

MIAT_STM32 USB 介面driver

浯陽科技有限公司



WU-YANG
Technology Co., Ltd.



Declared Version

Training Only

Declare

Document Number	
Document Version	1.00
Release Date	
Document Title	USB介面driver
Exercise Time	■
Platform	■ MIAT_STM32.V2 ■ MIAT IOB.V1
Peripheral	■
Author	■ WU-YANG Technology Co., Ltd.



實驗目的(一)

- 使用MIAT_STM32實驗板透過USB進行資料存取實驗，並利用PC端程式傳送資料顯示於LCD或由STM32取回資料。



實驗原理

☐ USB

- 簡介
- 傳輸類型
- 資料傳輸
- 裝置列舉
- Endpoint

☐ LCD



USB 簡介

- ☐ 支持熱插拔和隨插即用
- ☐ 包括Hub在內最多可連接127台裝置
- ☐ 一個控制器下最多可以有5級hub
- ☐ 傳輸數率
 - High-speed
 - ☐ 傳輸速率25Mbps ~ 400Mbps (最大480Mbps)
 - Full-speed
 - ☐ 傳輸速率500Kbps~10Mbps (最大12Mbps)
 - Low-speed
 - ☐ 傳輸速率10Kbps~100Kbps (最大1.5Mbps)



USB傳輸類型

- 巨量傳輸 (Bulk Transfer)
 - 使用剩下的頻寬傳輸大量資料(沒有保證延遲、連續性、頻寬和速度)
- 同步傳輸 (Isochronous Transfer)
 - 保證一定的頻寬傳輸(沒有保證資料的正確性)
- 控制傳輸 (Control Transfer)
 - 簡單的命令和狀態傳輸，用於匯流排的命令與設定
- 中斷傳輸 (Interrupt Transfer)
 - 定時詢問回報資料或是支援中斷



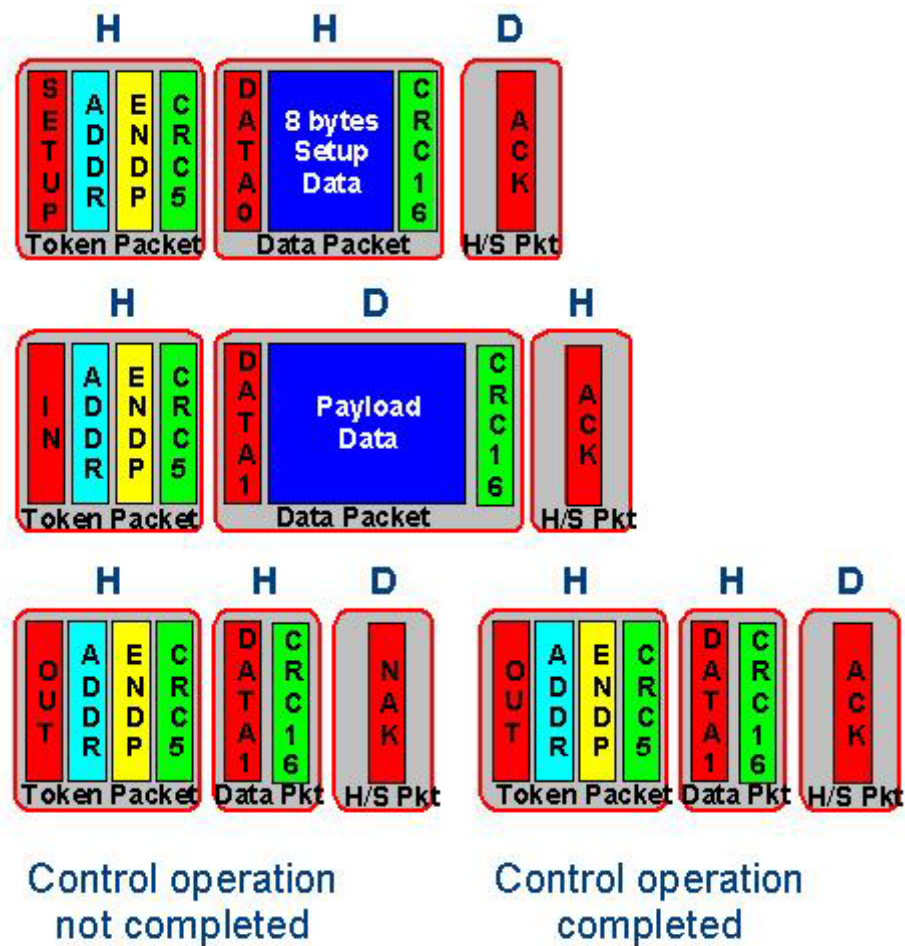
USB 資料傳輸

- ☐ 所有的傳輸接由Host端開始，一般Host為PC
- ☐ 裝置(Device)回應Host的需求
- ☐ Host傳送資料至Device稱為OUT
- ☐ Device送資料回Host稱為IN



Control Transfer

Control In Protocol



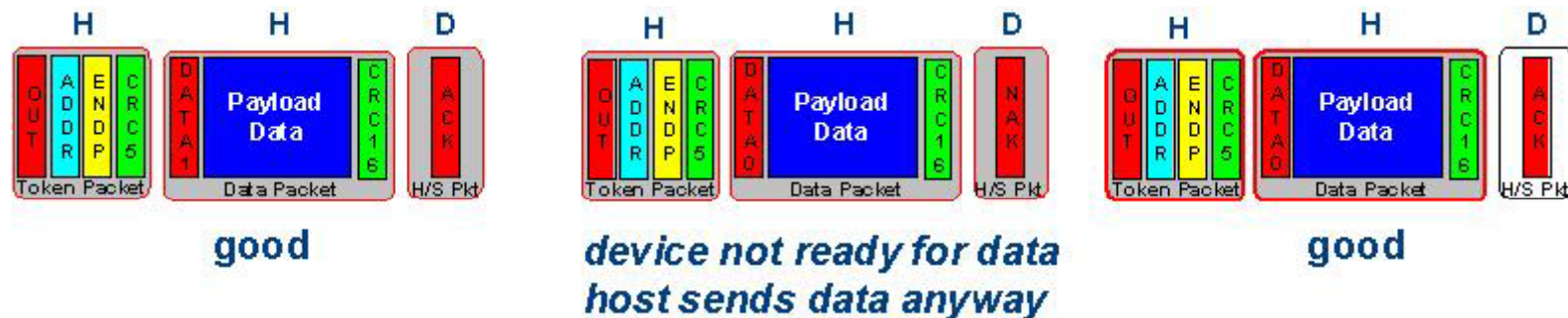


Bulk Transfer

Bulk IN



Bulk OUT





裝置列舉

- USB Plug and Play，第一個動作就是裝置列舉 (Enumeration) 電腦主機會利用控制傳輸和裝置溝通，以便取得裝置的資訊，如此電腦作業系統可以載入適當的驅動程式，讓裝置開始運作。
 - 取得裝置描述元
 - 設定裝置的新位址
 - 使用新位址取得裝置描述元
 - 取得配置描述元
 - 設定配置描述元



Endpoint

- ❑ STM32有 16 個Endpoint
- ❑ 每個Endpoint都有自己的方向與所指到的資料記憶體儲存空間
- ❑ 控制傳輸用為雙向，其餘為單向
- ❑ 每個 USB 裝置都有Endpoint0，只用來做控制傳輸用
- ❑ 裝置列舉的過程，Host 與Device 經由Endpoint0作控制傳輸



PC端驅動程式

- ☐ Windows XP 內建驅動程式
 - HID Device
 - ☐ Low-speed
 - 傳輸速率10Kbps~100Kbps (最大1.5Mbps)
 - Mass Storage Device
 - ☐ High-speed
 - 傳輸速率25Mbps ~ 400Mbps (最大480Mbps)



STM32 USB通訊

- ☐ 大量資料傳輸
- ☐ 採用Bulk Transfer
- ☐ 採用USB Mass Storage Device模式
- ☐ SCSI Command新增D0、D3、D4



PC端SCSI指令

- status = SCSI_CMD(fileHandle,
SCSI_IOCTL_DATA_IN,CMD,datalength,databuffer);
 - fileHandle : 磁碟機
 - SCSI_IOCTL_DATA_IN : 資料讀取
 - CMD : SCSI命令
 - datalength : 資料長度
 - databuffer : 資料存放空間
- SCSI_CMD(fileHandle,
SCSI_IOCTL_DATA_OUT,CMD,datalength,databuffer)
 - SCSI_IOCTL_DATA_OUT : 資料送出



CMD封包

C	AAAA	R	NN	
----------	-------------	----------	-----------	--

field	Description
C	SCSI Command
AAAA	Logical Block Address of First Block
R	reserved
NN	Number of Blocks to transfer

□ Example :

0xD4,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,.....

- SCSI Command D4
- First Logical Block Address : 0x00000000
- One Block to transfer
- Get Data from 0x00000000 ~ 0x00000200



硬體電路配置

Mapping Table

<i>Num.</i>	<i>MIAT_STM32V2</i>	<i>MIAT_IOBV1</i>
1	PE6 (1.5)	LCD_EN
2	PF6 (1.18)	LCD_R/W
3	PF7 (1.19)	LCD_RS
4	PF8(1.20)	LCD_D4
5	PF9 (1.21)	LCD_D5
6	PF10 (1.22)	LCD_D6
7	PF11 (2.13)	LCD_D7
8	VCC5V (1.36)	VCC5V
9	GND (1.35)	GND



實驗步驟

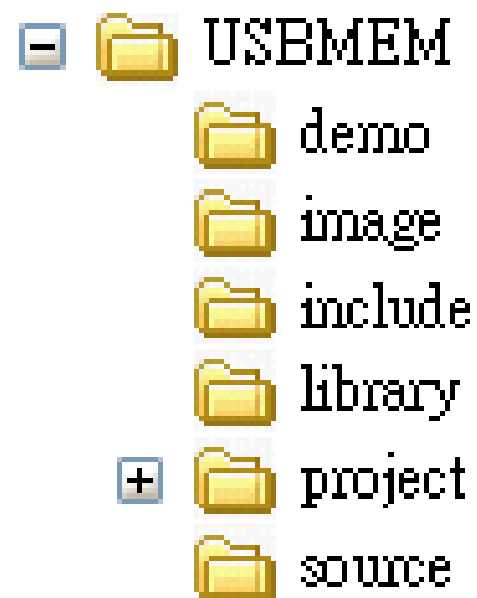
- ☐ 範例目錄架構
- ☐ 函式庫說明
- ☐ 範例說明
- ☐ 參數說明
- ☐ 燒錄MIAT_STM32
- ☐ PC端程式



範例目錄架構

□ 範例目錄

- 測試程式
- 測試映像檔
- 含括檔
- 函式庫
- 專案檔
- 原始碼





USBMEM函式庫

□ void USB_Init(void)

USB初始化函式

□ void USB_Connect(BOOL con)

USB連線建立

□ void USB_SetBuffer(unsigned char *Buffer,unsigned int Size)

設定SCSI Command D0傳回的記憶體區塊位置與大小



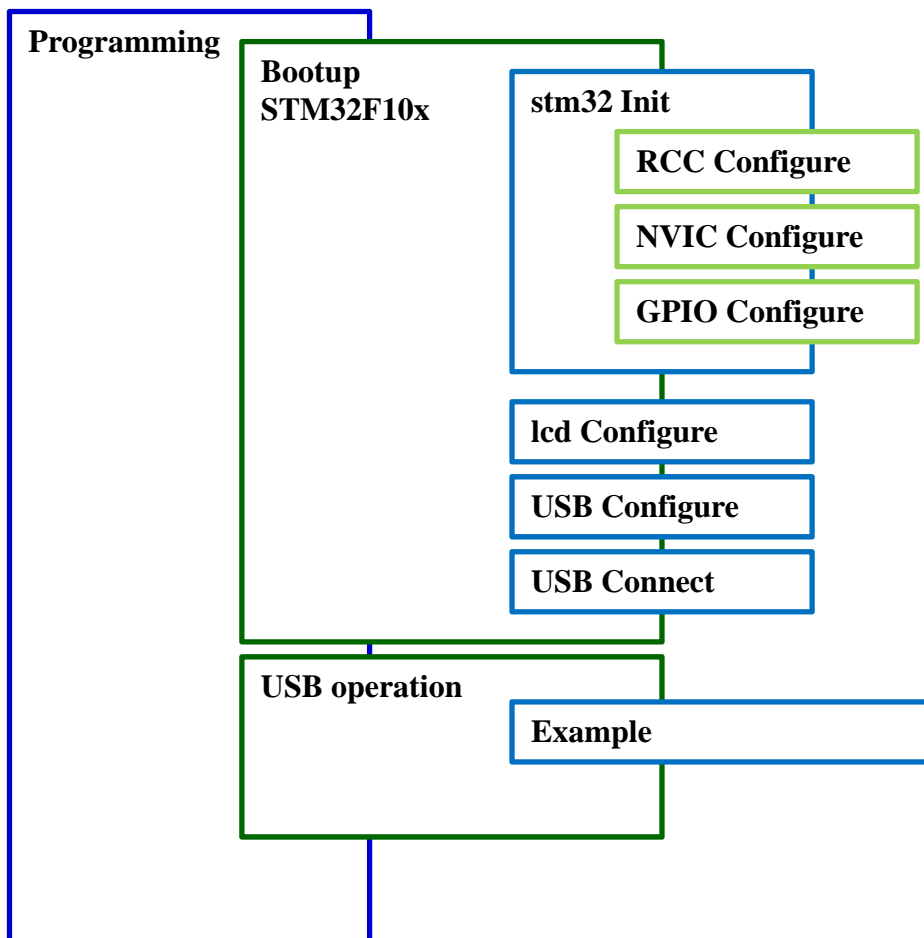
USBMEM函式庫

- ❑ `void USB_GetBuffer(void (*function)(unsigned char *))`
當執行SCSI Command D0指令時可呼叫一外部程式
- ❑ `void USB_SetInBuffer(unsigned char *Buffer,unsigned int Size)`
設定SCSI Command D3傳入的記憶體區塊位置與大小
- ❑ `void USB_SetOutBuffer(unsigned char *Buffer,unsigned int Size)`
設定SCSI Command D4傳回的記憶體區塊位置與大小



範例說明

Embedded Software Side



Bootup STM32F10x

```
int main (void)
{
    unsigned char *pIN_Buffer;
    unsigned short i;

    stm32_Init ();           // STM32 Initialization

    lcd_init();              // LCD Initialization
    lcd_clear();

    USB_SetOutBuffer(OUT_Buffer,512);
    USB_SetInBuffer(IN_Buffer,512);

    USB_Init();              // USB Initialization
    USB_Connect(TRUE);       // USB Connect
}
```



範例說明

Embedded Software Side

USB operation

Example

USB operation

```
while (1)
{
    // Loop forever
    pIN_Buffer=IN_Buffer;
    set_cursor (0, 0);
    for(i=0;i<16;i++)
    {
        lcd_putchar(*pIN_Buffer++);
    }
    set_cursor (0, 1);
    for(i=0;i<16;i++)
    {
        lcd_putchar(*pIN_Buffer++);
    }
} // end while
} // end main
```

LCD Line1顯示
IN_Buffer[0~15]

LCD Line2顯示
IN_Buffer[16~31]



參數說明

- ☐ unsigned char IN_Buffer[512]
 - 資料輸入
 - 大小必須是512的倍數
- ☐ unsigned char OUT_Buffer[512]
 - 資料輸出
 - 大小必須是512的倍數
- ☐ USB_Connect
 - TRUE : USB 連線
 - FALSE : USB斷線

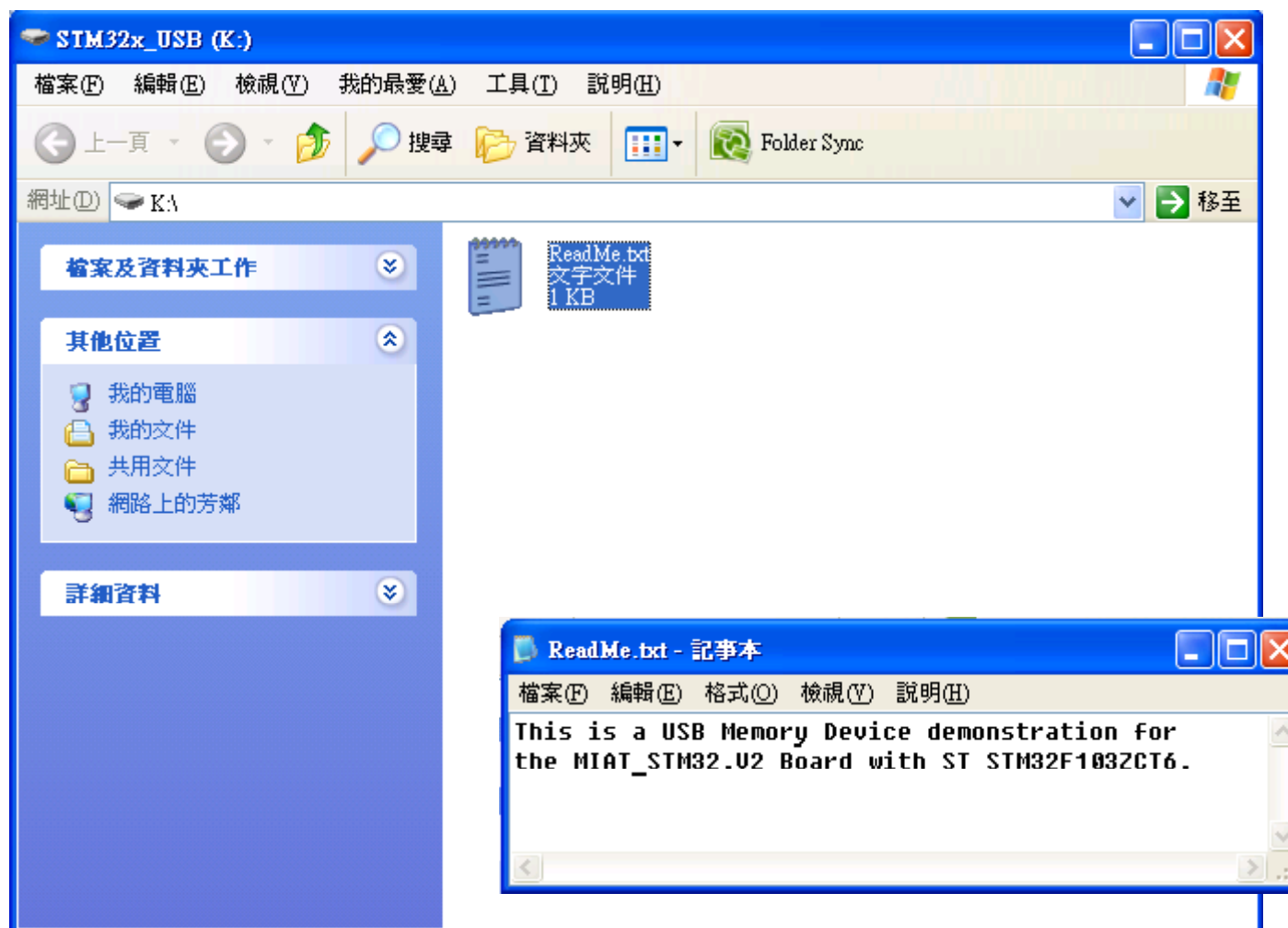


燒錄MIAT_STM32

- ☐ Rebuilder all target files產生HEX
- ☐ DFU File Manager轉換HEX產生DFU
- ☐ DfuSe Demonstration燒錄DFU
- ☐ Leave DFU mode

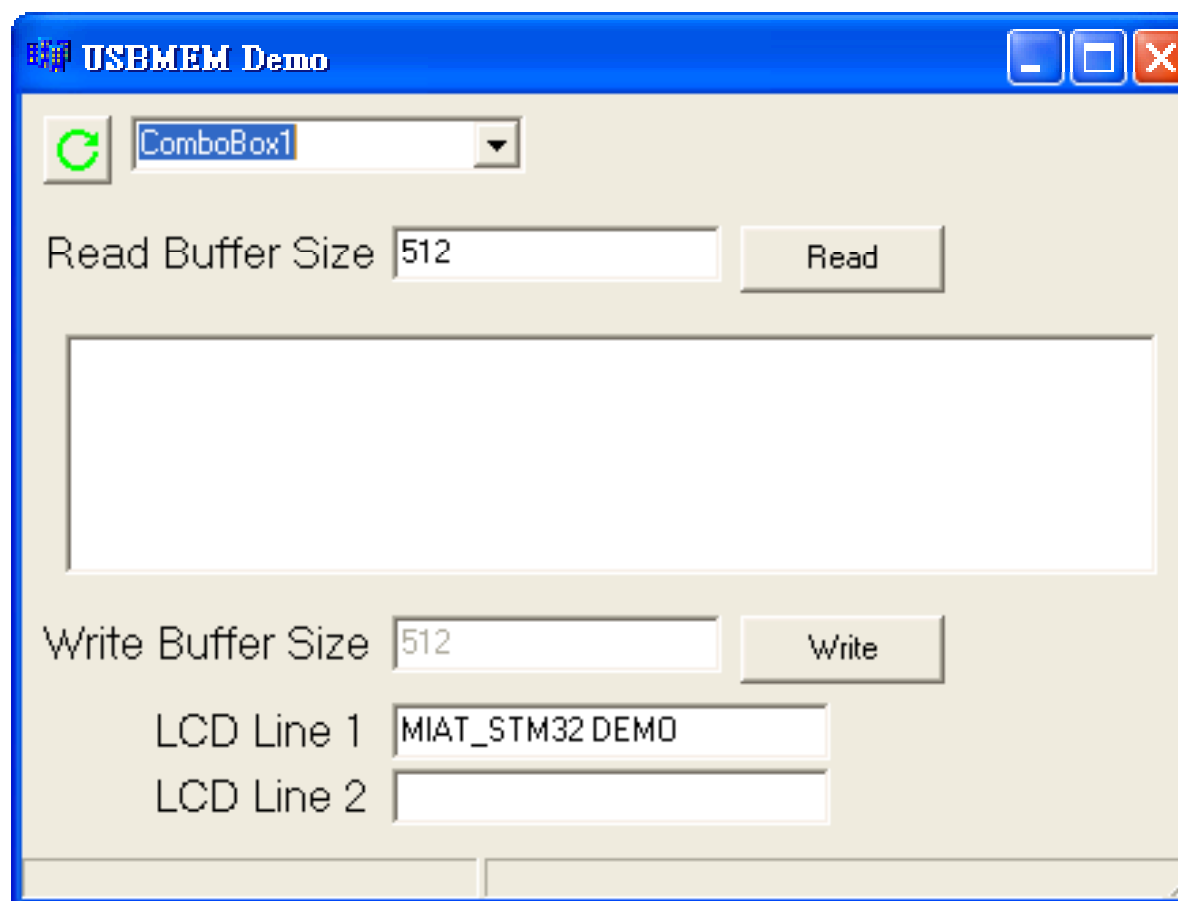


USB Mass Storage Device






PC端程式






PC端程式

- ☐  搜尋MIAT_STM32裝置
- ☐ Read Buffer Size : 取得資料大小
- ☐ Read : 取得資料
- ☐ Write Buffer Size : 送出資料大小(固定512)
- ☐ LCD Line1 : 顯示於LCD Line1的文字
- ☐ LCD Line2 : 顯示於LCD Line2的文字
- ☐ Write : 送出資料



取得資料

USBMEM Demo

 K:(0000008C)

Read Buffer Size

MIAT_STM32 DEMO

Write Buffer Size

LCD Line 1

LCD Line 2

USB 資料存取實驗

實驗一



WU-YANG
Technology Co., Ltd.



實驗一練習

☐ 練習:

- 測試從PC傳入資料至LCD顯示
- 測試從PC取回STM32資料
- 修改OUT_Buffer大小測試是否正常
- 修改OUT_Buffer內容測試是否正常



實驗目的(二)

- 使用MIAT_STM32實驗板透過MIAT_IOB實驗板整合2MPixels影像感測器進行影像擷取實驗，並利用PC端程式取回擷取影像。



硬體電路配置

Mapping Table

<i>Num.</i>	<i>MIAT_STM32V2</i>	<i>MIAT_IOBV1</i>	<i>Num.</i>	<i>MIAT_STM32V2</i>	<i>MIAT_IOBV1</i>
1	PC0 (1.24)	CMOS_D0	10	PC1 (1.25)	CMOS_D1
2	PC2 (1.26)	CMOS_D2	11	PC3 (1.27)	CMOS_D3
3	PC4 (2.8)	CMOS_D4	12	PC5 (2.9)	CMOS_D5
4	PC6 (3.24)	CMOS_D6	13	PC7 (3.25)	CMOS_D7
5	PE4 (1.3)	CMOS_STBYN	14	PA1 (1.33)	CMOS_MCLK
6	VCC5V (1.36)	VCC5V	15	GND (1.35)	GND
7	PE3 (1.2)	CMOS_VSYNC	16	PA6 (2.6)	CMOS_PCLK
8	PE2 (1.1)	CMOS_HSYNC	17	PE5 (1.4)	CMOS_RST
9	PB10 (2.33)	CMOS_SCL	18	PB11 (2.34)	CMOS_SDA

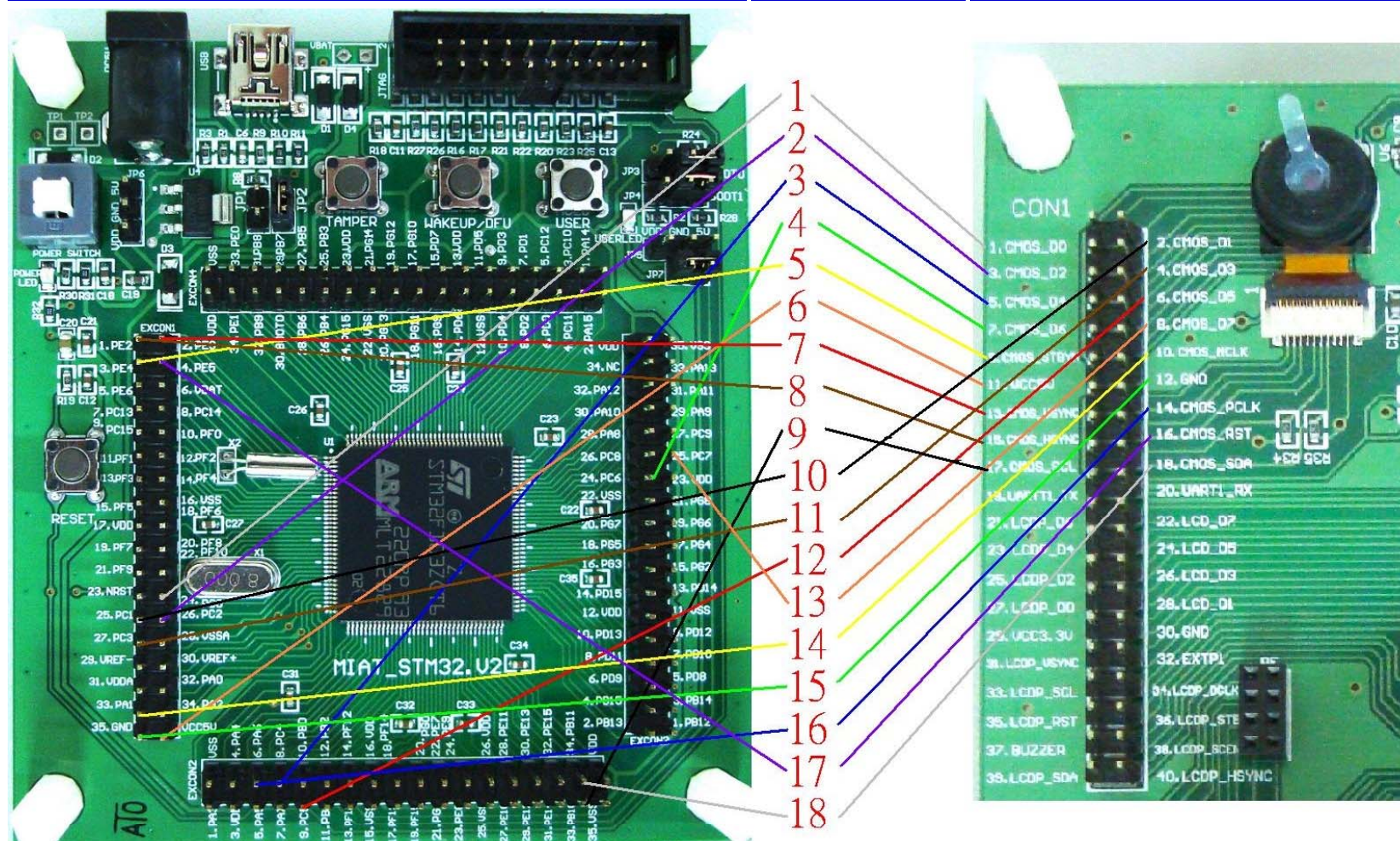


硬體電路配置

MIAT_STM32.V2 Board

Mapping Num.

MIAT_IOB.V1 Board





實驗步驟

- ☐ 範例目錄架構
- ☐ 函式庫說明
- ☐ 範例說明
- ☐ 參數說明
- ☐ PC端程式



CMOS函式庫

- ❑ void CMOS_Init(void)
CMOS初始化函式

- ❑ void CMOS_Exposure(unsigned short Shutter_Width)
調整CMOS曝光時間

- ❑ void CMOS_SetWindowOrigin(unsigned short Row_Start,
unsigned short Column_Start)
設定取像視窗起始位置



CMOS函式庫

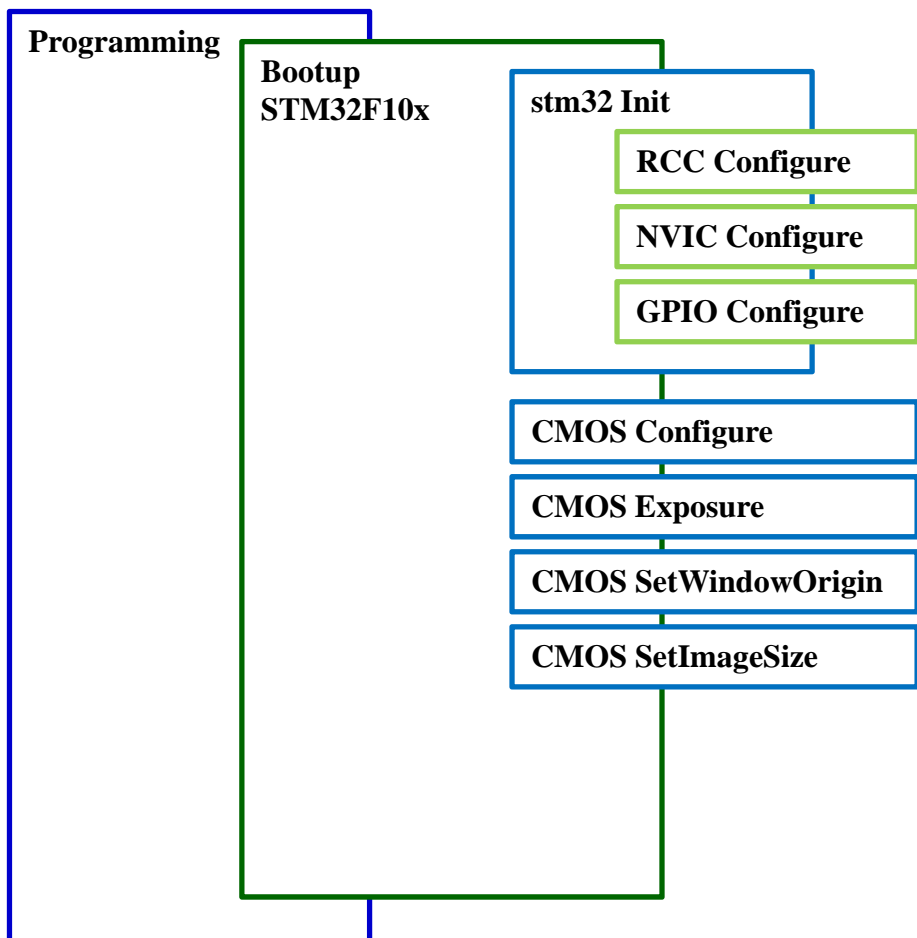
- ❑ void CMOS_SetImageSize(unsigned short Row_Width,
unsigned short Column_Width)
設定取像視窗大小

- ❑ void CMOS_Capture(unsigned char *CAM_Buffer)
取得CMOS影像存入CAM_Buffer記憶體區塊



範例說明

Embedded Software Side



Bootup STM32F10x

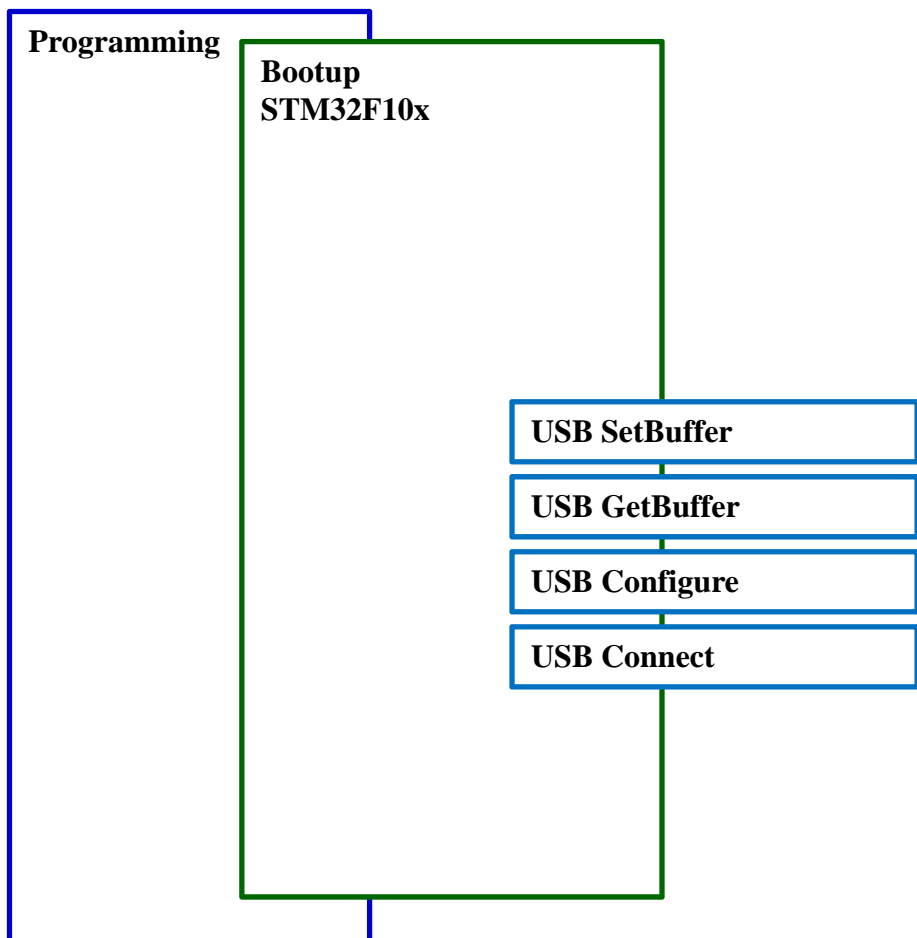
```
int main (void) {  
    CAM_Buffer=(unsigned char *)0x60040000;  
  
    stm32_Init ();           // STM32 Initialization  
    CMOS_Init();  
    CMOS_Exposure(0x0040);  
    CMOS_SetWindowOrigin(548,476); //548,476  
    CMOS_SetImageSize(200,768);  
}
```

設定CMOS參數



範例說明

Embedded Software Side



Bootup STM32F10x

```
USB_SetBuffer(CAM_Buffer,153600);
USB_GetBuffer(CMOS_Capture);

USB_Init();           // USB Initialization
USB_Connect(TRUE);    // USB Connect

while (1) {           // Loop forever
    ;
} // end while
} // end main
```

設定USB參數

啟動USB



參數說明

- Row_Start
 - Row_Start CMOS取像的Row起始位置
 - Row_Start必須大於等於 28
- Column_Start
 - Column_Start CMOS取像Column的起始位置
 - Column_Start必須大於等於 60
- Row_Width
 - CMOS取像Row的大小
 - Row_Width + Row_Start必須小於等於1228
- Column_Width
 - CMOS取像Column的大小
 - Column_Width + Column_Start必須小於等於1660



參數說明

- CAM_Buffer
 - CMOS影像輸出Buffer
 - 大小必須是512的倍數
 - 最大不可超過256K
- void USB_GetBuffer(void (*function)(unsigned char *))
 - 當執行SCSI Command D0指令時可呼叫一外部程式
例: CMOS_Capture(unsigned char *CAM_Buffer)



範例目錄架構

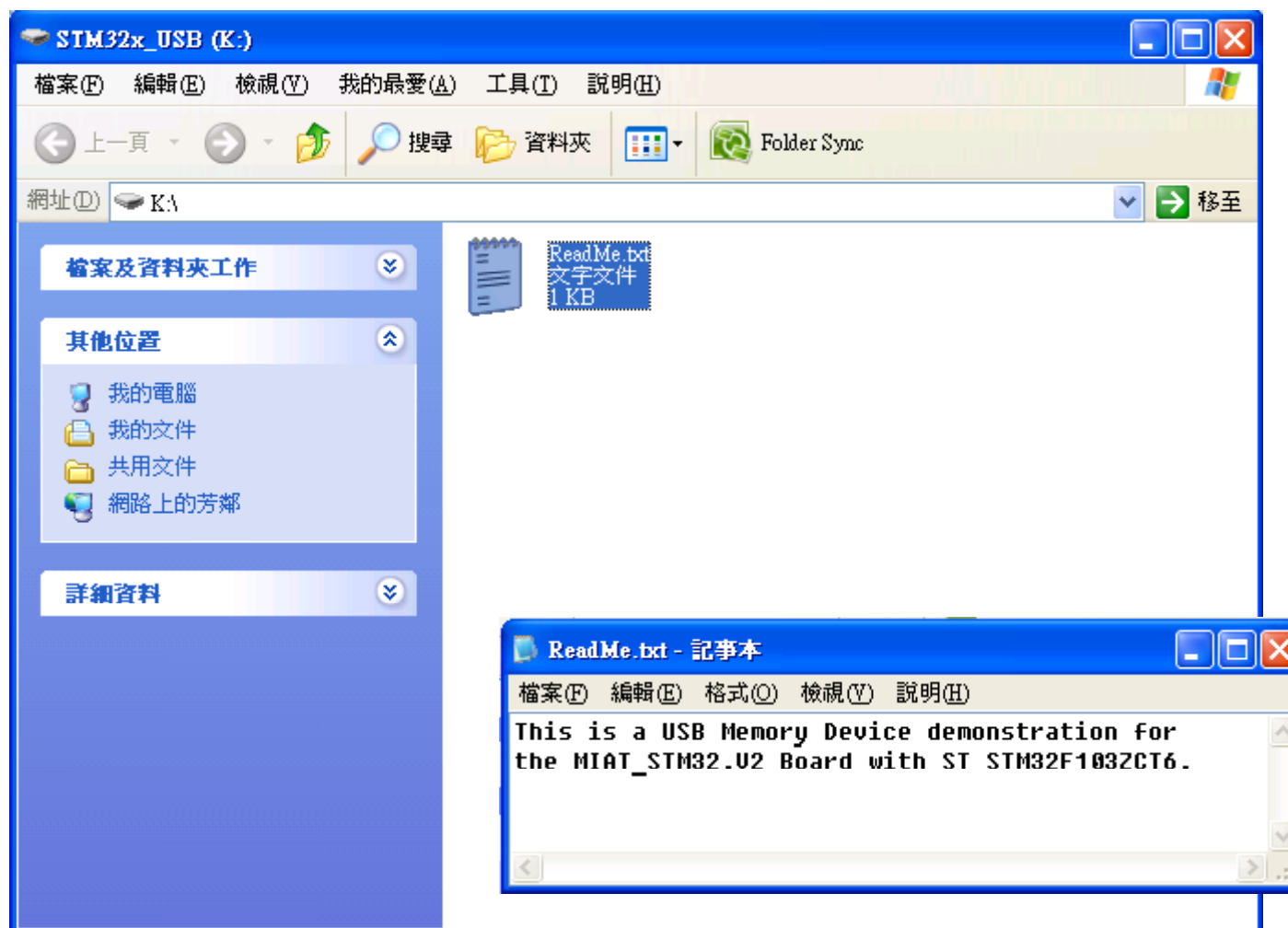
☐ 範例目錄

- 測試程式
- 測試映像檔
- 含括檔
- 函式庫
- 專案檔
- 原始碼



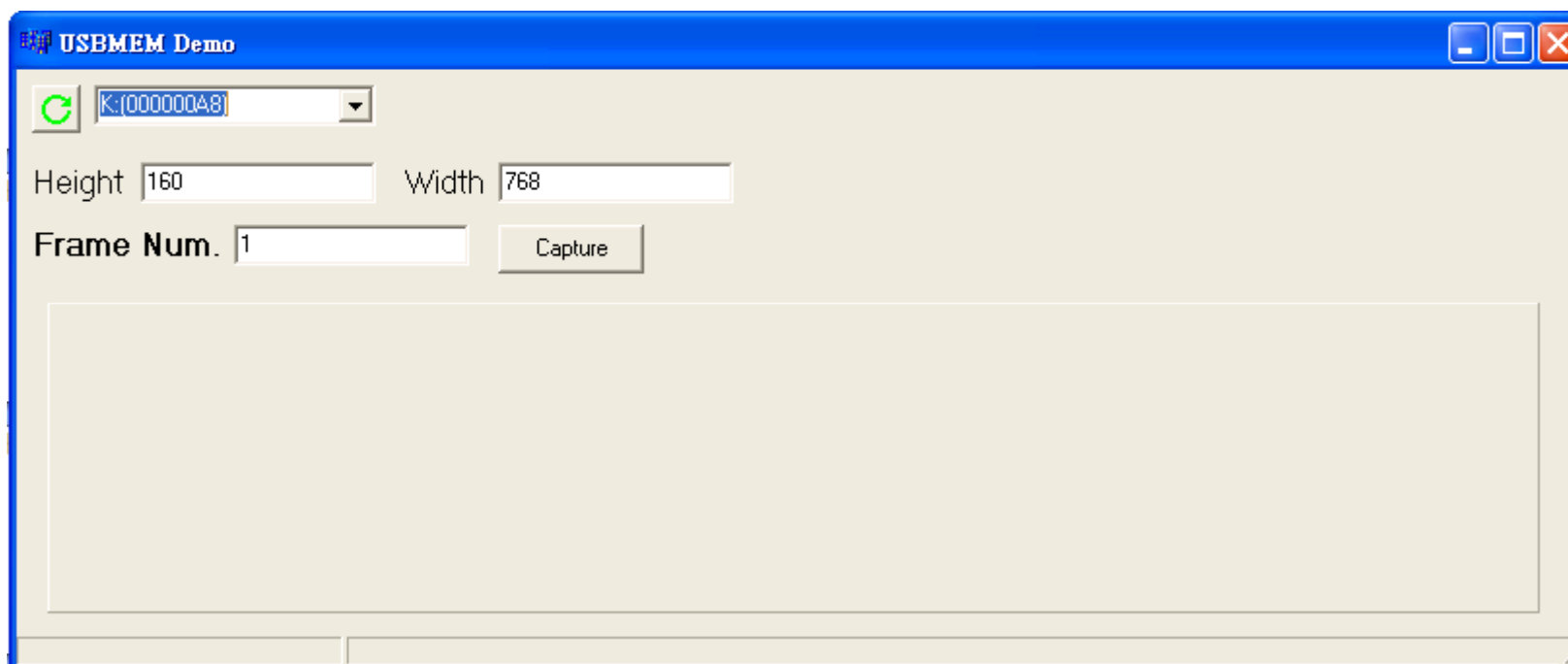


USB Mass Storage Device






PC端程式



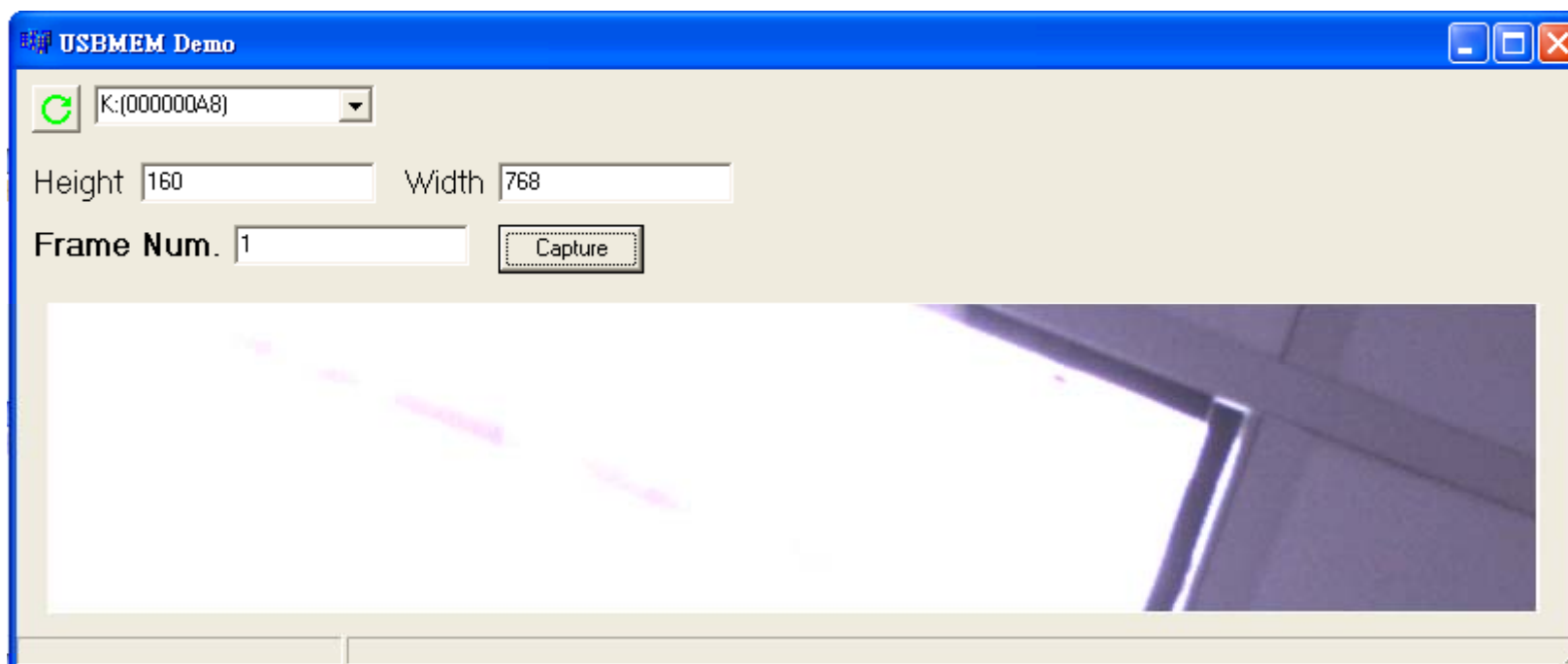


PC端程式

- ☐  搜尋MIAT_STM32裝置
- ☐ Height : 取得影像高
- ☐ Width : 取得影像寬
- ☐ Frame Num. : 連續取得張數
- ☐ Capture : 取得影像



取得影像



CMOS取像與USB資料傳輸實驗

實驗二



WU-YANG
Technology Co., Ltd.



實驗二練習

☐ 練習:

- 測試從PC取回CMOS影像資料
- 修改CAM_Buffer與CMOS取像大小測試是否正常
- 修改CMOS曝光時間、取像原點測試是否正常

Q & A



WU-YANG
Technology Co., Ltd.