =
PASSED;
  Data = 0x15041925;

  /* NVIC configuration */
  NVIC_Configuration();
  .......
```

範例說明

Flash FwLib Functions List

Function name	Description
FLASH_Unlock	Unlocks the FLASH Program Erase Controller.
FLASH_ClearFlag	Clears the FLASH pending flags.
FLASH_ErasePage	Erases a specified FLASH page.
FLASH_ProgramWord	Programs a word at a specified address.

Flash memory R/W operation

- **FLASH Unlock**
- **FLASH ClearFlag**
- **FLASH ErasePage**
- **FLASH ProgramWord**

```
/* Unlock the Flash Program Erase controller */
FLASH_Unlock();
 /* Define the number of page to be erased */
  NbrOfPage = (EndAddr - StartAddr) / FLASH_PAGE_SIZE;
 /* Clear All pending flags */
FLASH_ClearFlag(FLASH_FLAG_BSY | FLASH_FLAG_EOP
| FLASH_FLAG_PGERR | FLASH_FLAG_WRPRTERR);
/* Erase the FLASH pages */
for(EraseCounter = 0; (EraseCounter < NbrOfPage) &&
(FLASHStatus == FLASH_COMPLETE); EraseCounter++)
{
     FLASHStatus = FLASH_ErasePage(StartAddr +
(FLASH_PAGE_SIZE * EraseCounter));
}
/*  FLASH Word program of data 0x15041979 at addresses
defined by StartAddr and EndAddr*/
 Address = StartAddr;
while((Address < EndAddr) && (FLASHStatus ==
FLASH_COMPLETE))
{
   FLASHStatus = FLASH_ProgramWord(Address, Data);
   temp=(u32 *)Address;
   printf("0x%x data is 0x%x \r\n",Address,*temp);
   Address = Address + 4;
}
```

預設定義說明

- #define StartAddr ((u32)0x08008000)
  - 定義Flash使用起始點
  - 使用起始點必需大於0x8003000 + Code Size
    - 附註: 0x8000000~0x8003000為DFU程式區塊，使用此區塊將造成無法燒錄程式
    - Code Size可由產生的HEX檔得知
- #define EndAddr ((u32)0x0800C000)
  - 定義Flash使用結束點
  - 使用起始點必需小於0x8040000
- Data = 0x15041975;
  - 寫入資料
  - 32Bit
- #define FLASH_PAGE_SIZE ((u16)0x400)
  - 每一個Page有1KByte

Intel HEX Format

:	//	aaaa	tt	dd	cc

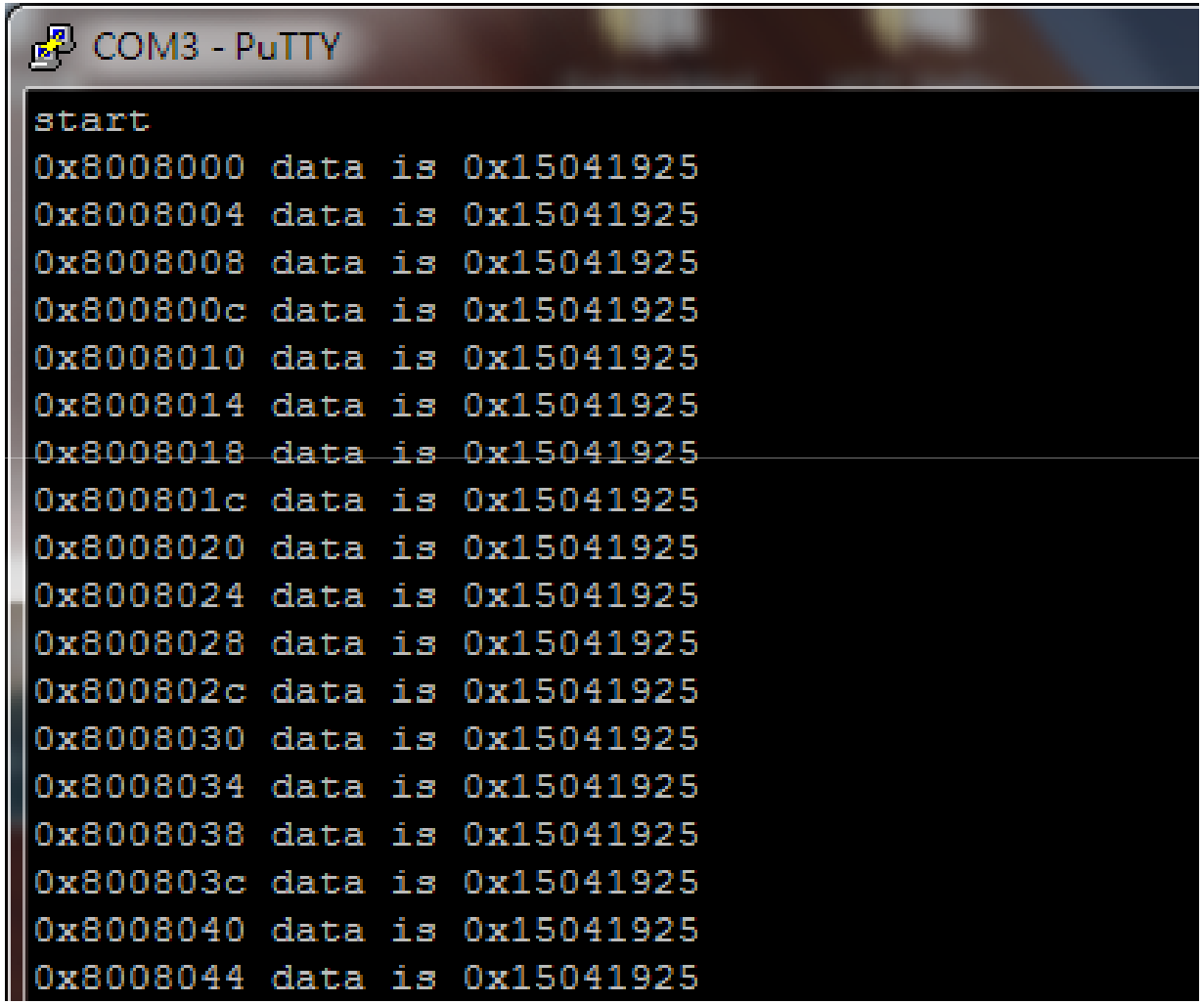
field	Description
:	the colon that starts every Intel HEX record.
//	the record-length field that represents the number of data bytes (**dd**) in the record.
aaaa	the address field that represents the starting address for subsequent data in the record.
tt	the field that represents the HEX record type, which may be one of the following: **00** - data record **01** - end-of-file record **02** - extended segment address record **04** - extended linear address record
dd	a data field that represents one byte of data. A record may have multiple data bytes. The number of data bytes in the record must match the number specified by the // field.
cc	the checksum field that represents the checksum of the record. The checksum is calculated by summing the values of all hexadecimal digit pairs in the record modulo 256 and taking the two's complement.

HEX Example

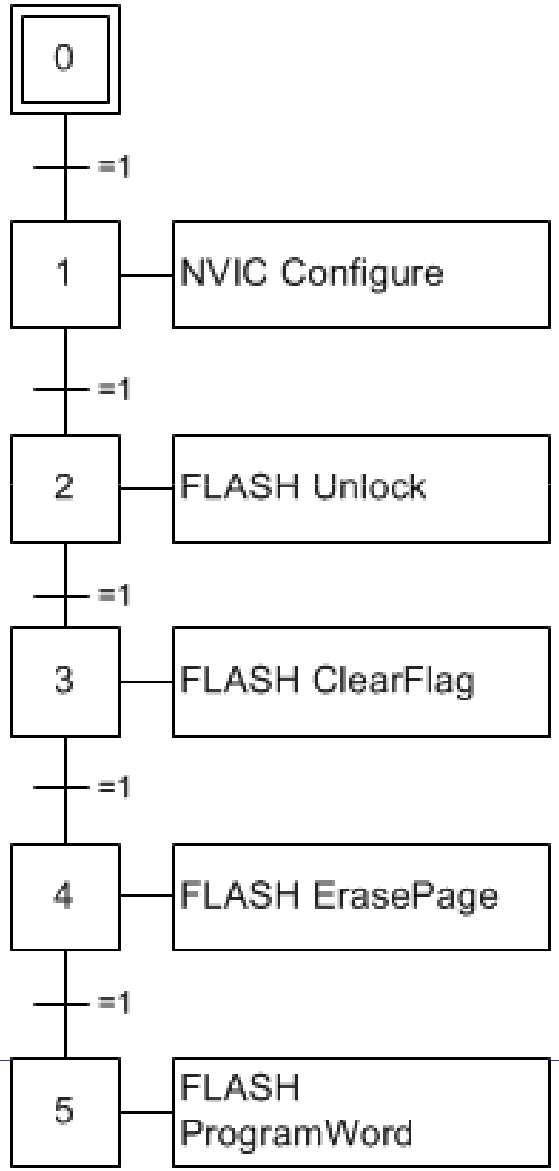
```
:020000040800F2
:10300000100E00200131000891 3E0008DD3D00084F
……
:105C00001A449729282910690 8011B3C020406CC74
:0C5C10002D010203040485060 7080900AA
:04000005080030EDD2
:00000001FF
```

- 資料填入起始位置為0x08003000
- 每一行有0x10(16)個Bytes
- 最後一筆資料為04000005080030EDD2
- 資料結束點為0x08005C10
- (0x08005C10 -0x08000000)/0x400 = 23
- 需以0x400為單位，所以可存的flash位置起點為0x08006000

程式執行結果


COM3 - PuTTY
start
0x8008000 data is 0x15041925
0x8008004 data is 0x15041925
0x8008008 data is 0x15041925
0x800800c data is 0x15041925
0x8008010 data is 0x15041925
0x8008014 data is 0x15041925
0x8008018 data is 0x15041925
0x800801c data is 0x15041925
0x8008020 data is 0x15041925
0x8008024 data is 0x15041925
0x8008028 data is 0x15041925
0x800802c data is 0x15041925
0x8008030 data is 0x15041925
0x8008034 data is 0x15041925
0x8008038 data is 0x15041925
0x800803c data is 0x15041925
0x8008040 data is 0x15041925
0x8008044 data is 0x15041925

Flash存取控制流程



實驗

實驗1：flash memory資料存取

實驗2：非揮發性資料寫入

WU-YANG
Technology Co., Ltd.

實驗1：flash memory資料存取

WU-YANG

Technology Co., Ltd.

實驗1說明

- #define StartAddr ((u32)0x08020000) /*定義Flash使用起始點*/
  使用起始點必需大於0x8003000 + Code Size
  - 0x8000000~0x8003000為DFU程式區塊，使用此區塊將造成無法燒錄程式
  - Code Size可由產生的HEX檔得知
- u32 Address = 0x08020000;
  - 定義Address初始值
  - 初始值必需與Flash使用起始點相同

實驗2：非揮發性資料寫入

觀察Flash寫入後電源關閉，再開啟後寫入資料是否仍存在

WU-YANG
Technology Co., Ltd.

實驗2說明

Flash FwLib Functions List

Function name	Description
FLASH_Unlock	Unlocks the FLASH Program Erase Controller.
FLASH_ClearFlag	Clears the FLASH pending flags.
FLASH_ErasePage	Erases a specified FLASH page.
FLASH_ProgramWord	Programs a word at a specified address.

Flash memory R/W operation

FLASH ErasePage

FLASH ProgramWord

```
 /* Erase the FLASH pages */
 for(EraseCounter = 0; (EraseCounter < NbrOfPage) &&
(FLASHStatus == FLASH_COMPLETE); EraseCounter++)
 {
  FLASHStatus = FLASH_ErasePage(StartAddr +
(FLASH_PAGE_SIZE * EraseCounter));
 }

 /*  FLASH Word program of data 0x15041979 at addresses
defined by StartAddr and EndAddr*/
 Address = StartAddr;

 while((Address < EndAddr) && (FLASHStatus ==
FLASH_COMPLETE))
 {
                FLASHStatus = FLASH_ProgramWord(Address,
Data);
               temp=(u32 *)Address;
                printf("0x%x data is 0x%x \r\n",Address,*temp);
                Address = Address + 4;
 }
```

Q & A

WU-YANG

Technology Co., Ltd.