TC3210 PMIC and EDID AUX Update Application Note

Project identification	:	TC3210
Document number	:	<document number>
Issue date	:	2018-8-30
Version	:	0.1
Status	:	Draft
Classification	:	Confidential
Issued by	:	Parade Technologies, Inc

Revision History

Version	Date	Author	Brief Description
0.1	2018-08-30		Initial version

1. Introduction

This document shows how to use the DP AUX channel to update the PMIC and EEPROM (EDID) which is connected to TC3210Master I2C bus.

The DPCD registers used in the document are listed below:

DPCD Address	Description
DPCD00480	Aux Update Enable
DPCD00482	[7] EEPROM WP configure [6] EEPROM WP value
DPCD0048B	0x90: Selection Path to Master I2C Bus
DPCD0048E	[7:1] I2C Device address [0] Enable this I2C address
DPCD00490	Aux Control TC3210 Register Enable
DPCD00491	TC3210 internal register Page selection
DPCD00492	TC3210 internal register offset
DPCD00493	TC3210 internal register value

2. Write PMIC over AUX

Make sure the PMIC IC is not Write Protected before the following sequence.
Make sure the PMIC IC is connected to TC3210's master I2C bus.

1) Open the I2C over Aux channel

a) Mapping DPCD register
- Single byte write (0x50, 0x41, 0x52, 0x41, 0x44, 0x45, 0x2d, 0x46, 0x57, 0x2D, 0x44, 0x50, 0x00, 0x06, 0x03, 0x03) to DPCD00480
- Read DPCD00480, if the read value is 1, then go to step b), else if the value is 0, repeat Step a)
- The above setting is used to enable writing the DPCD00482, DPCD0048B and DPCD0048E

b) Select DDC bus
- Write 0x90 to DPCD0048B

c) Set up the I2C device address which will be passed to DDC bus
- Write [the I2C read address of the PMIC] to DPCD0048E, taking 0x9E/0x9F as example

d) Use I2C over AUX command to directly write PMIC.

```
Example Code:


Reset_AUX:
   IICWrite(0x10, 0x75, 0x78); //For Parade DP501
```

```
        IICWrite(0x10, 0x75, 0x00); //For Parade DP501

    DPCDSEQUENCE:
        WriteDPCD(DPCD00480, 0x50);
        WriteDPCD(DPCD00480, 0x41);
        WriteDPCD(DPCD00480, 0x52);
        WriteDPCD(DPCD00480, 0x41);
        WriteDPCD(DPCD00480, 0x44);
        WriteDPCD(DPCD00480, 0x45);
        WriteDPCD(DPCD00480, 0x2d);
        WriteDPCD(DPCD00480, 0x46);
        WriteDPCD(DPCD00480, 0x57);

        WriteDPCD(DPCD00480, 0x2d);
        WriteDPCD(DPCD00480, 0x44);
        WriteDPCD(DPCD00480, 0x50);
        WriteDPCD(DPCD00480, 0x00);
        WriteDPCD(DPCD00480, 0x06);
        WriteDPCD(DPCD00480, 0x03);
        WriteDPCD(DPCD00480, 0x03);
        If ReadDPCD(DPCD00480)==0:
           Goto DPCDSEQUENCE

        WriteDPCD(DPCD0048B, 0x90);
        WriteDPCD(DPCD0048E, 0x9F);

        Aux_IICWrite(0x9E, offset, data)
```

2) Close the I2C over Aux channel

a) Protect EEPROM (TC3210 EEPROM_WP High)
i. Single byte Write 0xC0 to DPCD00482

b) Select EDID bus
ii. Write 0xe0 to DPCD0048B

c) Reset the I2C address which be passed to DDC bus
iii. Write 0x00 to DPCD0048E

d) Un-Mapping DPCD register
iv. Write 0x00 to DPCD00480

```
Example Code:
DPCDSEQUENCE:
   WriteDPCD(DPCD00480, 0x50);
   WriteDPCD(DPCD00480, 0x41);
   WriteDPCD(DPCD00480, 0x52);
   WriteDPCD(DPCD00480, 0x41);
   WriteDPCD(DPCD00480, 0x44);
   WriteDPCD(DPCD00480, 0x45);
   WriteDPCD(DPCD00480, 0x2d);
   WriteDPCD(DPCD00480, 0x46);
   WriteDPCD(DPCD00480, 0x57);
   WriteDPCD(DPCD00480, 0x2d);
   WriteDPCD(DPCD00480, 0x44);
   WriteDPCD(DPCD00480, 0x50);
   WriteDPCD(DPCD00480, 0x00);
   WriteDPCD(DPCD00480, 0x06);
   WriteDPCD(DPCD00480, 0x03);
   WriteDPCD(DPCD00480, 0x03);
      If ReadDPCD(DPCD00480)==0:
         Goto DPCDSEQUENCE

  WriteDPCD(DPCD00482, 0xc0);
  WriteDPCD(DPCD0048b, 0xe0);
  WriteDPCD(DPCD0048e, 0x00);
  WriteDPCD(DPCD00480, 0x00);
```

3. Read PMIC over AUX

Make sure the PMIC IC is connected to TC3210's master I2C bus.

1) Open the I2C over Aux channel

   a) Mapping DPCD register
      - Single byte write (0x50, 0x41, 0x52, 0x41, 0x44, 0x45, 0x2d, 0x46, 0x57, 0x2D, 0x44, 0x50, 0x00, 0x06, 0x03, 0x03) to DPCD00480
      - Read DPCD00480, if the read value is 1, then go to step b), else if the value is 0, repeat Step a)

   b) Select DDC bus
      - Write 0x90 to DPCD0048B

   c) Set up the I2C device address which be passed to DDC bus
      - Write I2C_Addr = ***0x9F*** to DPCD0048E

   d) Use I2C over AUX command to directly read PMIC registers

```
Example Code:
DPCDSEQUENCE:
   WriteDPCD(DPCD00480, 0x50);
   WriteDPCD(DPCD00480, 0x41);
   WriteDPCD(DPCD00480, 0x52);
   WriteDPCD(DPCD00480, 0x41);
   WriteDPCD(DPCD00480, 0x44);
   WriteDPCD(DPCD00480, 0x45);
   WriteDPCD(DPCD00480, 0x2d);
   WriteDPCD(DPCD00480, 0x46);
   WriteDPCD(DPCD00480, 0x57);
   WriteDPCD(DPCD00480, 0x2d);
   WriteDPCD(DPCD00480, 0x44);
   WriteDPCD(DPCD00480, 0x50);
   WriteDPCD(DPCD00480, 0x00);
   WriteDPCD(DPCD00480, 0x06);
   WriteDPCD(DPCD00480, 0x03);
   WriteDPCD(DPCD00480, 0x03);
   If ReadDPCD(DPCD00480)==0:
      Goto DPCDSEQUENCE
```

```
WriteDPCD(DPCD0048B, 0x90);
WriteDPCD(DPCD0048E, 0x9F);

AUX_IICRead(0x9F, offset, data)
```

2) Close the I2C over Aux channel

e) Protect EEPROM (TC3210 EEPROM_WP High)
i. Single byte Write 0xC0 to DPCD00482

f) Select EDID bus
ii. Write 0xe0 to DPCD0048B

g) Reset the I2C address which be passed to DDC bus
iii. Write 0x00 to DPCD0048E

h) Un-Mapping DPCD register
iv. Write 0x00 to DPCD00480

```
Example Code:
DPCDSEQUENCE:
  WriteDPCD(DPCD00480, 0x50);
  WriteDPCD(DPCD00480, 0x41);
  WriteDPCD(DPCD00480, 0x52);
  WriteDPCD(DPCD00480, 0x41);
  WriteDPCD(DPCD00480, 0x44);
  WriteDPCD(DPCD00480, 0x45);
  WriteDPCD(DPCD00480, 0x2d);
  WriteDPCD(DPCD00480, 0x46);
  WriteDPCD(DPCD00480, 0x57);
  WriteDPCD(DPCD00480, 0x2d);
  WriteDPCD(DPCD00480, 0x44);
  WriteDPCD(DPCD00480, 0x50);
  WriteDPCD(DPCD00480, 0x00);
  WriteDPCD(DPCD00480, 0x06);
  WriteDPCD(DPCD00480, 0x03);
  WriteDPCD(DPCD00480, 0x03);
    If ReadDPCD(DPCD00480)==0:
       Goto DPCDSEQUENCE
```

```
WriteDPCD(DPCD00482, 0xc0);
WriteDPCD(DPCD0048b, 0xe0);
WriteDPCD(DPCD0048e, 0x00);
WriteDPCD(DPCD00480, 0x00);
```

4. Write EDID over AUX

Make sure the Write Protection is connected to TC3210's Pin EEPROM_WP, otherwise, please make sure the EDID EEPROM is not Write Protected before the following sequence
Make sure the EDID EEPROM IC is connected to TC3210's master I2C bus.
EDID and Initial Code are saved in the same 32kbit EEPROM, the I2C address should be A0/A1.

1) To perform Write operation over AUX:

b) Mapping DPCD register
   i. Single byte write (0x50, 0x41, 0x52, 0x41, 0x44, 0x45, 0x2d, 0x46, 0x57, 0x2D, 0x44, 0x50, 0x00, 0x06, 0x03, 0x03) to DPCD00480
   ii. Read DPCD00480, if the read value is 1, then go to step b), else if the value is 0, repeat Step a)

c) Unprotect EEPROM (TC3210 EEPROM_WP Low)
   1. Write 0x80 to DPCD00482

d) Select DDC bus
   i. Write 0x90 to DPCD0048B

e) Set up the I2C address which will be passed to DDC bus
   i. Write I2C_Addr = **0xA1** to DPCD0048E

f) Write 256 bytes of data to EEPROM with I2C address **0xA0**

```
Example Code:
Reset_AUX:
   IICWrite(0x10, 0x75, 0x78); //For Parade DP501
   IICWrite(0x10, 0x75, 0x00); //For Parade DP501

DPCDSEQUENCE:
   WriteDPCD(DPCD00480, 0x50);
   WriteDPCD(DPCD00480, 0x41);
   WriteDPCD(DPCD00480, 0x52);
   WriteDPCD(DPCD00480, 0x41);
   WriteDPCD(DPCD00480, 0x44);
   WriteDPCD(DPCD00480, 0x45);
   WriteDPCD(DPCD00480, 0x2d);
   WriteDPCD(DPCD00480, 0x46);
   WriteDPCD(DPCD00480, 0x57);
   WriteDPCD(DPCD00480, 0x2d);
   WriteDPCD(DPCD00480, 0x44);
   WriteDPCD(DPCD00480, 0x50);
   WriteDPCD(DPCD00480, 0x00);
   WriteDPCD(DPCD00480, 0x06);
   WriteDPCD(DPCD00480, 0x03);
   WriteDPCD(DPCD00480, 0x03);
   If ReadDPCD(DPCD00480)==0:
      Goto DPCDSEQUENCE

   WriteDPCD(DPCD00482, 0x80);
   WriteDPCD(DPCD0048B, 0x90);
   WriteDPCD(DPCD0048E, 0xA9);
   for(j=0; j<256; j++)
      {
      Aux_IICWrite( 0xA8, j, data[j] );
      Wait(10ms)
      }
```

2) Close the I2C over Aux channel

i) Protect EEPROM (TC3210 EEPROM_WP High)
   i. Single byte Write 0xC0 to DPCD00482

j) Select EDID bus
   ii. Write 0xe0 to DPCD0048B

k) Reset the I2C address which be passed to DDC bus
   iii. Write 0x00 to DPCD0048E

l) Un-Mapping DPCD register
   iv. Write 0x00 to DPCD00480

```
Example Code:
DPCDSEQUENCE:
   WriteDPCD(DPCD00480, 0x50);
   WriteDPCD(DPCD00480, 0x41);
   WriteDPCD(DPCD00480, 0x52);
   WriteDPCD(DPCD00480, 0x41);
   WriteDPCD(DPCD00480, 0x44);
   WriteDPCD(DPCD00480, 0x45);
   WriteDPCD(DPCD00480, 0x2d);
   WriteDPCD(DPCD00480, 0x46);
   WriteDPCD(DPCD00480, 0x57);
   WriteDPCD(DPCD00480, 0x2d);
   WriteDPCD(DPCD00480, 0x44);
   WriteDPCD(DPCD00480, 0x50);
   WriteDPCD(DPCD00480, 0x00);
   WriteDPCD(DPCD00480, 0x06);
   WriteDPCD(DPCD00480, 0x03);
   WriteDPCD(DPCD00480, 0x03);
      If ReadDPCD(DPCD00480)==0:
         Goto DPCDSEQUENCE

   WriteDPCD(DPCD00482, 0xc0);
   WriteDPCD(DPCD0048b, 0xe0);
   WriteDPCD(DPCD0048e, 0x00);
   WriteDPCD(DPCD00480, 0x00);
```

5. Read EDID over AUX

Make sure the EDID EEPROM IC is connected to TC3210's master I2C bus.

Standard EDID read by I2C method can be used.

a) Read 256 byte of EDID data from EEPROM with I2C address 0xA1

```
        AUX_IICRead( 0xA1, Offset, LENGTH, data);
```

6. Write SW_AL_VCOM over AUX

a) Mapping DPCD register
- Single byte write (0x50, 0x41, 0x52, 0x41, 0x41, 0x55, 0x58, 0x2d, 0x52, 0x45, 0x47) to DPCD00490
- Read DPCD00490, if the value of DPCD00490h is 0x00, repeat the above sequence

b) Write value "V" into the register at page "P" offset "O"
- Write hex(P) to DPCD00491
- Write hex(O) to DPCD00492
- Write hex(V) to DPCD00493
-

SW AL_VCOM enable = Page4.0xef[4]
SW_AL_VCOM[8:0]= Page4.0xef[5]+Page4.0xd4[7:0]

Example Code (Set the register Page4.0xef = 0x10 to enable SW_AL_VCOM, the following sequence illustrate the sequence of this register setting through AUX CH)

```
IICWrite(0x10, 0x75, 0x78); //For Parade DP501
IICWrite(0x10, 0x75, 0x00); //For Parade DP501
EnableAuxUpdateInterface1:
  WriteDPCD(DPCD00490, 0x50);
  WriteDPCD(DPCD00490, 0x41);
  WriteDPCD(DPCD00490, 0x52);
  WriteDPCD(DPCD00490, 0x41);
  WriteDPCD(DPCD00490, 0x41);
  WriteDPCD(DPCD00490, 0x55);
  WriteDPCD(DPCD00490, 0x58);
  WriteDPCD(DPCD00490, 0x2d);
  WriteDPCD(DPCD00490, 0x52);
  WriteDPCD(DPCD00490, 0x45);
  WriteDPCD(DPCD00490, 0x47);


  ucValue = ReadDPCD(DPCD00490);
```

```
if (ucValue==0)
   goto EnableAuxUpdateInterface1;

 WriteDPCD(DPCD00491, 0x04); // select page 4
 WriteDPCD(DPCD00492, 0xef); // offset 0xef
 WriteDPCD(DPCD00493, 0x10); // set P4.0xef = 0x10
```

7. Write AL_VCOM into EEPROM over AUX

To perform Write operation over AUX:

a) Mapping DPCD register
Single byte write (0x50, 0x41, 0x52, 0x41, 0x44, 0x45, 0x2d, 0x46, 0x57, 0x2D, 0x44, 0x50, 0x00, 0x06, 0x03, 0x03) to DPCD00480
- Single byte Read DPCD00480 to check the read value is 1, if not, repeat step a)

b) Unprotect EEPROM
- Write 0x80 to DPCD00482

c) Select DDC bus
- Write 0x90 to DPCD0048B

d) Set up the I2C address which be passed to DDC bus
-Write I2C_Addr = 0xA1 to DPCD0048E

e) Write bytes of data to EEPROM with I2C address 0xA0

f) Protect EEPROM
- Write 0xC0 to DPCD00482

g) Reset the I2C address which be passed to DDC bus
-Write 0x00 to DPCD0048E

h) UnMapping DPCD register
-Write 0x00 to DPCD00480

```
//example code for TC3210:
        WriteDPCD(DPCD00480, 0x50);
        WriteDPCD(DPCD00480, 0x41);
        WriteDPCD(DPCD00480, 0x52);
        WriteDPCD(DPCD00480, 0x41);
        WriteDPCD(DPCD00480, 0x44);
        WriteDPCD(DPCD00480, 0x45);
        WriteDPCD(DPCD00480, 0x2d);
        WriteDPCD(DPCD00480, 0x46);
        WriteDPCD(DPCD00480, 0x57);
        WriteDPCD(DPCD00480, 0x2d);
        WriteDPCD(DPCD00480, 0x44);
        WriteDPCD(DPCD00480, 0x50);
        WriteDPCD(DPCD00480, 0x00);
        WriteDPCD(DPCD00480, 0x06);
        WriteDPCD(DPCD00480, 0x03);
        WriteDPCD(DPCD00480, 0x03);

        ReadDPCD(DPCD00480)
        //check the value of DPCD00480, it should be 1

        WriteDPCD(DPCD00482, 0x80);
        WriteDPCD(DPCD0048B, 0x90);
        WriteDPCD(DPCD0048E, 0XA1);

        IICWrite( 0xA0, 0x0F2C, VCOM_L );
        IICWrite( 0xA0, 0x0F2D, VCOM_H );
```

//Note:

1. I2C-over-AUX command should be 00-00-5x-01-j-data[j] (MOT set to 0), the translated I2C command is I2C-START+Ax+ACK+j+ACK+data[j]+ACK+I2C-STOP
2. The document use single byte write mode to update EEPROM to achieve maximum flexibility supporting all vendors' device. Burst write can also be implemented here.
3. Most vendors' EEPROM write time Twr is 5ms in maximum. After one write, a Twr delay should be inserted. The document uses 10ms as the example.

```
WriteDPCD(DPCD00482, 0xC0);
WriteDPCD(DPCD0048E, 0x00);
WriteDPCD(DPCD00480, 0x00);
```