嵌入式系統設計期中作業

SD CARD

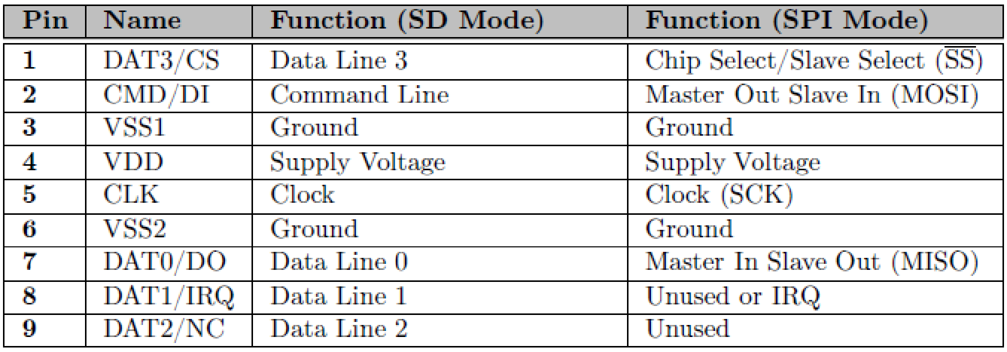
班級：資訊工程研究所在職專班

學號：995302020

姓名：宋俊賢

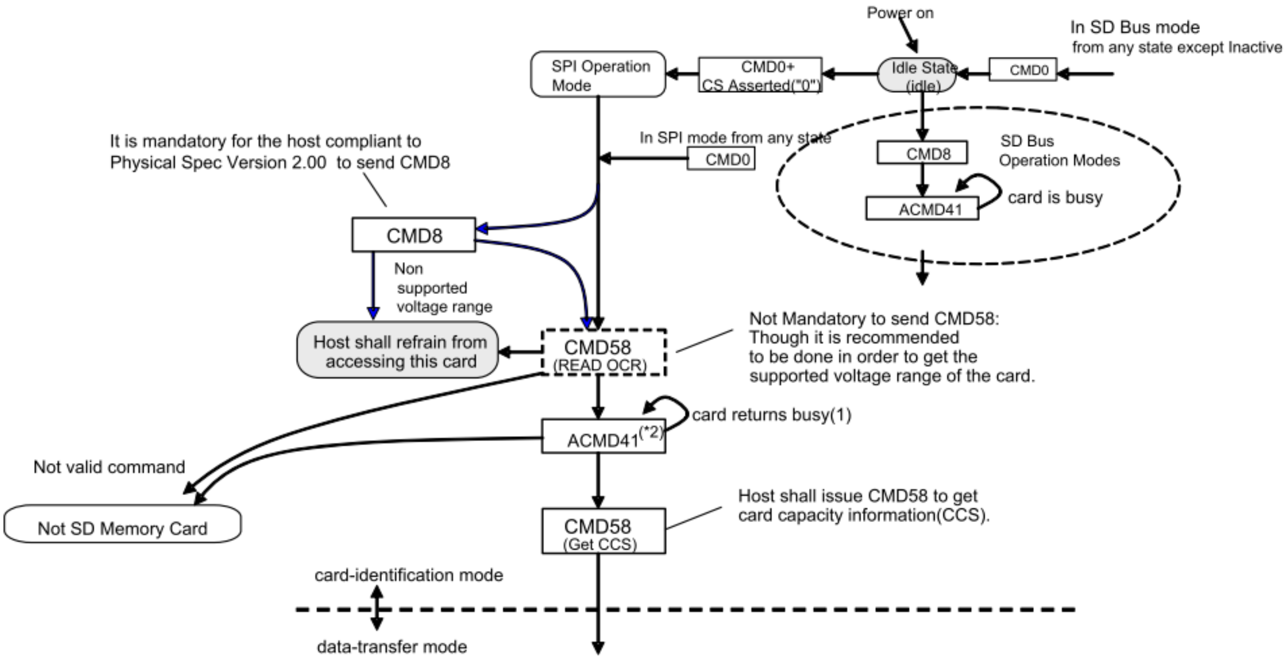
● 硬體週邊原理

1. SD Card Specification (SPI Mode)



Pin	Name	Function (SD Mode)	Function (SPI Mode)
1	DAT3/CS	Data Line 3	Chip Select/Slave Select ($\overline{SS}$)
2	CMD/DI	Command Line	Master Out Slave In (MOSI)
3	VSS1	Ground	Ground
4	VDD	Supply Voltage	Supply Voltage
5	CLK	Clock	Clock (SCK)
6	VSS2	Ground	Ground
7	DAT0/DO	Data Line 0	Master In Slave Out (MISO)
8	DAT1/IRQ	Data Line 1	Unused or IRQ
9	DAT2/NC	Data Line 2	Unused

a. SD Memory Card State Diagram

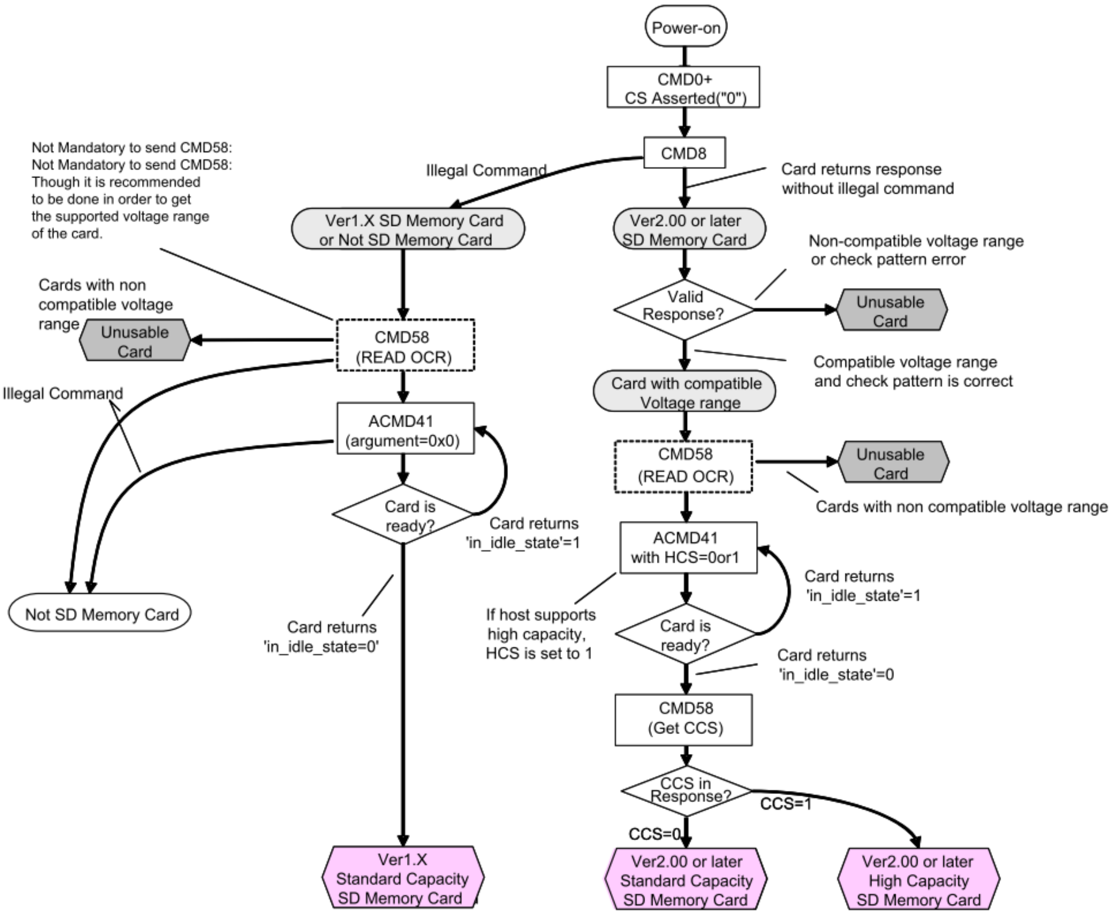

Power on
In SD Bus mode
from any state except Inactive
SPI Operation Mode
CMD0+ CS Asserted("0")
Idle State (idle)
CMD0
In SPI mode from any state
CMD0
CMD8
SD Bus Operation Modes
ACMD41
card is busy
It is mandatory for the host compliant to Physical Spec Version 2.00 to send CMD8
CMD8
Non supported voltage range
Host shall refrain from accessing this card
CMD58 (READ OCR)
Not Mandatory to send CMD58: Though it is recommended to be done in order to get the supported voltage range of the card.
card returns busy(1)
ACMD41(*2)
Not valid command
Not SD Memory Card
CMD58 (Get CCS)
Host shall issue CMD58 to get card capacity information(CCS).
card-identification mode
data-transfer mode

(*1) Note: Card returns busy when

- Card executes internal initialization process.
- If the card is High capacity SD Memory Card, there are 2 cases
  1. CMD8 was not issued before ACMD41
  2. ACMD41 is issued with HCS=0

(*2) Note: 2.1mm SD Memory Card can be initialized using CMD1 and Thin (1.4mm) SD Memory Card can be initialized using CMD1 only after firstly initialized by using CMD0 and ACMD41. In any of the cases CMD1 is not recommended because it may be difficult for the host to distinguish between MultiMediaCard and SD Memory Card.
If the SD card is initialized by CMD1 and the host treat it as MMC card, not SD card, the Data of the card may be damaged because of wrong interpretation of CSD and CID registers.

b. Mode Selection and Initialization



c. Bus Transfer Protection

I. Every SD Card command transferred on the bus is protected by CRC bits.

II. The SPI interface is initialized in the CRC OFF mode in default.

III. the RESET command (CMD0) that is used to switch the card to SPI mode, is received by the card while in SD mode and, therefore, shall have a valid CRC field. A valid reset command is: 0x95

IV. The CMD8 CRC verification is always enabled. The Host shall set correct CRC in the argument of CMD8. If CRC error is detected, card returns CRC error in R1 response regardless of command index

d. Command Format

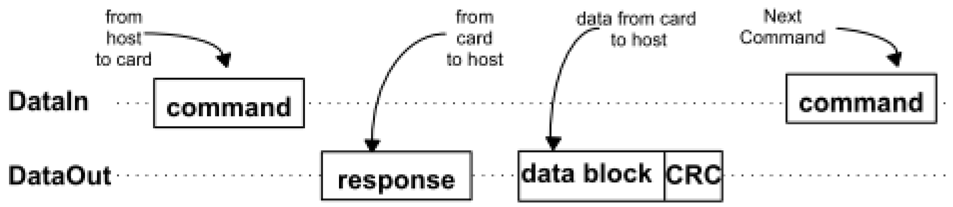
Bit position	47	46	[45:40]	[39:8]	[7:1]	0
Width (bits)	1	1	6	32	7	1
Value	'0'	'1'	x	x	x	'1'
Description	start bit	transmission bit	command index	argument	CRC7	end bit

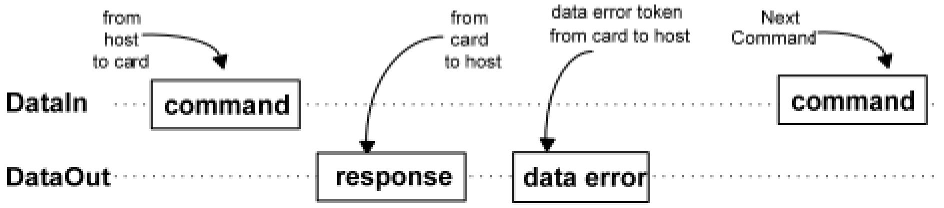
e. Data Read

I. The maximum block length is given by 512 Bytes regardless of READ_BL_LEN, defined in the CSD.

II. If partial block access is enabled in Standard Capacity Card (i.e. the CSD parameter READ_BL_PARTIAL equals 1), the block length can be any number between 1 and 512 Bytes.

III. The High Capacity SD Memory Card only supports 512-byte block length.





2. OCR：Operation Conditions Register (32 Bits)

Bit position	47	46	[45:40]	[39:8]	[7:1]	0
Width (bits)	1	1	6	32	7	1
Value	‘0’	‘0’	‘111111’	x	‘1111111’	‘1’
Description	start bit	transmission bit	reserved	OCR register	reserved	end bit

Bit 30 - Card capacity status (CCS) bit

a. this status bit is set to 1 if card is High Capacity SD Memory Card.

b. 0 indicates that the card is Standard Capacity SD Memory Card.

OCR bit position	OCR Fields Definition
0-3	reserved
4	reserved
5	reserved
6	reserved
7	Reserved for Low Voltage Range
8	reserved
9	reserved
10	reserved
11	reserved
12	reserved
13	reserved
14	reserved
15	2.7-2.8
16	2.8-2.9
17	2.9-3.0
18	3.0-3.1
19	3.1-3.2
20	3.2-3.3
21	3.3-3.4
22	3.4-3.5
23	3.5-3.6
24-29	reserved
30	Card Capacity Status (CCS)[1]
31	Card power up status bit (busy)[2]

VDD Voltage Window

[1] This bit is valid only when the card power up status bit is set.

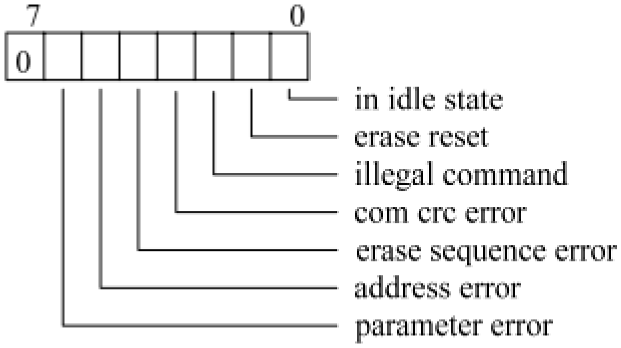
[2] This bit is set to LOW if the card has not finished the power up routine.

3. Detailed Command Description

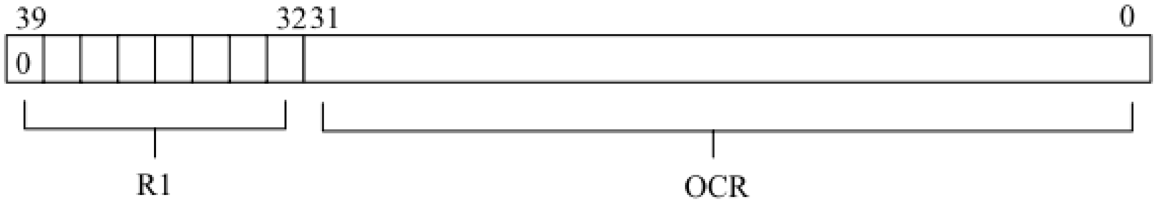
CMD INDEX	SPI Mode	Argument	Resp	Abbreviation	Command Description
CMD0	Yes	[31:0] stuff bits	R1	GO_IDLE_STATE	Resets the SD Memory Card
CMD1	Yes[1]	[31]Reserved bit [30]HCS [29:0]Reserved bits	R1	SEND_OP_ COND	Sends host capacity support information and activates the card's initialization process. HCS is effective when card receives SEND_IF_COND command. Reserved bits shall be set to '0'.
CMD8[9]	Yes	[31:12]Reserved bits [11:8]supply voltage(VHS) [7:0]check pattern	R7	SEND_IF_COND	Sends SD Memory Card interface condition that includes host supply voltage information and asks the accessed card whether card can operate in supplied voltage range. Reserved bits shall be set to '0'.
CMD55	Yes	[31:0] stuff bits	R1	APP_CMD	Defines to the card that the next command is an application specific command rather than a standard command
CMD58	Yes	[31:0] stuff bits	R3	READ_OCR	Reads the OCR register of a card. CCS bit is assigned to OCR[30].
ACMD41	Yes	[31]Resetved bit [30]HCS [29:0]Reserved bits	R1	SD_SEND_OP_COND	Sends host capacity support information and activates the card's initialization process. Reserved bits shall be set to '0'

4. Responses

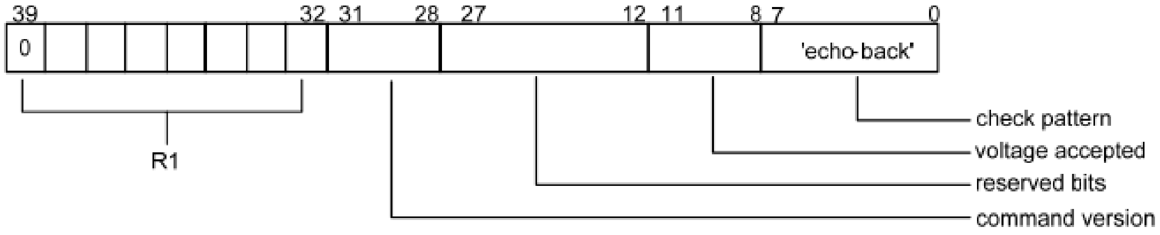
a. Format R1



b. Format R3

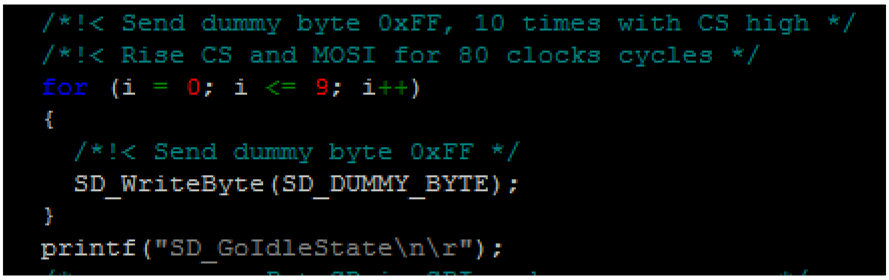


c. Format R7



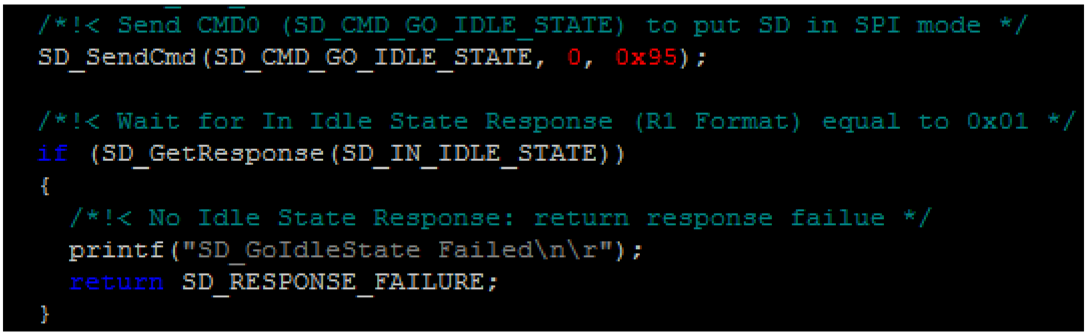
- # SIOC 的 Driver 設計及主要程式說明

1. SPI 的 CLK 不能超過 400KHz，所以要先將 SPI 的設定低於 400KHz，接下來要產生"大於"74 個 SPICLK，以穩定電壓



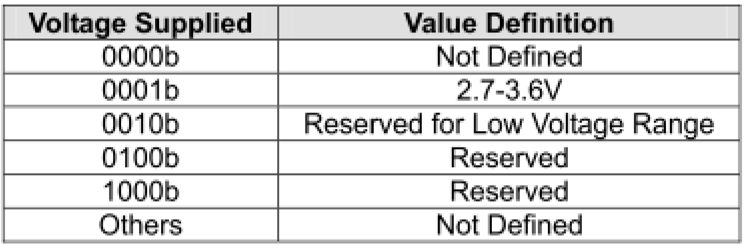
```
/*!< Send dummy byte 0xFF, 10 times with CS high */
/*!< Rise CS and MOSI for 80 clocks cycles */
for (i = 0; i <= 9; i++)
{
  /*!< Send dummy byte 0xFF */
  SD_WriteByte(SD_DUMMY_BYTE);
}
printf("SD_GoIdleState\n\r");
```

2. 爲了使 SD 卡初始化進入 SPI 模式，我們需要使用的命令有 3 個：
   a. CMD0,ACMD41,CMD55
   b. 使用 ACMD 類的指令前應先發 CMD55，CMD55 起到一個切換到 ACMD 類命令的作用。
3. CMD 0 (命名爲 GO_IDLE_STATE)，當指令送出去後，回從 R1 暫存器反回一個值，來 check 我們所送去 SD 卡的指令是否正確，CRC 值須使用有效的 reset value ：0x95



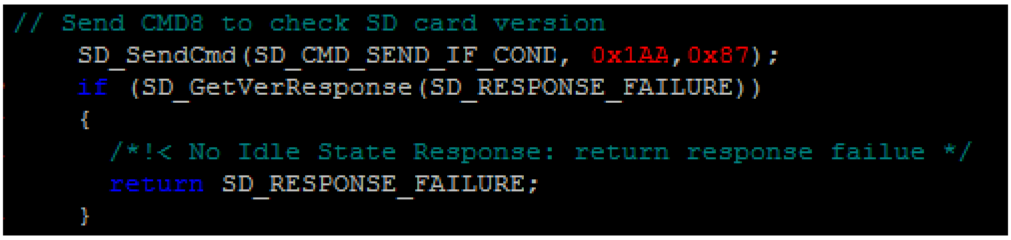
```
/*!< Send CMD0 (SD_CMD_GO_IDLE_STATE) to put SD in SPI mode */
SD_SendCmd(SD_CMD_GO_IDLE_STATE, 0, 0x95);

/*!< Wait for In Idle State Response (R1 Format) equal to 0x01 */
if (SD_GetResponse(SD_IN_IDLE_STATE))
{
  /*!< No Idle State Response: return response failue */
  printf("SD_GoIdleState Failed\n\r");
  return SD_RESPONSE_FAILURE;
}
```

4. CMD 8 (SEND_IF_COND)可以用來做 SD 卡的型別判斷，辨別 SD 1.x 或 SD 2.x，0x1AA 在這裡的用途是判斷 SD 2.x 的最重要依據，如果 R1 返回的值是 0x1，表示 CMD 8 的 argument 被接受，代表此卡片爲 SD 2.x 的，否則要看 CMD 58 (READ_OCR)的 return 決定。
   a. The card assumes the voltage specified in VHS as the current supplied voltage. Only 1-bit of VHS shall be set to 1 at any given time.



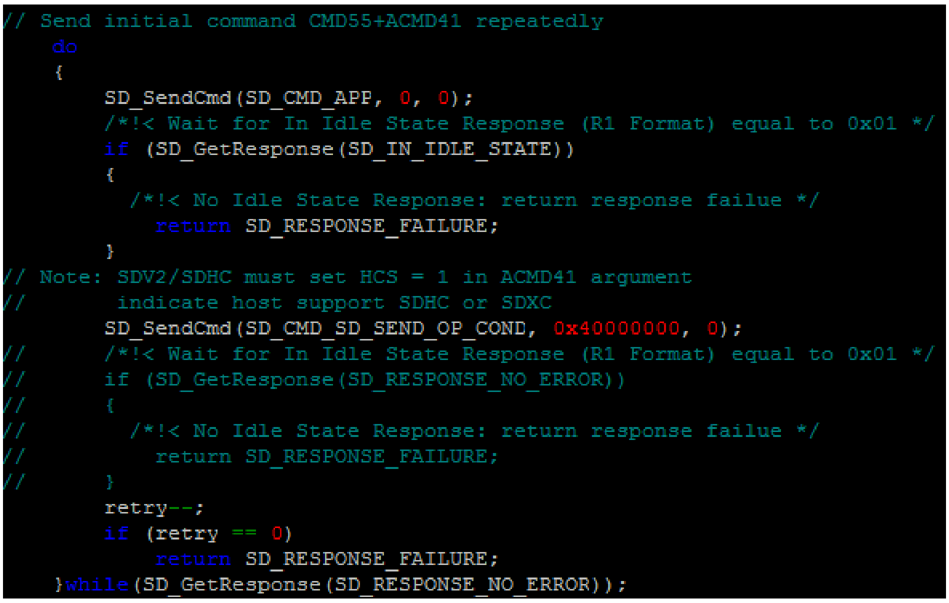
Voltage Supplied	Value Definition
0000b	Not Defined
0001b	2.7-3.6V
0010b	Reserved for Low Voltage Range
0100b	Reserved
1000b	Reserved
Others	Not Defined

   b. It is recommended to use ‘10101010b’ for the ‘check pattern’

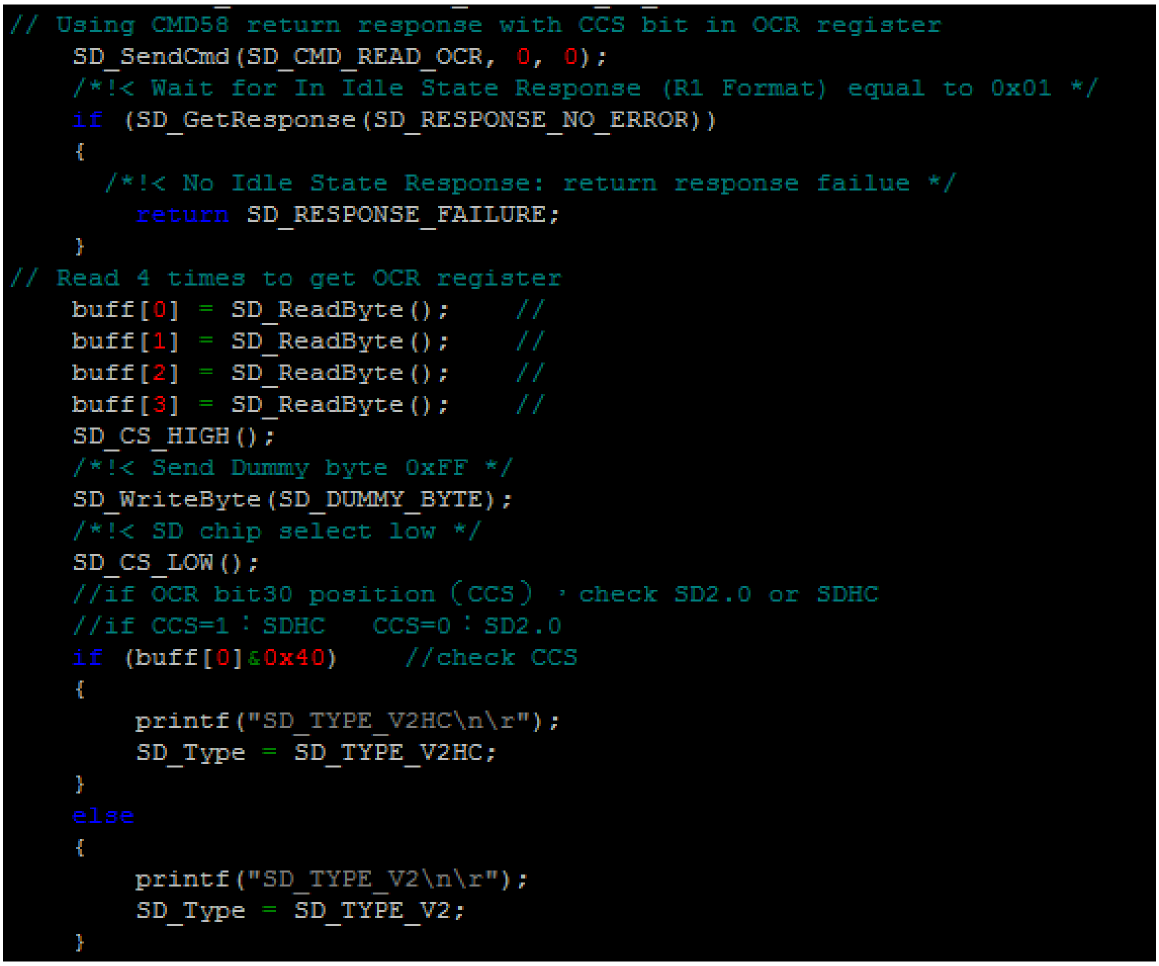


```
// Send CMD8 to check SD card version
    SD_SendCmd(SD_CMD_SEND_IF_COND, 0x1AA,0x87);
    if (SD_GetVerResponse(SD_RESPONSE_FAILURE))
    {
      /*!< No Idle State Response: return response failue */
      return SD_RESPONSE_FAILURE;
    }
```

5. ACMD 這一類的指令是不能單獨存在的，一定要搭配 CMD 55 (APP_CMD)
此 ACMD 41 所要輸入的參數為 0x40000000，才能進行 SD 2.x 與 SDHC 的判斷，判斷完之後利用 CMD 58 來讀取 SD 卡內的 OCR 暫存器



```
// Send initial command CMD55+ACMD41 repeatedly
    do
    {
        SD_SendCmd(SD_CMD_APP, 0, 0);
        /*!< Wait for In Idle State Response (R1 Format) equal to 0x01 */
        if (SD_GetResponse(SD_IN_IDLE_STATE))
        {
          /*!< No Idle State Response: return response failue */
            return SD_RESPONSE_FAILURE;
        }
// Note: SDV2/SDHC must set HCS = 1 in ACMD41 argument
//        indicate host support SDHC or SDXC
        SD_SendCmd(SD_CMD_SD_SEND_OP_COND, 0x40000000, 0);
//      /*!< Wait for In Idle State Response (R1 Format) equal to 0x01 */
//      if (SD_GetResponse(SD_RESPONSE_NO_ERROR))
//      {
//        /*!< No Idle State Response: return response failue */
//          return SD_RESPONSE_FAILURE;
//      }
        retry--;
        if (retry == 0)
            return SD_RESPONSE_FAILURE;
    }while(SD_GetResponse(SD_RESPONSE_NO_ERROR));
```

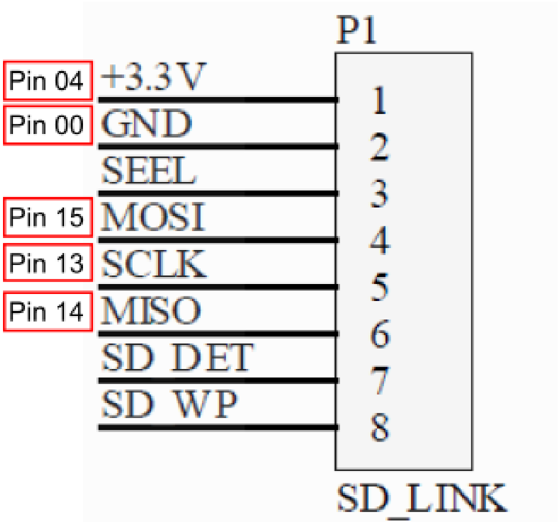


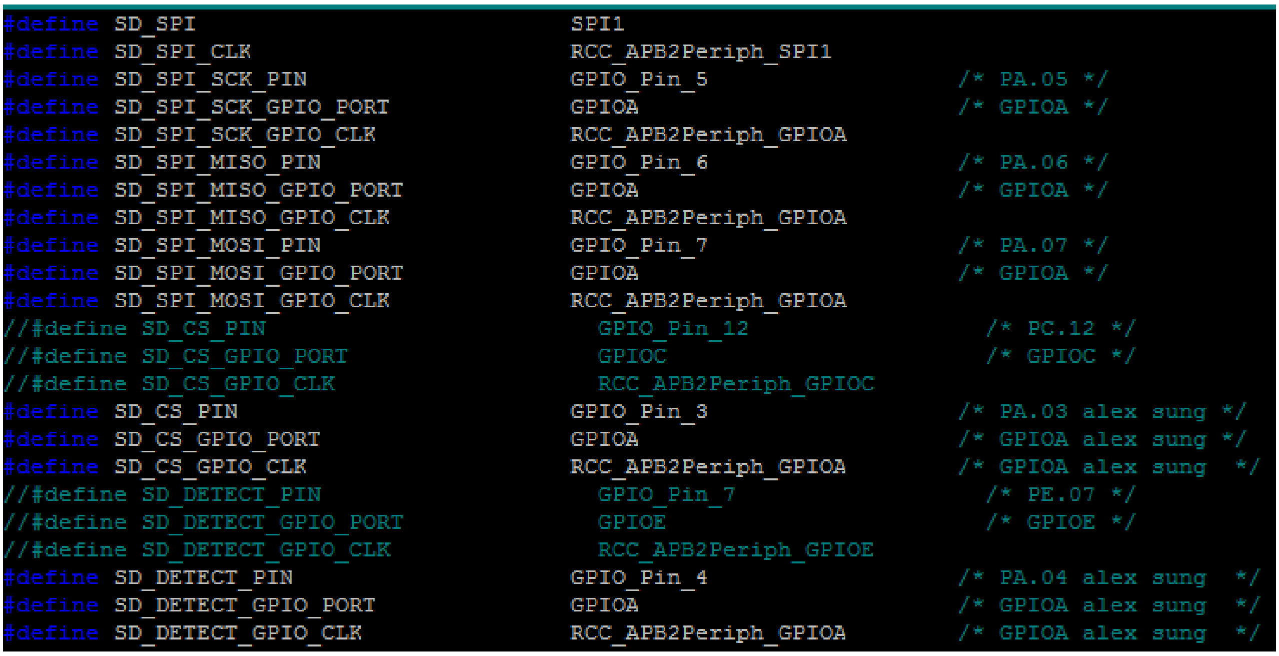
```
// Using CMD58 return response with CCS bit in OCR register
    SD_SendCmd(SD_CMD_READ_OCR, 0, 0);
    /*!< Wait for In Idle State Response (R1 Format) equal to 0x01 */
    if (SD_GetResponse(SD_RESPONSE_NO_ERROR))
    {
      /*!< No Idle State Response: return response failue */
        return SD_RESPONSE_FAILURE;
    }
// Read 4 times to get OCR register
    buff[0] = SD_ReadByte();    //
    buff[1] = SD_ReadByte();    //
    buff[2] = SD_ReadByte();    //
    buff[3] = SD_ReadByte();    //
    SD_CS_HIGH();
    /*!< Send Dummy byte 0xFF */
    SD_WriteByte(SD_DUMMY_BYTE);
    /*!< SD chip select low */
    SD_CS_LOW();
    //if OCR bit30 position（CCS），check SD2.0 or SDHC
    //if CCS=1：SDHC   CCS=0：SD2.0
    if (buff[0]&0x40)    //check CCS
    {
        printf("SD_TYPE_V2HC\n\r");
        SD_Type = SD_TYPE_V2HC;
    }
    else
    {
        printf("SD_TYPE_V2\n\r");
        SD_Type = SD_TYPE_V2;
    }
```

6. 正確的回應內容應該是：
   a. CMD0——0x01（SD 卡處於 in-idle-state）
   b. CMD55——0x01（SD 卡處於 in-idle-state）
   c. ACMD41——0x00（SD 卡跳出 in-idle-state，完成初始化準備接受下一條指令）

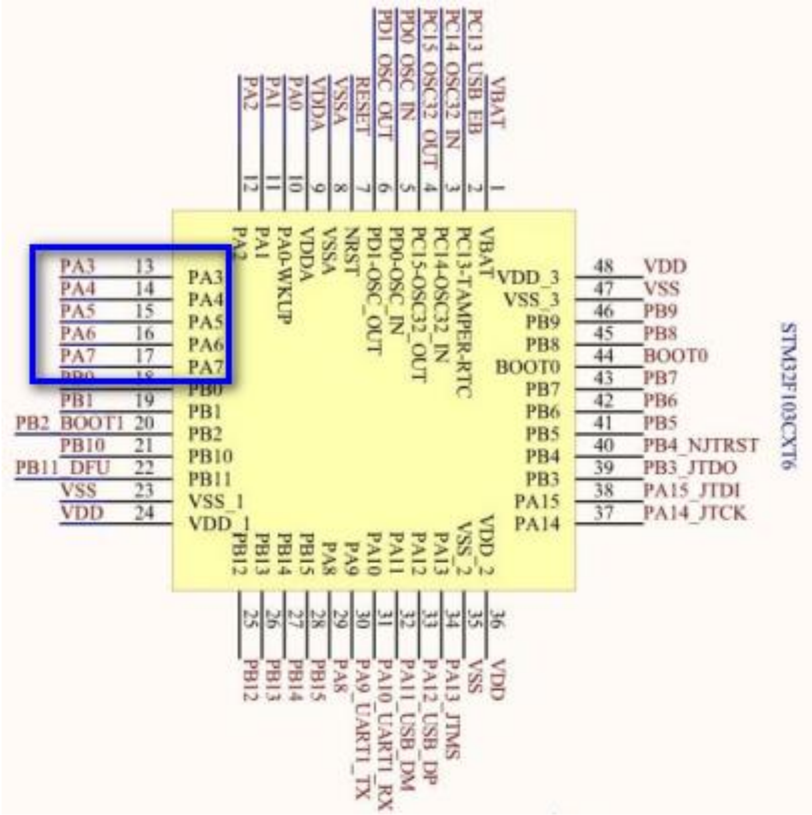
(如果最後的回應內容還是 0x01 的話，可以循環發送 CMD55+ACMD41，直到回應的內容 0x00。)

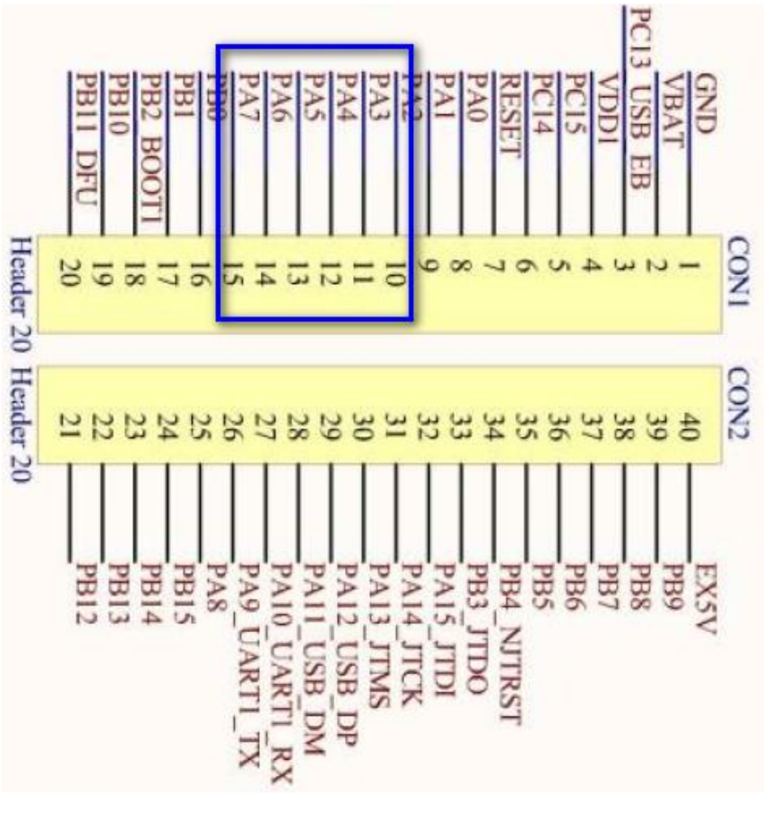
- SIOC 與週邊的接線圖



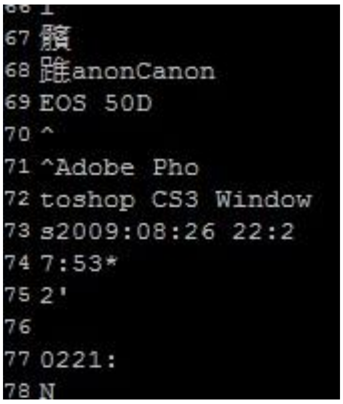


```
#define SD_SPI                           SPI1
#define SD_SPI_CLK                       RCC_APB2Periph_SPI1
#define SD_SPI_SCK_PIN                   GPIO_Pin_5                   /* PA.05 */
#define SD_SPI_SCK_GPIO_PORT             GPIOA                        /* GPIOA */
#define SD_SPI_SCK_GPIO_CLK              RCC_APB2Periph_GPIOA
#define SD_SPI_MISO_PIN                  GPIO_Pin_6                   /* PA.06 */
#define SD_SPI_MISO_GPIO_PORT            GPIOA                        /* GPIOA */
#define SD_SPI_MISO_GPIO_CLK             RCC_APB2Periph_GPIOA
#define SD_SPI_MOSI_PIN                  GPIO_Pin_7                   /* PA.07 */
#define SD_SPI_MOSI_GPIO_PORT            GPIOA                        /* GPIOA */
#define SD_SPI_MOSI_GPIO_CLK             RCC_APB2Periph_GPIOA
//#define SD_CS_PIN                        GPIO_Pin_12                  /* PC.12 */
//#define SD_CS_GPIO_PORT                  GPIOC                        /* GPIOC */
//#define SD_CS_GPIO_CLK                   RCC_APB2Periph_GPIOC
#define SD_CS_PIN                        GPIO_Pin_3                   /* PA.03 alex sung */
#define SD_CS_GPIO_PORT                  GPIOA                        /* GPIOA alex sung */
#define SD_CS_GPIO_CLK                   RCC_APB2Periph_GPIOA         /* GPIOA alex sung  */
//#define SD_DETECT_PIN                    GPIO_Pin_7                   /* PE.07 */
//#define SD_DETECT_GPIO_PORT              GPIOE                        /* GPIOE */
//#define SD_DETECT_GPIO_CLK               RCC_APB2Periph_GPIOE
#define SD_DETECT_PIN                    GPIO_Pin_4                   /* PA.04 alex sung  */
#define SD_DETECT_GPIO_PORT              GPIOA                        /* GPIOA alex sung  */
#define SD_DETECT_GPIO_CLK               RCC_APB2Periph_GPIOA         /* GPIOA alex sung  */
```

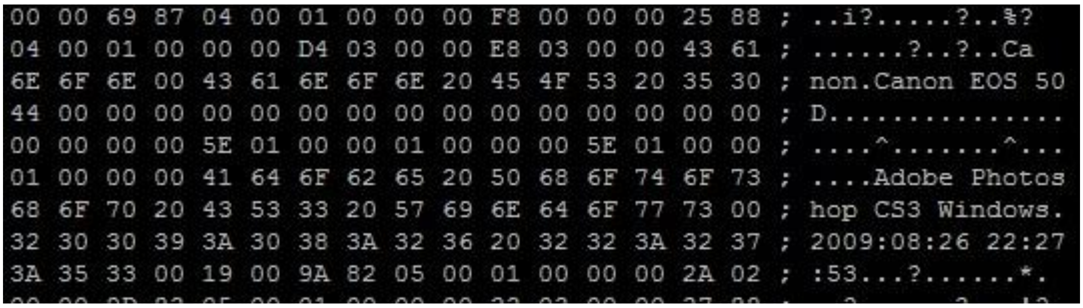

STM32F103CXT6
VBAT
PC13_USB_EB
PC14_OSC32_IN
PC15_OSC32_OUT
PD0_OSC_IN
PD1_OSC_OUT
RESET
VSSA
VDDA
PA0
PA1
PA2
PC13-TAMPER-RTC
PC14-OSC32_IN
PC15-OSC32_OUT
PD0-OSC_IN
PD1-OSC_OUT
NRST
PA0-WKUP
PA3
PA4
PA5
PA6
PA7
PB0
PB1
PB2_BOOT1
PB10
PB11_DFU
VSS
VDD
VSS_1
VDD_1
PB12
PB13
PB14
PB15
PA8
PA9_UART1_TX
PA10_UART1_RX
PA11_USB_DM
PA12_USB_DP
PA13_JTMS
VSS_2
VDD_2
VDD_3
VSS_3
PB9
PB8
BOOT0
PB7
PB6
PB5
PB4_NJTRST
PB3_JTDO
PA15_JTDI
PA14_JTCK


CON1
Header 20
CON2
Header 20
GND
VBAT
PC13_USB_EB
VDDI
PC15
PC14
RESET
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PB0
PB1
PB2_BOOT1
PB10
PB11_DFU
EX5V
PB9
PB8
PB7
PB6
PB5
PB4_NJTRST
PB3_JTDO
PA15_JTDI
PA14_JTCK
PA13_JTMS
PA12_USB_DP
PA11_USB_DM
PA10_UART1_RX
PA9_UART1_TX
PA8
PB15
PB14
PB13
PB12

- SIOC 驗證



```
1
髕
踓anonCanon
EOS 50D
^
^Adobe Pho
toshop CS3 Window
s2009:08:26 22:2
7:53*
2'

0221:
N
```



```
00 00 69 87 04 00 01 00 00 00 F8 00 00 00 25 88 ; ..i?.....?..%?
04 00 01 00 00 00 D4 03 00 00 E8 03 00 00 43 61 ; ......?..?..Ca
6E 6F 6E 00 43 61 6E 6F 6E 20 45 4F 53 20 35 30 ; non.Canon EOS 50
44 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; D...............
00 00 00 00 5E 01 00 00 01 00 00 00 5E 01 00 00 ; ....^.......^...
01 00 00 00 41 64 6F 62 65 20 50 68 6F 74 6F 73 ; ....Adobe Photos
68 6F 70 20 43 53 33 20 57 69 6E 64 6F 77 73 00 ; hop CS3 Windows.
32 30 30 39 3A 30 38 3A 32 36 20 32 32 3A 32 37 ; 2009:08:26 22:27
3A 35 33 00 19 00 9A 82 05 00 01 00 00 00 2A 02 ; :53...?......*.
```