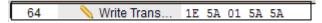
TEF6686 MCU 程序制作

- 1. TEF6686 上电初始化。
- (1) TEF6686 软件复位 1E 5A 01 5A 5A。



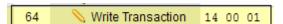
- (2) 延时 100ms。
- (3) TEF6686 Patch 初始化

(3) 12	1 0000 1 aten [/] x	- 10																							
64	Nrite Trans	1C	00	00																					
64	Nrite Trans	1C	00	74																					
64	Nrite Trans	1B	F0	00	38	2F	D0	80	F0	00	38	37	D0	80	43	B2	38	3A	D0	80	F0	00	70	00	C2
64	Nrite Trans	1B	F0	00	38	6B	D0	80	80	FC	39	0E	D0	80	F0	00	39	13	D0	80	F0	00	39	16	D0
64	Nrite Trans	1B	F0	00	39	ВС	D0	80	F0	00	39	32	D0	80	F0	00	39	ΒE	D0	80	C4	A2	02	11	60
64	Nrite Trans	1B	F0	00	39	9A	D0	80	90	01	39	34	D0	80	F0	00	38	F0	D0	80	F0	00	39	3B	D2
64	Nrite Trans	1B	F0	00	39	3F	D0	80	F0	00	39	49	D0	80	F0	00	39	4C	D0	80	9E	30	18	F9	D2
64	Nrite Trans	1B	F0	00	39	51	D0	80	F0	00	39	54	D0	80	F0	00	39	57	D0	80	32	00	39	59	D0
64	Nrite Trans	1B	F0	00	39	5B	D0	80	A8	01	39	AA	D0	80	F0	00	39	C3	D0	80	F0	00	39	C5	D0
64	Nrite Trans	1B	F0	00	39	C7	D0	80	F0	00	39	DA	D0	80	F0	00	39	DD	D0	80	F0	00	39	E0	D0
64	Nrite Trans	1B	F0	00	39	E3	D0	80	F0	00	70	00	F0	00	F0	00	39	E8	D0	80	F0	00	39	B5	D2
64	Nrite Trans	1B	F0	00	39	F6	D0	80	00	43	3A	05	D9	80	F0	00	3A	0F	D0	80	F0	00	3A	45	D0
64	Nrite Trans	1B	F0	00	3A	48	D0	80	F0	00	3A	59	D0	80	2E	40	3A	F8	D0	80	F0	00	3B	02	D0
64	Nrite Trans	1B	F0	00	0E	3F	60	00	50	10	28	D8	D2	80	91	01	01	36	60	00	F0	00	70	00	A0
64	Nrite Trans	1B	F0	00	70	00	A0	D8	F0	00	70	00	A0	E8	F0	00	70	00	A1	6F	F0	00	70	00	A1
64	Nrite Trans	1B	F0	00	70	00	A1	C4	F0	00	70	00	A1	DC	F0	00	20	31	D0	80	F0	00	04	C1	60
64	Nrite Trans	_																						22	
64	Nrite Trans	1B	F0	00	00	FF	60	03	F0	00	01	42	D2	80	90	03	40	02	F0	00	90	43	01	70	D1
	A MARIE Town										_														
64	Write Trans																							89	
64	Write Trans																							28	
64	Write Trans																							A8	
64	Write Trans																							0B	
64	Write Trans																							05	
64	Write Trans																							20	
64	Write Trans								10	00	1E	60	60	00	E.O.	00	10	00	r0	00	r0	00	10	00	10
64	Write Trans		F0		70	00	DO	08																	
64	Write Trans	1C	00	00																					

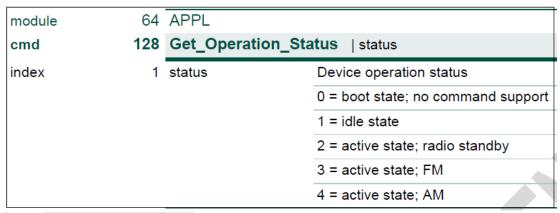
(3) LutByteValues[]数组写入 25*3+23(一次性写入 94 字节)字节 Byte.

64	Write Trans	1C	00	75																					
64	Write Trans	1B	40	13	40	2F	41	68	41	C1	42	14	44	DE	47	C5	47	FC	4E	56	4E	58	4E	5B	4D
64	Write Trans	1B	4E	0A	4E	49	4E	53	4F	92	4F	ΕA	50	74	50	80	56	60	56	D4	56	D9	59	В4	5A
64	Write Trans	1B	5B	79	5D	DC	60	67	60	88	60	A8	61	9F	61	В6	61	DF	62	14	62	59	62	9B	62
64	Write Trans	1B	63	69	64	ЗВ	66	09	66	E7	67	0B	67	1A	67	29	68	87	68	99	68	A7	68	B2	
64	Write Trans	1C	00	00																					

(4) 命令开始命令



- (5) 延时 50ms。
- (6) 状态检测模式检测 0X01。



64	Nrite Transaction	40 80 01
64	Read Transaction	00 00*
64	Write Transaction ■ Output Description Description ■ Output Description ■ Output Description ■ Output Description De	40 80 01
64	Read Transaction	00 01*

(7) IC 型号与版本检测

64	Nrite Trans	40	82	01					
64	Read Trans	09	0E	01	00	02	00	02	11*

(8) 设定模组内晶体频率 9.216MHz.



(9) devTEF665x_APPL_Activate(0x0001)

```
64 Nrite Transaction 40 05 01 00 01
```

- (10)) 延时 100ms。
- (11) 状态检测模式检测 0X02。

	g			_
64	Nrite Trans	40	80 01	
64	Read Trans	00	02*	

(12) TEF6686 FM band 收音设置命令

```
void devTEF6686_FM_para_command_boot(void)
{
   int i,j,k=1;
   unsigned char num;
   unsigned char buf[50];

for(i=0;i<sizeof(TEF6686_set_FM_command);i+=(TEF6686_set_FM_command[i]+1))
   {
      num=TEF6686_set_FM_command[i];   k=i+1;
      for(j=0;j<num;j++,k++) buf[j]=TEF6686_set_FM_command[k];
      devTEF665x_Write(buf, num);
   }
}</pre>
```

```
static const unsigned char TEF6686 set FM command[]=
0x07,0x20,0x01,0x01,0x00,0x01,0x26,0x52,
                                              // FM_Tune_To
0x05,0x20,0x16,0x01,0x00,0x01,
                                         // FM_Set_ChannelEqualizer
                                         // FM_Set_MphSuppression
0x05,0x20,0x14,0x01,0x00,0x01,
0x0B,0x20,0x28,0x01,0x00,0x78,0x01,0xF4,0x00,0x0A,0x00,0x14, // FM Set SoftMute Time
                                                      // FM_Set_SoftMute_Mph
0x09,0x20,0x2C,0x01,0x00,0x00,0x00,0xC8,0x03,0xE8,
                                                      // FM_Set_SoftMute_Noise
0x09,0x20,0x2B,0x01,0x00,0x00,0x00,0xC8,0x03,0xE8,
0x0B,0x20,0x32,0x01,0x00,0xC8,0x07,0xD0,0x00,0x0A,0x00,0x50, // FM_Set_HighCut_Time
0x09,0x20,0x36,0x01,0x00,0x03,0x00,0x78,0x00,0xA0, // FM_Set_HighCut_Mph
                                                      // FM Set HighCut Noise
0x09,0x20,0x35,0x01,0x00,0x03,0x00,0x96,0x00,0xC8,
0x07,0x20,0x37,0x01,0x00,0x01,0x09,0x60,
                                                // FM_Set_HighCut_Max
                                                // FM_Set_LowCut_Max
0x07,0x20,0x39,0x01,0x00,0x01,0x00,0x64,
0x0B,0x20,0x3C,0x01,0x00,0xC8,0x0F,0xA0,0x00,0x14,0x00,0x50, // FM_Set_Stereo_Time
0x09,0x20,0x40,0x01,0x00,0x03,0x00,0x64,0x00,0x96,
                                                      // FM_Set_Stereo_Mph
                                                      // FM_Set_Stereo_Noise
0x09,0x20,0x3F,0x01,0x00,0x03,0x00,0x78,0x00,0xA0,
                                                      // FM_Set_StHiBlend_Mph
0x09,0x20,0x4A,0x01,0x00,0x03,0x00,0x50,0x00,0x8C,
                                                      // FM_Set_StHiBlend_Noise
0x09,0x20,0x49,0x01,0x00,0x03,0x00,0x50,0x00,0x8C,
0x09,0x40,0x03,0x01,0x00,0x00,0x00,0x21,0x00,0x03,
                                                      // APPL_Set_GPIO
0x09,0x40,0x03,0x01,0x00,0x01,0x00,0x21,0x00,0x00,
                                                      // APPL_Set_GPIO
                                                      // APPL_Set_GPIO
0x09,0x40,0x03,0x01,0x00,0x02,0x00,0x21,0x00,0x00,
0x09,0x40,0x03,0x01,0x00,0x00,0x00,0x20,0x00,0x03,
                                                      // APPL_Set_GPIO
                                                      // APPL_Set_GPIO
0x09,0x40,0x03,0x01,0x00,0x01,0x00,0x20,0x00,0x00,
                                                      // APPL_Set_GPIO
0x09,0x40,0x03,0x01,0x00,0x02,0x00,0x20,0x00,0x00,
0x0D,0x30,0x16,0x01,0x00,0x20,0x00,0x00,0x20,0x00,0x11,0x3A, // AUDIO_Set_Dig_IO
0x05,0x30,0x0B,0x01,0x00,0x00,
                                            // AUDIO Set Mute
                                // AUDIO_Set_Volume
0X05,0x30,0x0A,0x01,0x00,0x32,
0x07,0x30,0x0D,0x01,0x00,0x80,0x00,0xE0 // AUDIO_Set_Output_Source
};
```

64	📏 Write Trans	20	01	01	00	01	26	52				
64	Nrite Trans	20	16	01	00	01						
64	Nrite Trans	20	14	01	00	01						
64	Nrite Trans	20	28	01	00	78	01	F4	00	0A	00	14
64	Nrite Trans	20	2C	01	00	00	00	C8	03	E8		
64	Nrite Trans	20	2B	01	00	00	00	C8	03	E8		
64	Nrite Trans	20	32	01	00	C8	07	D0	00	0A	00	50
64	Nrite Trans	20	36	01	00	03	00	78	00	A0		
64	Nrite Trans	20	35	01	00	03	00	96	00	C8		
64	Nrite Trans	20	37	01	00	01	09	60				
64	Nrite Trans	20	39	01	00	01	00	64				
64	Nrite Trans	20	3C	01	00	C8	0F	A0	00	14	00	50
64	Nrite Trans	20	40	01	00	03	00	64	00	96		
64	Nrite Trans	20	3F	01	00	03	00	78	00	A0		
64	Nrite Trans	20	4A	01	00	03	00	50	00	8C		
64	Nrite Trans	20	49	01	00	03	00	50	00	8C		
64	Nrite Trans	40	03	01	00	00	00	21	00	03		
64	Nrite Trans	40	03	01	00	01	00	21	00	00		
64	Nrite Trans	40	03	01	00	02	00	21	00	00		

64	Nrite Trans	40	03	01	00	00	00	20	00	03				
64	Nrite Trans	40	03	01	00	01	00	20	00	00				
64	Nrite Trans	40	03	01	00	02	00	20	00	00				
64	Nrite Trans	30	16	01	00	20	00	00	00	20	00	00	11	3A
64	Nrite Trans	30	16	01	00	21	00	00	00	20	00	00	11	3A
64	Nrite Trans	30	0B	01	00	00								
64	Nrite Trans	30	0A	01	00	32								
64	Nrite Trans	30	0D	01	00	80	00	ΕO						

(13) 状态检测模式检测 0X03。

_	-		
64	Write Trans	40	80 01
64	Read Trans	00	03*

(14) FM 频率设置。

64 Nrite Trans	20 01 01 00 01 22 2E
----------------	----------------------

软件纠错处理: 收音高频头 9pin:SDA 或者 10pin:SCL 对地 GND 接一个按键开关(50~100ms),开机时,按一下按键开关,相当于干扰严重一样,模拟通讯数据出错,MCU 无纠错处理,此时收音无声 不能收台,收音完全不能工作。如果 MCU 软件有纠错处理,接一个按键开关,收音可以正常工作。

2. TEF6686 波段切换

```
void devTEF6686_AM_para_command_boot(void)
{
  int i,j,k=1;
  unsigned char num;
  unsigned char buf[50];

  for(i=0;i<sizeof(TEF6686_set_AM_command);i+=(TEF6686_set_AM_command[i]+1))
  {
    num=TEF6686_set_AM_command[i];  k=i+1;
    for(j=0;j<num;j++,k++) buf[j]=TEF6686_set_AM_command[k];
    devTEF665x_Write(buf, num);
  }
}</pre>
```

								
64	Write Trans	21	01	01	00	01	03	E7
64	Nrite Trans	21	52	01	00	C8	00	00
64	Nrite Trans	21	52	01	00	00	00	00
64	Nrite Trans	30	0A	01	00	32		
64	Write Trans	21	01	01	00	02	02	0A
64	Nrite Trans	21	01	01	00	07		

```
      FM_Tune_To (1, 1, 8930)
      Preset tuning to FM 89.3 MHz

      AM_Tune_To (1, 2, 990)
      Search tuning (from FM) to AM 990 kHz

      AM_Tune_To (1, 7)
      End (release mute of AM Search action)

      [ w 20 01 01 0001 22E2 ]
      Preset tuning to FM 89.3 MHz

      [ w 21 01 01 0002 03DE ]
      Search tuning (from FM) to AM 990 kHz

      [ w 21 01 01 0007 ]
      End (release mute of Search action)
```

- 3. TEF6686 频率切换
- (1) FM 频率设定

```
0x20,0x01,0x01,0x00,0x01,0x22,0xE2, // FM_Tune_To
```

(2) AM 频率设定

OX21, OX01, OX01, OX00, OX01, OX03, OXDE, // AM_Tune_To

4. FM AM SEEK 通过读取 LEVEL USN WAM OFFSET 停台。

64	Write Trans	20 80 01
64	Read Trans	00 41*
64	Nrite Trans	20 80 01
64	Read Trans	00 7D*
64	Nrite Trans	20 81 01
64	Read Trans	03 E8 00 8A 00 14 00 56 FE DB 02 D0 00 6E*

```
64
       Write Trans... 20 01 01 00 02 22 7E
64
        Write Trans... 20 80 01
       Read Trans... 00 41*
64
       Write Trans... 20 80 01
64
64
       Read Trans... 00 7D*
64
       Write Trans... 20 81 01
       🐤 Read Trans... 03 E8 01 E1 00 1E 00 1D FF DA 09 38 00 2D*
64
64
       Write Trans... 20 01 01 00 01 22 7E
64
        Write Trans... 20 85 01
                                 立体声检测pilot
64
       Read Trans... 80 00*
```

```
//AM auto SEEK
    if((Band==MW)||(Band==LW)||(Band==SW))
    {
        LCDDisplayFreq();
        devTEF665x_SeekMode_freq (RF_Freq);
        delay_ms(20);
        fm=0;
        devTEF665x_Radio_Get_Quality_Status (fm,&status);
            tl=status;
        delay_ms(2);
        devTEF665x_Radio_Get_Quality_Status (fm,&status);
        t2=status;
        if((t1==0)||(t2==0)) continue;
    }

if((Band==MW)||(Band==LW)||(Band==SW))
    {
        delay_ms(30);
        devTEF665x_Radio_Get_Quality_Data (fm,&level,&usn,&wam,&offset,&bandwidth);
        if(offset<0x8000) IFC=offset;
        else IFC=0x10000-offset;
        if((level>=0xf000)||(IFC>=0x10)) continue;
        else if((level>=0xf000)||(IFC>=0x10)) & (IFC<0x08))
        { LitLED(TUNING,On);
            devTEF665x_Radio_freq (RF_Freq);
            break;
        }
    }
    *while(1);</pre>
```