Base-band Driver Configuration

2004/1/3

Ian

Agenda

- MCP/Nand and Scatter file
- Audio Relative settings
- GPIO
- UEM configuration
- HW Default Setting
- ADC Scheduler
- EINT
- Aux Task
- UARTUSB
- Charging Parameter
- Keypad
- Mass storage disk mount
- PWM Control
- META
- Compile option
- Build command

MCP

- *@ mcu\custom\system\board version\custom_emi.c*
- MT6218B

```
#define EMI_NUMBER_OF_ENTRIES   2

/* struct EMI_NODE_T {
  kal_uint16      chip_selct;   (Chip Select)
  EMI_MODE_TYPE   mode;         (Normal, Page, Burst Mode)
  EMI_WAIT_STATE   RLT;         (Ready Latency Time)
  EMI_WAIT_STATE       WST;         (Write Wait State)
  EMI_PAGE_SIZE    page_size;    (Page/Burst size for Page/Burst Mode Operation)
  EMI_WAIT_STATE   PRLT;        (Read Latency within the same page or in Burst Mode Operation)
  EMI_DATA_WIDTH   data_width;   (8bits or 16bits device)
  EMI_RBLN         RBLN;
  EMI_SETUP_TIME   C2WS, C2RS;
  EMI_HOLD_TIME    C2WH;
} EMI_NODE_TYPE; */

/* \CS0 ~ \CS7 must be defined orderedly */
const EMI_NODE_TYPE  EMI_Node[EMI_NUMBER_OF_ENTRIES] =
#if defined(MCU_52M)
{
 /* Fujitsu MB84VP24491HK-70 (16MByte Flash) */
 {EMI_CS0, EMI_PAGE_MODE, EMI_4_WAIT_STATE, EMI_2_WAIT_STATE, EMI_16B_PER_PAGE, EMI_1_WAIT_STATE, EMI_16BIT_DEVICE,
  EMI_RBLN_ENABLE, EMI_SETUP_TIME_1, EMI_SETUP_TIME_1, EMI_HOLD_TIME_0},
 /* Fujitsu MB84VP24491HK-70 (4MByte FCRAM) */
 {EMI_CS1, EMI_BURST_MODE, EMI_4_WAIT_STATE, EMI_2_WAIT_STATE, EMI_16B_PER_PAGE, EMI_1_WAIT_STATE, EMI_16BIT_DEVICE,
  EMI_RBLN_ENABLE, EMI_SETUP_TIME_1, EMI_SETUP_TIME_1, EMI_HOLD_TIME_0}
#else
{
 /* Fujitsu MB84VP24491HK-70 (16MByte Flash) */
 {EMI_CS0, EMI_PAGE_MODE, EMI_2_WAIT_STATE, EMI_1_WAIT_STATE, EMI_16B_PER_PAGE, EMI_1_WAIT_STATE, EMI_16BIT_DEVICE,
  EMI_RBLN_ENABLE, EMI_SETUP_TIME_1, EMI_SETUP_TIME_1, EMI_HOLD_TIME_0},
 /* Fujitsu MB84VP24491HK-70 (4MByte FCRAM) */
 {EMI_CS1, EMI_BURST_MODE, EMI_2_WAIT_STATE, EMI_1_WAIT_STATE, EMI_16B_PER_PAGE, EMI_1_WAIT_STATE, EMI_16BIT_DEVICE,
  EMI_RBLN_ENABLE, EMI_SETUP_TIME_1, EMI_SETUP_TIME_1, EMI_HOLD_TIME_0}
#endif
};
```

MCP

- *@ mcu\custom\system\board version\custom_emi.c*
- MT6205B

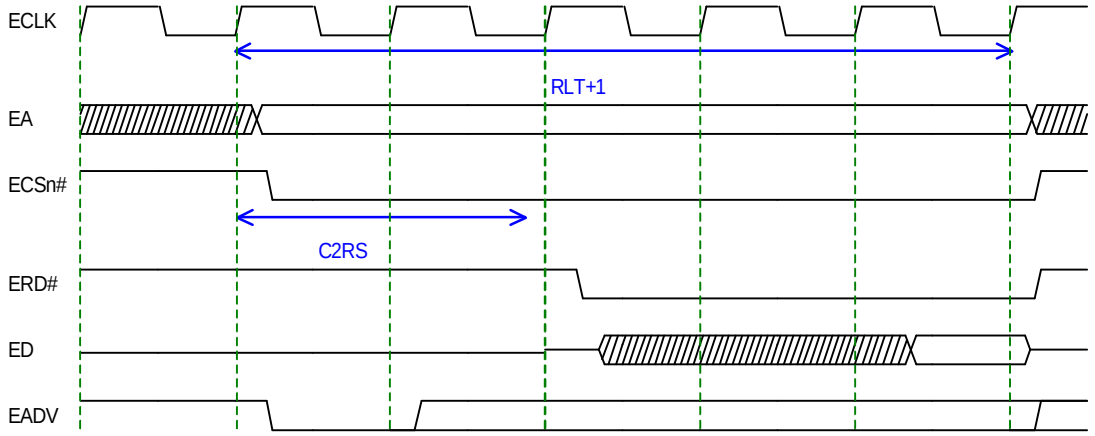
```
#define EMI_NUMBER_OF_ENTRIES   2

/* struct EMI_NODE_T {
   kal_uint16      chip_selct;   (Chip Select)
   EMI_MODE_TYPE   mode;         (Normal, Page, Burst Mode)
   EMI_WAIT_STATE   RLT;         (Ready Latency Time)
   EMI_WAIT_STATE       WST;         (Write Wait State)
   EMI_PAGE_SIZE    page_size;    (Page/Burst size for Page/Burst Mode Operation)
   EMI_WAIT_STATE   PRLT;        (Read Latency within the same page or in Burst Mode Operation)
   EMI_DATA_WIDTH   data_width;   (8bits or 16bits device)
   EMI_RBLN        RBLN;
   EMI_SETUP_TIME   C2WS, C2RS;
   EMI_HOLD_TIME    C2WH;
} EMI_NODE_TYPE; */

/* \CS0 ~ \CS7 must be defined orderedly */
const EMI_NODE_TYPE  EMI_Node[EMI_NUMBER_OF_ENTRIES] =
#if defined(MCU_13M)
{
 /* Flash */
 {EMI_CS0, EMI_NORMAL_MODE, EMI_1_WAIT_STATE, EMI_0_WAIT_STATE, EMI_NO_PAGE, EMI_0_WAIT_STATE, EMI_16BIT_DEVICE,
 EMI_RBLN_ENABLE, EMI_CSE_NO},
 /* SRAM */
 {EMI_CS1, EMI_NORMAL_MODE, EMI_1_WAIT_STATE, EMI_0_WAIT_STATE, EMI_NO_PAGE, EMI_0_WAIT_STATE, EMI_16BIT_DEVICE,
 EMI_RBLN_ENABLE, EMI_CSE_NO}
#else
{
 /* Flash */
 {EMI_CS0, EMI_NORMAL_MODE, EMI_2_WAIT_STATE, EMI_1_WAIT_STATE, EMI_NO_PAGE, EMI_0_WAIT_STATE, EMI_16BIT_DEVICE,
 EMI_RBLN_ENABLE, EMI_CSE_NO},
 /* SRAM */
 {EMI_CS1, EMI_NORMAL_MODE, EMI_2_WAIT_STATE, EMI_1_WAIT_STATE, EMI_NO_PAGE, EMI_0_WAIT_STATE, EMI_16BIT_DEVICE,
 EMI_RBLN_ENABLE, EMI_CSE_NO}
#endif
};
```

EMI Configuration

@mcu\init\include\init.h

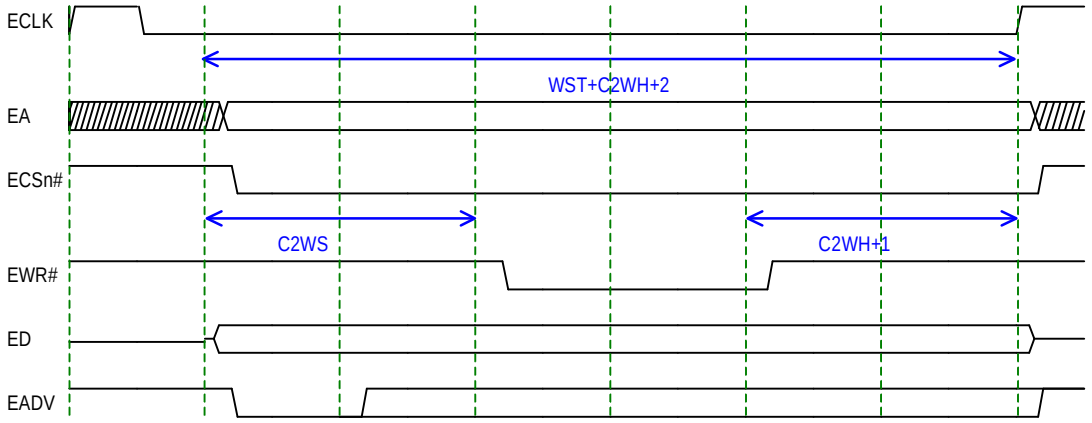
```
typedef struct EMI_NODE_T
{
  kal_uint16    chip_select;
   EMI_MODE_TYPE          mode;
   EMI_WAIT_STATE         RLT;
   EMI_WAIT_STATE         WST;
   EMI_PAGE_SIZE          page_size;
   EMI_WAIT_STATE         PRLT;
   EMI_DATA_WIDTH         data_width;
   EMI_RBLN RBLN;
   EMI_SETUP_TIME         C2WS;
   EMI_SETUP_TIME         C2RS;
   EMI_HOLD_TIME          C2WH;
}
EMI_NODE_TYPE;
```

READ



RLT=4, C2RS=2

WRITE



WST=3, C2WS=2, C2WH=1

EMI recommendation settings (by CS2)

Flash:

Setting	Page mode	Non-page mode
Mode setting	**Page mode**	Burst mode
RLT (read wait states)	3T, 4T	3T, 4T
PRLT (page read wait states)	1T ~ 2T	3T, 4T
C2RS (Read setup time)	1T	1T
WST (write wait states)	2T ~ 3T	2T ~ 3T
C2WS (Write setup time)	1T	1T

Red: Must not change!
Blue: Please refer to datasheet!

RAM/PSRAM:

Setting	Page mode	Non-page mode
Mode setting	**Burst mode**	Burst mode
RLT (read wait states)	3T, 4T	3T, 4T
PRLT (page read wait states)	1T ~ 2T	3T, 4T
C2RS (Read setup time)	1T	1T
WST (write wait states)	2T ~ 3T	2T ~ 3T
C2WS (Write setup time)	1T	1T

Suggested Setting

Before W04.36, @ mcu\custom\drv\misc_drv\board version\custom_drv_init.c, otherwise @ mcu\custom\system\board version\custom_emi.c

@ 52MHz, 70ns non-page mode MCP

```
const    EMI_NODE_TYPE  EMI_Node[2]
{
 {EMI_CS0,……..}
  /* SRAM */
   {EMI_CS1,
    EMI_BURST_MODE,
    EMI_3_WAIT_STATE,   /* RLT */
    EMI_2_WAIT_STATE,   /* WST */
    EMI_NO_PAGE,
    EMI_3_WAIT_STATE,   /* PRLT */
    EMI_16BIT_DEVICE,
    EMI_RBLN_ENABLE,
    EMI_SETUP_TIME_1,
    EMI_SETUP_TIME_1,
    EMI_HOLD_TIME_0},
};
```

Read latency (3+1)x19.2ns ~ 76.8ns
Write latency (2+2)x19.2ns ~ 76.8ns

@ 52MHz, 70ns page mode (30ns) MCP

```
const    EMI_NODE_TYPE  EMI_Node[2]
{
 {EMI_CS0,……..}
  /* SRAM */
   {EMI_CS1,
    EMI_BURST_MODE,
    EMI_4_WAIT_STATE,   /* RLT */
    EMI_2_WAIT_STATE,   /* WST */
    EMI_8B_PER_PAGE,
    EMI_2_WAIT_STATE,   /* PRLT */
    EMI_16BIT_DEVICE,
    EMI_RBLN_ENABLE,
    EMI_SETUP_TIME_1,
    EMI_SETUP_TIME_1,
    EMI_HOLD_TIME_0},
};
```

Random read latency (4+1)x19.2ns ~ 96ns
Page read latency (2+1)x19.2ns ~ 57.6ns
Write latency (2+2)x19.2ns ~ 76.8ns

Am49PDL127BH/Am49PDL129BH

Stacked Multi-Chip Package (MCP) Flash Memory and pSRAM

128 Megabit (8 M x 16-Bit) CMOS 3.0 Volt-only, Simultaneous Operation Flash Memory and 32 Mbit (2 M x 16-Bit) CMOS Pseudo Static RAM with Page Mode

DISTINCTIVE CHARACTERISTICS

MCP Features

- Power supply voltage of 2.7 to 3.3 volt
- High performance
  - Access time as fast as 65 ns initial / 25 ns page
- Package
  - 73-Ball FBGA
- Operating Temperature
  - –40°C to +85°C

Flash Memory Features

- Both top and bottom boot blocks in one device
- Manufactured on 0.13 µm process technology
- 20-year data retention at 125°C
- Minimum 1 million erase cycle guarantee per sector

PERFORMANCE CHARACTERISTICS

- High Performance
  - Page access times as fast as 25 ns
  - Random access times as fast as 65 ns

Read-Only Operations – Am29PDL129H

Parameter		Description		Test Setup		Speed Options		Unit
JEDEC	Std.					66	85	
t_{AVAV}	t_{RC}	Read Cycle Time (Note 1)			Min	65	85	ns
t_{AVQV}	t_{ACC}	Address to Output Delay (Note 3)		CE#f1, OE# = V_{IL}	Max	65	85	ns
t_{ELQV}	t_{CE}	Chip Enable to Output Delay (Note 4)		OE# = V_{IL}	Max	65	85	ns
	t_{PACC}	Page Access Time			Max	25	30	ns
t_{GLQV}	t_{OE}	Output Enable to Output Delay			Max	25	30	ns
t_{EHQZ}	t_{DF}	Chip Enable to Output High Z (Notes 1, 5, 6)			Max	16		ns
t_{GHQZ}	t_{DF}	Output Enable to Output High Z (Notes 1, 5)			Max	16		ns
t_{AXQX}	t_{OH}	Output Hold Time From Addresses, CE#f1/CE#f2 or OE#, Whichever Occurs First (Notes 5, 6)			Min	5		ns
	t_{OEH}	Output Enable Hold Time (Note 1)	Read		Min	0		ns
			Toggle and Data# Polling		Min	10		ns

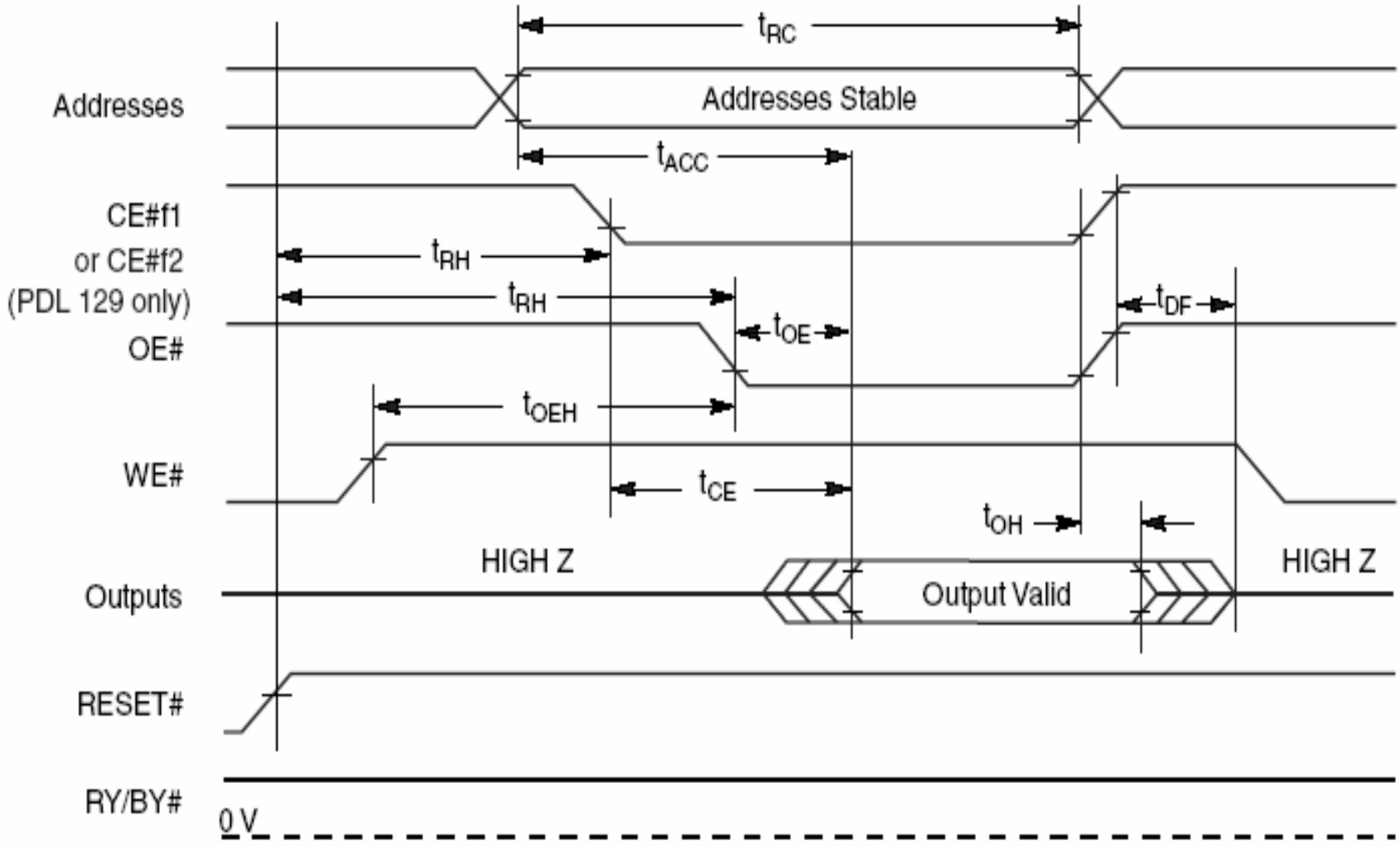


Figure 14. Read Operation Timings

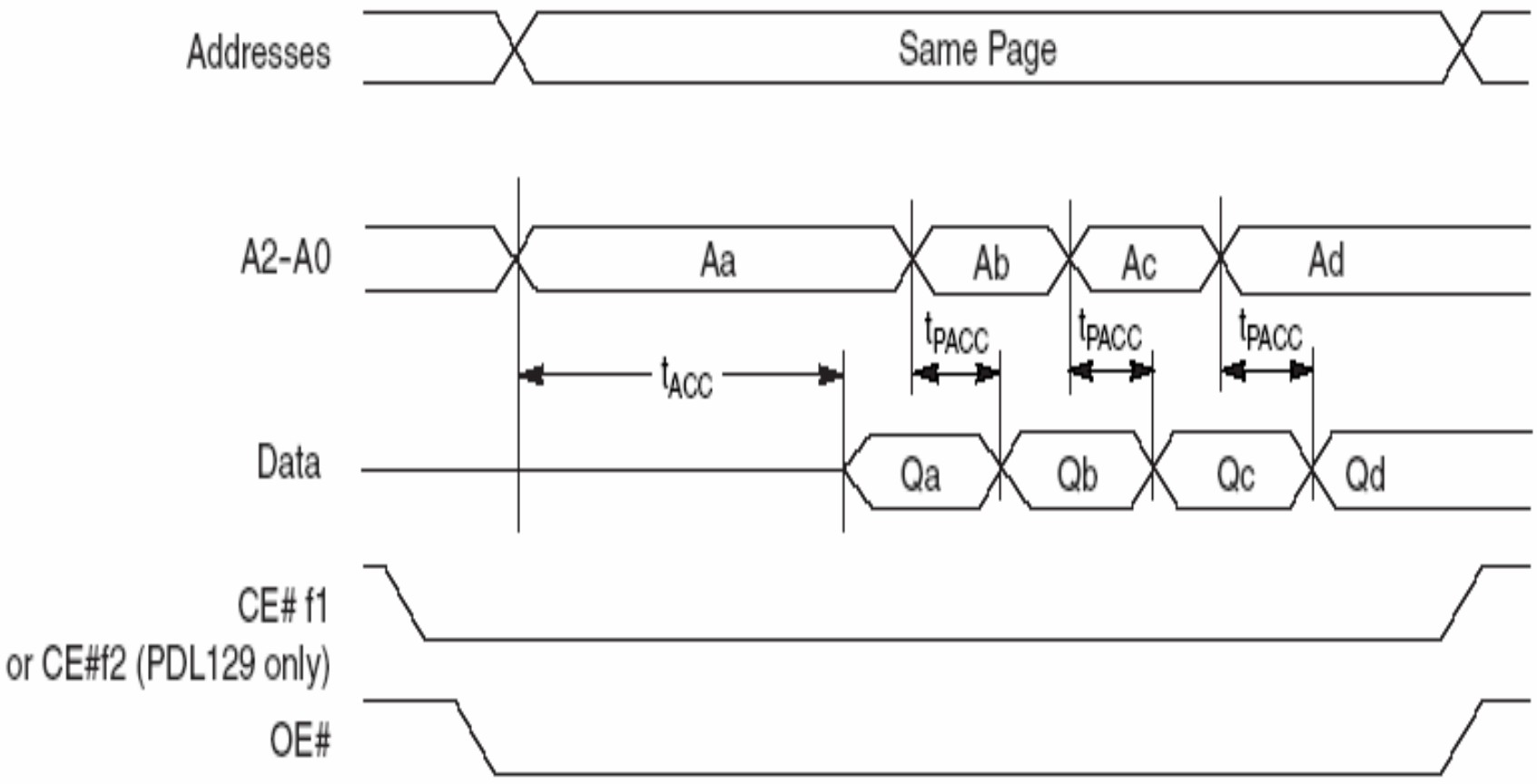


Figure 15. Page Read Operation Timings

Driving

Valid for W04.36 and later, @ mcu\custom\system\board version\custom_emi.c

```
const EMI_GENERAL EMI_general_setting =
{
   EMI_DRIVING_8mA,    ; Driving strength of address and data bus
   EMI_DRIVING_8mA,    ; Driving strength of ECLK (burst mode only)
   0                   ; clock delay control for burst mode flash
};
```

Singal Bin Scatter file overview

- *@ mcu\custom\system\board version\ScatBoard.txt*
- *Singal Bin 6218B*

```
ROM 0x08000000 0xe00000
{
  ROM 0x08000000 0x400000
  {
    ... (+RO-CODE)
  }
  ROM2 +0x0 0x400000
  {
    ... (+RO-CODE)
  }
  ROM3 +0x0
  {
    ... (+RO-DATA)
  }
  EXTSRAM 0x00000140 0x003FFEC0
  {
    * (+RW, +ZI)
  }
  INTSRAM_CODE 0x40000000 0x10000
  {
```

0x08000000 is the start address of \CS0

4MB execution region constraint of ADS 1.2 (Linker Error L6286E)

1. 0x0 is the start address of \CS1

2. 320 bytes (0x14) is reserved for vector table and a little code

0x40000000 is the start address of BB internal RAM

Singal Bin Scatter File

- *@ mcu\custom\system\board version\ScatBoard.txt*
- *Singal Bin 6218B*

```
ROM 0x08000000 0x00e00000 {
   ROM 0x08000000 0x400000
   { bootarm.obj (C$$code,+First)
    … }
   ROM2 +0x0 0x400000
   { .ANY (+RO-CODE) }
   ROM3 +0x0
   { .ANY (+RO-DATA) }
   EXTSRAM 0x00000140 0x003FFEC0
   {
       * (+RW, +ZI) }
   INTSRAM_CODE 0x40000000 0x00014000
   { isrentry.obj (INTERNCODE)
    …}
   INTSRAM_DATA 0x40014000 0x0000c000
   { init_memory_stack.obj (INTERNCONST, INTERNRW, INTERNZI)
    好多好多OBJ}
 }
```

Spansion 128+32 (bits)
0x00e00000 (bytes) **è** Size of the NOR – LAST BANK size(FAT Flashconf.c)
0x003FFEC0 (bytes) **è** PSRAM size – 0x00000140

Singal Bin Scatter File

- *@ \\\mtkrd5fs1\public\sa\RP2\Hank\Flash Verify Status\NOR-based MCP DATASHEET\SPANAION\AMD\S71PL127_129JB0_00_A0.pdf*

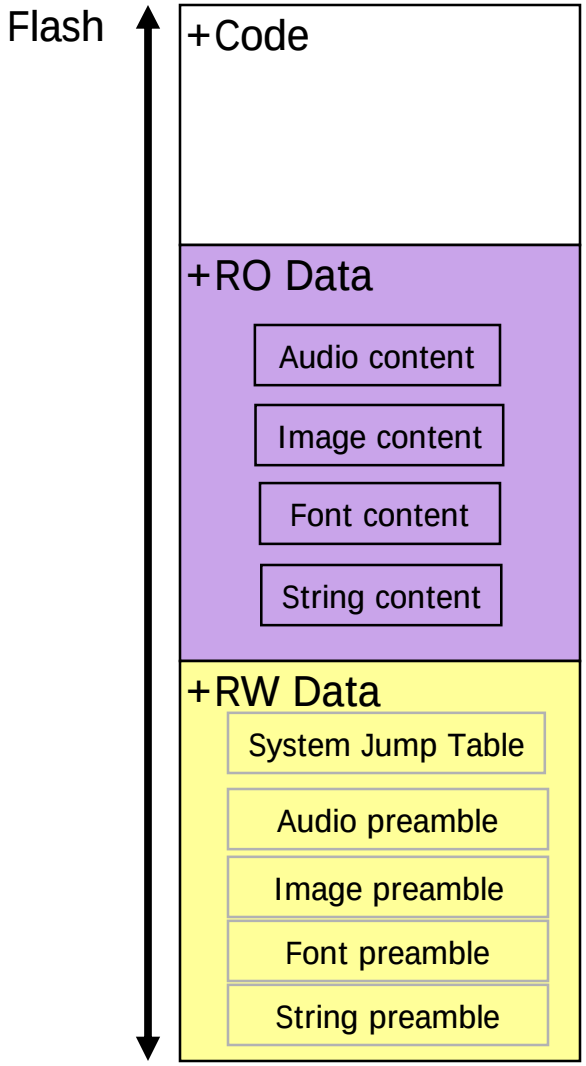
program operations for battery-powered applications

- **Dual Chip Enable inputs (PL129J)**
  - Two CE# inputs control selection of each half of the memory space
- **Simultaneous Read/Write Operation**
  - Data can be continuously read from one bank while executing erase/program functions in another bank
  - Zero latency switching from write to read operations
- **FlexBank Architecture**
  - 4 separate banks, with up to two simultaneous operations per device
  - Bank A: 16Mbit (4Kw x 8 and 32Kw x 31)
  - Bank B: 48Mbit (32Kw x 96)
  - Bank C: 48 Mbit (32Kw x 96)
  - Bank D: 16Mbit (4Kw x 8 and 32Kw x 31)

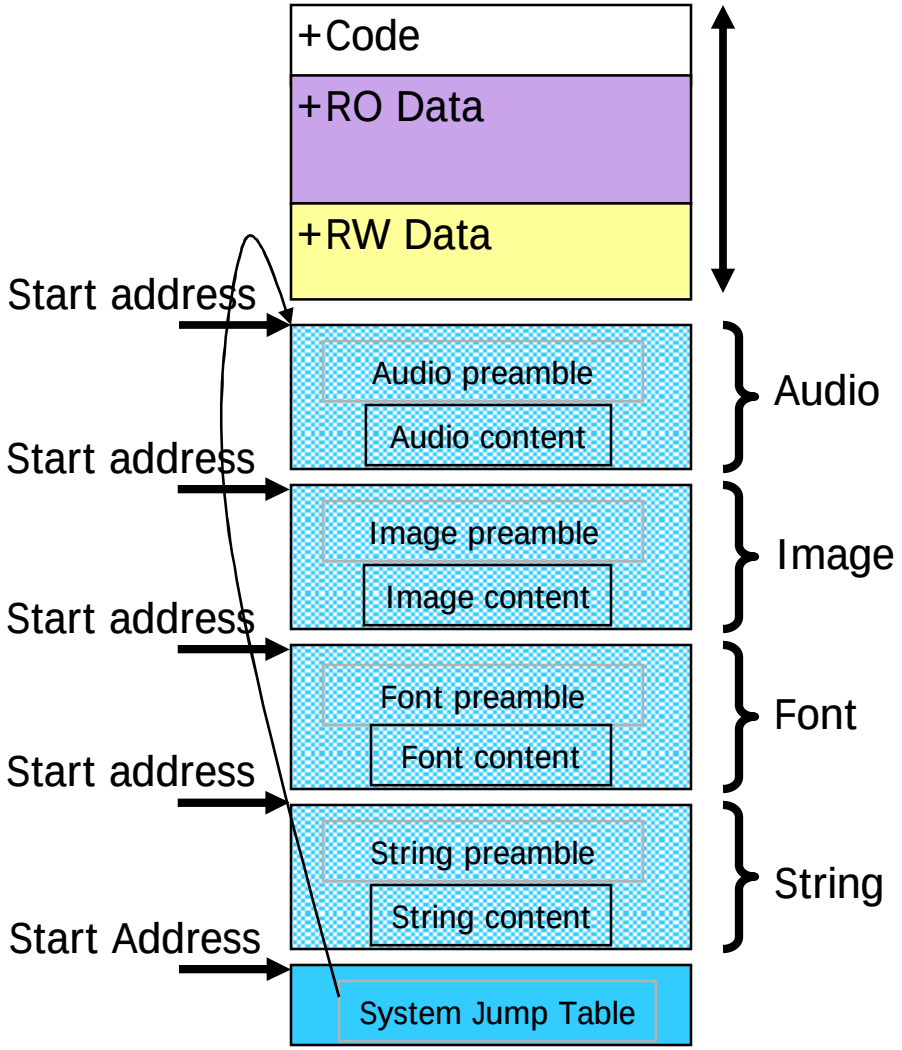
0x00e00000 (bytes)

(16+48+48)*1024*1024/8=14680064=0xe00000

Single-bin v.s. Multi-bin



Memory layout of Single-bin

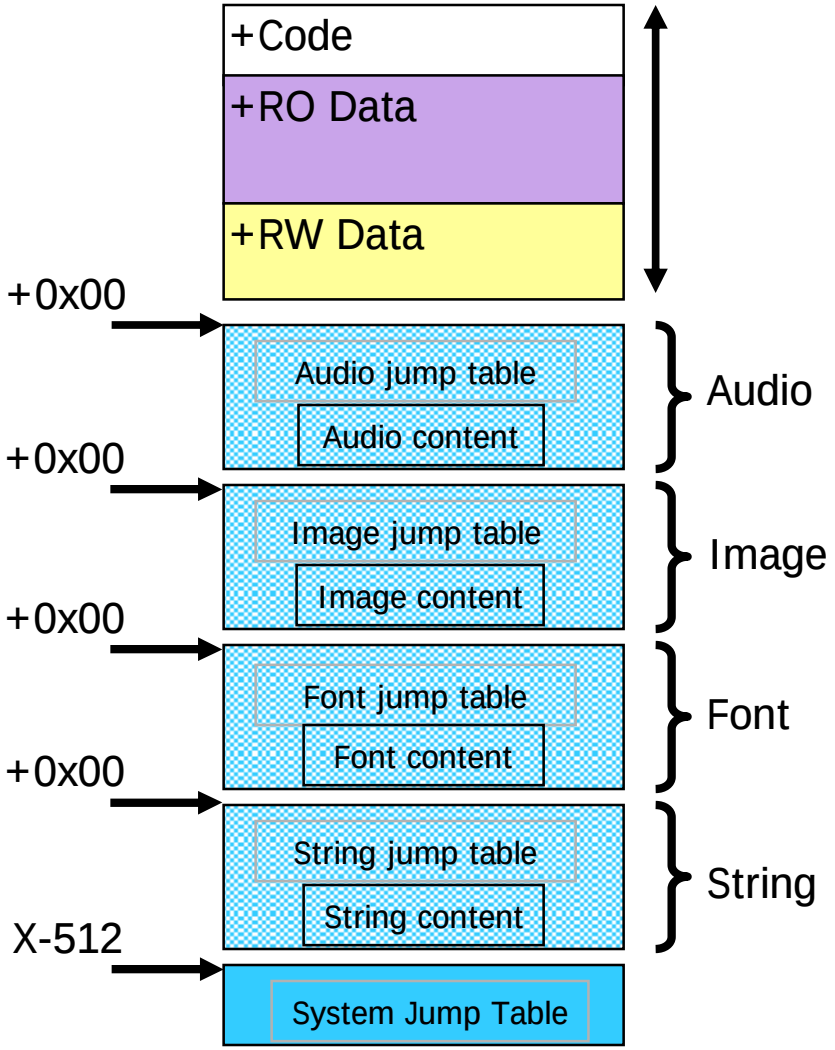


Memory layout of Multi-bin

Adjacent Multi-bin

```
ROM 0x08000000
{
}
AUDIO_ROM +0x0
{
}
IMAGE_ROM +0x0
{
}
FONT_ROM +0x0
{
}
STR_ROM +0x0
{
}
JUMP_TABLE 0x08dFFE00 0x200
{
{
```

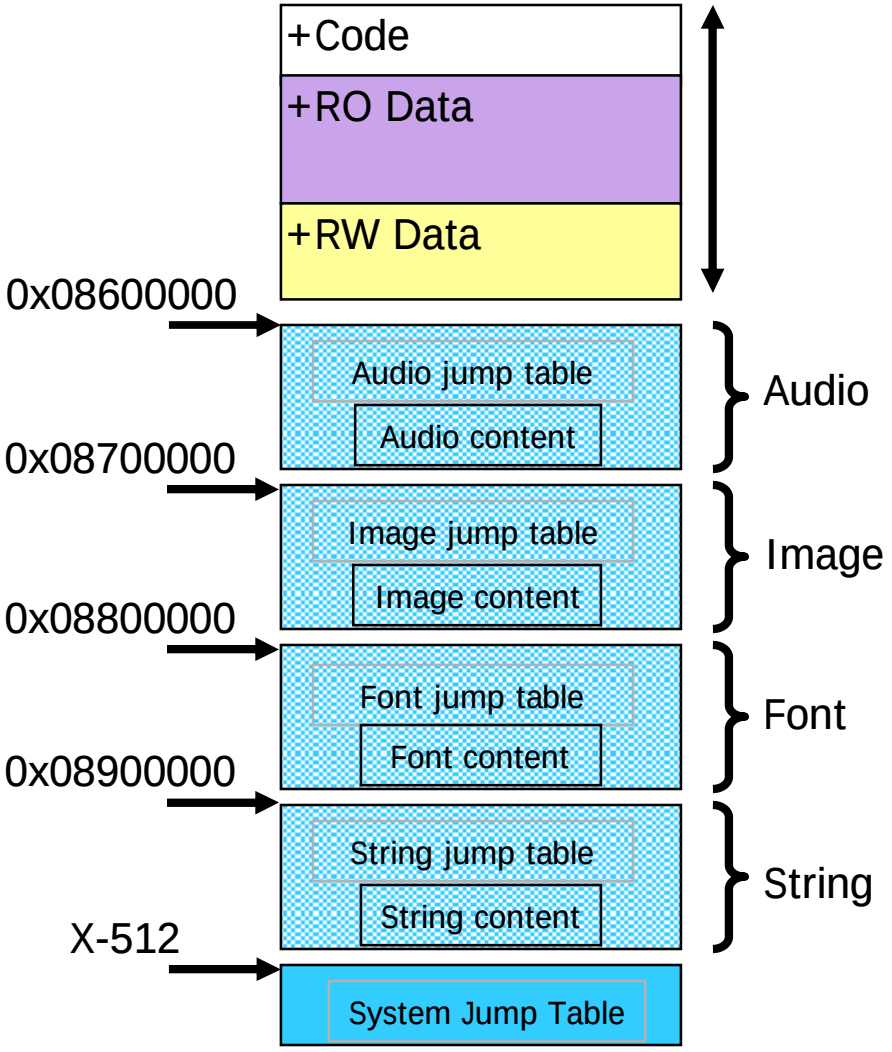
512 bytes prior to FAT



X: Start address of FAT

Absolute Multi-bin

```
ROM 0x08000000
{
}
AUDIO_ROM 0x08600000
{
}
IMAGE_ROM 0x08700000
{
}
FONT_ROM 0x08800000
{
}
STR_ROM 0x08900000
{
}
JUMP_TABLE 0x08dFFE00 0x200
{
{
```



X: Start address of FAT

Switch from Single-bin to Multi-bin (1/3)

@ mcu\custom\system\board version\custom_jump_tbl.h **(DRIVER WILL TAKE CARE)**

- Local compile option

#define MULTIPLE_BINARY_FILES

```
#ifdef MULTIPLE_BINARY_FILES

/* Macros below specific for multiple binary files or multiple load segments only */
/* Collection of audio */
extern kal_uint32  Load$$AUDIO_ROM$$Base;
extern kal_uint32  Image$$AUDIO_ROM$$Length;
extern kal_uint32  Image$$AUDIO_ROM_CONTENT$$Length;

/* macro of resource address and size for const arrary initialization */
#define CONST_RES_AUDIO_ADDR                (&Load$$AUDIO_ROM$$Base)
#define CONST_RES_AUDIO_TBL_SIZE            ((kal_uint32)&Image$$AUDIO_ROM$$Length)
#define CONST_RES_AUDIO_CONTENT_SIZE        ((kal_uint32)&Image$$AUDIO_ROM_CONTENT$$Length)

#else

extern Type_Preamble_Content AudioResPreamble;

#define CONST_RES_AUDIO_ADDR                ((kal_uint32 *)&AudioResPreamble)
#define CONST_RES_AUDIO_TBL_SIZE            ((kal_uint32)0)
#define CONST_RES_AUDIO_CONTENT_SIZE        ((kal_uint32)0)

#endif /* MULTIPLE_BINARY_FILES */
```

Switch from Single-bin to Multi-bin (2/3)

@ mcu\custom\system\board version\ScatBoard.txt

Split out resources as independent load regions.

```
RESOURCE_ROM +0x0
{
  RESOURCE_ROM +0x0
  {
    /* Resource Preamble + Resource jump table */
    resource_resource_jtbl.obj (RESOURCE_JUMP_TBL, +First)
    resource_resource_jtbl.obj (+RW)
  }
  RESOURCE_ROM_CONTENT +0x00
  {
    /* Resource content */
    resource_RESOURCE_content.obj (+RO)
  }
}
```

- Split out system jump table as independent load region too.
  - Even though only one resource to be separated!

```
JUMP_TABLE 0x08dFFE00 0x200
{
  JUMP_TABLE +0x00
  {
    custom_jump_tbl.obj (+RO, +RW, +ZI)
  }
}
```

Switch from Single-bin to Multi-bin (3/3)

- *@ mcu\custom\system\CANNON_PLUS_BB\scatCANNON_PLUS.txt*
- *Dicision*
  - *DRIVER TAKE CARE OF BULTI-BIN SCATTER FILES???*

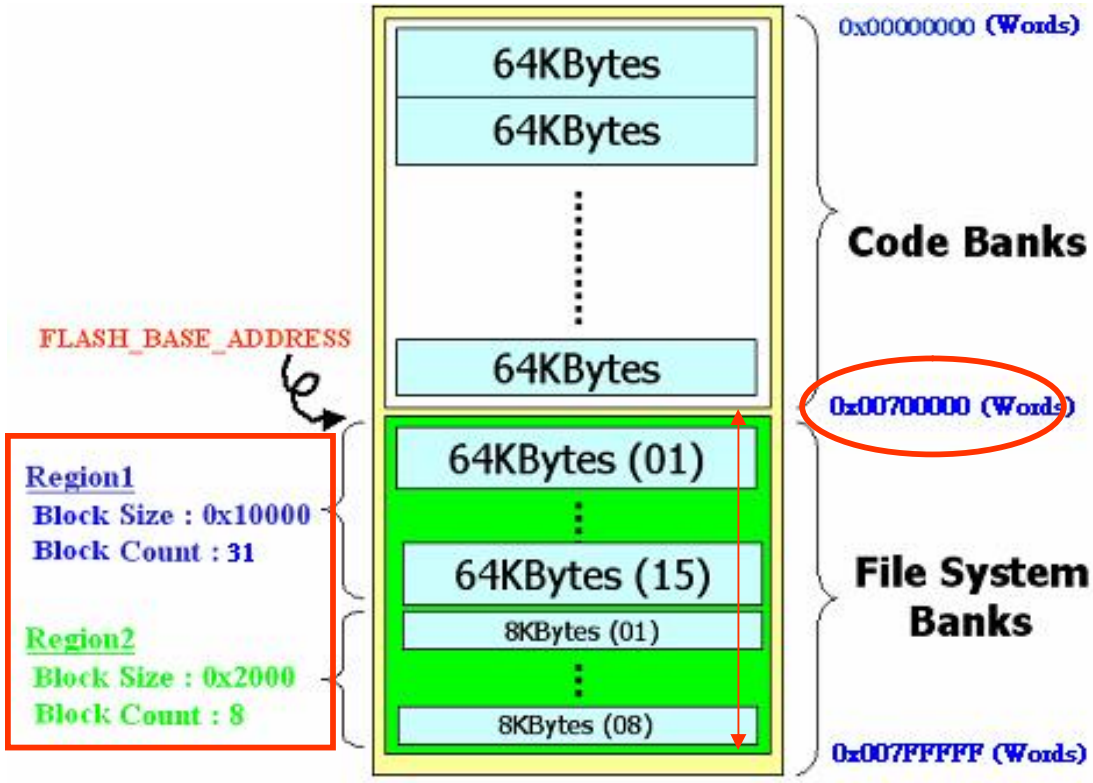
ASK Designer's help

```
ROM 0x08000000
{
   ROM 0x08000000 0x400000
   {…}
   ROM2 +0x0 0x400000
   {…}
   ROM3 +0x0
   {…}
   EXTSRAM 0x00000140
0x003FFEC0
   {…}
   INTSRAM_CODE 0x40000000
0x00014000
   {…}
   INTSRAM_DATA 0x40014000
0x0000c000
   {…}
}
```

```
AUDIO_ROM +0x00
{
  AUDIO_ROM +0x00
  {
    resource_audio_jtbl.obj (RESOURCE_JUMP_TBL, +First)
    resource_audio_jtbl.obj (+RW)
    resource_audio.obj (+RW)
  }
  AUDIO_ROM_CONTENT +0x00
  {
    resource_audio.obj (+RO)
  }
}
…
JUMP_TABLE 0x08dFFE00 0x200
{
  JUMP_TABLE +0x00
  {
    custom_jump_tbl.obj (+RO, +RW, +ZI)
  }
}
```

Flash Configuration

- NOR Flash
  - *@mcu\custom\flash\board version\FlashConf.c*
  - 5 step configuration



Step5.

Set PARTITION_SECTORS 0 (If no NOR partition)

Set PARTITION_SECTORS N (If NOR Pat.)
N=(FS BANK - 200K- NVRAM size)/512 byte

Step1.

FLASH_BASE_ADDRESS
0x00e00000 (Bytes)

Step3.

Set RegionInfo[]

Step4.

Set TOTAL_BLOCKS 39

Step2.

ALLOCATED_FAT_SPACE = 0x200000 (Bytes)

Flash Configuration

- NOR Flash
- *@mcu\custom\flash\board version\FlashConf.c*

```
/************
 * Step 1. *
 ************/
#define FLASH_BASE_ADDRESS 0x0E00000
/************
 * Step 2. *
 ************/
#define ALLOCATED_FAT_SPACE 0x200000 //0x200000 == 2Mbytes
/************/
/* Step 3. */
/************/
static const FlashRegionInfo RegionInfo[] =
{
  {0x10000, 31},
  {0x2000, 8}, // Spare one for SoC
  EndRegionInfo /* Don't modify this line */
};
/************/
/* Step 4. */
/************/
#define TOTAL_BLOCKS     39 //Maximum 127

/************/
/* Step 5. */
/************/
/* Partition Sectors in the First Partitions (Unit Sectors) one sector 512 Byte */
/* 0 for no partition */
#define PARTITION_SECTORS  1024
```

- ***How many User disk on NOR now?***

Flash Configuration

- NOR User Disk configuration
- *@mcu\custom\drv\msic_drv\board version\custom_drv_init.c*
- *Mount the NOR/Nand user disk*

```
void custom_drv_init(void)
{
  GPIO_init(); /* configure GPIO for debugging */
  spi_ini(); /* For LCD module */
  LCD_FunConfig();
  Alter_init();
#ifndef __L1_STANDALONE__
  PWM_initialize();
#endif
#ifdef __USB_ENABLE__
  #if ( (defined(__MSDC_MS__)) || (defined(__MSDC_SD_MMC__)) )
    USB_Ms_Register_DiskDriver(&USB_MSDC_drv);
  #endif
  #ifdef __USB_RAMDISK__
    USB_Ms_Register_DiskDriver(&USB_RAM_drv);
  #endif  /*__USB_RAMDISK__*/
  /* Hide the NOR Flash to the USER until NOR has partitions */
  /* By James Fu */
    USB_Ms_Register_DiskDriver(&USB_NOR_drv);
  #ifdef NAND_SUPPORT
    USB_Ms_Register_DiskDriver(&USB_NAND_drv);
  #endif
#endif  /*__USB_ENABLE__*/
} ? end custom_drv_init ?
```

Flash Configuration

- NAND Flash
  - CS1 Maintain, Mapping ID/Settings (timing,size…etc.)
  - If problems occurs, please contact driver partner for support.
  - ***@mcu\drv\src\NAND_MTD.c***

Audio Relative settings

- *@ mcu\custom\audio\board version\afe.c*
- ***Step1. For MT6218B set the proper GPIO Pin for Loud Speaker***
  - ***Function:***
  - ***AFE_TurnONExtAmplifier, AFE_TurnOffExtAmplifier, SwitchExtAmp***

```
/*************************************************************
* FUNCTION
*  AFE_TurnOnExtAmplifier
*  AFE_TurnOffExtAmplifier
* DESCRIPTION
*   These two function are to turn on/off external amplifier individually
*************************************************************/
void AFE_TurnOnExtAmplifier( void )
{
  afe.gpio_lock = KAL_TRUE;
  GPO_WriteIO( 1, 1 );
}

void AFE_TurnOffExtAmplifier( void )
{
  GPO_WriteIO( 0, 1 );
  afe.gpio_lock = KAL_FALSE;
}
```

```
void SwitchExtAmp( kal_uint8 device )
{
  if( !afe.gpio_lock ) {
    if( device & L1SP_BUFFER_EXT )
      GPO_WriteIO( 1, 1 );
    else
      GPO_WriteIO( 0, 1 );
  }
}
```

Audio Relative settings

- *@ mcu\custom\audio\board version\afe.c*
- ***Step1. For MT6205B set the proper GPIO Pin for Loud Speaker***
  - ***Function:***
  - ***UpdateVAPDN***

```
static void UpdateVAPDN( kal_uint8 device )
{
  ...
#if defined( YAMAHA_MA3 ) && defined( PCB01 )
  if( device & L1SP_BUFFER_EXT )
    YamahaMa3_SpkAmpOn();
  else
    YamahaMa3_SpkAmpOff();
#else
  if( device & L1SP_BUFFER_EXT )
    GPIO_WriteIO( 1, 1 );
  else
    GPIO_WriteIO( 0, 1 );
#endif

}
```

- ***Step2. For MT6205B set Differential mode/ Single ended for Speaker***
  - ***Function:***
  - ***AFEinit***

```
void AFE_Init( void )
{
...
  AFE_AC_CON0     = 0x4B;

  AFE_AC_CON1     = 0x80;
  AFE_APDN_CON    = 0;
  AFE_DAI_CON     = 0;
  AFE_LB_CON      = 0;
...
}
```

Definition of AFE_AC_CON0 and is at
Chapter 5.2 page 121/175 of 6205B datasheet
VDSEND0 for Handset & VDSEND1 for Headset
1: single-ended 0: differential mode

Audio Relative settings

- *@ mcu\custom\audio\board version\audcoeff.c*

```
const unsigned char L1SP_MICROPHONE1  = L1SP_LNA_0;
const unsigned char L1SP_MICROPHONE2  = L1SP_LNA_1;

const unsigned char L1SP_SPEAKER1     = L1SP_BUFFER_0;
const unsigned char L1SP_SPEAKER2     = L1SP_BUFFER_ST;
const unsigned char L1SP_LOUD_SPEAKER = L1SP_BUFFER_ST|L1SP_BUFFER_EXT;
const unsigned char L1SP_SPEAKER_ST   = L1SP_BUFFER_ST;

const signed short Audio_FIR_Input_Coeff[30] = {
  32767,    0,    0,    0,    0,
      0,    0,    0,    0,    0,
      0,    0,    0,    0,    0,
      0,    0,    0,    0,    0,
      0,    0,    0,    0,    0
};

const signed short Audio_FIR_Output_Coeff[30] = {
    573,  -368,   308,  -760,   305,
   -492,   540,   814, -4020,  4962,
  -2912,  -294, 15931, -32767, 24101,
  24101, -32767, 15931,  -294, -2912,
```

NO USE and absent in MT6205B

Step 1
Path decided by Layout
Normal mode Microphone path
Headset mode Microphone path
Normal mode Speaker path
Headset mode speaker path
Loud speaker path
Stereo output path

Step2
32767 by pass filter

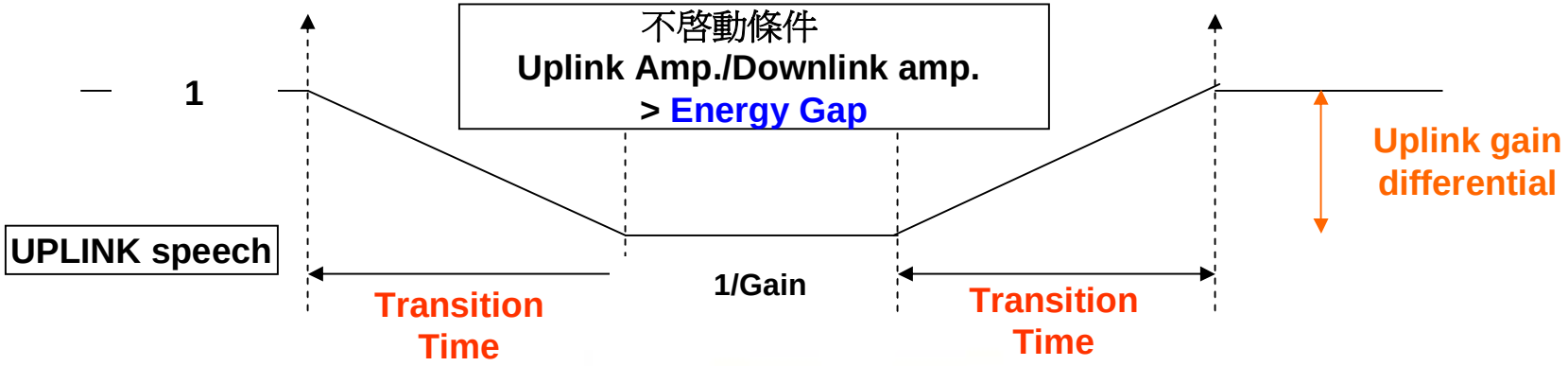
Step3
Please use matlab & meta make up other settings

Audio Relative settings

- *@ mcu\custom\audio\board version\audcoeff.c*

```
const unsigned short ES_TimeConst  = 0x0004;
const unsigned short ES_VolConst   = 0x0001;
const unsigned short ES_TimeConst2 = 32767; // 4098;
const unsigned short ES_VolConst2  = 32767; // 7000;
```

- ***Step4 Adjust Echo suppression settings***
  - ***ES_TimeConst = 0xYYZZ for Handfree mode Echo suppression***
    - ***0xYY*20 (ns) = transition time***
    - ***0xZZ*6 (db) = Energy gap***
  - ***ES_volConst = 0xAAAA for Handfree mode Echo supression***
    - ***uplink gain= uplink gain * 0xAAAA/32767***
  - ***ES_TimeConst2/ES_VolConst2 Normal mode vol 7 echo suppression settings***



Audio Relative settings

- *@ mcu\custom\audio\board version\audcoeff.c*
- ***Step5 Others (Should be same as cannon+)***
  - ***Adjustable @ mcu\custom\audio\board version\nvram_default_audio.c***

```
const unsigned short Media_Playback_Maximum_Swing  = 12800;

const unsigned short  DG_DL_Speech   = 0x1000;
const unsigned short  DG_Microphone  = 0x2000;
const unsigned short  DG_DAF       = 16384;
const unsigned short  DG_AAC       = 32767;
const unsigned short  DG_AMR       = 0x1000;
const unsigned short  DG_VM       = 0x1000;
const unsigned short  DG_MIDI      = 304;
const unsigned short  DG_Tone      = 3072;
const unsigned short  DG_WAV       = 16384;
```

Audio Relative settings

- *@ mcu\custom\audio\board version\nvram_default_audio.c*
- ***Setting Gains for different mode***
  - ***MICROPHONE_VOLUME***
  - ***SIDE_TONE_VOLUME***
  - ***GAIN_NOR_CTN_VOL0-7 (Normal mode Call tone)***
    - Call Tone is used for tone volume other than keypad tones, i.e. warning tone (plug-in charger), success tone ...etc.
  - ***GAIN_NOR_KEY_VOL0-7 (Normal mode Keypad tone)***
  - ***GAIN_NOR_MIC_VOL0-7 (Normal mode Microphone)***
  - ***GAIN_NOR_GMI_VOL0-7 (Normal mode GMI)***
    - GMI Tone is used for FM Radio volume now
  - ***GAIN_NOR_SPH_VOL0-7 (Normal mode Speech tone)***
  - ***GAIN_NOR_SID_VOL0-7 (Normal mode Side tone)***
  - ***GAIN_NOR_MED_VOL0-7 (Normal mode melody)***
  - ***Same as Headset mode and Hand-free more***
- *@ mcu\custom\audio\board version\nvram_default_audio.h*
  - ***Doesn't need change***

GPIO

- *@ mcu\custom\drv\misc_drv\board version\gpio_drv.c*
- Void GPIO_init(void)
  - DRV_WriteReg(addr, data)
    - GPIO_MODE1~6, GPO_MODE
    - Ex. DRV_WriteReg(GPIO_MODE2, 0x0601);
  - DRV_Reg(addr)
    - GPIO_DIR1~3
    - EX. DRV_Reg(GPIO_DIR1) |= 0x02e7; (1:output 0:input)

```
void GPIO_init(void)
{
  DRV_WriteReg(GPIO_MODE1,0x0000);
  DRV_WriteReg(GPIO_MODE2,0x0601);
  DRV_WriteReg(GPIO_MODE3,0x5600);
  DRV_WriteReg(GPIO_MODE4,0x1555);
  DRV_WriteReg(GPIO_MODE5,0x0141);
  DRV_WriteReg(GPIO_MODE6,0x5540);
  DRV_WriteReg(GPO_MODE, 0x0001);

  DRV_Reg(GPIO_DIR1) |= 0x02ff;
  DRV_Reg(GPIO_DIR2) |= 0x8003;
  DRV_Reg(GPIO_DIR3) |= 0x0704;
  #ifdef __USB_ENABLE__
    GPIO_WriteIO(0,16);
  #endif/*__USB_ENABLE__*/
  DRV_WriteReg(0x80000200,0x000c);

}
```

UEM configuration

- *@ mcu\custom\drv\misc_drv\board version\custom_equipment.c*
- GPIO used in Engineering Mode
  - Data structure

```
GPIO_MAP_ENTRY gpio_map_tbl[] = {
/*GPIO_LABEL_REN */     {GPIO_VAILD, GPIO_PORT_4, netname[0], NULL },
/*GPIO_LABEL_OPOFFB */   {GPIO_VAILD, 1, netname[1], NULL },
/*GPIO_LABEL_SUBBLEN */  {GPIO_INVAILD, 0, NULL, NULL },
/*GPIO_LABEL_LCDBL */    {GPIO_VAILD, GPIO_PORT_3, netname[3], NULL },
/*GPIO_LABEL_CHRCNTL */  {GPIO_VAILD, GPIO_PORT_18, netname[4], NULL },
/*GPIO_LABEL_GEN */     {GPIO_VAILD, GPIO_PORT_5, netname[5], NULL },
/*GPIO_LABEL_BEN */     {GPIO_VAILD, GPIO_PORT_6, netname[6], NULL },
/*GPIO_LABEL_VIBEN */    {GPIO_INVAILD, GPIO_PORT_15, netname[7], NULL },
/*GPO_LABEL_BEN */      {GPIO_INVAILD, 0, netname[8], NULL },
/*GPIO_LABELID_9 */     {GPIO_INVAILD, GPIO_PORT_9, netname[GPIO_PORT_9], NULL },
/*GPIO_LABELID_10 */     {GPIO_INVAILD, GPIO_PORT_10, netname[GPIO_PORT_10], NULL }
/*GPIO_LABELID_11 */     {GPIO_INVAILD, GPIO_PORT_11, netname[GPIO_PORT_11], NULL }
/*GPIO_LABELID_12 */     {GPIO_INVAILD, GPIO_PORT_12, netname[GPIO_PORT_12], NULL }
/*GPIO_LABELID_13 */     {GPIO_INVAILD, GPIO_PORT_13, netname[GPIO_PORT_13], NULL }
/*GPIO_LABELID_14 */     {GPIO_INVAILD, GPIO_PORT_14, netname[GPIO_PORT_14], NULL }
/*GPIO_LABELID_15 */     {GPIO_INVAILD, GPIO_PORT_15, netname[GPIO_PORT_15], NULL }
/*GPIO_LABELID_16 */     {GPIO_INVAILD, GPIO_PORT_16, netname[GPIO_PORT_16], NULL }
/*GPIO_LABELID_17 */     {GPIO_INVAILD, GPIO_PORT_17, netname[GPIO_PORT_17], NULL }
/*GPIO_LABELID_18 */     {GPIO_INVAILD, GPIO_PORT_18, netname[GPIO_PORT_18], NULL }
/*GPIO_LABELID_19 */     {GPIO_INVAILD, GPIO_PORT_19, netname[GPIO_PORT_19], NULL }
/*GPIO_LABELID_20 */     {GPIO_INVAILD, GPIO_PORT_20, netname[GPIO_PORT_20], NULL }
/*GPO_LABELID_0 */     {GPIO_INVAILD, 0, netname[21], NULL },
/*GPO_LABELID_1 */     {GPIO_INVAILD, 0, netname[22], NULL },
/*GPO_LABELID_2 */     {GPIO_INVAILD, 0, netname[23], NULL }
};
```

UEM configuration

– Netname

```
const unsigned char netname[][MAX_NETNAME_TEXT] = {
/*0, GPIO_LABEL_REN */     "GP4_REN",
/*1, GPIO_LABEL_OPOFFB */   "GPO1_OPOFFB",
/*2, GPIO_LABEL_SUBBLEN */  "GP2_SUBBLEN",
/*3, GPIO_LABEL_LCDBL */    "GP3_LCDBL",
/*4, GPIO_LABEL_CHRCNTL */  "GP18_CHRCNTL",
/*5, GPIO_LABEL_GEN */      "GP5_GEN",
/*6, GPIO_LABEL_BEN */      "GP6_BEN",
/*7, GPIO_LABEL_VIBEN */    "GP15_VIBEN",
/*8, GPO_LABEL_BEN */       "GPO3_BEN",
/*9,  GPIO_LABELID_9 */     "GPIO#9",
/*10,  GPIO_LABELID_10 */   "GPIO#10",
/*11,  GPIO_LABELID_11 */   "GPIO#11",
/*12,  GPIO_LABELID_12 */   "GPIO#12",
/*13,  GPIO_LABELID_13 */   "GPIO#13",
/*14,  GPIO_LABELID_14 */   "GPIO#14",
/*15,  GPIO_LABELID_15 */   "GPIO#15",
/*16,  GPIO_LABELID_16 */   "GPIO#16",
/*17,  GPIO_LABELID_17 */   "GPIO#17",
/*18,  GPIO_LABELID_18 */   "GPIO#18",
/*19,  GPIO_LABELID_19 */   "GPIO#19",
/*20,  GPIO_LABELID_20 */   "GPIO#20"
/*21,  GPO_LABELID_0 */    "GPO#0",
/*22,  GPO_LABELID_1 */    "GPO#1",
/*23,  GPO_LABELID_2 */    "GPO#2"
};
```

UEM configuration

- *@ mcu\custom\drv\misc_drv\board version\custom_equipment.h*
- GPIO_LABELID_ENUM in the header file

```
typedef enum {
  GPIO_LABEL_REN = 0x0,
  GPIO_LABEL_OPOFFB,
  GPIO_LABEL_SUBBLEN,
  GPIO_LABEL_LCDBL,
  GPIO_LABEL_CHRCNTL,
  GPIO_LABEL_GEN,
  GPIO_LABEL_BEN,
  GPIO_LABEL_VIBEN,
  GPO_LABEL_BEN,
  GPIO_LABELID_9,
  GPIO_LABELID_11,
  GPIO_LABELID_10,
  GPIO_LABELID_12,
  GPIO_LABELID_13,
  GPIO_LABELID_14,
  GPIO_LABELID_15,
  GPIO_LABELID_16,
  GPIO_LABELID_17,
  GPIO_LABELID_18,
  GPIO_LABELID_19,
  GPIO_LABELID_20,
  GPO_LABELID_0,
  GPO_LABELID_1,
  GPO_LABELID_2,

  /* shound NOT modify this value*/
  GPIO_LABELID_MAX
} ? end GPIO_LABELID_ENUM ? GPIO_LABELID_ENUM;
```

UEM configuration

- *@ mcu\custom\drv\misc_drv\board version\custom_equipment.c*
- EINT Mapping Table

```
const unsigned char eintname[][MAX_NETNAME_TEXT] = {
  "Charger",
  "Cable",
  "SendKey",
  "Earphone",
  "ClamShell"
};

GPIO_MAP_ENTRY eint_map_tbl[] = {
/*EINT_LABELID_0 */     {GPIO_VAILD, 0, eintname[0], NULL },
/*EINT_LABELID_1 */     {GPIO_VAILD, 1, eintname[1], NULL },
/*EINT_LABELID_2 */     {GPIO_VAILD, 1, eintname[2], NULL },
/*EINT_LABELID_3 */     {GPIO_VAILD, 1, eintname[3], NULL },
/*EINT_LABELID_4 */     {GPIO_INVAILD, 2, eintname[4], NULL },
};
```

- ADC Mapping Table

```
const unsigned char adcname[][MAX_NETNAME_TEXT] = {
  "VBAT",
  "BTemp",
  "VAUX",
  "Current",
  "VChgr"
};

GPIO_MAP_ENTRY adc_map_tbl[] = {
/*ADC_LABELID_0 */     {GPIO_VAILD, 0, adcname[0], NULL },
/*ADC_LABELID_1 */     {GPIO_VAILD, 1, adcname[1], NULL },
/*ADC_LABELID_2 */     {GPIO_VAILD, 2, adcname[2], NULL },
/*ADC_LABELID_3 */     {GPIO_VAILD, 3, adcname[3], NULL },
/*ADC_LABELID_4 */     {GPIO_VAILD, 4, adcname[4], NULL },
};
```

HW Default Setting

- *@ mcu\custom\drv\misc_drv\board version\custom_hw_default.c*
- Set PWM1, PWM2, Alter Level.
  - LCD backlight, keypad backlight, flashlight.
  - Data structure

```
kal_uint32 PWM1_Level_Info[PWM_MAX_LEVEL][2] =
{
  /*Freq,duty*/
  {200,20},   /*Level 1*/
  {200,40},   /*Level 2*/
  {200,60},   /*Level 3*/
  {200,80},   /*Level 4*/
  {200,100}   /*Level 5*/
};

kal_uint32 const NVRAM_EF_CUST_HW_LEVEL_TBL_DEFAULT[] = {
              64, 20, /* PWM 1  Level 1 */
              64, 40, /* PWM 1  Level 2*/
              64, 60, /* PWM 1  Level 3*/
              64, 80, /* PWM 1  Level 4*/
              64, 100,/* PWM 1 Level 5 */
              100, 10,  /* PWM 2  Level 1 */
              100, 30,  /* PWM 2  Level 2*/
              100, 50,  /* PWM 2  Level 3*/
              100, 70,  /* PWM 2  Level 4*/
              100, 100,/* PWM 2 Level 5 */
              64, 20,  /* PWM 3  Level 1 */
              64, 40,  /* PWM 3  Level 2*/
              64, 60,  /* PWM 3  Level 3*/
              64, 80,  /* PWM 3  Level 4*/
              64, 100,/* PWM 3 Level 5 */
```

HW Default Setting

- LCD default contrast setting

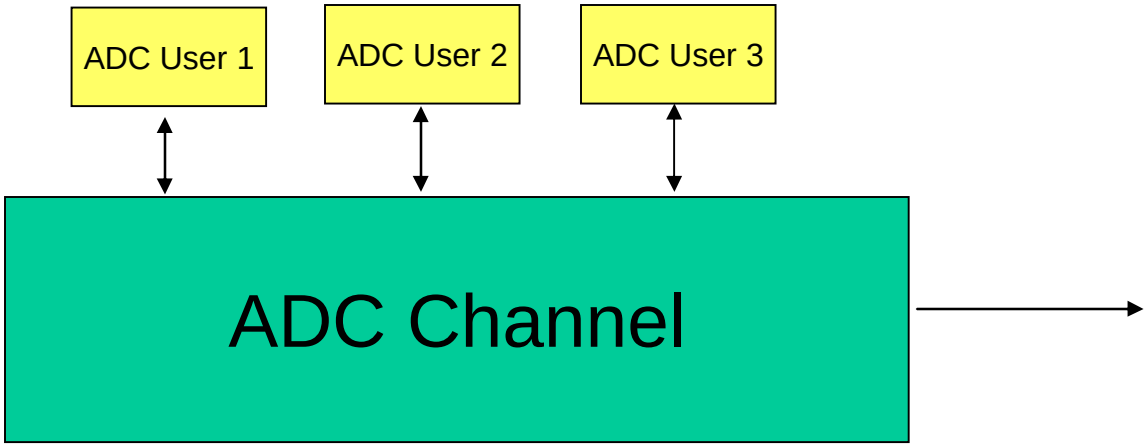
```
/* Main LCD contrast level 1 ~ 15 */
1, 2, 3, 4, 5, 6, 7,
8, 9, 10, 11, 12, 13, 14, 15,
/* Main LCD Bias Param (Reserved) */
0, 0, 0, 0, 0,
/* Main LCD Linerate Param (Reserved) */
0, 0, 0, 0, 0,
/* Main LCD Temperature Param (Reserved) */
0, 0, 0, 0, 0,
/* Sub LCD contrast level 1 ~ 15 */
17, 18, 19, 20, 21, 22, 23,
24, 25, 26, 27, 28, 29, 30, 31,
/* Sub LCD Bias Param (Reserved) */
0, 0, 0, 0, 0,
/* Sub LCD Linerate Param (Reserved) */
0, 0, 0, 0, 0,
/* Sub LCD Temperature Param (Reserved) */
0, 0, 0, 0, 0,
```

- Battery Level setting

```
/* Battery voltage Level */
3300000, /* Low Battery Power off */
3400000, /* MO Limit */
3500000, /* Low Battery */
/* battery level 1 ~ 8 */
3600000, 3740000, 3880000, 9999999,
9999999, 9999999, 9999999
```

ADC scheduler

- A channel used by Auxiliary Device.
  - Battery monitoring
  - Charger control
  - peripheral device detection
  - temperature measurement .....
- One physical channel have many logical owner.



ADC scheduler

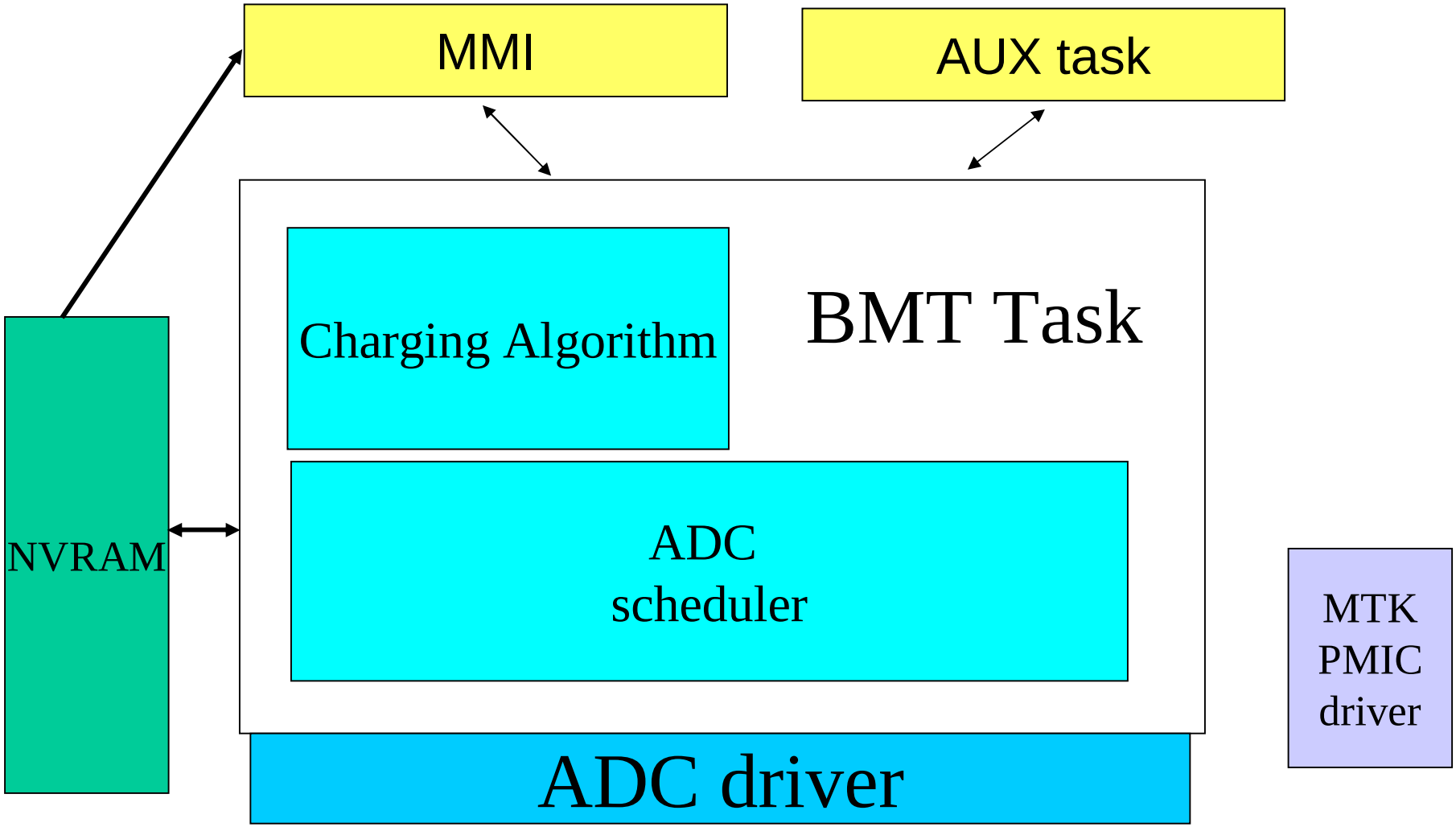
Customizations

@ mcu\custom\drv\misc_drv\board version\adc_channel.c

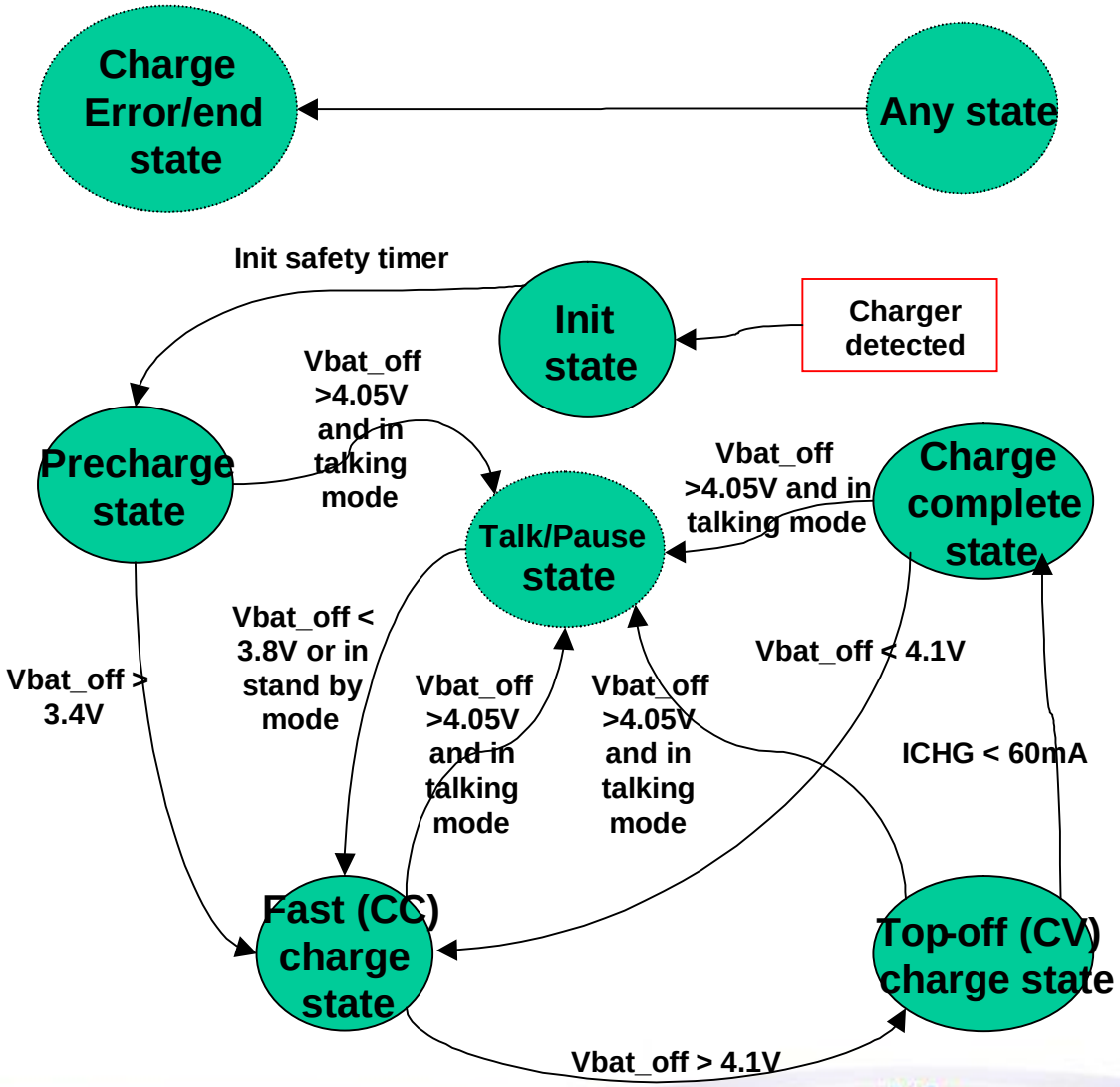
- 5 ADC channels in MT6205B, and 7 in MT6218B
- Customized ADC channel definitions
  e.g. ADC_VBAT = 0; ADC_VBATTMP = 2; ADC_ACCESSORYID=3; ADC_VISENSE = 1; ADC_VCHARGER = 4;

```
const kal_uint8 ADC_VBAT = 0;
const kal_uint8 ADC_VBATTMP = 2;
const kal_uint8 ADC_ACCESSORYID = 3;
const kal_uint8 ADC_VISENSE = 1;
const kal_uint8 ADC_VCHARGER = 4;
```

BMT



BMT



ADC check for abnormal condition

a. Vbat > 4.5V Battery over voltage

b. Vchg> 6.5 orVchg< 4.5 Invalid charger

c. BATEMP < 0oC or BATEMP > 45oC Battery temp. too cold or over heat

d. BATID N.A.

e. ICHGon> 750mA orICHGon< 20mA Invalid charger or battery

f. ICHGoff> 50mA Charger fail

g. Check ADC fail,

h Safety timer timeout (6hr)

i. Charger plugged out

j. Bad contact of Charger

All boundary values above can be customized

P.S. XXX_on : ADC value is measured during pulse on charging and RF off state.

xxx_off : ADC value is measured during pulse off charging and RF off state.

xxx: ADC value is measured during pulse on and pulse off charging and RF off state.

BMT

ADC scheduler

- Stated above

Charging Algorithm

@ mcu\custom\drv\misc_drv\board version\chr_parameter.c

- This file is defined for charging parameters.

@ mcu\custom\drv\misc_drv\board version\batvalue.h

- No need change

Charging Parameter

- *@ mcu\custom\drv\misc_drv\board version\chr_parameter.c*
- Setting charging parameter **à** bmt_custom_chr_def
  - Step 1. GPIO configuration for charging control , battery detection and vibrator.

```
bmt_customized_struct bmt_custom_chr_def =
{
  18,/*GPIO_CHRCTRL*/
  14,/*GPIO_BATDET*/
  15,/*GPIO_VIBRATOR*/
```

  - Step 2.Charging parameters

```
/*charing parameters*/
/*Check Phy parameters,Maybe changed*/
  1100000,/*Typical_LI_BATTYPE*/
  1100000,/*Typical_NI_BATTYPE*/
  1000000,/*ICHARGE_ON_HIGH*/
  20000,/*ICHARGE_ON_LOW*/
  1000000,/*ICHARGE_OFF_HIGH*/ //?250000
  4050000,/*V_FAST2TOPOFF_THRES*/
  2500000,/*BATTMP_MINUS_40C*/
  1469409,/*BATTMP_0C*/
  414568,/*BATTMP_45C*/
  4500000,/*MAX_VBAT_LI*/
  5500000,/*MAX_VBAT_NI*/
  3400000,/*V_PRE2FAST_THRES*/
  250000,/*I_TOPOFF2FAST_THRES*/  /*250ma,TOPOFF->FAST*/
  120000,/*I_TOPOFF2FULL_THRES*/  /*120ma,TOPOFF->BATFULL*/
  4110000,/*V_FULL2FAST_THRES*/  /*BATFULL->FAST*/
  414557,/*V_TEMP_FAST2FULL_THRES_NI*/  /*50oC,FAST->BATFULL*/
  4050000,/*V_FULL2FAST_THRES_NI*/
  600000,/*FAST_ICHARGE_HIGHLEVEL*/  /*600ma,for table search*/
  400000,/*FAST_ICHARGE_LOWLEVEL*/  /*400ma,for table search*/
  4050000,/*V_PROTECT_HIGH_LI*/
  3800000,/*V_PROTECT_LOW_LI*/
  6500000,/*VCHARGER_HIGH*/
  0,/*VCHARGER_LOW*/
```

Voltage of ADC2 at Temp -40C = 2.8*(188.5/188.5+24) =2.48376V
Voltage of ADC2 at Temp 0C = 2.8*(27.278/27.278+24) =1.489496V
Voltage of ADC2 at Temp 45C = 2.8*(4.913/4.913+24) =0.475785V

Charging Parameter

- *@ mcu\custom\drv\misc_drv\board version\chr_parameter.c*
- Step 3. Calibration (6218B *7, 6205B *5 adc channels)

```
//ADC_CALIDATA adc_cali_param
{
  {
    5524,
    5524,
    5524,
    5524,
    5524,
    5524,
    5524
  },
  {
    (23286),
    (23286),
    (23286),
    (23286),
    (23286),
    (23286),
    (23286)
  }
};
```

```
/*ratio = adc_volt_factor/100*/
//const kal_uint16 adc_volt_factor[ADC_MAX_CHANNEL]
{
  100,
  100,
   50,
   50,
  250,
  100,/*NC*/
  100 /*NC*/
},

//const kal_uint8 TONOFFTABLE[6][2] =
{
  {7,1},
  {8,1},
  {9,1},
  {7,1},/*talk */
  {8,1},/*talk */
  {9,1}/*talk */
},
```

ADC channel 1: Vcc=1:1

First 3 is CC mode duty
Last 3 is talking mode dutys
X>600 ma
400ma<X<600ma
X<400ma

- Step 4. Define resistor value of diff. Temp.
- **#define NTCR1 27278.0 /*resistor at 0C*/**
- **#define NTCR2 10000.0 /*resistor at 25C*/**
- **#define NTCR3 4913.0 /*resistor at 45C*/**

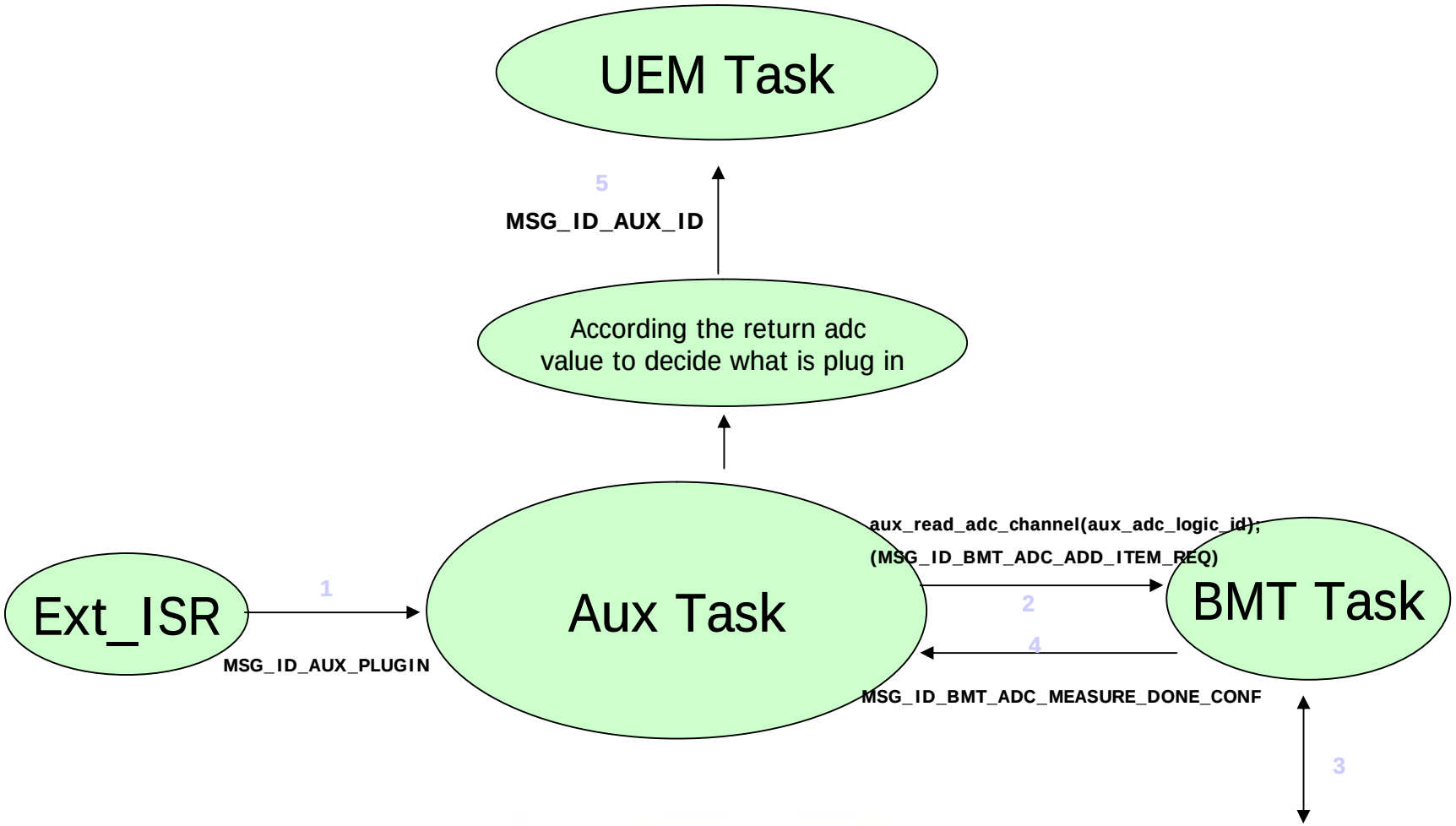
EINT

- *@ mcu\custom\drv\misc_drv\board version\eint_def.c*
- External Interrupt channel assignment
  - Channel assignment.
  - De-bounce time delay setting.
  - Example.

```
const kal_uint8 AUX_EINT_NO = 0;
const kal_uint8 CHRDET_EINT_NO = 2;
const kal_uint8 CLAMDET_EINT_NO=1;
const kal_uint8 USB_EINT_NO = 3;
/*Unit: 10ms*/
kal_uint8 custom_eint_sw_debounce_time_delay[EINT_MAX_CHANNEL] =
{
  50,  /*EINT 0,500ms*/
  50,  /*EINT 1,500ms*/
  50,  /*EINT 2,500ms*/
  50   /*EINT 3,500ms*/
};
```

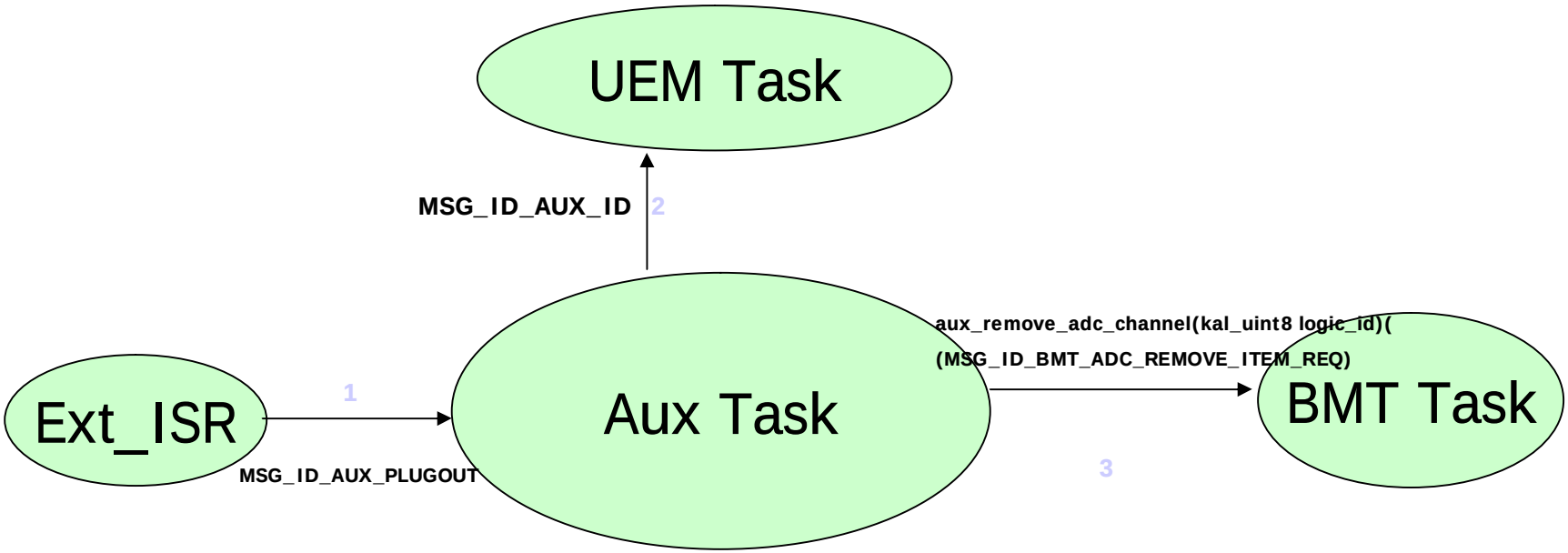
Aux Task

Plug-in Detection



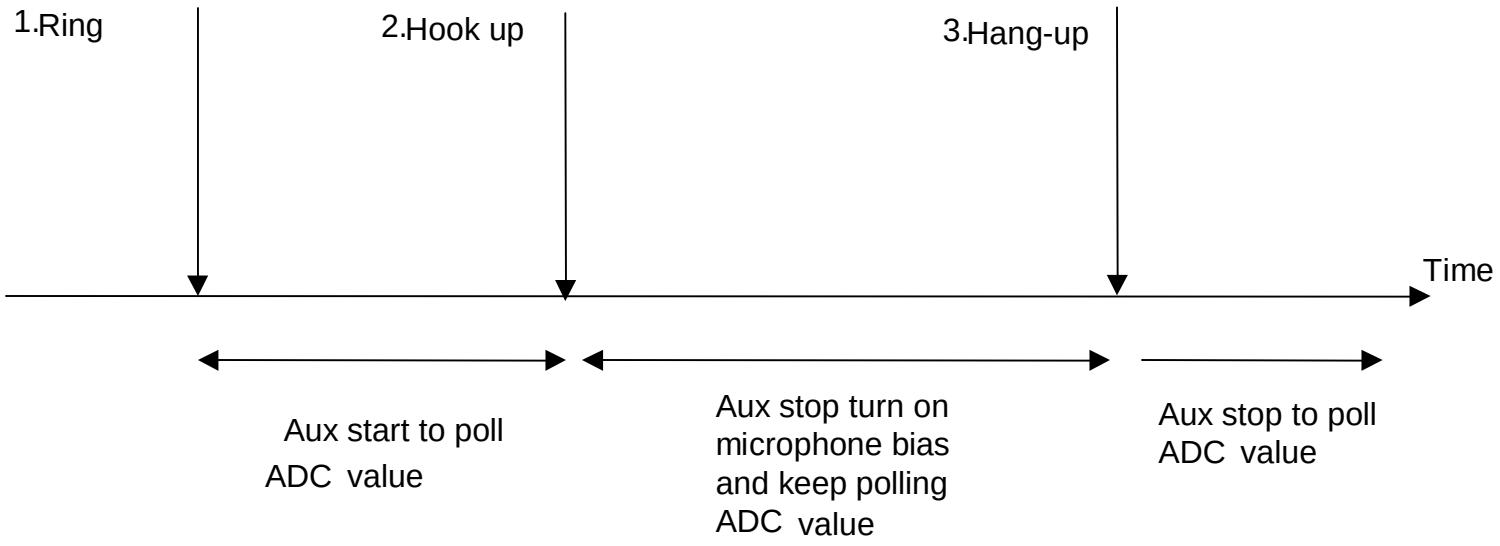
Aux Task

Plug- out Detection



Aux Task

Send Key Detection



Aux Task

Customization

- *@ mcu\custom\drv\misc_drv\board version\aux_main.c*

AUX Task is to

- § detect earphone, uart plug-in and plug out
- § detect send/end key press
- § Support ADC report in Engineer mode

```
#define UART_ADC              9900000
#define NORMAL_EARPHONE_ADC_HIGH  3000000/*Normal mode(2.2v)*/
#define NORMAL_EARPHONE_ADC_LOW   500000
#define SINGLE_EARPHONE_ADC_HIGH  3000000/*Single-ended mode(1.9v)*/
#define SINGLE_EARPHONE_ADC_LOW   500000
#define SENDKEY_ADC           300000

#define POLLING_INTERVAL      50 /*polling interval*/
#define TURN_ON_BIAS_INTERVAL  10 /*measure adc interval*/
#define PLUGIN_DEBOUNCE_TIME   100 /*unit=10ms*/
#define PLUGOUT_DEBOUNCE_TIME  0
#define AUX_UART_PORT         uart_port1
#define AUX_UART_DSR_PIN      0xff
#define AUX_UART_GPIO         0xff
#define POLLING_TIMER_INDEX    1
#define ADC_TIMER_INDEX       0
```

9.9 v à UART
0.5 ~ 3 v à Earphone
Sendkey à 0.3v

This driver detect uart_1 only

Separate Earphone/Uart by GPIO? (TGH)

Aux Task

Customization

- *@ mcu\custom\drv\misc_drv\board version\aux_main.c*

AUX Task is to

- Detect Clam on/off (Clam shell project only)
- Step 1 Set the clam_state as the open level
  - **kal_bool clam_state = LEVEL_HIGH;**
- Step 2 Set the proper Clam Close Level
  - In function **void CLAM_EINT_HISR(void)**

```
void CLAM_EINT_HISR(void)
{
  ilm_struct        *clam_ilm;
  aux_id_struct      *aux_id_data;

   if (clam_state == LEVEL_LOW)
   {
     aux_id_data = (aux_id_struct*)
                   construct_local_para(sizeof(aux_id_struct), TD_CTRL);

     aux_id_data->aux_id = AUX_ID_CLAM_CLOSE;

     DRV_BuildPrimitive(clam_ilm,
                MOD_EINT_HISR,
                MOD_UEM,
                MSG_ID_AUX_ID,
                aux_id_data);
   }
   else
   {
     aux_id_data = (aux_id_struct*)
                   construct_local_para(sizeof(aux_id_struct), TD_CTRL);

     aux_id_data->aux_id = AUX_ID_CLAM_OPEN;
```

Set CLAM_CLOSE level

Here is CLAM_CLOSE command

Here is CLAM_OPEN command

UART

UART

- *@ mcu\custom\drv\misc_drv\board version\uart_def.c*
- Set UART Flow Control
  - fc_none: no flow control
  - fc_sw: software flow control
  - fc_hw: hardware flow control

```
UART_flowCtrlMode UART_GetFlowCtrl(UART_PORT uart_port)
{
 UART_flowCtrlMode flow_ctrl;

  switch(uart_port)
  {
    case uart_port1:
      flow_ctrl=fc_hw;
      break;
    case uart_port2:
      flow_ctrl=fc_none;
      break;
#ifdef __UART3_SUPPORT__
    case uart_port3:
      flow_ctrl=fc_none;
      break;
#endif
```

USB

- *@ mcu\custom\drv\misc_drv\board version\usb_custom.c*
- Step 1. Device description parameter
  - USB_MANUFACTURER_STRING[]
  - USB_PRODUCT_STRING[]

```
static const kal_uint16 USB_MANUFACTURER_STRING[] =
{
    0x031a,
    'M',
    'e',
    'd',
    'i',
    'a',
    'T',
    'e',
    'k',
    ' ',
    'I',
    'n',
    'c'
};

static const kal_uint16 USB_PRODUCT_STRING[] =
{
    0x0310,
    'F',
    'I',
    'R',
    'E',
    'F',
    'L',
    'Y'
};
```

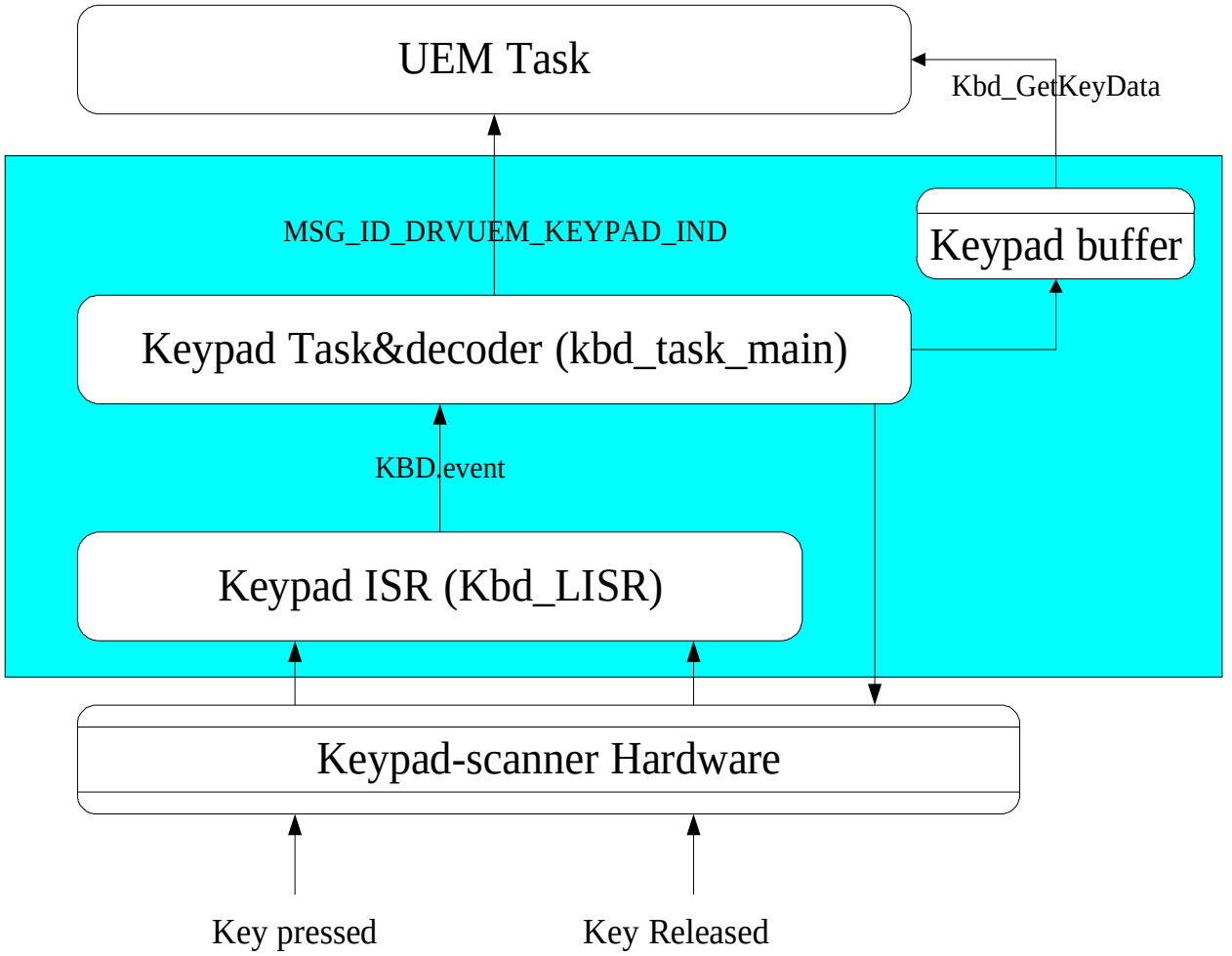
Length of(“Mediatek Inc”)=12
(12+1)*2 = 26 = 0x1a
Null string **è** +1 Unicode **è** *2
03勿動

USB

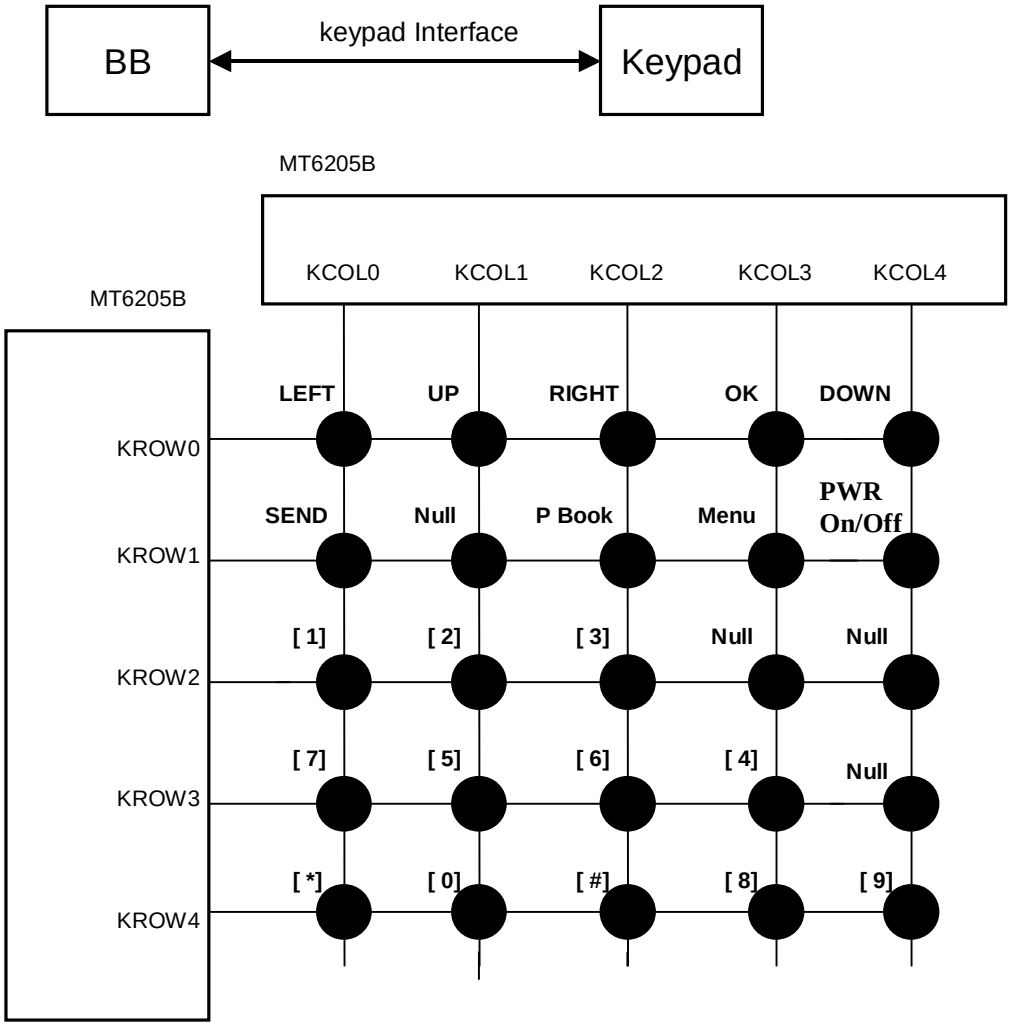
- Step 2. GPIO as power control pin
  - Ex. GPIO16 should be reserved in gpio_drv.c

```
void USB_PowerControl(kal_bool enable)
{
    #ifdef __USB_ENABLE__
        if (enable == KAL_TRUE)
            GPIO_WriteIO(1,16);
        else
            GPIO_WriteIO(0,16);
    #endif  /*__USB_ENABLE__*/
}
```

Keypad



Keypad



Keypad

- *@ mcu\custom\drv\misc_drv\board version\keypad_def.c*
- Keypad mapping definition

```
#define Custom_Keypress_Period 1000
kal_uint8 powerkey_position=DEVICE_KEY_END;
const keypad_struct  keypad_custom_def = {
    /*keypad mapping*/
    {
     /* row 0 */
       DEVICE_KEY_SEND,/*send*/
       DEVICE_KEY_VOL_UP,/*volume up*/
       DEVICE_KEY_FUNCTION,
       DEVICE_KEY_VOL_DOWN,/*volume down*/
       DEVICE_KEY_NONE,
       DEVICE_KEY_NONE,
       DEVICE_KEY_END,
```

```
     /* row 5 */
       DEVICE_KEY_SK_LEFT,/*OK*/
       DEVICE_KEY_STAR,/***/
       DEVICE_KEY_0,/*0*/
       DEVICE_KEY_HASH,/*#*/
       DEVICE_KEY_NONE,
       DEVICE_KEY_NONE,
       DEVICE_KEY_END
    },
    /*power on period*/
    Custom_Keypress_Period,
    /*powerkey position*/
    DEVICE_KEY_END
};
```

Mass Storage Disk Mount

- *@ mcu\custom\drv\misc_drv\board version\custom_drv_init.c*

```
void custom_drv_init(void)
{
  GPIO_init(); /* configure GPIO for debugging */
  spi_ini(); /* For LCD module */
  LCD_FunConfig();
  Alter_init();
#ifndef __L1_STANDALONE__
  PWM_initialize();
#endif
#ifdef __USB_ENABLE__
  #if ( (defined(__MSDC_MS__)) || (defined(__MSDC_SD_MMC__)) )
    USB_Ms_Register_DiskDriver(&USB_MSDC_drv);
  #endif
  #ifdef __USB_RAMDISK__
    USB_Ms_Register_DiskDriver(&USB_RAM_drv);
  #endif  /*__USB_RAMDISK__*/
  /* Hide the NOR Flash to the USER until NOR has partitions */
  /* By James Fu */
  //USB_Ms_Register_DiskDriver(&USB_NOR_drv);
  #ifdef NAND_SUPPORT
    USB_Ms_Register_DiskDriver(&USB_NAND_drv);
  #endif
#endif  /*__USB_ENABLE__*/
} ? end custom_drv_init ?
```

Mount MSDC

Mount NOR

Mount NAND

PWM Control

- *@ mcu\custom\drv\misc_drv\board version\PWMdrv.c*
- ***Setting PWM1 and PWM2 ClockSource***
  - ***PWM_Init(pwmclk_32k,pwmclk_8MHZ); /*clok = 32k/8*/***
  - ***PWM2_Init(pwmclk_32k,pwmclk_8MHZ); /*clok = 32k/8*/***
  - ***pwmclk_13M/pwmclk_32k***
  - ***pwmclk_1MHZ/case pwmclk_2MHZ/case pwmclk_4MHZ/case pwmclk_8MHZ***

```
kal_uint32 PWM1_Level_Info[PWM_MAX_LEVEL][2] =
{
  /*Freq,duty*/
  {200,20},   /*Level 1*/
  {200,40},   /*Level 2*/
  {200,60},   /*Level 3*/
  {200,80},   /*Level 4*/
  {200,100}   /*Level 5*/
};
```

Frequency Range

Driver will calculate proper Count

$$freq = \frac{ClockSource}{Count+1}$$

$$\Rightarrow freq_{Min} \approx \frac{ClockSource}{Count_{Max}} \approx \frac{ClockSource}{2^{13}}$$

$$\Rightarrow freq_{Min} \le freq \le ClockSource$$

$$Ex: ClockSource = 13000000$$

$$freq_{Min} = \frac{13000000}{8192} \approx 1586\,(Hz)$$

$$1586 \le freq \le 13000000$$

DutyCycle Resolution

$$DutyCycle_{step} = \frac{1}{Count+1} * 100$$

META

- *@ mcu\custom\meta\board version\ft_customize.c*

```
63 #include "ft_customize.h"
64 #include "gpio_sw.h"
65
66 #if !defined(BULL888_BB)
67     #error "custom\\meta\\FIREFLY_[illegible] mismatch with compile option!"
68 #endif
69
70 kal_uint8 FT_CustomInit() {
71         return 0;
72 }
73
```

Compile option

- ***What should be deleted what should not?***
- ***void GPIO_init(void)***
- ***{***
- ***#ifdef MT6205B***
- ***#if defined(KLM2003_BB)***
- ***...***
- ***#elif defined(CHICAGO2003_BB)***
- ***...***
- ***#else /*Default*/***
- ***DRV_WriteReg(GPIO_PULLEN,0x000c);***
- ***DRV_WriteReg(GPIO_PULLEN2,0x000a);***
- ***#ifdef ORDNANCE***
- ***DRV_WriteReg(GPIO_MODE1,0x9554);***
- ***#else /*!ORDNANCE*/***
- ***DRV_WriteReg(GPIO_MODE1,0x9555);***
- ***#endif /*ORDNANCE*/***
- ***DRV_WriteReg(GPIO_MODE2,0x0155);***
- ***DRV_WriteReg(GPIO_MODE3,0x0155);***
- ***DRV_WriteReg(GPIO_MODE4,0x0052);***
- ***DRV_Reg(GPIO_DIR) |= 0xe000;***
- ***DRV_WriteReg(GPIO_DIR2,0x0020);***
- ***DRV_WriteReg(0x80000200,0x000c);***
- ***#endif***
- ***#endif /*MT6205B*/***
- ***#if ((defined(MT6208)) || (defined(FPGA)))***
- ***...***
- ***#endif /*(MT6208,FPGA)*/***
- ***}***

```
void GPIO_init(void)
{
  /*ESR3*/
    DRV_WriteReg(GPIO_PULLEN,0x000c);
    DRV_WriteReg(GPIO_PULLEN2,0x000a);
    /*GPIO mode */
    DRV_WriteReg(GPIO_MODE1,0x4500);
    DRV_WriteReg(GPIO_MODE2,0x0055);
    DRV_WriteReg(GPIO_MODE3,0x0155);
    DRV_WriteReg(GPIO_MODE4,0x001a);
    /*GPIO direction*/
      DRV_Reg(GPIO_DIR) |= 0xf04f;
    DRV_WriteReg(GPIO_DIR2,0x0000);
    GPIO_WriteIO(0, 15);
    DRV_WriteReg(0x80000200,0x000c);
}
```

Build Command

- **Clean**
  - Clean all modules **à** Make custom=PROJECTNAME gprs **clean**
  - Clean module **à** Make custom =PROJECTNAME gprs **clean module_name**
- **New**
  - Make custom =PROJECTNAME gprs **new**
- **Update**
  - Update all modules **à** make custom= =PROJECTNAME gprs **update**
  - Update module **à** make custom= =PROJECTNAME gprs **update module_name**
- **Remake**
  - Remake all modules **à** make custom= =PROJECTNAME gprs **remake**
  - Remake module **à** make custom= =PROJECTNAME gprs **remake module_name**

Q&A !!