YMU762C Installation Guide

MA-3 Sound Middle Ware is for YMU762, however changing of files allow you to use it for YMU762C.
We itemize these portions of the change and some cautions for it as follows.

(1) Change of Wait time in Write-Read operation.

Difference:

YMU762 : needs 450ns at least
YMU762C : needs 600ns at least

Affected file: mamachdep.c
Way to change:

Change machdep_ Wait (450) to machdep_Wait(600) in machdep_WriteStatusFlagReg and machdep_WriteDataReg functions.

* However, to keep the wait time corresponds to the value in the parentheses is depending upon its Installation.

(2) Using in TCXO mode, you need to set CKSEL bit to"1" in the Intermediate register Bank=1,ID=5.

Affected file: mamachdep.h
Way to change:

In the following definition, set a numerical number where bit7 is "1".

```
#define    MA_ADJUST1_VALUE    (???)      /* register bank 1, ID #5 */
```

(3) Using in TCXO mode, there are differences in hardware initialization and power down sequence.

Affected file: madevdrv.c
Way to change:

In hardware initialization sequence (case=0 and case=1) and power down release sequence (case=3) in the following MaDevDrv_PowerManagement function,

You need to add 2.0ms wait time before the each settings.

1. /* set PLLPD bit of power management (A) setting register to '0' */ …(case=0)
2. /* set PLLPD and AP0 bits of REG_ID #6 power management (A) setting register to '0' */…(case=1)
3. /* set AP0 and PLLPD bits of REG_ID #6 power management (A) setting register to '0' */…(case=3)

* However, clock input to CLKI terminal needs to be stable sufficiently before the wait time.

```
/*****************************************************************************
 *      MaDevDrv_PowerManagement
 *
 *      Description:
 *                      Power management.
 *      Argument:
 *                      mode    0: Hardware initialize sequence (power down)
 *                                      1: Hardware initialize sequence (normal)
 *                                      2: Power down change sequence
 *                                      3: Power down release sequence
 *      Return:
 *                      0               success
 *                      < 0             error code
 *
 *****************************************************************************/
SINT32 MaDevDrv_PowerManagement
(
        UINT8   mode                                            /* operation mode */
)
{
        UINT8   count;                                          /* loop counter */
        SINT32  result = MASMW_SUCCESS;         /* result of function */

        MADEVDRV_DBGMSG(("MaDevDrv_PowerManagement: mode=%d¥n", mode));

        switch ( mode )
        {
        case 0:

                /* sequence of hardware initialize when power downed */

                /* set BANK bits of REG_ID #4 basic setting register to '0' */
                machdep_WriteStatusFlagReg( MA_BASIC_SETTING_REG );
                machdep_WriteDataReg( 0x00 );

                /* set DP0 bit of REG_ID #5 power management (D) setting register to '0' */
                machdep_WriteStatusFlagReg( MA_POWER_MANAGEMENT_DIGITAL_REG );
                machdep_WriteDataReg( MA_DP3 | MA_DP2 | MA_DP1 );


                machdep_Wait( 2 * 1000 * 1000 );


                /* set PLLPD bit of power management (A) setting register to '0' */
                machdep_WriteStatusFlagReg( MA_POWER_MANAGEMENT_ANALOG_REG );
                machdep_WriteDataReg( MA_AP4R | MA_AP4L | MA_AP3 | MA_AP2 | MA_AP1 |
              MA_AP0 );


                        (Omitted)

        case 1:

                /* sequence of hardware initialize when normal */

                /* set BANK bits of REG_ID #4 basic setting register to '0' */
```

```
        machdep_WriteStatusFlagReg( MA_BASIC_SETTING_REG );
        machdep_WriteDataReg( 0x00 );


        /* set DP0 bit of REG_ID #5 power management (D) setting register to '0' */
        machdep_WriteStatusFlagReg( MA_POWER_MANAGEMENT_DIGITAL_REG );
        machdep_WriteDataReg( MA_DP3 | MA_DP2 | MA_DP1 );


        machdep_Wait( 2 * 1000 * 1000 );


        /* set PLLPD and AP0 bits of REG_ID #6 power management (A) setting register to '0' */
        machdep_WriteStatusFlagReg( MA_POWER_MANAGEMENT_ANALOG_REG );
        machdep_WriteDataReg( MA_AP4R | MA_AP4L | MA_AP3 | MA_AP2 | MA_AP1 );


                (Omitted)


case 3:


        /* set BANK bits of REG_ID #4 basic setting register to '0' */
        machdep_WriteStatusFlagReg( MA_BASIC_SETTING_REG );
        machdep_WriteDataReg( 0x00 );

        /* set DP0 bit of REG_ID #5 power management (D) setting register to '0' */
        machdep_WriteStatusFlagReg( MA_POWER_MANAGEMENT_DIGITAL_REG );
        machdep_WriteDataReg( MA_DP3 | MA_DP2 | MA_DP1 );


        machdep_Wait( 2 * 1000 * 1000 );


        /* set AP0 and PLLPD bits of REG_ID #6 power management (A) setting register to '0' */
        machdep_WriteStatusFlagReg( MA_POWER_MANAGEMENT_ANALOG_REG );
        machdep_WriteDataReg( MA_AP4R | MA_AP4L | MA_AP3 | MA_AP2 | MA_AP1 );


                (Omitted)
```