

V69 Pm 编译说明

V69 pm 代码和 v59pm 是同一套代码，在使用的时候需要选用不同的配置，包括 32M 和 64M 的不同，100pin/128pin/156pin 的区别等，在这里统一说明一下如何配置。另外需要注意的是，由于硬件 I/O 口的不一致性，对不同的 pcb 需要不同的配置，因此在代码中添加了 PCB 版本型号的定义 PCB_VERSION 加以区分，当选用不同芯片时注意修改对应的 pcb 型号。

首先介绍下 pm 的基本配置项，进入 sboot 文件夹，make clean 清除，make menuconfig 开始配置 pm，进入画面：

```
Platform Configuration ---> 修改项1
General Configuration ---> 修改项2
Build Options --->
Debugging Options --->
Installation Options --->
Module Options --->
---
Load an Alternate Configuration File
Save Configuration to an Alternate File
```

-----以上标记的地方是需要修改的，其他的不需要。

Platform Configuration 设置

```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
MStar Chip Family (Macaw12) ---> 1. 芯片类型
CPU Core Selection (AEON R2 Only) --->
Booting mode (CPU Booting from external SPI) --->
CPU Clock Selection (216MHz) --->
Board Selection (BD_MST031B_10AL0_11523) 2. 具体芯片型号
MStar Panel Selection (PNL_WXGA_AU20_T200XW02) --->
Memory Map Type Selection (Macaw12 64MB) 3. 内存大小
Memory Frequency Selection (800 MHz) --->
BIST Length Selection (0x1001 units) --->
(0x81000000) SBOOT BOOTRAM Memory Address
(0x41200000) UBOOT BOOTRAM Memory Address
(256) URAM Size for UBoot relocation (MB)
[ 1 ] C... MStar... CTD
```

- *****第一项是修改芯片类型，v59/v69 都是 Macaw12
- *****第二修改项是配置不同的 board, board 是用来区分不同芯片的，如 32M 和 64M, 100pin 和 128pin, 带 dtv 还不带 dtv 等，一定要选择使用的芯片。
- *****第三修改项选择 mem 类型，是 64M 还是 32M
- *****其他项使用默认设置，无需修改

General Configure 设置

```
Buffer allocation policy (Allocate with Malloc) --->
[ ] UART1 Enable
[ ] MIPS boot from eMMC
[ ] Sboot to kernel
[ ] Show terse applet usage messages
[*] Build bootloader without U-Boot (Non-OS)
[ ] Secure Boot
[ ] Build bootloader with compressed U-Boot
[*] Build PM binary 选择时编译为pm，不选时编译为sboot
```

- *****按照上面的设置即可
- *****最后一项是选择编译 pm 代码还是 sboot 代码，选择是 pm，取消是 sboot。

下面是 v59/v69 用到的 board 选择, 对应 Board Selection 项。

1. v69 DTMB

- a) 这个属于 156pin , 64M 类型 , 配置 :
BD_MST034B_10AL6_12071

2. v59 64M

- a) 100pin: BD_MST031B_10AL0_11523
b) 128pin : BD_MST149B_D01A_S

3. V59 32M

- a) 100pin: BD_MST031B_10AL0_11523_32
b) 128pin: BD_MST149B_D01A_S_32

在 menuconfig 可视化菜单中添加新的选项

以新添加的 Board BD_MST149B_D01A_S_32 为例，如下：

在 config.in 文件中修改，添加新项如下：

```
854         bool "BD_MST031B_10AL0_11523"
855     config MSTAR_MACAW12_BD_MST034B_10AL6_12071
856         bool "BD_MST034B_10AL6_12071"
857     config MSTAR_MACAW12_BD_MST034B_20AL6_12071
858         bool "BD_MST034B_20AL6_12071"
859     config MSTAR_MACAW12_BD_MST031B_10AL0_11523_32
860         bool "BD_MST031B_10AL0_11523_32"
861     config MSTAR_MACAW12_BD_MST149B_D01A_S_32
862         bool "BD_MST149B_D01A_S_32"
863     endif
864
865 endif
```

然后再做 make menuconfig 就会有这个选项了

```
3      ( ) BD_MST034B_20AL6_12071
3      ( ) BD_MST031B_10AL0_11523_32
3      (X) BD_MST149B_D01A_S_32 新增加的项
```

在代码中要添加对应的设置

```
00197: #define BD_MST031B_10AL0_11523_32 0x085F // F
00198: #define BD_MST030B_10AL8_12052_32 0x0860
00199: #define BD_MST149B_D01A_S_32 0x0861
00200:
00201: #define BD_MST034B_20AL6_12071 0x0862
```

```
4: #elif defined (CONFIG_MSTAR_MACAW12_BD_MST149B_D01A_S_32)
5:     #define MS_BOARD_TYPE_SEL BD_MST149B_D01A_S_32
6: #elif defined (CONFIG_MSTAR_MACAW12_BD_MST034B_10AL6_12071)
7:     #define MS_BOARD_TYPE_SEL BD_MST034B_10AL6_12071
8: #elif defined (CONFIG_MSTAR_MACAW12_BD_MST034B_20AL6_12071)
```

```
304: #elif (MS_BOARD_TYPE_SEL == BD_MST149B_D01A_S)
305:     #include "BD_MST149B_D01A_S.h"
306: #elif (MS_BOARD_TYPE_SEL == BD_MST149B_D01A_S_32)
307:     #if (PCB_VERSION == MSTU59_128P_V60)
308:     #include "BD_MST149B_D01A_S_TSU59S_V60.h"
309:     #else
310:     #error "not defined pcb version!"
311:     #endif
```

内容修改

Pm 代码编译后生成的 bin 有两个，分别是 pm.bin 和 sbboot.bin。

sbboot.bin 作用于开机，在上面介绍的内容都设置正确的时候，只需要设置 gpio 正确就可以了。

Pm.bin 作用于待机状态，负责待机状态下的 gpio 状态设置，外面信号的检测唤醒（CEC,IR,VGA,USB 等等），一般出问题都是在 pm.bin。配置的时候一般只需修改 gpio 和按键板，有时候需要修改 ir。

修改 1: gpio 设定

```
:  
: #define PIN_FLASH_WP0 PIN_131  
:  
  
//-----GPIO setting(default GPIO pin level)-----  
#define PIN_61_IS_GPIO GPIO_OUT_HIGH // MUTE_  
#define PIN_132_IS_GPIO GPIO_OUT_HIGH // PWR  
#define PIN_140_IS_GPIO GPIO_OUT_LOW // VBL  
#define PIN_65_IS_GPIO GPIO_OUT_LOW // PANEL  
#define PIN_131_IS_GPIO GPIO_OUT_LOW // SPI  
#define PIN_59_IS_GPIO GPIO_IN // Tuner_  
#define PIN_60_IS_GPIO GPIO_IN // Tuner
```

Gpio 设定，pm 和 sbboot 共用，修改此处就可以了。文件为对应的 bd_xxxx.h。如下,PCB_VERSION 一定要定义正确。

```
: #include "BD_MST149B_D01A_S.h"  
: #elif (MS_BOARD_TYPE_SEL == BD_MST149B_D01A_S_32)  
:     #if (PCB_VERSION == MSTU59_128P_V60)  
:         #include "BD_MST149B_D01A_S_TSU59S_V60.h"  
:     #else  
:         #error "not defined pcb version!"  
:     #endif
```

修改 2: keypad 设置

```
//-----MST Keypad definition-----
#define KEYPAD_CHANNEL_SUPPORT          4 //Maximun supported
#define ADC_KEY_CHANNEL_NUM             2 //Real supported ke
#define ADC_KEY_LAST_CHANNEL            ADC_KEY_CHANNEL_NUM
//config which keypad channel enabled
#define ENABLE_KPDCHAN_1                ENABLE
#define ENABLE_KPDCHAN_2                DISABLE
#define ENABLE_KPDCHAN_3                DISABLE
#define ENABLE_KPDCHAN_4                DISABLE
```

该段代码下的 keypad 相关代码都要修改，可参考 APcode 配置

```
if(abs(Key_Value - tADCKeyLevel[u8ChIdx][j]) < ADCKeyValueOffset)
if (Key_Value < tADCKeyLevel[u8ChIdx][j])
```

两种按键判断方式，默认我们使用上面的这种。

修改 3: IR 设置

如果遥控器是 NEC 码的，不需要修改 pm 代码。因为所有的 NEC 码解码方式是相同的，只需要从 AP 层获取唤醒键的键值就可以。如果遥控器不是 NEC 码的，则需要修改 pmcode 中的解码方式，代码和 ap 层相同，可以参考 ap 层解码方式。然后同样是获取 ap 层传递的唤醒键键值。Pm 从 AP 层获取唤醒键值方式如下：

在文件

```
Bootaeonsysinit.c (x:\work6m48\y59\mboot_m12_496632\mboot_m12\src\macaw12)
Bootaeonsysinit.c (x:\work6m48\y59\mboot_m12_496632\mboot_m12\src\macaw12\pm\core)
Bss.h (x:\work6m48\y59\mboot_m12_496632\mboot_m12\src\include\config\feature\buffers\golin)
```

获取函数


```
// Init Wakeup Device
MDrv_CheckWakeupDeviceFromAP();
```

根据唤醒标志初始化参数，包括 IR

```
: void MDrv_CheckWakeupDeviceFromAP(void)
: {
:     wPM_WakeUpDevice = MDrv_Read2Byte(PM_SLEEP_WAKEUP_DEVICE_FLAG);
:     WKDBG(printf("wPM_WakeUpDevice = 0x%x \r\n", wPM_WakeUpDevice));
:
:     if (wPM_WakeUpDevice & PM_WAKEUP_BY_KEYPAD)
:     {
:         // init Keypad by default
:     }
:
:     if (wPM_WakeUpDevice & PM_WAKEUP_BY_IR)
:     {
:         // init IR by default
:     }
: }
```

获取 AP 层传递的唤醒 IR 键值

```
static void MDrv_IR_ConfigWakeup(void)
{
    // store IR wakeup keys
    g_u8IRWakeupKey[0] = MDrv_ReadByte(STNADBY_WAKEUP_IR_KEY0);
    g_u8IRWakeupKey[1] = MDrv_ReadByte(STNADBY_WAKEUP_IR_KEY1);
    g_u8IRWakeupKey[2] = MDrv_ReadByte(STNADBY_WAKEUP_IR_KEY2);
    g_u8IRWakeupKey[3] = MDrv_ReadByte(STNADBY_WAKEUP_IR_KEY3);
}
```

判断检测到的 IR 值是否是 AP 指定的唤醒键

```
BOOL MDrv_IR_IsWakeup(U8 u8Key, U8 IRCode0)
{
    U8 i;
    if(IRCode0)
    { //FOR IR2 POWER
        for(i=2; i<MAX_IR_WAKEUP_KEYS; i++)
        {
            if(u8Key==g_u8IRWakeupKey[i])
                return TRUE;
        }
    }
}
```

匹配则唤醒

```
//check IR wakeup key
if(stKeyStatus.keytype == KEY_TYPE_IR)
{
    // Check if it is wakeup key
    if(MDrv_IR_IsWakeup(stKeyStatus.keydata, stKeyStatus.keyrepeat)
    {
        //force it to power key
        u8KeyCode = KEY_WAKEUP;
        return;
    }
}
```

```
break;
case KEY_WAKEUP:
    if ( stKeyStatus.keytype == KEY_TYPE_IR )
    {
        msAPI_Power_NotifyWakeupDeviceForAP(WakeUp_by_IR);
        //printf(" KEY_WAKEUP WakeUp >>\r\n");
        SystemWholeChipReset();
    }
}
```

AP 层传递参数方式:

```
void MDrv_Power_StandbyPowerOff(void)
{
    printf("=> StandBy \n"); 设置唤醒方式, 如vga, cec, ir,
    MDrv_AUDIO_SetPowerOn(0); keypad等

    MDrv_Power_Standby_CheckWakeupDevice();
    msAPI_Power_SetWakeUpDevice();
    //configure wakeup devices
    _MDrv_IR_ConfigWakeUp(); 设置IR唤醒键键值
}
```

```
static void _MDrv_IR_ConfigWakeUp(void)
{
    // set header code for HW NEC decode
    MDrv_WriteByte(STNADBY_IR_HEADER_CODE0_1, IR_HEADER_CODE0);
    MDrv_WriteByte(STNADBY_IR_HEADER_CODE0_2, IR_HEADER_CODE1);
    //support 2nd NEC Full decode
    MDrv_WriteByte(STNADBY_IR_HEADER_CODE1_1, IR_HEADER_CODE_2ND0);
    MDrv_WriteByte(STNADBY_IR_HEADER_CODE1_2, IR_HEADER_CODE_2ND1);
    MDrv_WriteRegBit(STNADBY_IR_HEADER_CODE1_EN, IR_2ND_NEC_DE
    // set power-on key
    #if 1 //Configure IR_WAKEUP_KEYX in IR_XXX_YYY.h
    MDrv_WriteByte(STNADBY_WAKEUP_IR_KEY0, IR_WAKEUP_KEY0); //FOR IF
    MDrv_WriteByte(STNADBY_WAKEUP_IR_KEY1, IR_WAKEUP_KEY1); //FOR IF
    MDrv_WriteByte(STNADBY_WAKEUP_IR_KEY2, IR_WAKEUP_KEY2); //FOR IF
    MDrv_WriteByte(STNADBY_WAKEUP_IR_KEY3, IR_WAKEUP_KEY3); //FOR IF
    #endif
}
```


待机后无法开机

待机后无法开机分两种情况：

1. AP 层代码进入 PM 失败，PM 没有正常启动。

这时候需要确认一下是否在 AP 层修改了什么，还原一下看是否会 ok。

2. 进入 PM，但无法唤醒

这种情况，请按照上面介绍的 IR 配置方式和 KEYPAD 配置方式核对一下，是否正常，如果没有发现问题，可以在 pm 中添加打印来确认。

3. 如何确认 PM 是否在运行。

默认情况下，PM 的打印信息都是打开的，波特率为 38400.打开 MSTV tools，看看是否有打印信息，是否可以读写寄存器，如果可以读写那么就说明 PM 运行正常，否则就是 PM 启动有问题了。