

# HT66FV1x0 內置音頻放大器應用須知

文件編碼：AN0486TC

## 簡介

HOLTEK MCU HT66FV1x0 系列，內建了一個支援數位元音量調節的 16-bit DAC、一組 AB 類的音頻功率放大器、一組硬體 SPI 以及硬體 UART 等資源，可應用於家電、醫療、安防等需要語音功能的消費性電子產品的開發。上述 MCU 內建 1.5W 高功率音頻功率放大器以及數位元音量控制功能，滿足客戶對於高質量聲音需求，省去外置一顆功放 IC、能精簡外部零件、整體成本更具競爭優勢，非常適合應用在各種語音播放產品。

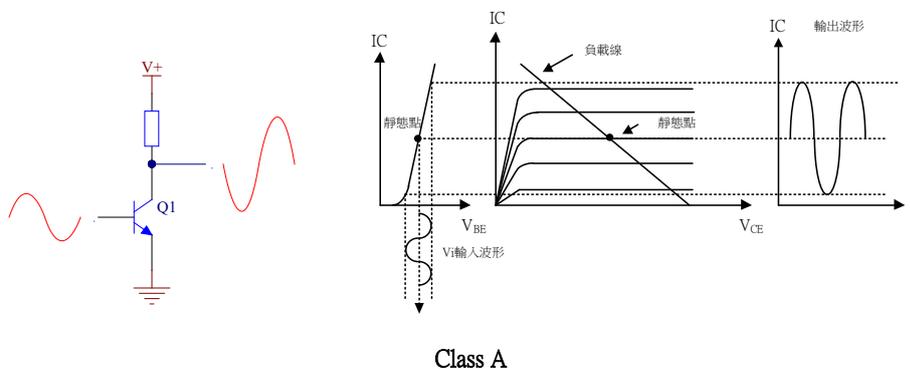
以下以 HT66FV1x0 系列為例，將介紹 MCU 內建的音頻功率放大器之原理與特性及使用方法，希望能給用戶帶來幫助。

## 功能說明

功率放大器是音頻系統中最基本的設備。作用是將音源或前級放大器輸入的微弱信號進行放大後，產生足夠大的電流去推動喇叭進行聲音的播放。常見功率放大器類型分為：Class A、Class B、Class AB、Class D。

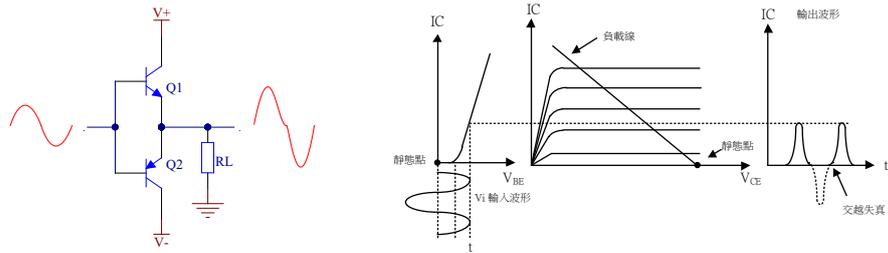
### Class A 原理

Class A 為最簡單的放大器類型，電晶體在輸入信號的整個週期內均導通，導通角度為 360 度。由於放大器工作在特性曲線的線性範圍內，所以瞬態失真和交替失真較小，但效率很低。放大器的靜態工作點設定在負載線的中點附近，負載特性曲線直觀的反應出負載電壓電流變化關係，負載線與電晶體輸出特性曲線的交點即是靜態工作點，用於分析電路的靜態工作點及失真情況。



### Class B 原理

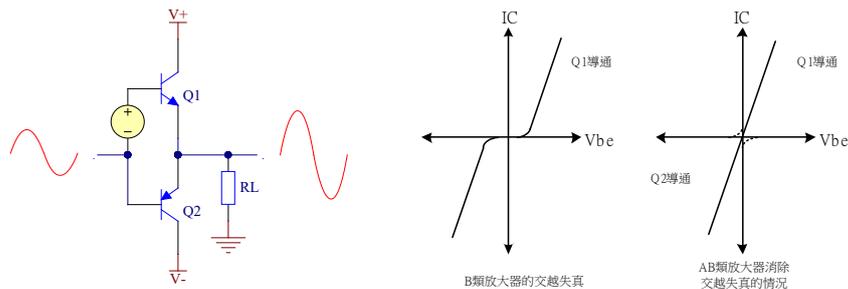
Class B 靜態工作點選在基極電流為零的那條輸出特性曲線上，即負載線的截止點，輸出級的電晶體只在信號波形的半個週期(180 度)導通。為了對整個信號進行放大，使用了兩個電晶體，一個用於正輸出信號，另一個用於負輸出信號，並在輸出端接假性負載 RL 替代功率放大器輸出埠接收功率的元件，通常接假性負載做功放電路的調試，調試完再接真負載(喇叭)。Class B 的效率遠遠高於 Class A，但由於兩個電晶體從通到斷過程中存在交越點，失真較大。



Class B

### Class AB 原理

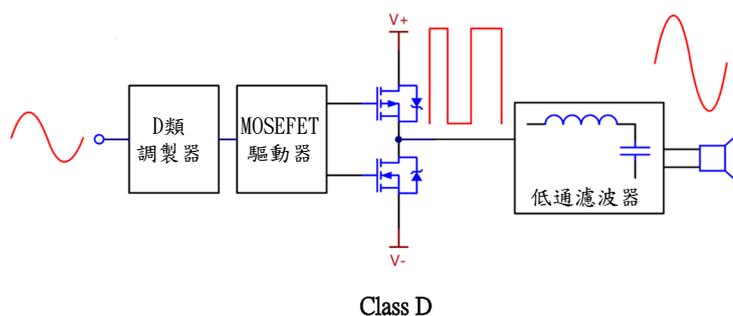
Class AB 是兼具 Class A 和 Class B 優勢的設計，Class AB 的靜態工作點低於 Class A、高於 Class B，選在負載線的中點和截止點之間，效率高於 Class A，失真低於 Class B，導通角度為 180 度到 360 度。通過對電路中的兩個電晶體進行偏置，信號接近零的時候，兩個電晶體都是導通的，但其中的電流很小，類似於 Class A；大信號時，由於信號的作用使其中的一隻電晶體截止的時候，另一隻電晶體則一定是導通的，兩個電晶體始終是交替導通的，類似於 Class B。Class B 的交越失真是由於信號大小在  $-0.6V < V_i < 0.6V$  之間時，兩個電晶體都無法導通所引起的，而 Class AB 在兩個電晶體間加  $V_{BB}$  偏壓，以降低交越失真。



Class AB

### Class D 原理

Class D 工作原理是基於開關 MOS 電晶體，可在極短的時間內完全導通和完全截止，因此效率非常高，但是開關工作模式增加了輸出信號的失真。D 類調製器是由比較器和三角波發生器組成的 PWM 調製器，用輸入信號經過積分器濾波後的修正信號對三角波進行調製產生調製輸出方波，經 MOSFET 驅動器進行大功率電壓電流放大，放大後的數位信號經過低通濾波器濾波後，還原出音頻信號。



### Class AB 與其他功率放大器優缺點比較

Class AB 的效率高於 Class A，失真度低於 Class B。而與 Class D 比較，由於 Class D 的工作模式完全不同於 Class AB，會產生某些高頻諧波，EMI 電磁干擾比較明顯，另外 Class AB 較 Class D 音頻放大器在成本上有明顯優勢，Class D 一般是 Class AB 的 2 至 3 倍價格。Class AB 功率放大器具有效率高、失真小、價格低，是目前音頻功率放大器應用中最為廣泛的設計。

下表列出了各類放大器的優缺點：

	Class A	Class B	Class AB	Class D
工作點位置	負載線中點	負載線截止點	負載線中點與截止點之間	功放管工作在開關狀態
導通角度	$\theta = 360^\circ$	$\theta = 180^\circ$	$180^\circ < \theta < 360^\circ$	開關狀態
失真度	失真小	失真度高於 AB 類有交叉失真	可消除交叉失真	失真大
功率轉移效率	效率最低，在 50% 以下	效率約為 50% 至 78.5%	效率約為 50% 至 78.5%	效率最高，在 85% 以上
主要用途	失真度低的小功率放大器	大功率放大器	一般的音響擴大機	高功率或效率放大器

### 功率放大器電氣特性

HT66FV1x0 內建的 Class AB 功率放大器的電氣特性，如下表所示 ( $T_a = 25^\circ\text{C}$ )。

#### 總諧波失真 (THD, Total Harmonic Distortion)

當某一特定頻率的正弦信號輸入到功率放大器時，因放大器內部線路或外部元件非線性失真等因素產生了基於輸入頻率倍數的諧波，這些諧波幅值的均方根值與輸入頻率幅值間的比值稱為總諧波失真。

#### 噪聲 (N, Noise)

功放功率放大輸出的信號除了上面描述的諧波噪聲外，還有一些因電路和器件引起的其他頻率的雜波，如熱噪聲等。

通常將總諧波失真和噪聲合起來表述輸出的噪聲指標，通常是越小越好。HT66FV1x0 MCU，(THD+N)/S 典型值可達到 0.2%。

#### 最大輸出功率 ( $P_{out}$ )

反應了一個音頻功率放大器的負載能力，通常音頻放大器廠家會提供產品在工作電壓一定條件和額定負載下的最大輸出功率。MCU 內置功放在工作電壓為 5V，(THD+N)/S=10% 時，功率可以達到 1.5W。

用戶需根據功放的輸出功率和阻抗來選擇合適的喇叭，一般喇叭的功率略高於功放的功率，

喇叭的阻抗參數建議與下表中的功放輸出功率的負載參數相同。

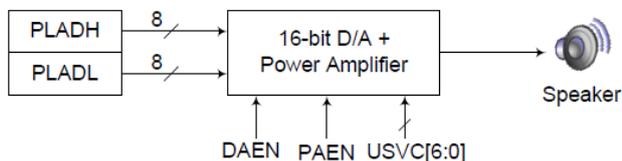
符號	參數	測試條件		最小	典型	最大	單位
		V <sub>DD</sub>	條件				
AV <sub>DD,PA</sub>	音頻功率放大器工作電壓	—	—	2.2	—	5.5	V
(THD+N)/S	(THD+N)/S	5V	8Ω 負載 輸出功率=500mW	—	0.2	—	%
P <sub>OUT</sub>	輸出功率	3V	8Ω 負載 (THD+N)/S=1%	—	410	—	mW
			8Ω 負載 (THD+N)/S=10%	—	550	—	mW
		5V	8Ω 負載 (THD+N)/S=1%	—	1200	—	mW
			8Ω 負載 (THD+N)/S=10%	—	1500	—	mW

註：正弦波輸入@1kHz & -6dB。

## 工作原理

HT66FV1x0 內建 Class AB 類高輸出功率的音頻放大器，下面將介紹 Class AB。由 HT66FV1x0 內建的 SPI 介面讀取外掛 Flash ROM IC 中的數位音訊資料給 16-bit DAC 的資料暫存器 PLADH & PLADL，經 DAC 轉換成對應的音頻類比電壓信號，由喇叭播出聲音。

HT66FV1x0 語音播放功能的方塊圖如下圖所示，DAEN 和 PAEN 分別是 DAC 和功放的使能位元，語音停止播放時，可除能來降低功耗；USVC[6:0]用於數位元式的音量調節。



### 語音播放控制器相關暫存器

HT66FV1x0 的語音播放功能由一系列暫存器控制，兩個控制暫存器用於控制 16 位 D/A 轉換器和功率放大器，再加上喇叭的靜音控制。兩對數據暫存器用於儲存要播放的數據。

#### USVC 暫存器

Bit	7	6	5	4	3	2	1	0
Name	MUTEB	USVC6	USVC5	USVC4	USVC3	USVC2	USVC1	USVC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

- Bit7 **MUTEB**：喇叭靜音控制  
0：喇叭輸出靜音  
1：喇叭輸出使能
- Bit6~0 **USVC6~USVC0**：喇叭音量控制

這些位元用於調節輸出的音量，其範圍為-32dB~6dB。

**PLAC 暫存器**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	PAEN	DAEN
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	0

- Bit7~2 未定義，讀為 0
- Bit1 **PAEN**：功率放大器使能控制  
0：除能  
1：使能
- Bit0 **DAEN**：16 位 D/A 轉換器使能控制  
0：除能  
1：使能

需要注意的是，當微控制器進入空閒/休眠，16 位元 D/A 轉換器和功率放大器都將除能。

**PLADL 暫存器**

Bit	7	6	5	4	3	2	1	0
Name	P_D7	P_D6	P_D5	P_D4	P_D3	P_D2	P_D1	P_D0
R/W								
POR	0	0	0	0	0	0	0	0

- Bit7~0 **P\_D7~P\_D0**：放音數據低位元組暫存器 bit7~bit0

該暫存器用於儲存 16 位元放音數據的低位元組。

**PLADH 暫存器**

Bit	7	6	5	4	3	2	1	0
Name	P_D15	P_D14	P_D13	P_D12	P_D11	P_D10	P_D9	P_D8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

- Bit7~0 **P\_D15~P\_D8**：放音數據高位元組暫存器 bit7~bit0

該暫存器用於儲存 16 位元放音數據的高位元組。注意，如果需要更新 16 位元放音數據，

應在放音數據高位元組暫存器被寫入前先修改放音數據低位元組暫存器。



## PCB Layout 注意事項

- 元件擺放時優先考慮電源濾波電容，其擺放盡可能靠近 MCU，SPI Flash ROM 之相關位置擺放方式也儘可能靠近 MCU 為原則，特別是 SPI Clock 拉線盡可能越短越好。
- 避免 Audio 功率放大器操作時，瞬間大電流所產生的噪聲干擾，需區分為數位電源( $V_{DD}$ )及類比電源( $AV_{DD,PA}$ )，共兩組獨立電源。
- Audio 功率放大器電源端引腳  $AV_{DD,PA}$  需直接從電源正端拉線，且走線線寬不小於 12mil。
- 避免 Audio 功率放大器操作時，瞬間大電流所產生的噪聲干擾，需區分為數位接地( $V_{SS}$ )及類比接地( $AV_{SS,PA}$ )，共兩組接地。
- Audio 功率放大器接地端引腳  $AV_{SS,PA}$  需直接從電源負端接地點拉線，且走線線寬不小於 12mil。
- $V_{SS}$ 、 $AV_{SS,PA}$  兩組獨立接地須以鋪銅方式完成。
- 零件擺放時須預留電源及接地的走線寬度。
- 功率放大器輸出 SP+/SP-走線要粗且儘可能不要有灌孔。
- 由於直角容易累積電荷，有尖端放電的效應，容易影響 PCB 的穩定度，因此盡量以 45 度的斜角或弧角的方式進行走線。

## 軟體說明

### 語音播放控制器的設定

HT66FV1x0 經由 MCU 的暫存器，控制語音播放，說明如下：

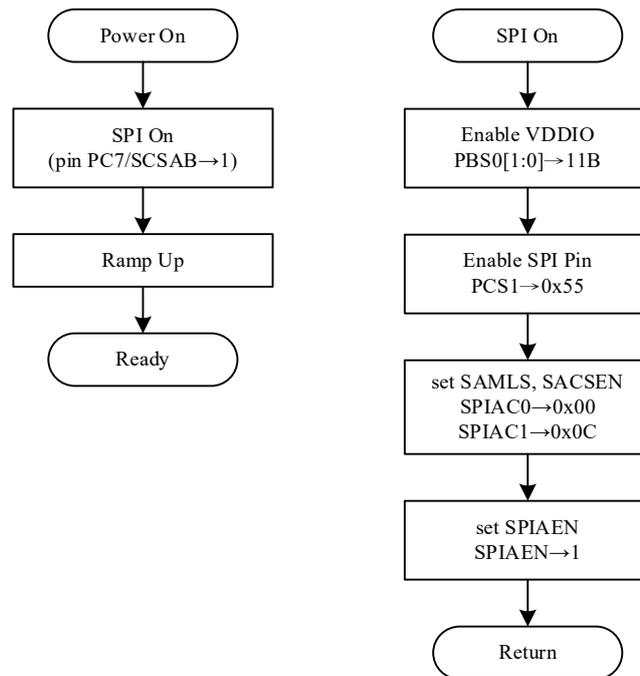
1. PAEN(PLAC.1)設定為"1"，使能功率放大器。DAEN(PLAC.0)設定為"1"，使能 16 位元 D/A 轉換器。
2. MUTEB(USVC.7) 設定為"1"，使能喇叭功能。
3. 音量控制有兩種方式
  - 類比式音量調節：在 DAC 輸出管腳 AUD 與功放類比信號輸入管腳 AUD\_IN 之間外掛一可變電阻，將 DAC 輸出信號用電阻分壓的方式進行衰減再進入功放，方便用於採用旋鈕的方式來調整音量。
  - 數位元式音量調節：通過 USVC6~USVC0 軟體控制喇叭音量，方便用戶用按鍵等方式，調整範圍從+6dB ~ -32dB，每一階為 0.5dB(大音量)或 1dB(小音量)。
4. 由外掛 SPI Flash ROM 儲存語音資料，MCU 經由內建的 SPI 介面讀取儲存在 SPI Flash 中的語音資料。
5. 16 位元的放音數據儲存在 PLADL/PLADH 暫存器中，軟體利用設定 Timer Mode 固定時間產生中斷，中斷時間到，軟體從 SPI Flash 讀取的語音數據寫到 16-bit DAC，再經內建 Power Amplifier 放大輸出推動喇叭發出聲音。

## Power On/Off Pop Noise

Pop Noise 是音頻器件在上電、斷電瞬間，各種操作帶來的瞬態衝擊所產生的爆破聲；為壓抑 pop noise，需留意以下流程：

### 開始播放語音的流程

#### (1) Power On Sequence

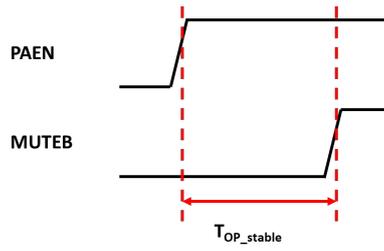


開始讀取 Flash 時，將 pin(PC7/SCSAB)由 1 設為 0 會造成 V33 瞬間變化，進而影響 VCC，使輸出不穩定。因此須先將 pin(PC7/SCSAB)由 1 設為 0，再啟動 Ramp Up 流程，以確保播放語音時的電源穩定，以減少爆音。

#### (2) Ramp Up

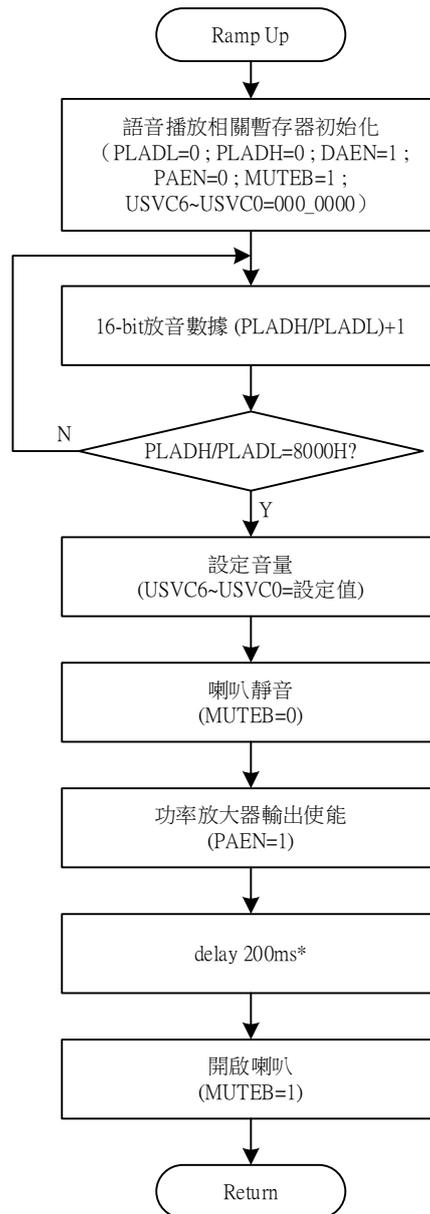
開機時，先將輸出腳位(SP+/SP-)漸進充電至  $1/2 V_{DD}$  的準位，並給予一定時間供功率放大器穩定輸出後，再對喇叭輸出聲音以壓抑 pop noise，流程敘述如下。

- 初始狀態：(PLADH/PLADL)=0000H, DAEN=1, PAEN=0, MUTE=1, USVC6~USVC0=000\_0000
- (PLADH/PLADL)從 0000H 逐步增加到 8000H
- 設定音量：USVC6~USVC0=設定值
- MUTE=0
- 開啟功率放大器(PAEN=1)，等待一定時間供功率放大器穩定輸出
- 開啟喇叭(MUTE=1)，準備播放語音



Note：一般而言，此時間大於 200ms 以上，能有效抑制 pop noise；進一步延長此時間可進一步壓抑 pop noise。此時間依不同喇叭而異。

● Ramp Up 軟體流程圖



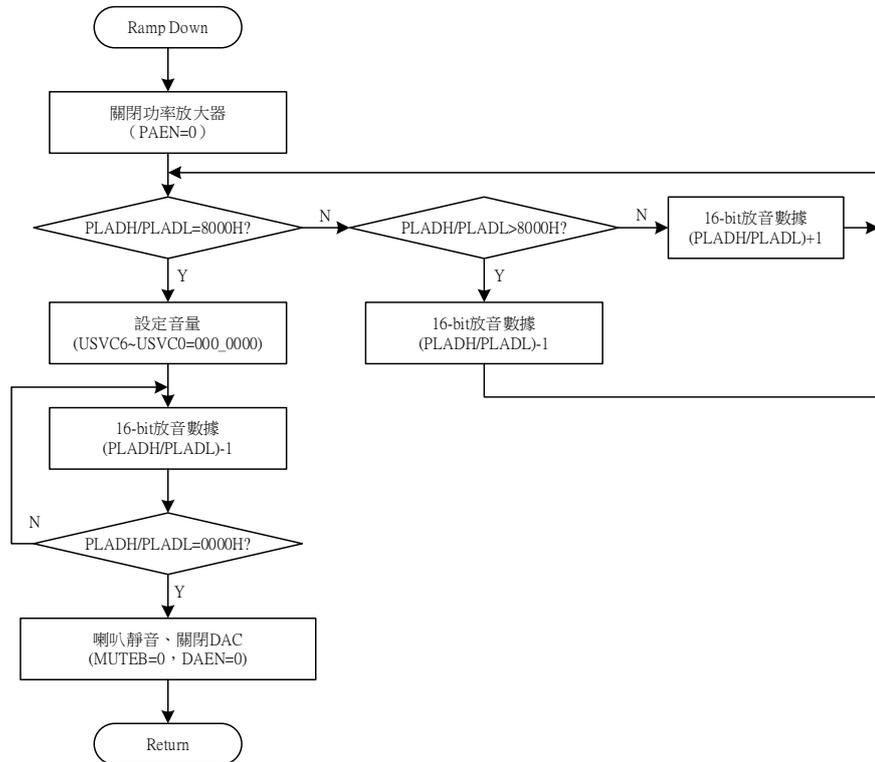
\*：時間不固定，需根據實際情況調整。

結束語音播放的流程

(1) Ramp Down

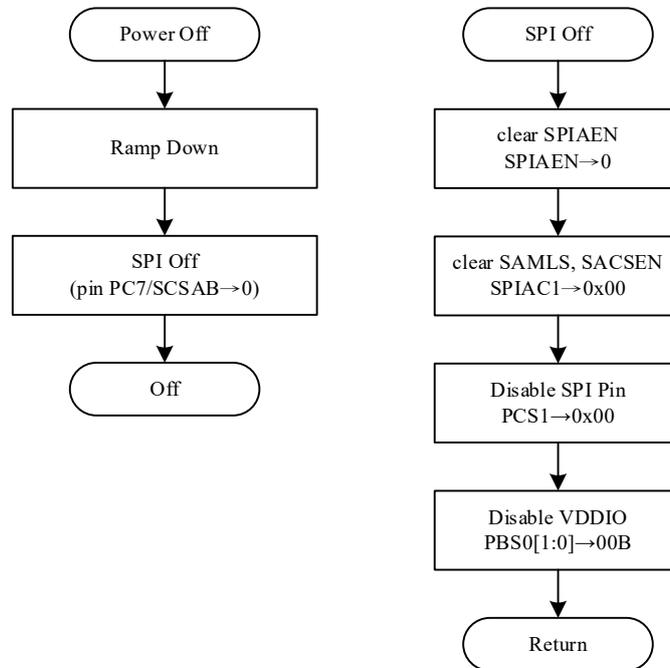
關機時

- 關閉功率放大器(PAEN=0)
- 將(PLADH/PLADL) 逐步調整到 8000H
- 設定音量：USVC6~USVC0=000\_0000
- (PLADH/PLADL)從 8000H 逐步減少到 0000H
- 關閉 DAC 與揚聲器(DAEN=0，MUTE=0)
- Ramp Down 軟體流程圖如下所示



## (2) Power Off Sequence

與 Power On 類似，當 Pin(PC7/SCSAB)由 0 設為 1 時，會造成 V33 瞬間變化，進而影響 VCC，使輸出不穩定。因此，當 SPI 為啟用狀態(Pin(PC7/SCSAB)為 0)時，須於 Ramp Down 流程結束後，再將存取 Flash 用之 Pin(PC7/SCSAB)由 0 設為 1，關閉 SPI 功能區塊，以減少爆音。



範例程式為設定 Ramp Up 與 Ramp Down 的具體流程，避免 DAC 與 Audio Power Amplifier On/Off 時造成的 Pop Noise。範例程式設定如下( $f_{sys}=8\text{MHz}$ )。

### (1) Power On Sequence 使用 ASM 語言設定

```

MAIN_START:
    CALL SPI_ON
    CALL RAMP_UP
    
```

### (2) SPI On 使用 ASM 語言設定

```

SPI_ON:
    ; enable VDDIO
    MOV A,PBS0
    OR A,03H
    MOV PBS0,A
    ; enable SPI pin
    MOV A,01010101B
    MOV PCS1,A
    ; set SAMLS, SACSEN
    CLR SPIAC0
    MOV A,0CH
    MOV SPIAC1,A
    ; set SPIAEN
    SET SPIAEN
    RET
    
```

## (3) Ramp Up 使用 ASM 語言設定

```

RAMPUP:
    MOV    A,00H
    MOV    PLADL,A                ; 初始化 DAC 16-bit 數據
    MOV    PLADH,A
    SET    DAEN                  ; 16-bit D/A 轉換器使能
    CLR    PAEN                  ; 功率放大器除能
    MOV    A,80H                 ; MUTEB=1 揚聲器使能；音量設為 000_0000
    MOV    USVC, A
LOOP1:
    CALL  DELAY_10US
    MOV    A,1                   ; PLADH/PLADL 從 0000H 逐步增加到 8000H
    ADDM  A,PLADL
    MOV    A,0
    ADCM  A,PLADH
    SNZ   PLADH.7
    JMP   LOOP1
    MOV    A,00H                 ; 音量設為目標值；MUTEB=0 揚聲器靜音
    MOV    USVC, A
    SET    PAEN                  ; 功率放大器使能
    CALL  DELAY_200MS           ; 延時等待功率放大器穩定
    SET    MUTEB                 ; MUTEB=1 開啟揚聲器
    RET
    
```

## (4) Power On Sequence 使用 V3 C 語言設定

```

SPI_ON();
RAMP_UP();
    
```

## (5) SPI On 使用 V3 C 語言設定

```

void SPI_ON(void)
{
    _pbs0 = _pbs0 | 0x03;        // enable VDDIO
    _pcs1 = 0x55;               // enable SPI pin
                                // set SAMLS, SACSSEN

    _spiac0 = 0x00;
    _spiac1 = 0x0C;
    _spiaen = 1;                // set SPIAEN
}
    
```

## (6) Ramp Up 使用 V3 C 語言設定

```

void RAMP_UP(void)
{
    unsigned int j;
    _pladl = 0x00;               // 初始化 DAC 16-bit 數據
    _pladh = 0x00;
    _daen = 1;                  // 16-bit D/A 轉換器使能
    _paen = 0;                  // 功率放大器除能
    _usvc = 0x80;               // MUTEB=1 揚聲器使能；音量設為 000_0000
    PLAD = 0x0000;
    while(PLAD != 0x8000)       // PLADH/PLADL 從 0000H 逐步增加到 8000H
    {
        GCC_CLRWDT();
        GCC_DELAY(10);
        PLAD+=1;
        _pladl = PLAD;
    }
}
    
```

```

        _pladh = PLAD >> 8;
    }
    _usvc = 0x00;           // 音量設為目標值；MUTE=0 揚聲器靜音
    _plac = 0x03;           // 功率放大器使能
    for(i= 0;j<300;j++)     // 延時等待功率放大器穩定
    {
        GCC_CLRWDT0;
        GCC_DELAY(4000);
    }
    _usvc = 0x80;           // MUTE=1 開啟揚聲器
}
    
```

## (7) Ramp Down 使用 ASM 語言設定

```

RAMPDOWN:
    CLR PAEN                ; 功率放大器除能
COMPARE_EQUAL:              ; PLADH/PLADL 逐步調整到 8000H
    MOV A,PLADH
    XOR A,80h
    SNZ Z
    JMP COMPARE_LARGER
    MOV A,PLADL
    XOR A,00h
    SNZ Z
    JMP COMPARE_LARGER
    JMP VOICE_OFF
COMPARE_LARGER:
    CALL DELAY_10US
    SZ PLADH.7
    JMP DECREASE            ; PLAD>8000H
    JMP INCREASE            ; PLAD<8000 H
DECREASE:
    MOV A,OFFH
    ADDM A,PLADL
    MOV A,OFFH
    ADCM A,PLADH
    JMP COMPARE_EQUAL
    RET
INCREASE:
    MOV A,1
    ADDM A,PLADL
    MOV A,0
    ADCM A,PLADH
    JMP COMPARE_EQUAL
VOICE_OFF:
    MOV A,80H                ; 音量設為 000_0000
    MOV USVC, A
LOOP2:
    CALL DELAY_10US
    MOV A,OFFH                ; PLADH/PLADL 從 8000H 逐步減少到 0000H
    ADDM A,PLADL
    MOV A,OFFH
    ADCM A,PLADH
    SZ PLADL
    JMP LOOP2
    SZ PLADH
    JMP LOOP2
    
```

```

CLR  MUTEB           ; 揚聲器輸出除能
CLR  DAEN           ; 16-bit D/A 轉換器除能
RET
    
```

(8) Power Off Sequence 使用 ASM 語言設定

```

MAIN_START:
CALL  RAMP_DOWN
CALL  SPI_OFF
    
```

(9) SPI Off 使用 ASM 語言設定

```

SPI_OFF:
CLR  SPIAEN         ; clear SPIAEN
CLR  SPIAC1         ; clear SAMLS, SACSEN
CLR  PCS1           ; disable SPI pin
MOV  A,PBS0         ; disable VDDIO
AND  A,0FCH
MOV  PBS0,A
RET
    
```

(10) Ramp Down 使用 V3 C 語言設定

```

void RAMP_DOWN(void)
{
    _paen=0;           // 功率放大器除能
    PLAD=((unsigned int)_pladh<<8) | _pladl;
    while(PLAD != 0x8000) // PLADH/PLADL 逐步調整到 8000H
    {
        GCC_CLRWDT0;
        GCC_DELAY(10);
        if(PLAD > 0x8000)
        {
            PLAD--;
            _pladl = PLAD;
            _pladh = PLAD >>8;
        }
        else
        {
            PLAD++;
            _pladl = PLAD;
            _pladh = PLAD >>8;
        }
    }
    _usvc = 0x80;       // 音量設為 000_0000
    while(PLAD)        // PLADH/PLAD 從 8000H 逐步減少到 0000H
    {
        GCC_CLRWDT0;
        GCC_DELAY(10);
        PLAD-=1;
        _pladl = PLAD;
        _pladh = PLAD >>8;
    }
    _usvc = 0x00;       // 揚聲器輸出靜音
    _daen = 0;         // 16-bit D/A 轉換器除能
}
    
```

(11) Power Off Sequence 使用 V3 C 語言設定

```
RAMP_DOWN();
SPI_OFF();
```

(12) SPI Off 使用 V3 C 語言設定

```
void SPI_OFF(void)
{
    _spiaen = 0x00;           // clear SPIAEN
    _spiac1 = 0x00;         // clear SAMLS, SACSEN
    _pcs1 = 0x00;           // disable SPI pin
    _pbs0 = _pbs0 & 0xFC;   // disable VDDIO
}
```

## 結論

本文通過對 HT66FV1x0 系列內建音頻功率放大器功能的介紹，使用者能清楚瞭解 Holtek 8-bit MCU 內建 Power Amplifier 功能實現。

## 版本及修改資訊

日期	作者	發行	修訂說明
2021.3.12	李玉梅	V1.10	解決爆音流程處理不足問題
2018.4.30	藍愛娣	V1.00	第一版發行

## 參考資料

參考文件 HT66FV130/140/150/160 Datasheet。

如需進一步瞭解，敬請瀏覽 Holtek 官方網站 [www.holtek.com](http://www.holtek.com)。

## 免責聲明

本網頁所載的所有資料、商標、圖片、連結及其他資料等（以下簡稱「資料」），只供參考之用，盛群半導體股份有限公司及其關聯企業（以下簡稱「本公司」）將會隨時更改資料，並由本公司決定而不作另行通知。雖然本公司已盡力確保本網頁的資料準確性，但本公司並不保證該等資料均為準確無誤。本公司不會對任何錯誤或遺漏承擔責任。

本公司不會對任何人士使用本網頁而引致任何損害（包括但不限於電腦病毒、系統固障、資料損失）承擔任何賠償。本網頁可能會連結至其他機構所提供的網頁，但這些網頁並不是由本公司所控制。本公司不對這些網頁所顯示的內容作出任何保證或承擔任何責任。

### 責任限制

在任何情況下，本公司並不須就任何人由於直接或間接進入或使用本網站，並就此內容上或任何產品、資訊或服務，而招致的任何損失或損害負任何責任。

### 管轄法律

本免責聲明受中華民國法律約束，並接受中華民國法院的管轄。

### 免責聲明更新

本公司保留隨時更新本免責聲明的權利，任何更改於本網站發佈時，立即生效。