

# ATE305 Control Guide Part 1

Daniel 2017/4/17 Initial Version  
2017/4/19 修正一些 Address & Error

## New PEB Pin Driver 結構

每一個 PEB Channel 都必須要靠 Pin Driver 做輸出輸入，PEB Board 的 Pin Driver 採用 ATE305 這顆 IC，每一顆 IC 具有 2 個 Channel。

<http://www.analog.com/media/en/technical-documentation/data-sheets/ADATE305.pdf>

PEB Channel Board 結構上每片有 16 個 Channel，所以共需要 8 顆 ATE305。PEB Channel Board 的 Register Map 中會有 8-bit 用作 Chip Select，而控制 ATE305 的 Protocol 中又有 2bit 用來選擇每顆 ATE305 的 2 Channel，最後組成 16 Channel 的控制。ATE305 的 Protocol 中有自己的 Address(5-bit) 來控制下面的功能：

Table 20. Register Selection

DATA[15:0]	CH[1:0]	R/W	ADDR[4:0]	Register Selected	Reset State
N/A <sup>1</sup>	N/A	N/A	0x00	NOP	N/A
DATA[13:0]	CH[1:0]	R/W	0x01	VH DAC level	4096d
DATA[13:0]	CH[1:0]	R/W	0x02	VL DAC level	4096d
DATA[13:0]	CH[1:0]	R/W	0x03	VT/VCOM DAC level	4096d
DATA[13:0]	CH[1:0]	R/W	0x04	VOL DAC level	4096d
DATA[13:0]	CH[1:0]	R/W	0x05	VOH DAC level	4096d
DATA[13:0]	CH[1:0]	R/W	0x06	VCH DAC level	4096d
DATA[13:0]	CH[1:0]	R/W	0x07	VCL DAC level	4096d
DATA[13:0]	CH[1:0]	R/W	0x08	V(IOH) DAC level	4096d
DATA[13:0]	CH[1:0]	R/W	0x09	V(IOL) DAC level	4096d
DATA[13:0]	CH[1]	R/W	0x0A	OVD high level	4096d
DATA[13:0]	CH[0]	R/W	0x0A	OVD low level	4096d
DATA[15:0]	CH[1:0]	R/W	0x0B	PMUDAC level	16384d
DATA[2:0]	CH[1:0]	R/W	0x0C	PE/PMU enable	000b
DATA[2:0]	CH[1:0]	R/W	0x0D	Channel state	000b
DATA[9:0]	CH[1:0]	R/W	0x0E	PMU state	0d
DATA[2:0]	CH[1:0]	R/W	0x0F	PMU measure enable	000b
DATA[0]	CH[1:0]	R/W	0x10	Differential comparator enable	0b
DATA[1:0]	CH[1:0]	R/W	0x11	16-bit DAC monitor	00b
DATA[1:0]	CH[1:0]	R/W	0x12	OVD_CHx alarm mask	01b
DATA[2:0]	CH[1:0]	R	0x13	OVD_CHx alarm state	N/A
N/A	N/A	N/A	0x14 to 0x1F	Reserved	N/A

PEB Channel Board, ATE305 Register Map:

W	0x37	ATE_CTRL	// ATE_CTRL[15:8]: ATE 305 Chip Select
			// ATE_CTRL[7]: R(0)/W(1) Control
			// ATE_CTRL[6:5]: Channel Select
			// ATE_CTRL[4:0]: ATE 305 Control Address
R	0x37	ATE_BUSY	// ATE_BUSY[0]: ATE Controller Working
W	0x38	ATE_DATA	// Write Data for ATE_CTRL
R	0x38	ATE_DATA	// Read Data from ATE_CTRL

上表中關於 ATE\_CTRL[6:5] (Channel Select) 有一個例外：“Adress = 0x0A” OVD

由於 OVD 在每一顆 ATE305 中為兩 channel 共用，所以此時 Channel Select 反而變成 OVD 的 high / low Select。

範例 1a，設定 Channel[3] 的 VH = 5V (輸出給 DUT 邏輯 high 時的實際電壓)

VH 的設定範圍為-2.5V ~ 7.5V，DAC\_Data 為 14-bit

運算公式為  $VH = [(10/16384) \times DAC\_DATA] - 2.5$ ,  $DAC\_DATA = [(VH + 2.5) \times 16384] / 10$

VH=5，代入後  $DAC\_DATA = 12288 = 0x3000$

```
W 0x38 0x3000 // 先將要寫入的值寫在 ATE_DATA 中
W 0x37 {8'b00000010, 1'b1, 2'b10, 5'h01}
           // Channel[3]屬第二顆 ATE305，故將其 Chip Select 設定為 8'b00000010
           // Write Command，故 bit[7] = 1
           // Channel[3]屬第二顆 ATE305 的第二 Channel，故 bit[6:5] = 2'b10
           // VH 的控制 Address = 5'h01，故 bit[4:0] = 5'h01
R 0x37 ATE_WORKING // Polling ATE_WORKING[0] 直到為 0 代表設定結束
```

範例 1b，讀取 Channel[3] 的 VH 值

```
W 0x37 {8'b00000010, 1'b0, 2'b10, 5'h01} // Read Command，故 bit[7] = 0
R 0x37 ATE_WORKING = 0
R 0x38 ATE_DATA // ATE_DATA Should be 0x3000
```

$ATE\_DATA = DAC\_DATA = 0x3000 = 12288$  代入後  $VH = [(10/16384) \times 12288] - 2.5 = 5V$

範例 2，設定 Channel[0] 的 VL = 0V (輸出給 DUT 邏輯 low 時的實際電壓)

```
W 0x38 0x1000 // 0x1000 為 0V
W 0x37 {8'h01, 1'b1, 2'b01, 5'h02} // VL 的控制 Address = 5'h02
R 0x37 ATE_WORKING = 0
```

範例 3，設定 Channel[0] 的 VT = 2.5V (VTERM 的實際電壓)

```
W 0x38 0x1000 // 0x2000 為 2.5V
W 0x37 {8'h01, 1'b1, 2'b01, 5'h03} // VT 的控制 Address = 5'h03
R 0x37 ATE_WORKING = 0
```

範例 4，設定 Channel[0] 的 VOL = 1.25V (DUT 輸出電壓低於 VOL 時算是邏輯 low)

```
W 0x38 0x1800 // 0x1800 為 1.25V
W 0x37 {8'h01, 1'b1, 2'b01, 5'h04} // VOL 的控制 Address = 5'h04
R 0x37 ATE_WORKING = 0
```

範例 5，設定 Channel[0] 的 VOH = 3.75V (DUT 輸出電壓高於 VOH 時算是邏輯 high)

```
W 0x38 0x2800 // 0x2800 為 3.75V
W 0x37 {8'h01, 1'b1, 2'b01, 5'h05} // VOH 的控制 Address = 5'h05
R 0x37 ATE_WORKING = 0
```

範例 6，設定 Channel[0] 的 VCH = 5V (輸出給 DUT 的最高電壓，就算  $V_H > V_{CH}$ ，最多也只輸出  $V_{CH}$ )

```
W 0x38 0x3000 // 0x3000 為 5V
W 0x37 {8'h01, 1'b1, 2'b01, 5'h06} // VCH 的控制 Address = 5'h06
R 0x37 ATE_WORKING = 0
```

範例 7，設定 Channel[0] 的 VCL = 0V (輸出給 DUT 的最低電壓，就算  $V_L < V_{CL}$ ，最低也只輸出  $V_{CL}$ )

```
W 0x38 0x1000 // 0x1000 為 0V
W 0x37 {8'h01, 1'b1, 2'b01, 5'h07} // VCL 的控制 Address = 5'h07
R 0x37 ATE_WORKING = 0
```

範例 8，設定 Channel[0] 的 IOH = 6mA (DUT 輸出邏輯 high 時，應該流出的電流)

IOH 的設定範圍為 -6mA ~ 18mA，DAC\_Data 為 14-bit

運算公式為  $IOH = [(24/16384) \times DAC\_DATA] - 6$ ， $DAC\_DATA = [(IOH + 6) \times 16384] / 24$

IOH=6，代入後  $DAC\_DATA = 8192 = 0x2000$

```
W 0x38 0x2000 // 0x2000 為 6mA
W 0x37 {8'h01, 1'b1, 2'b01, 5'h08} // IOH 的控制 Address = 5'h08
R 0x37 ATE_WORKING = 0
```

範例 9，設定 Channel[0] 的 IOL = 6mA (DUT 輸出邏輯 low 時，應該流出的電流)

```
W 0x38 0x2000 // 0x2000 為 6mA
W 0x37 {8'h01, 1'b1, 2'b01, 5'h09} // IOL 的控制 Address = 5'h09
R 0x37 ATE_WORKING = 0
```

範例 10a，設定 Channel[1:0] 的 OVD High = 7V (DUT 輸出電壓超過 OVD\_High 時發出警告)

OVD\_H 的設定範圍為 -3V ~ 7V，DAC\_Data 為 14-bit

運算公式為  $OVD\_H = [(10/16384) \times DAC\_DATA] - 3$ ， $DAC\_DATA = [(OVD\_H + 3) \times 16384] / 10$

OVD\_H=7，代入後  $DAC\_DATA = 16383 = 0x3FFF$

```
W 0x38 0x3FFF // 0x3FFF 為 7V
W 0x37 {8'h01, 1'b1, 2'b10, 5'h0A} // Channel Select = 2'b10 時為設定 OVD_H
                                     // OVD 的控制 Address = 5'h0A
R 0x37 ATE_WORKING = 0
```

範例 10b，設定 Channel[1:0] 的 OVD Low = -3V (DUT 輸出電壓低於 OVD\_Low 時發出警告)

```
W 0x38 0x0000 // 0x0000 為 -3V
W 0x37 {8'h01, 1'b1, 2'b01, 5'h0A} // Channel Select = 2'b01 時為設定 OVD_L
                                     // OVD 的控制 Address = 5'h0A
R 0x37 ATE_WORKING = 0
```

範例 11a，設定 Channel[0] 的 PMUDAC = 2.5V (PMUDAC 有兩種用途，可能是 PMU Mode 的 Force V 或 Force I，其中電流檔位又有五種對應到五個公式，模式由其後的 0x0E 控制)  
 PMUDAC 的設定範圍為-2.5V~7.5V，DAC\_Data 為 16-bit  
 電壓運算公式為  $PMUDAC = [(10/65536) \times DAC\_DATA] - 2.5$ ,  $DAC\_DATA = [(PMUDAC + 2.5) \times 65536] / 10$   
 PMUDAC = 2.5，代入後  $DAC\_DATA = 32768 = 0x8000$

```
W 0x38    0x8000 // 0x8000 為 2.5V
W 0x37    {8'h01, 1'b1, 2'b01, 5'h0B} // PMUDAC 的控制 Address = 5'h0B
R 0x37    ATE_WORKING = 0
```

範例 11b，設定 Channel[0] 的 PMUDAC = 25mA(模式由其後的 0x0E 控制)

PMUDAC 的電流檔位範圍為:

Range A: -50mA ~ 50mA

Range B: -4mA ~ 4mA

Range C: -400uA ~ 400uA

Range D: -40uA ~ 40uA

Range E: -4uA ~ 4uA

Range A 公式為  $PMUDAC = [(100/65536) \times DAC\_DATA] - 50$ ,  $DAC\_DATA = [(PMUDAC + 50) \times 65536] / 100$   
 (單位: mA)

Range B 公式為  $PMUDAC = [(8/65536) \times DAC\_DATA] - 4$ ,  $DAC\_DATA = [(PMUDAC + 4) \times 65536] / 8$   
 (單位: mA)

Range C 公式為  $PMUDAC = [(800/65536) \times DAC\_DATA] - 400$ ,  $DAC\_DATA = [(PMUDAC + 400) \times 65536] / 800$   
 (單位: uA)

Range D 公式為  $PMUDAC = [(80/65536) \times DAC\_DATA] - 40$ ,  $DAC\_DATA = [(PMUDAC + 40) \times 65536] / 80$   
 (單位: uA)

Range E 公式為  $PMUDAC = [(8/65536) \times DAC\_DATA] - 4$ ,  $DAC\_DATA = [(PMUDAC + 4) \times 65536] / 8$   
 (單位: uA)

PMUDAC = 25mA，必定使用 Range A，代入後  $DAC\_DATA = 49152 = 0xC000$

```
W 0x38    0xC000 // 0xC000 為 25mA
W 0x37    {8'h01, 1'b1, 2'b01, 5'h0B} // PMUDAC 的控制 Address = 5'h0B
R 0x37    ATE_WORKING = 0
```