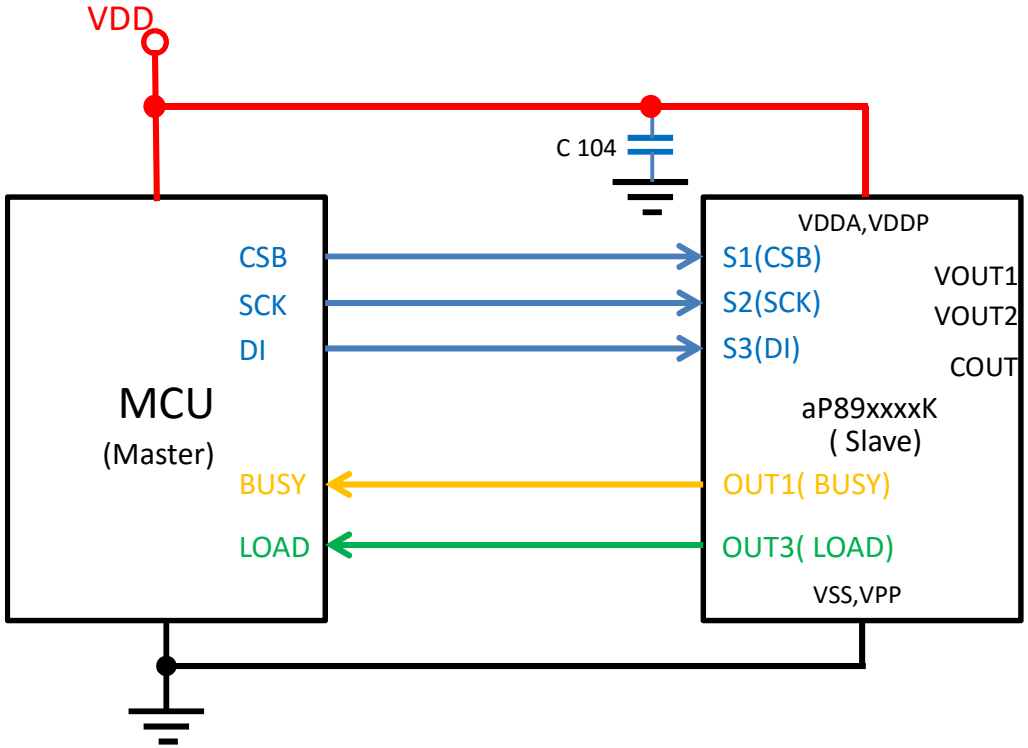
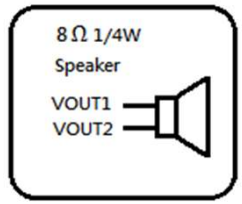
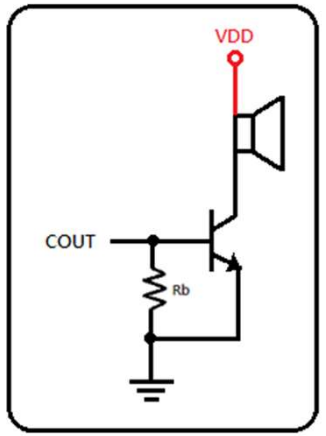
SPI Mode







CSB : Chip Select (active low, output from master).

SCK : Serial Clock (output from master).

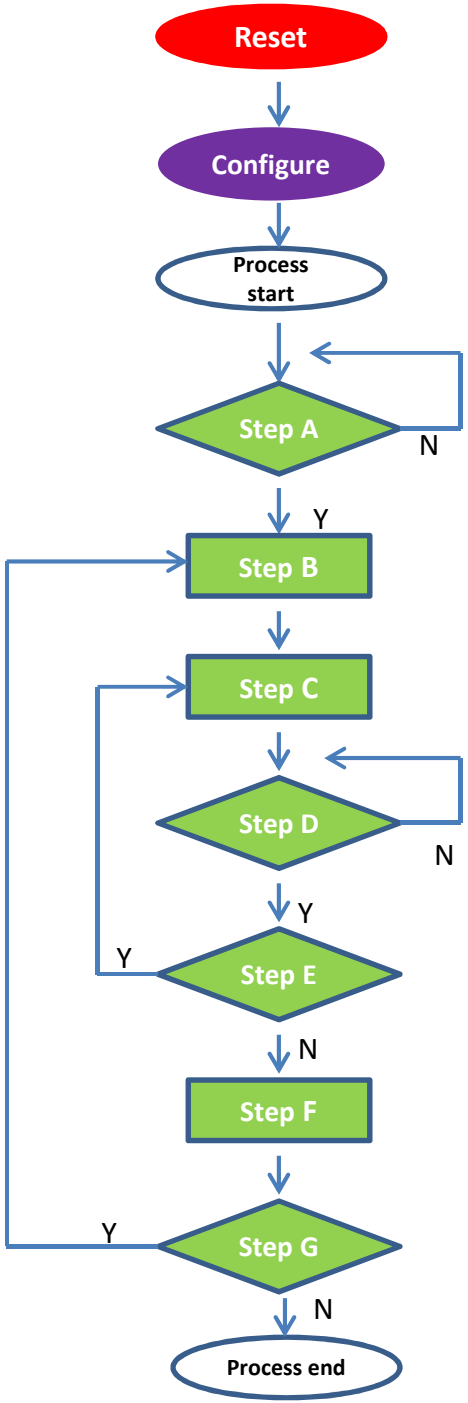
DI : Master Output, Slave Input (output from master).

BUSY : Master Input, Slave Output (output from slave).

OUT1 as output from the Slave chip to the Master CPU for feedback response.

LOAD : Master Input, Slave Output (output from slave).

OUT3 as output from the Slave chip to the Master CPU for feedback response.



Power On / External Reset

The Ext power on reset time depend on Crst .
The Internal power on reset time . : Typ = 5us

Device configure time : 2ms

Step A : Whether you want to play voices.

Step B : CMD PU1 / PU2.

Step C : CMD Load.

Step D : To detect whether the end of the LOAD signal.

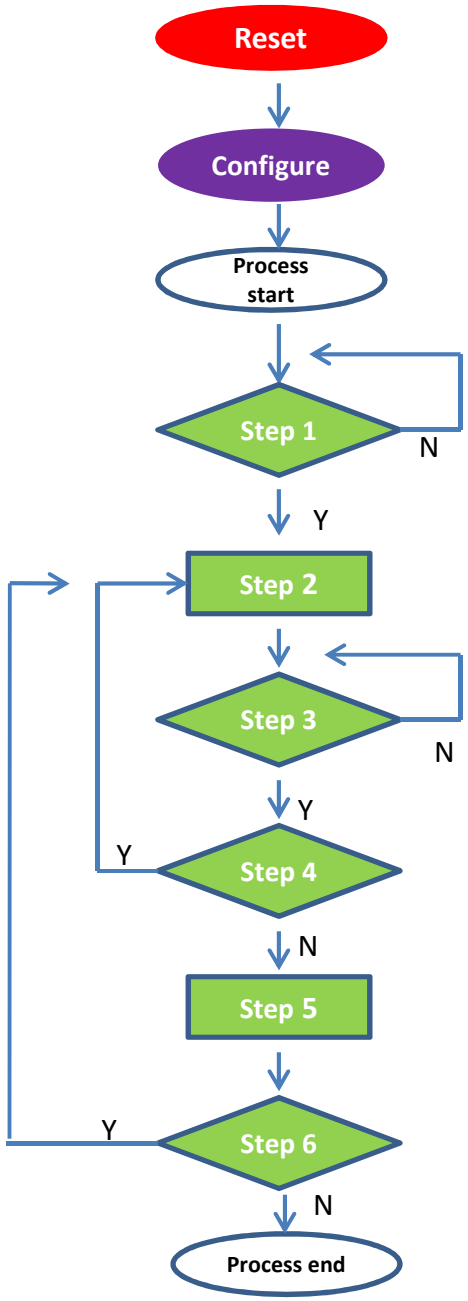
Step E : Do you want to continue to play voices.

Step F : CMD PD1 / PD2.

Step G : After power down would like to continue to play voices.

Active CMD : PU1/PU2,Load,Play

De-Active CMD : PD1/PD2



Power On / External Reset

The Ext power on reset time depend on Crst .
The Internal power on reset time . : Typ = 5us

Device configure time : 2ms

Step 1 : Whether you want to play voices.

Step 2 : CMD Play.

Step 3 : To detect whether the end of the BUSY signal.

Step 4 : Do you want to continue to play voices.

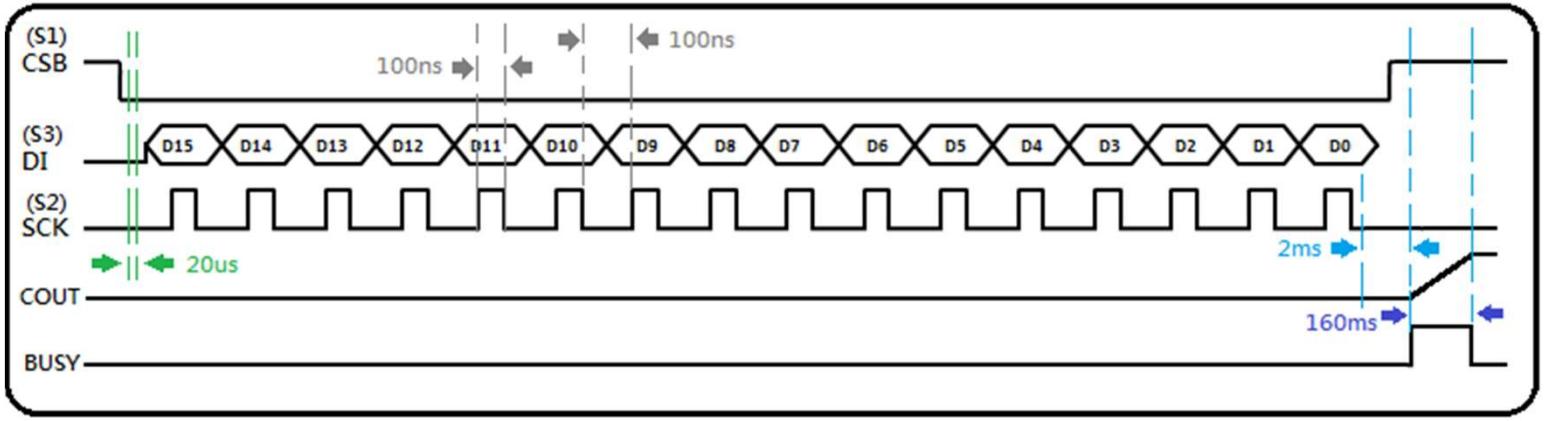
Step 5 : CMD PD1 / PD2.

Step 6 : After power down would like to continue to play voices.

Active CMD : PU1/PU2,Load,Play

De-Active CMD : PD1/PD2

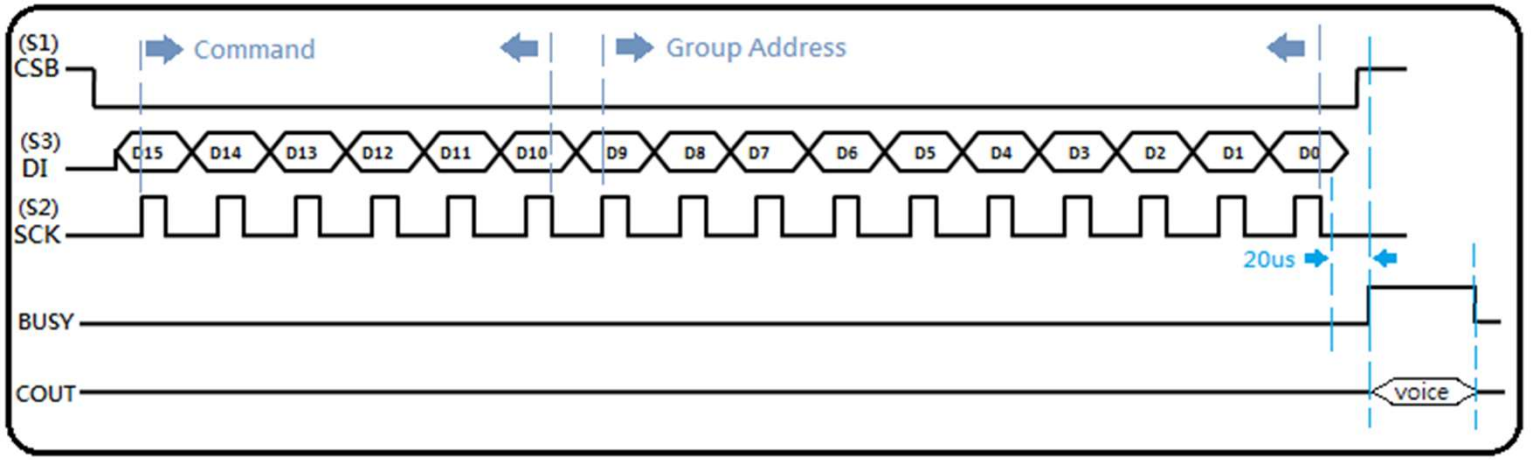
Step B : CMD PU1 (A400h) / PU2 (A800h)




```
void SPIdata(unsigned int cmd)
{
      unsigned int mask = 0;

      for(mask = 0x8000;mask > 0;mask >>=1)
      {
          DI = (cmd & mask) ? 1:0;
          SCK = 0;
          SCK = 1;
      }
      SCK = 0;
}
```

```
void SoundChip_Active(unsigned int cmd)
{
      CSB = 1;
      CSB = 0;            //start condition
      Delay20us();        //wait sleep to wake up state  (20us)
      SPIdata(cmd);       //power up chip with ramp-up or without ramp-up
      CSB = 1;             //stop condition
      Delay2ms();         //wait state exist wake up state  ( 2ms)
      while(BUSY);        //wait end of ramp  (160ms)
}
```

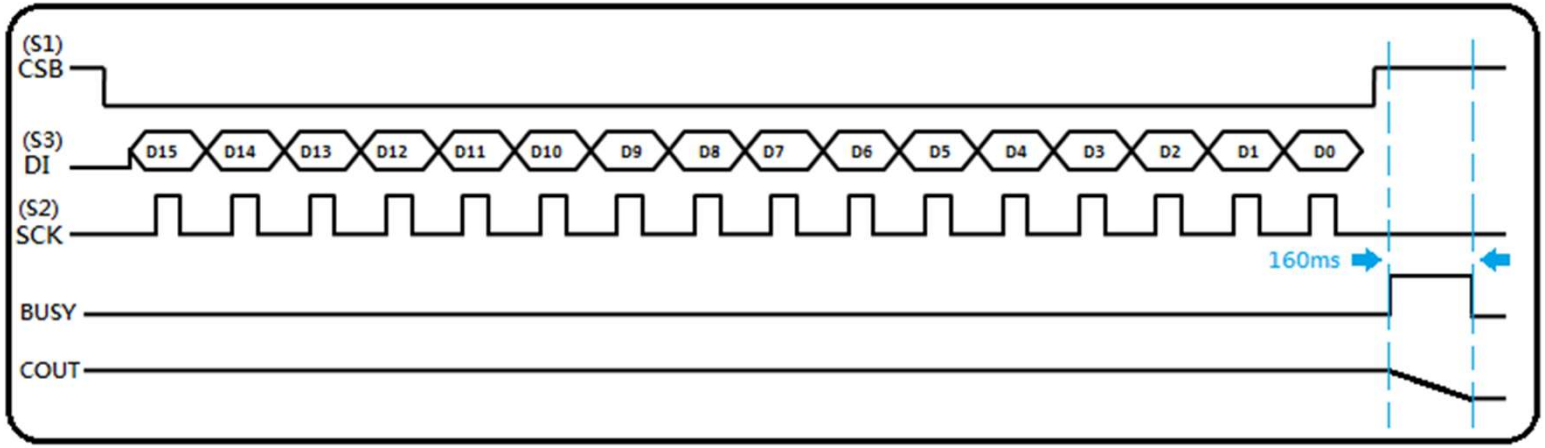
Step C : CMD Load (9400h) + Group Address




```
void SPIdata(unsigned int cmd)
{
      unsigned int mask = 0;

      for(mask = 0x8000;mask > 0;mask >>=1)
      {
          DI = (cmd & mask) ? 1:0;
          SCK = 0;
          SCK = 1;
      }
      SCK = 0;
}
```

```
void SoundChip_Load(unsigned int addr)
{
      unsigned int cmd = 0;
      cmd = 0x9400 + addr;

      CS = 1;
      CS = 0;           //start condition
      SPIdata(cmd);
      CS = 1;           //stop condition
      Delay20us();      //for max output delay of BUSY/FULL signal
}
```

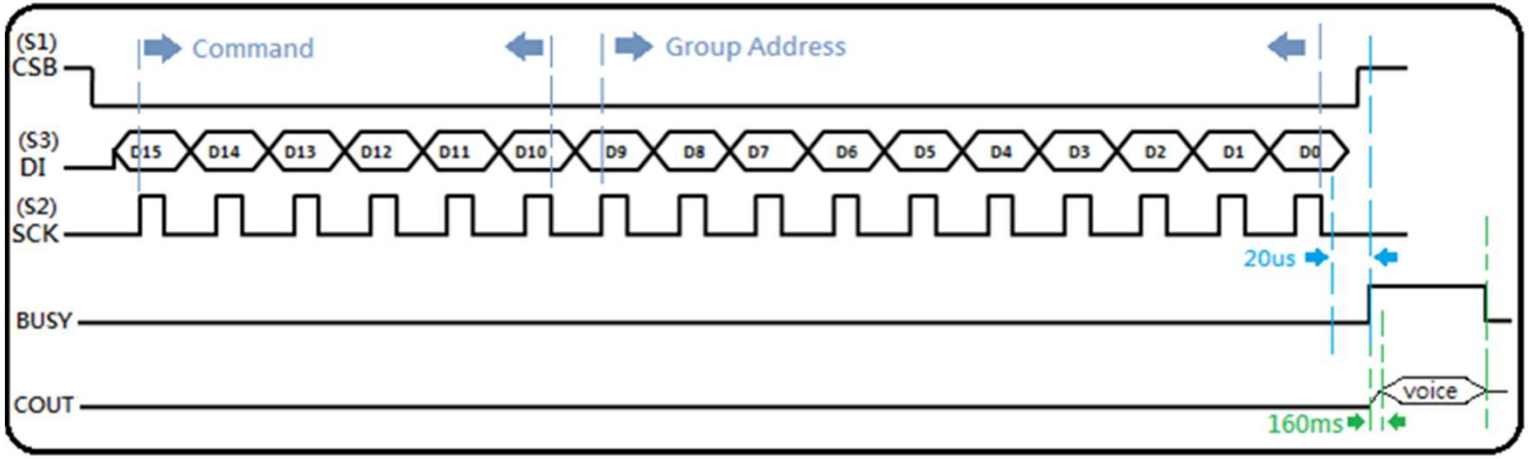
Step F : CMD PD1 (B400h) / PD2 (B800h)




```
void SPIdata(unsigned int cmd)
{
      unsigned int mask = 0;

      for(mask = 0x8000;mask > 0;mask >>=1)
      {
          DI = (cmd & mask) ? 1:0;
          SCK = 0;
          SCK = 1;
      }
      SCK = 0;
}
```

```
void SoundChip_DeActive(unsigned int cmd)
{
      while(LOAD);       //wait the last voice group be loaded
      while(BUSY);       //wait the end of last voice group

      CSB = 1;
      CSB = 0;           //start condition
      SPIdata(cmd)       //power down chip with ramp-down or without ramp-down
      CSB = 1;           //stop condition
      while(BUSY);       //wait end of ramp  (160ms)
}
```

Step 2 : CMD Play (9800h) + Group Address



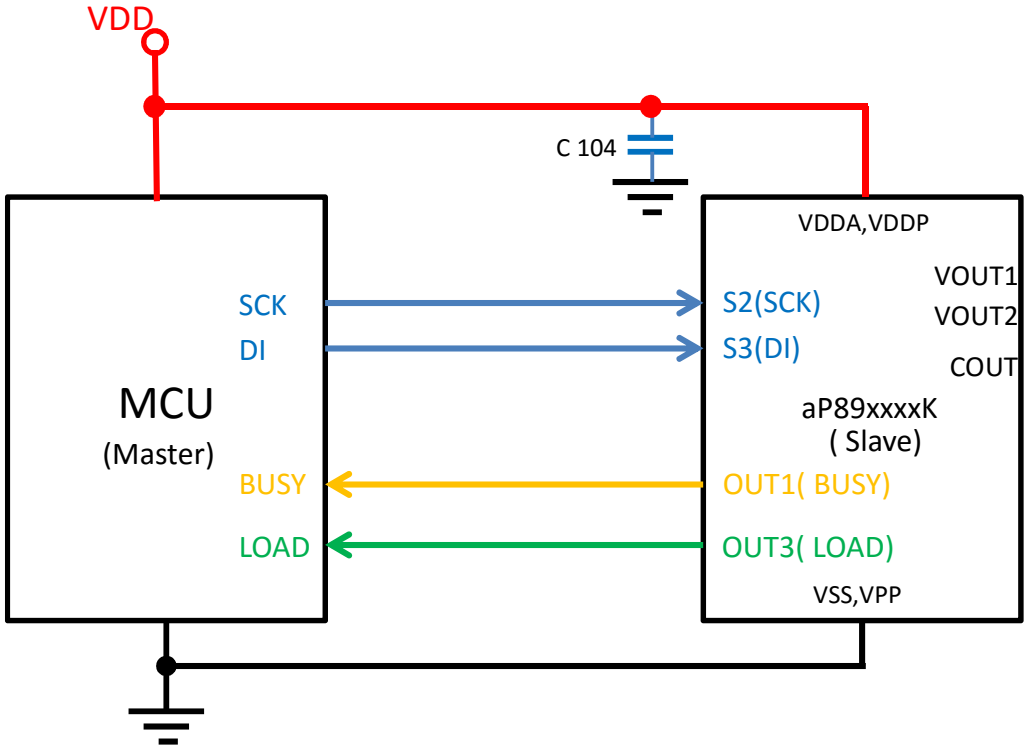

```
void SPIdata(unsigned int cmd)
{
      unsigned int mask = 0;

      for(mask = 0x8000;mask > 0;mask >>=1)
      {
          DI = (cmd & mask) ? 1:0;
          SCK = 0;
          SCK = 1;
      }
      SCK = 0;
}
```

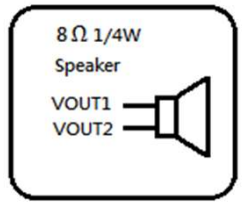
```
void SoundChip_Play(unsigned int addr)
{
      unsigned int cmd = 0;
      cmd = 0x9800 + addr;

      CSB = 1;
      CSB = 0;                  //start condition
      SPIdata(cmd);
      CSB = 1;                  //stop condition
      Delay20us();              //for max output delay of BUSY/FULL signal
}
```

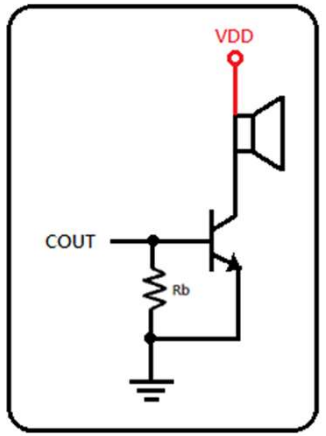
```
void main()
{
        //The voices are PWM output.
        //
       unsigned char CheckBtn = 0;
       InitPortD();
       InitPortB();
       Delay10ms();                    //waiting Reset time & Device configure time.
       while(1)
       {
                                       //Step A : Whether you want to play voices.
              PushBtn1(&CheckBtn); //Detecting whether a button is pressed.
              //---------------------------------
              if(CheckBtn == 1)        //When button is pushed.
              {
                  //----------------------------
                  //Step B : CMD PU1
                  SoundChip_Active(0xa400);
                  //----------------------------
                  //Step C : CMD Load
                  SoundChip_Load(0);
                  while(FULL);             //Step D : To detect whether the end of the LOAD signal.
                  SoundChip_Load(1);
                  while(FULL);
                  SoundChip_Load(2);
                  while(FULL);
                  //----------------------------
                  //When you do not want to play voices
                  //Step F : CMD PD1
                  SoundChip_DeActive(0xb400);

                  CheckBtn = 0;
               }
         }//while(1)
}
```

```
void main()
{
        //The voices are PWM output.
        //
       unsigned char CheckBtn = 0;
       InitPortD();
       InitPortB();
       Delay10ms();                    //waiting Reset time & Device configure time.
       while(1)
       {
                                       //Step 1 : Whether you want to play voices.
              PushBtn1(&CheckBtn); //Detecting whether a button is pressed.
              //---------------------------------
              if(CheckBtn == 1)        //When button is pushed.
              {
                  //----------------------------
                  //Step 2 : CMD Play
                  SoundChip_Play(0);
                  while(BUSY);          //Step 3 : To detect whether the end of the BUSY signal.
                  SoundChip_Play(1);
                  while(BUSY);
                  SoundChip_Play(2);
                  while(BUSY);
                  //----------------------------
                  //When you do not want to play voices
                  //Step 5 : CMD PD1
                  SoundChip_DeActive(0xb400);

                  CheckBtn = 0;
               }
         }//while(1)
}
```

I2C Mode







SCK : Serial Clock (output from master).

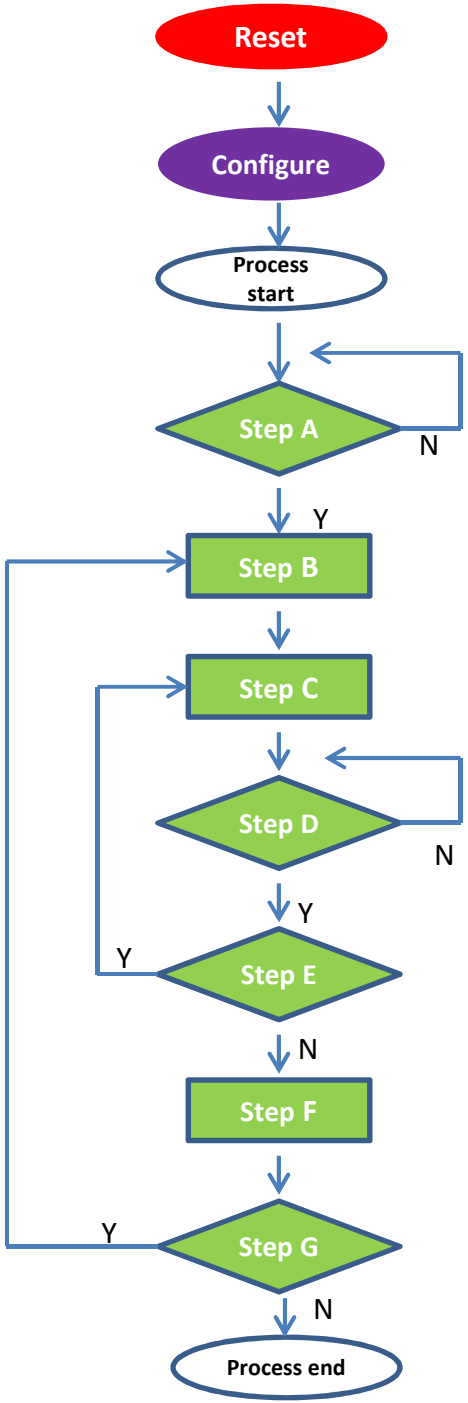
DI : Master Output, Slave Input (output from master).

BUSY : Master Input, Slave Output (output from slave).

OUT1 as output from the Slave chip to the Master CPU for feedback response.

LOAD : Master Input, Slave Output (output from slave).

OUT3 as output from the Slave chip to the Master CPU for feedback response.



Power On / External Reset

The Ext power on reset time depend on Crst .
The Internal power on reset time . : Typ = 5us

Device configure time : 2ms

Step A : Whether you want to play voices.

Step B : CMD PU1 / PU2.

Step C : CMD Load.

Step D : To detect whether the end of the LOAD signal.

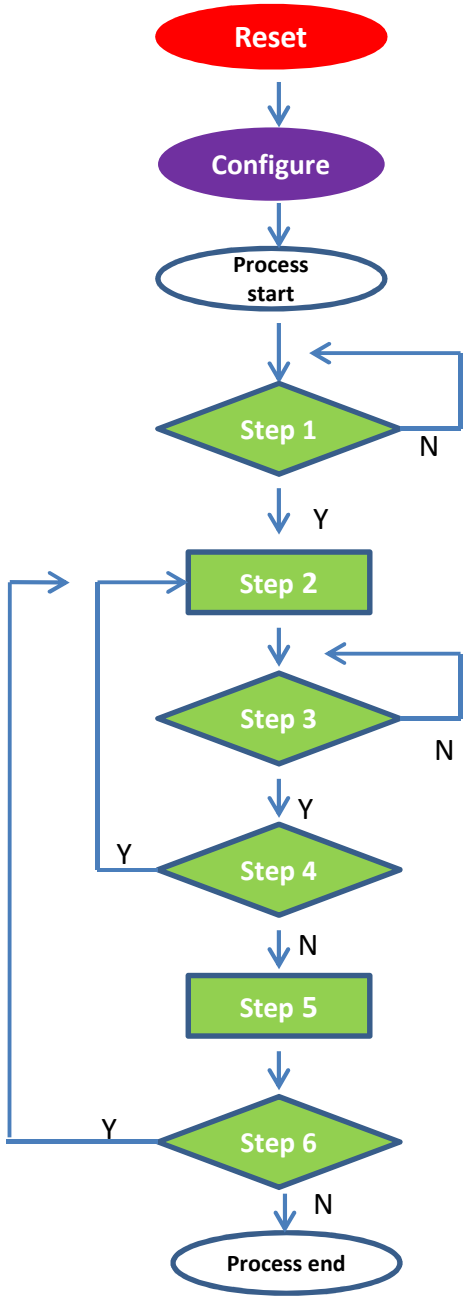
Step E : Do you want to continue to play voices.

Step F : CMD PD1 / PD2.

Step G : After power down would like to continue to play voices.

Active CMD : PU1/PU2,Load,Play

De-Active CMD : PD1/PD2



Power On / External Reset

The Ext power on reset time depend on Crst .
The Internal power on reset time . : Typ = 5us

Device configure time : 2ms

Step 1 : Whether you want to play voices.

Step 2 : CMD Play.

Step 3 : To detect whether the end of the BUSY signal.

Step 4 : Do you want to continue to play voices.

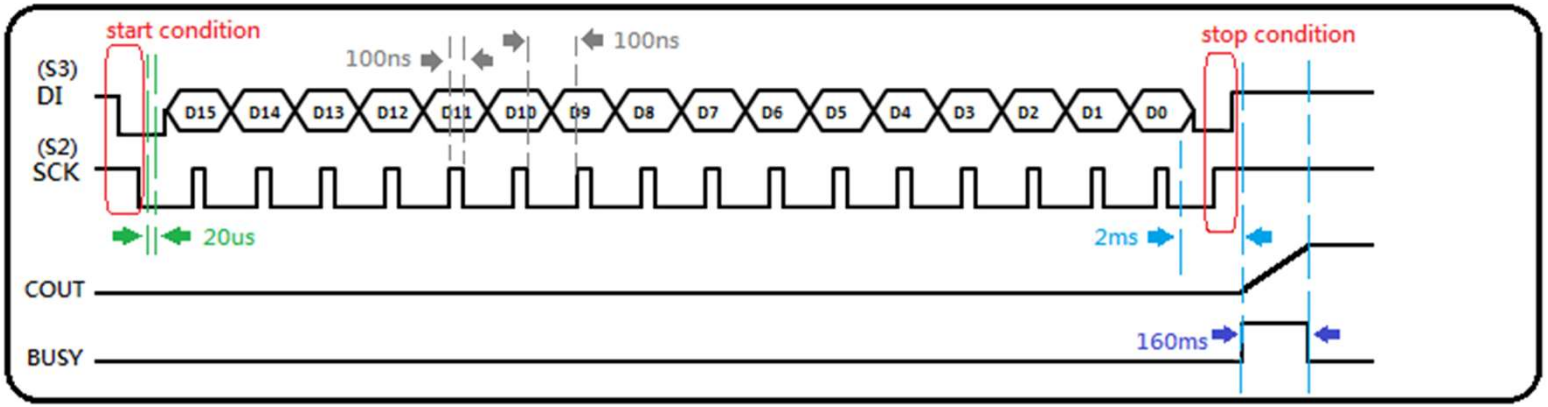
Step 5 : CMD PD1 / PD2.

Step 6 : After power down would like to continue to play voices.

Active CMD : PU1/PU2,Load,Play

De-Active CMD : PD1/PD2

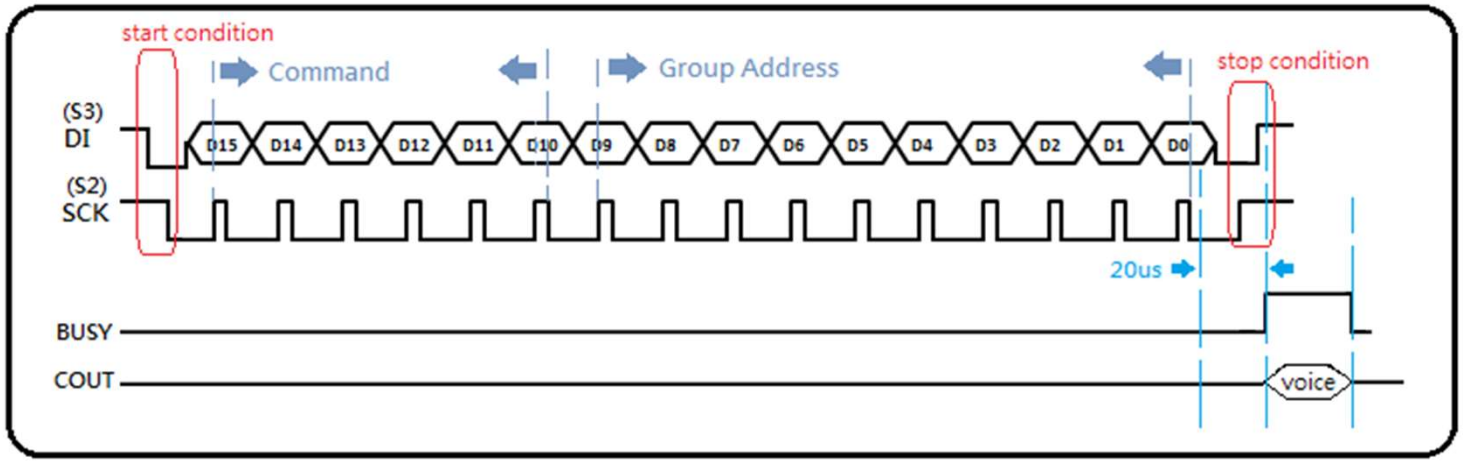
Step B : CMD PU1 (A400h) / PU2 (A800h)




```
void I2Cdata(unsigned int cmd)
{
      unsigned int mask = 0;
      for(mask = 0x8000;mask > 0;mask >>=1)
      {
          SCK = 0;
          DI = (cmd & mask) ? 1:0;
          SCK = 1;
      }
      SCK = 0;
}
```

```
void SoundChip_Active(unsigned int cmd)
{
      SCK = 1;       //start condition
      DI = 1;        //
      DI = 0;        //
      //--------------//
      Delay20us();       //wait sleep to wake up state  (20us)
      I2Cdata(cmd);      //power up chip with ramp-up or without ramp-up
      //--------------//stop condition
      DI = 0;        //
      SCK = 1;       //
      DI = 1;        //
      Delay2ms();        //wait state exist wake up state (2ms)
      while(BUSY);       //wait end of ramp  (160ms)
}
```

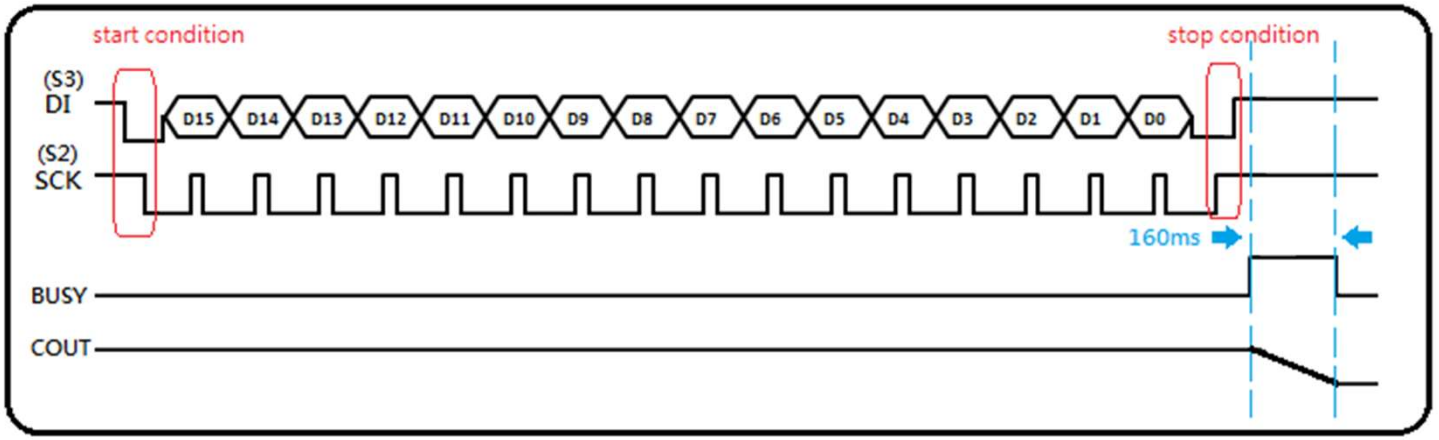
Step C : CMD Load (9400h) + Group Address




```
void I2Cdata(unsigned int cmd)
{
      unsigned int mask = 0;
      for(mask = 0x8000;mask > 0;mask >>=1)
      {
          SCK = 0;
          DI = (cmd & mask) ? 1:0;
          SCK = 1;
      }
      SCK = 0;
}
```

```
void SoundChip_Load(unsigned int cmd)
{
      unsigned int cmd = 0;
      cmd = 0x9400 + addr;

      SCK = 1;       //start condition
      DI = 1;        //
      DI = 0;        //
      //--------------//
      I2Cdata(cmd);     //power up chip with ramp-up or without ramp-up
      //--------------//stop condition
      DI = 0;        //
      SCK = 1;       //
      DI = 1;        //
      Delay20us();       //for max output delay of BUSY/FULL signal
}
```

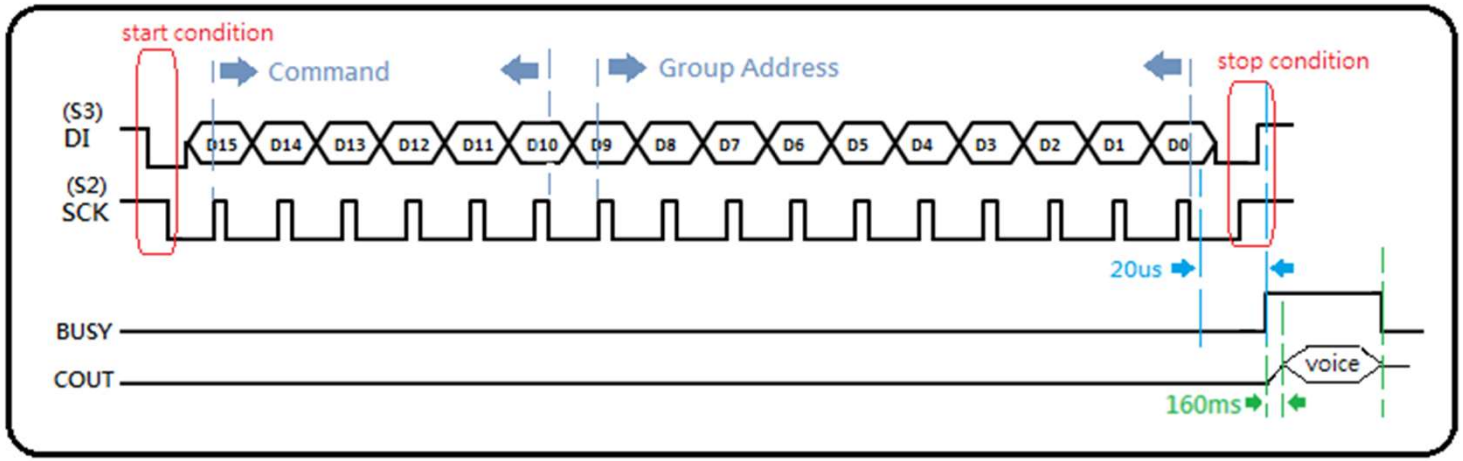
Step F : CMD PD1 (B400h) / PD2 (B800h)




```
void I2Cdata(unsigned int cmd)
{
      unsigned int mask = 0;
      for(mask = 0x8000;mask > 0;mask >>=1)
      {
          SCK = 0;
          DI = (cmd & mask) ? 1:0;
          SCK = 1;
      }
      SCK = 0;
}
```

```
void SoundChip_DeActive(unsigned int cmd)
{
      while(FULL);          //wait the last voice group be loaded
      while(BUSY);          //wait the end of last voice group

       SCK = 1;       //start condition
       DI = 1;        //
       DI = 0;        //
       //--------------//
       I2Cdata(cmd);        //power up chip with ramp-up or without ramp-up
       //--------------//stop condition
       DI = 0;        //
       SCK = 1;       //
       DI = 1;        //
       while(BUSY);         //wait end of ramp  (160ms)
}
```

Step 2 : CMD Play (9800h) + Group Address




```
void I2Cdata(unsigned int cmd)
{
      unsigned int mask = 0;
      for(mask = 0x8000;mask > 0;mask >>=1)
      {
          SCK = 0;
          DI = (cmd & mask) ? 1:0;
          SCK = 1;
      }
      SCK = 0;
}
```

```
void SoundChip_Play(unsigned int cmd)
{
      unsigned int cmd = 0;
      cmd = 0x9800 + addr;

      SCK = 1;      //start condition
      DI = 1;       //
      DI = 0;       //
      //-------------//
      I2Cdata(cmd);     //power up chip with ramp-up or without ramp-up
      //-------------//stop condition
      DI = 0;       //
      SCK = 1;      //
      DI = 1;       //
      Delay20us();      //for max output delay of BUSY/FULL signal
}
```

```
void main()
{
        //The voices are DAC output.
        //
       unsigned char CheckBtn = 0;
       InitPortD();
       InitPortB();
       Delay10ms();                    //waiting Reset time & Device configure time.
       while(1)
       {
                                       //Step A : Whether you want to play voices.
              PushBtn1(&CheckBtn); //Detecting whether a button is pressed.
              //---------------------------------
              if(CheckBtn == 1)        //When button is pushed.
              {
                  //----------------------------
                  //Step B : CMD PU2
                  SoundChip_Active(0xa800);
                  //----------------------------
                  //Step C : CMD Load
                  SoundChip_Load(0);
                  while(FULL);             //Step D : To detect whether the end of the LOAD signal.
                  SoundChip_Load(1);
                  while(FULL);
                  SoundChip_Load(2);
                  while(FULL);
                  //----------------------------
                  //When you do not want to play voices
                  //Step F : CMD PD2
                  SoundChip_DeActive(0xb800);

                  CheckBtn = 0;
               }
         }//while(1)
}
```

```
void main()
{
        //The voices are DAC output.
        //
       unsigned char CheckBtn = 0;
       InitPortD();
       InitPortB();
       Delay10ms();                    //waiting Reset time & Device configure time.
       while(1)
       {
                                       //Step 1 : Whether you want to play voices.
              PushBtn1(&CheckBtn); //Detecting whether a button is pressed.
              //---------------------------------
              if(CheckBtn == 1)        //When button is pushed.
              {
                  //----------------------------
                  //Step 2 : CMD Play
                  SoundChip_Play(0);
                  while(BUSY);          //Step 3 : To detect whether the end of the BUSY signal.
                  SoundChip_Play(1);
                  while(BUSY);
                  SoundChip_Play(2);
                  while(BUSY);
                  //----------------------------
                  //When you do not want to play voices
                  //Step 5 : CMD PD2
                  SoundChip_DeActive(0xb800);

                  CheckBtn = 0;
               }
         }//while(1)
}
```