



---

**HY16F3981**

**C 函式程式庫手冊**

## Table of Contents

<b>1. 導讀 .....</b>	<b>16</b>
1.1. C 函式程式庫簡介 .....	16
1.2. 相關文檔 .....	16
<b>2. 系統控制 .....</b>	<b>17</b>
2.1. 函數簡介 .....	17
2.2. 函數說明 .....	18
2.2.1. SYS_SleepFlagRead .....	18
2.2.2. SYS_SleepFlagClear .....	18
2.2.3. SYS_WdogFlagRead .....	19
2.2.4. SYS_WdogFlagClear .....	19
2.2.5. SYS_ResetFlagRead .....	20
2.2.6. SYS_ResetFlagClear .....	20
2.2.7. SYS_BOR_FlagRead .....	20
2.2.8. SYS_BOR_FlagClear .....	21
2.2.9. SYS_EnableGIE .....	21
2.2.10. SYS_DisableGIE .....	22
2.2.11. SYS_LowPower .....	23
2.2.12. SYS_INTPriority .....	23
<b>3. 晶片時鐘源 CLOCK .....</b>	<b>25</b>
3.1. 函數簡介 .....	25
3.2. 內部定義常量 .....	26
3.3. 函數說明 .....	27
3.3.1. DrvCLOCK_EnableHighOSC .....	27
3.3.2. DrvCLOCK_CloseEHOSC .....	27
3.3.3. DrvCLOCK_CloseIHOSC .....	28
3.3.4. DrvCLOCK_SelectIHOSC .....	28
3.3.5. DrvCLOCK_EnableLowOSC .....	29
3.3.6. DrvCLOCK_CloseELOSC .....	30
3.3.7. DrvCLOCK_SelectMCUClock .....	30
3.3.8. DrvCLOCK_TrimHAO .....	31
3.3.9. DrvCLOCK_CalibrateHAO .....	31

3.3.10.	DrvCLOCK_SelectOHS_HS.....	32
3.3.11.	DrvCLOCK_EnableENHAO .....	32
3.3.12.	DrvCLOCK_SelectIHOSC_CalHAO.....	33
<b>4.</b>	<b>定時計數器 TIMER/WDT .....</b>	<b>34</b>
4.1.	函數簡介 .....	34
4.2.	內部定義常量.....	36
4.3.	函數說明.....	38
4.3.1.	DrvWDT_Open .....	38
4.3.2.	DrvWDT_CounterRead .....	38
4.3.3.	DrvWDT_ClearWDT .....	39
4.3.4.	DrvWDT_ResetEnable .....	39
4.3.5.	DrvTMA_Open .....	40
4.3.6.	DrvTMA_Close .....	41
4.3.7.	DrvTMA_CounterRead .....	41
4.3.8.	DrvTMA_ClearTMA .....	42
4.3.9.	DrvTIMER_EnableInt .....	42
4.3.10.	DrvTIMER_DisableInt.....	42
4.3.11.	DrvTIMER_GetIntFlag .....	43
4.3.12.	DrvTIMER_ClearIntFlag .....	44
4.3.13.	DrvTMB_Open .....	44
4.3.14.	DrvTMBC_Clk_Source .....	45
4.3.15.	DrvTMBC_Clk_Disable .....	46
4.3.16.	DrvTMB_ClearTMB .....	46
4.3.17.	DrvTMB_CounterRead.....	47
4.3.18.	DrvTMB_Close .....	47
4.3.19.	DrvPWM0_Open .....	47
4.3.20.	DrvPWM1_Open .....	48
4.3.21.	DrvPWM_CountCondition .....	49
4.3.22.	DrvPWM0_Close.....	50
4.3.23.	DrvPWM1_Close.....	50
4.3.24.	DrvCAPTURE1_Open.....	51
4.3.25.	DrvCAPTURE2_Open.....	51
4.3.26.	DrvCAPTURE1_Read .....	52
4.3.27.	DrvCAPTURE2_Read .....	52
4.3.28.	DrvCAPTURE_IPort .....	53
4.3.29.	DrvTMB_TCI1Edge .....	54

4.3.30.	DrvTMB_CPI1Input .....	54
4.3.31.	DrvTMB2_Open .....	55
4.3.32.	DrvTMB2_Close .....	56
4.3.33.	DrvTMB2_Clk_Source.....	56
4.3.34.	DrvTMB2_Clk_Disable .....	57
4.3.35.	DrvTMB2_ClearTMB .....	57
4.3.36.	DrvTMB2_CounterRead.....	58
4.3.37.	DrvPWM2_Open .....	58
4.3.38.	DrvPWM3_Open .....	59
4.3.39.	DrvTMB2PWM_CountCondition.....	60
4.3.40.	DrvPWM2_Close.....	60
4.3.41.	DrvPWM3_Close.....	61
4.3.42.	DrvTMB2_CPI3Input .....	61
4.3.43.	DrvTMB2_TCI3Edge .....	62
<b>5.</b>	<b>晶片 IO 口 GPIO .....</b>	<b>63</b>
5.1.	函數簡介 .....	63
5.2.	內部定義常量.....	66
5.3.	函數說明.....	67
5.3.1.	DrvGPIO_Open .....	67
5.3.2.	DrvGPIO_SetBit .....	67
5.3.3.	DrvGPIO_ClrBit.....	68
5.3.4.	DrvGPIO_GetBit.....	68
5.3.5.	DrvGPIO_SetPortBits.....	69
5.3.6.	DrvGPIO_ClrPortBits.....	70
5.3.7.	DrvGPIO_GetPortBits .....	70
5.3.8.	DrvGPIO_IntTrigger.....	71
5.3.9.	DrvGPIO_ClkGenerator .....	71
5.3.10.	DrvGPIO_ClearIntFlag .....	72
5.3.11.	DrvGPIO_GetIntFlag .....	73
5.3.12.	DrvGPIO_Close.....	73
5.3.13.	DrvGPIO_LCDIOOpen .....	74
5.3.14.	DrvGPIO_LCDIOClose.....	75
5.3.15.	DrvGPIO_LCDIOSetPorts .....	75
5.3.16.	DrvGPIO_LCDIOClrPorts.....	76
5.3.17.	DrvGPIO_LCDIOSetBit .....	77
5.3.18.	DrvGPIO_LCDIOClrBit.....	78

5.3.19.	DrvGPIO_LCDIOGetBit.....	78
5.3.20.	DrvGPIO_EnableAnalogPin .....	79
5.3.21.	DrvGPIO_PT2_EnableINPUT .....	80
5.3.22.	DrvGPIO_PT2_DisableINPUT .....	80
5.3.23.	DrvGPIO_PT2_EnablePullHigh.....	81
5.3.24.	DrvGPIO_PT2_DisablePullHigh.....	81
5.3.25.	DrvGPIO_PT2_EnableOUTPUT .....	82
5.3.26.	DrvGPIO_PT2_DisableOUTPUT .....	82
5.3.27.	DrvGPIO_PT2_EnableINT .....	83
5.3.28.	DrvGPIO_PT2_DisableINT .....	83
5.3.29.	DrvGPIO_PT2_IntTriggerPorts .....	83
5.3.30.	DrvGPIO_PT2_IntTriggerBit.....	84
5.3.31.	DrvGPIO_PT2_GetIntFlag .....	85
5.3.32.	DrvGPIO_PT2_ClearIntFlag.....	85
5.3.33.	DrvGPIO_PT2_GetPortBits.....	86
5.3.34.	DrvGPIO_PT2_SetPortBits .....	86
5.3.35.	DrvGPIO_PT2_ClrPortBits.....	87
5.3.36.	DrvGPIO_PT3_EnableINPUT .....	87
5.3.37.	DrvGPIO_PT3_DisableINPUT .....	87
5.3.38.	DrvGPIO_PT3_EnablePullHigh.....	88
5.3.39.	DrvGPIO_PT3_DisablePullHigh.....	88
5.3.40.	DrvGPIO_PT3_EnableOUTPUT .....	89
5.3.41.	DrvGPIO_PT3_DisableOUTPUT .....	89
5.3.42.	DrvGPIO_PT3_GetPortBits.....	90
5.3.43.	DrvGPIO_PT3_SetPortBits .....	90
5.3.44.	DrvGPIO_PT3_ClrPortBits.....	91
5.3.45.	DrvGPIO_PT6_EnableINPUT .....	91
5.3.46.	DrvGPIO_PT6_DisableINPUT .....	92
5.3.47.	DrvGPIO_PT6_EnableOUTPUT .....	92
5.3.48.	DrvGPIO_PT6_DisableOUTPUT .....	93
5.3.49.	DrvGPIO_PT6_GetPortBits.....	93
5.3.50.	DrvGPIO_PT6_SetPortBits .....	94
5.3.51.	DrvGPIO_PT6_ClrPortBits.....	94
5.3.52.	DrvGPIO_PT7_EnableINPUT .....	95
5.3.53.	DrvGPIO_PT7_DisableINPUT .....	95
5.3.54.	DrvGPIO_PT7_EnableOUTPUT .....	95
5.3.55.	DrvGPIO_PT7_DisableOUTPUT .....	96
5.3.56.	DrvGPIO_PT7_GetPortBits.....	96
5.3.57.	DrvGPIO_PT7_SetPortBits .....	97

5.3.58.	DrvGPIO_PT7_ClrPortBits .....	97
5.3.59.	DrvGPIO_PT8_EnableINPUT .....	98
5.3.60.	DrvGPIO_PT8_DisableINPUT .....	98
5.3.61.	DrvGPIO_PT8_EnableOUTPUT .....	99
5.3.62.	DrvGPIO_PT8_DisableOUTPUT .....	99
5.3.63.	DrvGPIO_PT8_GetPortBits .....	100
5.3.64.	DrvGPIO_PT8_SetPortBits .....	100
5.3.65.	DrvGPIO_PT8_ClrPortBits .....	101
5.3.66.	DrvGPIO_PT9_EnableINPUT .....	101
5.3.67.	DrvGPIO_PT9_DisableINPUT .....	101
5.3.68.	DrvGPIO_PT9_EnableOUTPUT .....	102
5.3.69.	DrvGPIO_PT9_DisableOUTPUT .....	102
5.3.70.	DrvGPIO_PT9_GetPortBits .....	103
5.3.71.	DrvGPIO_PT9_SetPortBits .....	103
5.3.72.	DrvGPIO_PT9_ClrPortBits .....	104
5.3.73.	DrvGPIO_PT13_EnableINPUT .....	104
5.3.74.	DrvGPIO_PT13_DisableINPUT .....	105
5.3.75.	DrvGPIO_PT13_EnableOUTPUT .....	105
5.3.76.	DrvGPIO_PT13_DisableOUTPUT .....	106
5.3.77.	DrvGPIO_PT13_GetPortBits .....	106
5.3.78.	DrvGPIO_PT13_SetPortBits .....	107
5.3.79.	DrvGPIO_PT13_ClrPortBits .....	107
5.3.80.	DrvGPIO_PortIDIF .....	107
<b>6.</b>	<b>模數轉換器 ADC .....</b>	<b>109</b>
6.1.	函數簡介 .....	109
6.2.	內部定義常量 .....	110
6.3.	函數說明 .....	112
6.3.1.	DrvADC_PInputChannel .....	112
6.3.2.	DrvADC_NInputChannel .....	112
6.3.3.	DrvADC_SetADCInputChannel .....	113
6.3.4.	DrvADC_InputSwitch .....	114
6.3.5.	DrvADC_RefInputShort .....	114
6.3.6.	DrvADC_ADGain .....	115
6.3.7.	DrvADC_DCOffset .....	116
6.3.8.	DrvADC_RefVoltage .....	117
6.3.9.	DrvADC_FullRefRange .....	117

6.3.10.	DrvADC_OSR .....	118
6.3.11.	DrvADC_ClkEnable .....	119
6.3.12.	DrvADC_ClkDisable .....	119
6.3.13.	DrvADC_CombFilter .....	120
6.3.14.	DrvADC_EnableInt .....	120
6.3.15.	DrvADC_DisableInt .....	121
6.3.16.	DrvADC_ReadIntFlag .....	121
6.3.17.	DrvADC_Enable .....	122
6.3.18.	DrvADC_Disable .....	122
6.3.19.	DrvADC_GetConversionData .....	123
<b>7.</b>	<b>SPI32 串列通訊 .....</b>	<b>124</b>
7.1.	函數簡介 .....	124
7.2.	內部定義常量 .....	126
7.3.	函數說明 .....	127
7.3.1.	DrvSPI32_Open .....	127
7.3.2.	DrvSPI32_Close .....	128
7.3.3.	DrvSPI32_IsBusy .....	129
7.3.4.	DrvSPI32_SetClockFreq .....	129
7.3.5.	DrvSPI32_IsRxBufferFull .....	130
7.3.6.	DrvSPI32_IsTxBufferFull .....	131
7.3.7.	DrvSPI32_EnableRxInt .....	131
7.3.8.	DrvSPI32_EnableTxInt .....	132
7.3.9.	DrvSPI32_DisableRxInt .....	132
7.3.10.	DrvSPI32_DisableTxInt .....	133
7.3.11.	DrvSPI32_GetRxIntFlag .....	133
7.3.12.	DrvSPI32_GetTxIntFlag .....	134
7.3.13.	DrvSPI32_ClrIntRxFlag .....	134
7.3.14.	DrvSPI32_ClrIntTxFlag .....	135
7.3.15.	DrvSPI32_Read .....	135
7.3.16.	DrvSPI32_Write .....	136
7.3.17.	DrvSPI32_Enable .....	136
7.3.18.	DrvSPI32_BitLength .....	137
7.3.19.	DrvSPI32_GetDCFlag .....	137
7.3.20.	DrvSPI32_IsABFlag .....	138
7.3.21.	DrvSPI32_IsOVFlag .....	138
7.3.22.	DrvSPI32_IsRxFlag .....	139

7.3.23.	DrvSPI32_SetEndian .....	139
7.3.24.	DrvSPI32_SetCSO .....	140
7.3.25.	DrvSPI32_DisableIO .....	140
7.3.26.	DrvSPI32_EnableIO .....	141
<b>8.</b>	<b>非同步串列通訊 UART .....</b>	<b>142</b>
8.1.	函數簡介 .....	142
8.2.	內部定義常量 .....	145
8.3.	函數說明 .....	146
8.3.1.	DrvUART_Open .....	146
8.3.2.	DrvUART_Close .....	147
8.3.3.	DrvUART_EnableInt .....	147
8.3.4.	DrvUART_GetTxFlag .....	148
8.3.5.	DrvUART_GetRxFlag .....	149
8.3.6.	DrvUART_ClrTxFlag .....	149
8.3.7.	DrvUART_ClrRxFlag .....	149
8.3.8.	DrvUART_Read .....	150
8.3.9.	DrvUART_ClrABDOVF .....	150
8.3.10.	DrvUART_Write .....	151
8.3.11.	DrvUART_EnableWakeUp .....	151
8.3.12.	DrvUART_GetPERR .....	151
8.3.13.	DrvUART_GetFERR .....	152
8.3.14.	DrvUART_GetOERR .....	152
8.3.15.	DrvUART_GetABDOVF .....	153
8.3.16.	DrvUART_Enable_AutoBaudrate .....	153
8.3.17.	DrvUART_Disable_AutoBaudrate .....	154
8.3.18.	DrvUART_CheckTRMT .....	154
8.3.19.	DrvUART_ClkEnable .....	154
8.3.20.	DrvUART_ClkDisable .....	155
8.3.21.	DrvUART_Enable .....	156
8.3.22.	DrvUART_ConfigIO .....	156
8.3.23.	DrvUART_TRStatus .....	157
8.3.24.	DrvUART_IntType .....	157
8.3.25.	DrvUART_DisableWakeUp .....	158
8.3.26.	DrvUART_GetNERR .....	158
8.3.27.	DrvUART_ClrPERR .....	159
8.3.28.	DrvUART_ClrFERR .....	159



8.3.29.	DrvUART_ClrOERR .....	160
8.3.30.	DrvUART_ClrNERR .....	160
8.3.31.	DrvUART2_Open .....	160
8.3.32.	DrvUART2_Enable .....	162
8.3.33.	DrvUART2_Close .....	162
8.3.34.	DrvUART2_EnableInt .....	163
8.3.35.	DrvUART2_IntType .....	163
8.3.36.	DrvUART2_GetTxFlag .....	164
8.3.37.	DrvUART2_GetRxFlag .....	164
8.3.38.	DrvUART2_ClrTxFlag .....	165
8.3.39.	DrvUART2_ClrRxFlag .....	165
8.3.40.	DrvUART2_Read .....	166
8.3.41.	DrvUART2_Write .....	166
8.3.42.	DrvUART2_EnableWakeUp .....	166
8.3.43.	DrvUART2_DisableWakeUp .....	167
8.3.44.	DrvUART2_Enable_AutoBaudrate .....	167
8.3.45.	DrvUART2_Disable_AutoBaudrate .....	168
8.3.46.	DrvUART2_GetPERR .....	168
8.3.47.	DrvUART2_GetFERR .....	169
8.3.48.	DrvUART2_GetOERR .....	169
8.3.49.	DrvUART2_GetNERR .....	170
8.3.50.	DrvUART2_ClrPERR .....	170
8.3.51.	DrvUART2_ClrFERR .....	170
8.3.52.	DrvUART2_ClrOERR .....	171
8.3.53.	DrvUART2_ClrNERR .....	171
8.3.54.	DrvUART2_GetABDOVF .....	172
8.3.55.	DrvUART2_ClrABDOVF .....	172
8.3.56.	DrvUART2_TRStatus .....	173
8.3.57.	DrvUART2_CheckTRMT .....	173
8.3.58.	DrvUART2_ClkEnable .....	174
8.3.59.	DrvUART2_ClkDisable .....	174
8.3.60.	DrvUART2_ConfigIO .....	175
<b>9.</b>	<b>儀表放大器 IA .....</b>	<b>176</b>
9.1.	函數簡介 .....	176
9.2.	內部定義常量 .....	177
9.3.	函數說明 .....	178

9.3.1.	DrvIA_SetIInputChannel .....	178
9.3.2.	DrvIA_PInputChannel.....	179
9.3.3.	DrvIA_NInputChannel .....	179
9.3.4.	DrvIA_IAGain.....	180
9.3.5.	DrvIA_IACHM.....	180
9.3.6.	DrvIA_IAIS.....	181
9.3.7.	DrvIA_ENIA .....	181
<b>10.</b>	<b>低雜訊運算放大器 OPAMP .....</b>	<b>182</b>
10.1.	功能簡介 .....	182
10.2.	內部定義常量.....	183
10.3.	函數說明.....	184
10.3.1.	DrvOP_Open .....	184
10.3.2.	DrvOP_Close.....	184
10.3.3.	DrvOP_PInput .....	185
10.3.4.	DrvOP_NInput .....	186
10.3.5.	DrvOP_OPOutEnable .....	187
10.3.6.	DrvOP_OPOutDisable.....	187
10.3.7.	DrvOP_OuputFilter.....	187
10.3.8.	DrvOP_OutputPinEnable .....	188
10.3.9.	DrvOP_OutputPinDisable.....	188
10.3.10.	DrvOP_OutputInverse .....	189
10.3.11.	DrvOP_OutputWithCHPCK.....	190
10.3.12.	DrvOP_EnableInt .....	190
10.3.13.	DrvOP_DisableInt.....	191
10.3.14.	DrvOP_ReadIntFlag .....	191
10.3.15.	DrvOP_ClearIntFlag .....	192
10.3.16.	DrvOP_Feedback.....	192
10.3.17.	DrvOP_OPDEN.....	193
<b>11.</b>	<b>電源管理 PMU .....</b>	<b>194</b>
11.1.	函數簡介 .....	194
11.2.	內部定義常量.....	195
11.3.	函數說明.....	196
11.3.1.	DrvPMU_VDDA_Voltage.....	196

11.3.2.	DrvPMU_VDDA_LDO_Ctrl.....	196
11.3.3.	DrvPMU_BandgapEnable .....	197
11.3.4.	DrvPMU_BandgapDisable .....	198
11.3.5.	DrvPMU_REFO_Enable.....	198
11.3.6.	DrvPMU_REFO_Disable.....	198
11.3.7.	DrvPMU_AnalogGround.....	199
11.3.8.	DrvPMU_LDO_LowPower.....	200
11.3.9.	DrvPMU_EnableENLVD.....	200
11.3.10.	DrvPMU_DisableENLVD.....	200
11.3.11.	DrvPMU_SetLVDVS.....	201
11.3.12.	DrvPMU_SetLVD12 .....	201
11.3.13.	DrvPMU_SetLVDS .....	202
11.3.14.	DrvPMU_GetLVDO .....	203
<b>12.</b>	<b>數模轉換器 DAC .....</b>	<b>204</b>
12.1.	函數功能簡介.....	204
12.2.	內部定義常量.....	205
12.3.	函數說明.....	206
12.3.1.	DrvDAC_Open .....	206
12.3.2.	DrvDAC_Close .....	207
12.3.3.	DrvDAC_Enable .....	207
12.3.4.	DrvDAC_Disable .....	207
12.3.5.	DrvDAC_EnableOutput .....	208
12.3.6.	DrvDAC_DisableOutput .....	208
12.3.7.	DrvDAC_PInput.....	209
12.3.8.	DrvDAC_NInput.....	209
12.3.9.	DrvDAC_DABIT.....	210
12.3.10.	DrvDAC_DALH.....	210
<b>13.</b>	<b>即時時鐘 RTC.....</b>	<b>212</b>
13.1.	函數簡介.....	212
13.2.	內部定義常量.....	213
13.3.	函數說明.....	214
13.3.1.	DrvRTC_SetFrequencyCompensation.....	214
13.3.2.	DrvRTC_WriteEnable .....	214

13.3.3.	DrvRTC_WriteDisable .....	215
13.3.4.	DrvRTC_AlarmEnable .....	215
13.3.5.	DrvRTC_AlarmDisable .....	216
13.3.6.	DrvRTC_PeriodicTimeEnable .....	216
13.3.7.	DrvRTC_PeriodicTimeDisable .....	217
13.3.8.	DrvRTC_Enable .....	217
13.3.9.	DrvRTC_Disable .....	218
13.3.10.	DrvRTC_HourFormat .....	218
13.3.11.	DrvRTC_ReadState .....	219
13.3.12.	DrvRTC_ClearState .....	219
13.3.13.	DrvRTC_EnableInt .....	220
13.3.14.	DrvRTC_DisableInt .....	220
13.3.15.	DrvRTC_ReadIntFlag .....	221
13.3.16.	DrvRTC_ClearIntFlag .....	221
13.3.17.	DrvRTC_Write .....	221
13.3.18.	DrvRTC_Read .....	222
13.3.19.	DrvRTC_ClkConfig .....	223
13.3.20.	DrvRTC_EnableWUEn .....	224
13.3.21.	DrvRTC_DisableWUEn .....	224
<b>14.</b>	<b>IIC 串列通訊 I2C .....</b>	<b>226</b>
14.1.	函數簡介 .....	226
14.2.	內部定義常量 .....	227
14.3.	函數說明 .....	229
14.3.1.	DrvI2C_Open .....	229
14.3.2.	DrvI2C_Close .....	229
14.3.3.	DrvI2C_SlaveSet .....	230
14.3.4.	DrvI2C_SetIOPin .....	230
14.3.5.	DrvI2C_WriteData .....	231
14.3.6.	DrvI2C_Write3ByteData .....	232
14.3.7.	DrvI2C_ReadData .....	232
14.3.8.	DrvI2C_Ctrl .....	232
14.3.9.	DrvI2C_EnableInt .....	233
14.3.10.	DrvI2C_DisableInt .....	234
14.3.11.	DrvI2C_ReadIntFlag .....	234
14.3.12.	DrvI2C_ClearIntFlag .....	235
14.3.13.	DrvI2C_ClearEIRQ .....	235

14.3.14.	DrvI2C_ClearIRQ .....	236
14.3.15.	DrvI2C_GetStatusFlag .....	236
14.3.16.	DrvI2C_TimeOutEnable .....	237
14.3.17.	DrvI2C_TimeOutDisable .....	238
14.3.18.	DrvI2C_STSP .....	239
14.3.19.	DrvI2C_MGetACK .....	239
14.3.20.	DrvI2C_DisableIOPin .....	240
14.3.21.	DrvI2C_EnableSEn .....	240
14.3.22.	DrvI2C_DisableSEn .....	241
14.3.23.	DrvI2C_EnableI2CEn .....	241
<b>15.</b>	<b>LCD 顯示驅動器 .....</b>	<b>241</b>
15.1.	函數簡介 .....	241
15.2.	內部常量定義 .....	243
15.3.	函數說明 .....	244
15.3.1.	DrvLCD_EnableCLK .....	244
15.3.2.	DrvLCD_DisplayMode .....	244
15.3.3.	DrvLCD_LcdDuty .....	245
15.3.4.	DrvLCD_LCDBuffer .....	245
15.3.5.	DrvLCD_SwpCOMSEG .....	246
15.3.6.	DrvLCD_IOMode .....	246
15.3.7.	DrvLCD_WriteData .....	247
15.3.8.	DrvLCD_VLCDTrim .....	248
15.3.9.	DrvLCD_VLCDDMode .....	249
<b>16.</b>	<b>FLASH 讀寫 .....</b>	<b>250</b>
16.1.	函數簡介 .....	250
16.2.	函數說明 .....	251
16.2.1.	DrvFlash_Burn_Word .....	251
16.2.2.	ROM_BurnPage .....	251
16.2.3.	ReadWord .....	252
16.2.4.	ReadPage .....	252
16.2.5.	PageErase .....	253
16.2.6.	SectorErase .....	253
16.2.7.	ROM_BurnWordonly .....	254
16.2.8.	ROM_BurnPageWriteonly .....	255

---

16.3. Flash 儲存空間結構 .....	256
<b>17. ACE 指令讀寫.....</b>	<b>257</b>
17.1. 函數簡介 .....	257
17.2. 函數說明 .....	257
17.2.1. ace_mtar (Move to ACE Register) .....	257
17.2.2. ace_mfar (Move from ACE Register) .....	257
17.2.3. ace_BitRd (Bus Bit Read) .....	258
17.2.4. ace_BitWt (Bus Bit Write).....	258
17.2.5. ace_BitTg (Bus Bit Toggle) .....	259
17.2.6. ace_HByteWt (Bus Half Byte Write).....	260
17.2.7. ace_RBitWt (Regsiter Bit Write).....	261
17.2.8. ace_RBitTg (Regsiter Bit Toggle) .....	262
17.2.9. ace_RBitXor (Regsiter Bit Level XOR).....	262
<b>18. REVISION HISTORY .....</b>	<b>264</b>

注意：

- 1、本說明書中的內容，隨著產品的改進，有可能不經過預告而更改。請客戶及時到本公司網站下載更新 <http://www.hycontek.com>。
- 2、本規格書中的圖形、應用電路等，因第三方工業所有權引發的問題，本公司不承擔其責任。
- 3、本產品在單獨應用的情況下，本公司保證它的性能、典型應用和功能符合說明書中的條件。當使用在客戶的產品或設備中，以上條件我們不作保證，建議客戶做充分的評估和測試。
- 4、請注意輸入電壓、輸出電壓、負載電流的使用條件，使 IC 內的功耗不超過封裝的容許功耗。對於客戶在超出說明書中規定額定值使用產品，即使是瞬間的使用，由此所造成的損失，本公司不承擔任何責任。
- 5、本產品雖內置防靜電保護電路，但請不要施加超過保護電路性能的過大靜電。
- 6、本規格書中的產品，未經書面許可，不可使用在要求高可靠性的電路中。例如健康醫療器械、防災器械、車輛器械、車載器械及航空器械等對人體產生影響的器械或裝置，不得作為其部件使用。
- 7、本公司一直致力於提高產品的品質和可靠度，但所有的半導體產品都有一定的失效概率，這些失效概率可能會導致一些人身事故、火災事故等。當設計產品時，請充分留意冗餘設計並採用安全指標，這樣可以避免事故的發生。
- 8、本規格書中內容，未經本公司許可，嚴禁用於其他目的之轉載或複製。

## 1. 導讀

### 1.1. C 函式程式庫簡介

本檔用於描述 HYCON HY16F3981 系列 C 函式程式庫使用的參考手冊。系統端軟體發展人員可以通過使用 C 函式程式庫直接調用開發替換寄存器操作開發來有效的提高整個產品的開發效率。

### 1.2. 相關文檔

用戶可以在我們公司網站上下載以下所有文檔，獲取其他相關的資料。

下載文檔的網址：

<http://www.hycontek.com/>



## 2. 系統控制

### 2.1. 函數簡介

該部分函數描述晶片工作系統控制，包含：

- 工作模式 ( 休眠模式 ( sleep )、待機模式 ( Idle )、等待模式 ( Waite mode ) ) 的控制
- 晶片工作狀態標誌位元控制

序號	函數名稱	功能描述
01	SYS_SleepFlagRead	讀取休眠標誌位元
02	SYS_SleepFlagClear	清除休眠標誌位元
03	SYS_WdogFlagRead	讀取看門狗標誌位元
04	SYS_WdogFlagClear	清除看門狗標誌位元
05	SYS_ResetFlagRead	讀取重定標誌位元
06	SYS_ResetFlagClear	清除重定標誌位元
07	SYS_BOR_FlagRead	讀取低電壓重定(BOR) 標誌位元
08	SYS_BOR_FlagClear	清除低電壓重定(BOR) 標誌位元
09	SYS_EnableGIE	使能全域中斷 GIE 且開啟對應的中斷向量
10	SYS_DisableGIE	關閉全域中斷 GIE
11	SYS_LowPower	設置 IC 低功耗工作模式
12	SYS_INTPriority	設置對應中斷向量的中斷優先權級別

## 2.2. 函數說明

### 2.2.1. SYS\_SleepFlagRead

- 函數

unsigned int SYS\_SleepFlagRead (void);

- 函數功能

讀取休眠標誌位元 ( Sleep flag ) 的值;

讀取寄存器0x40104[3]的值。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/System.h

- 函數返回值

0：正常工作模式

1：晶片進入休眠模式

- 函數用法

```
/* 讀取休眠標誌位元 */
```

```
unsigned char temp_flag;   temp_flag=SYS_SleepFlagRead( );
```

### 2.2.2. SYS\_SleepFlagClear

- 函數

void SYS\_SleepFlagClear(void);

- 函數功能

清零休眠標誌位元；

清零寄存器0x40104[3]的值。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/System.h

- 函數返回值

無

- 函數用法

```
/* 清零sleep flag. */
```

```
SYS_SleepFlagClear( ); //set 0x40104[3]=0
```

### 2.2.3. SYS\_WdogFlagRead

- 函數

unsigned int SYS\_WdogFlagRead (void);

- 函數功能

讀取看門狗(WDT)標誌位元的值；  
讀取寄存器0x40104[2]的值。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/System.h

- 函數返回值

0：正常

1：看門狗(WDT)已經觸發

- 函數用法

/\* 讀取看門狗(WDT)標誌位元. \*/

```
unsigned char temp_flag ; temp_flag =SYS_WdogFlagRead( );
```

### 2.2.4. SYS\_WdogFlagClear

- 函數

void SYS\_WdogFlagClear(void);

- 函數功能

清零看門狗(WDT)標誌位元；  
清零寄存器0x40104[2]的值。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/System.h

- 函數返回值

無

- 函數用法

/\* 清零看門狗(WDT)的標誌位元 \*/

```
SYS_WdogFlagClear( ); //0x40104[2]=0
```

### 2.2.5. SYS\_ResetFlagRead

- 函數

unsigned int SYS\_ResetFlagRead (void);

- 函數功能

讀取重定標誌位元的值；  
讀取寄存器0x40104[1]的值。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/System.h

- 函數返回值

0：正常  
1：Reset PIN 外部復位已經觸發

- 函數用法

```
/* 讀取重置標誌位元. */  
unsigned char temp_flag ; temp_flag =SYS_ResetFlagRead( );
```

### 2.2.6. SYS\_ResetFlagClear

- 函數

void SYS\_ResetFlagClear(void);

- 函數功能

清零重定標誌位元的值；  
清零寄存器0x40104[1]的值。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/System.h

- 函數返回值

無

- 函數用法

```
/* 清零重定標誌位元 */  
SYS_ResetFlagClear( ); //0x40104[1]=0
```

### 2.2.7. SYS\_BOR\_FlagRead

- 函數

unsigned int SYS\_BOR\_FlagRead (void);

- 函數功能

讀取低電壓重定(BOR)標誌位元的值 ;  
讀取寄存器0x40104[0]的值。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/System.h

- 函數返回值

0 : 正常

1 : 低電壓重定(BOR) 功能已觸發

- 函數用法

/\* 讀取低電壓重定(BOR)標誌位元 . \*/

```
unsigned char temp_flag ; temp_flag =SYS_BOR_FlagRead( );
```

## 2.2.8. SYS\_BOR\_FlagClear

- 函數

void SYS\_BOR\_FlagClear(void);

- 函數功能

清零低電壓重定(BOR)標誌位元 ;  
清零寄存器0x40104[0]的值.

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/System.h

- 函數返回值

無

- 函數用法

/\* 清零低電壓重定(BOR) 標誌位元 \*/

```
SYS_BOR_FlagClear( ); //0x40104[0]=0
```

## 2.2.9. SYS\_EnableGIE

- 函數

```
unsigned int SYS_EnableGIE (unsigned int uPriority,unsigned short intvector);
```

● **函數功能**

使能全域中斷(GIE),使能對應中斷向量並設置相應優先權級別的中斷可以進行中斷嵌套回應,優先順序別高的先回應,中斷向量優先權級別可以通過函數SYS\_INTPriority()設置。

● **輸入參數**

uPriority [in]:設定允許開啟中斷向量的優先權級別,設置範圍是0~4

0: 不允許任何優先權級別的中斷向量回應

1: 只允許優先權級別被SYS\_INTPriority()函數設定為最高級別的中斷向量回應。

2: 只允許優先權級別被SYS\_INTPriority()函數設定為最高級別、次高級別的中斷向量回應。

3: 只允許優先權級別被SYS\_INTPriority()函數設定為最高級別、次高級別、低級別的中斷向量回應

4: 只允許優先權級別被SYS\_INTPriority()函數設定為最高級別、次高級別、低級別、最低級別的中斷向量回應

intvector[in]: 選擇中斷向量[HW9:HW8:HW7:HW6:HW5:HW4:HW3:HW2:HW1:HW0];輸入範圍為0~0x3FF。

每一位值對應一個中斷向量使能位HW9~HW0,只有對應位為1,才能使能對應的中斷向量;

invector=[HW9:HW8:HW7:HW6:HW5:HW4:HW3:HW2:HW1:HW0]

使能HW0/HW3/HW5,則invector=0x29 ( 101001B );

使能所有中斷向量,則invector=0x3FF ( 1111111111B );

不開啟任何中斷向量,則invector=0x000 ( 0000000000B )

● **包含標頭檔**

Peripheral\_lib/System.h

● **函數返回值**

0: 設置成功      1: 設置失敗

● **函數用法**

/\* 使能全域中斷GIE,並允許優先權級別為0, 1, 2,3的中斷向量回應,使能中斷向量HW0~HW9 \*/

```
SYS_EnableGIE(4,0x3FF);
```

## 2.2.10. SYS\_DisableGIE

● **函數**

```
void SYS_DisableGIE (void);
```

● **函數功能**

關閉全域中斷使能 ( GIE )。

● **輸入參數**

無

● **包含標頭檔**

Peripheral\_lib/System.h

● **函數返回值**

無

● **函數用法**

```
/* 關閉全域中斷使能(GIE). */  
SYS_DisableGIE( );
```

## 2.2.11. SYS\_LowPower

- 函數

unsigned char SYS\_LowPower(unsigned char umode)

- 函數功能

設置並啟動低功耗工作模式；開啟低功耗模式前需要開啟任何一個中斷向量。且需要切換為低頻晶震源。  
設置寄存器0x40104[4]。

- 輸入參數

umode[in]：輸入範圍0~2

0：休眠模式 ( Sleep mode )

1：待機模式 ( Idle mode )

2：等待模式 ( Waite mode )

- 包含標頭檔

Peripheral\_lib/System.h

- 函數返回值

0：設置成功

1：設置失敗

- 函數用法

```
/* 啟動休眠工作模式*/
```

```
DrvGPIO_Open(E_PT2,0xFF,E_IO_IntEnable); // 開啟PT2外部中斷向量
```

```
SYS_EnableGIE(4, 0x3FF); //開啟全域中斷控制
```

```
DrvCLOCK_SelectMCUClock(1,0); //切換頻率源為低頻
```

```
SYS_LowPower(0); //進入休眠工作模式
```

## 2.2.12. SYS\_INTPriority

- 函數

unsigned char SYS\_INTPriority(unsigned short intvector,unsigned short upriority);

- 函數功能

設置對應中斷向量的優先權級別。優先權級別為0~3,且0為最高級別。

注意：使用前，必須關閉所有的中斷使能，才能修改中斷優先權級別。

- 輸入參數

intvector[in]：中斷向量選擇，輸入範圍為0~9，分別對應HW0~HW9;

upriority [in]：設置開啟中斷向量的優先權級別，設置範圍是0~3

0: 優先權級別為最高級別

1: 優先權級別為次高級別

2: 優先權級別為低級級別.

3: 優先權級別為最低級別

在設置中斷優先權級別都為同一級別時，中斷回應的的先後順序為：

HW0 > HW1 > HW2 > ...> HW7 > HW9

● 包含標頭檔

Peripheral\_lib/System.h

● 函數返回值

0：設置成功

1：設置失敗

● 函數用法

/\* 設置中斷向量0優先權級別為 1 \* /

SYS\_ INTPriority(0,1);



### 3. 晶片時鐘源 CLOCK

#### 3.1. 函數簡介

函數描述 CPU 及其他功能模組的時鐘源操作，包含：

- 內部高速及低速晶振的控制
- 外部高速及低速晶振的控制
- CPU 時鐘源的切換

序號	函數名稱	功能描述
01	DrvCLOCK_EnableHighOSC	開啟高頻晶震
02	DrvCLOCK_CloseEHOSC	關閉外部高頻晶震
03	DrvCLOCK_CloseIHOSC	關閉內部高頻晶震
04	DrvCLOCK_SelectIHOSC	設置內部高頻晶震 HAO 的頻率
05	DrvCLOCK_EnableLowOSC	開啟低頻晶震
06	DrvCLOCK_CloseELOSC	關閉外部低頻晶震
07	DrvCLOCK_SelectMCUClock	設置 MCU 時鐘
08	DrvCLOCK_TrimHAO	內部高頻晶震校正
09	DrvCLOCK_CalibrateHAO	按照晶片出廠校正值進行 HAO 頻率校正
10	DrvCLOCK_SelectOHS_HS	外部高頻晶振模式(HSXT)選擇
11	DrvCLOCK_EnableENHAO	開啟內部高頻震盪器
12	DrvCLOCK_SelectIHOSC_CalHAO	設置內部高頻晶震 HAO 的頻率並且按照晶片出廠校正值進行 HAO 頻率校正

### 3.2. 內部定義常量

#### E\_CLOCK\_SOURCE

識別字	數值	功能意義
E_INTERNAL	0x0	內部
E_EXTERNAL	0x1	外部

#### E\_TRIM\_FREQUEN

識別字	數值	功能意義
TRIM_HAO2MHZ	0x0	校正HAO 2MHZ頻率
TRIM_HAO4MHZ	0x1	校正HAO 4MHZ頻率
TRIM_HAO10MHZ	0x2	校正HAO 10MHZ頻率
TRIM_HAO16MHZ	0x3	校正HAO 16MHZ頻率

### 3.3. 函數說明

#### 3.3.1. DrvCLOCK\_EnableHighOSC

- 函數

unsigned int DrvCLOCK\_EnableHighOSC(E\_CLOCK\_SOURCE uSource, unsigned int delay)

- 函數功能

開啟高速晶震，並選擇CPU時鐘來源為外部晶震(HSXT)或者內部晶震(HSRC)；

設定等待晶震達到穩定所需時間；

若CPU時鐘源選擇外部晶震，則寄存器0x40300[5]=1, 0x40300[1]=1;

若CPU時鐘源選擇為內部晶震，則寄存器0x40300[5]=0, 0x40300[0]=1;

- 輸入參數

uSource [in] : CPU時鐘源選擇. 設定範圍：0~1

0: 內部晶震

1: 外部晶震

delay[in] :設置等待晶震達到穩定所需時間.設定範圍：1~0xFFFFFFFF

需要注意當前CPU的運算速度CPU\_CLK；晶震達到穩定時間 $t=(1/CPU\_CLK)*4000*delay$ ；

函數內部已經有4000次運算速度的迴圈，參數delay是倍數設置，設置參數時需要參考當前運算速度及需要啟動的晶震頻率。

外部晶震(EXT OSC)達到穩定需要的時間t

4MHZ/8MHZ 約30ms；

內部晶震 ( HAO ) 達到穩定需要的時間t

2MHZ 約1.0ms

4MHZ 約0.5ms

10MHZ 約0.2ms

16MHZ 約0.1ms

- 包含標頭檔

Peripheral\_lib/DrvCLOCK.h

- 函數返回值

0：設置成功

其他：設置失敗

- 函數用法

// 開啟外部高速晶震,當前CPU\_CK=10MHZ/2，開啟外部4MHZ，設定延時 $t=40ms=(1/(10MHZ/2))*4000*50$

DrvCLOCK\_EnableHighOSC(E\_EXTERNAL,50);

#### 3.3.2. DrvCLOCK\_CloseEHOSC

- 函數

void DrvCLOCK\_CloseEHOSC()

- 函數功能

關閉外部高速晶震;注意：若被關閉的晶震當前是作為CPU時鐘源，必須先切換為有效時鐘源才能關閉該晶震。

寄存器0x40300[1]=0;

- 函數輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvCLOCK.h

- 函數返回值

無

- 函數用法

/\*開啟設定內部晶震作為CPU時鐘源，關閉外部高速晶\*/

DrvCLOCK\_EnableHighOSC(E\_INTERNAL,50); //開啟內部高速晶震

DrvCLOCK\_CloseEHOSC(); //關閉外部高速晶震

### 3.3.3. DrvCLOCK\_CloseIHOSC

- 函數

void DrvCLOCK\_CloseIHOSC()

- 函數功能

關閉內部高速晶震(HAO)；但是前提必須先將CPU時鐘源切換到有效的時鐘源。

寄存器0x40300[0]=0;

- 函數輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvCLOCK.h

- 函數返回值

無

- 函數用法

/\* 開啟外部高速晶震作為CPU時鐘源，關閉內部高速晶震\*/

DrvCLOCK\_EnableHighOSC(E\_EXTERNAL,50); //開啟外部高速時鐘源

DrvCLOCK\_CloseIHOSC(); //關閉內部高速晶震

### 3.3.4. DrvCLOCK\_SelectIHOSC

- 函數

unsigned int DrvCLOCK\_SelectIHOSC(uMode)

- 函數功能

設置內部晶震HAO的頻率模式;  
設置寄存器0x40300[4:3]。

- 輸入參數

uMode [in] : HAO頻率值, 輸入範圍 : 0~3  
0 : 2MHz, 0x40300[4:3]=00b  
1 : 4MHz, 0x40300[4:3]=01b  
2 : 10MHz, 0x40300[4:3]=10b  
3 : 16MHz, 0x40300[4:3]=11b

- 包含標頭檔

Peripheral\_lib/DrvCLOCK.h

- 函數返回值

0 : 設置成功  
其他 : 設置失敗

- 函數用法

```
/* 設置內部晶震(HAO)=4MHz*/  
DrvCLOCK_SelectIHOSC(1);
```

### 3.3.5. DrvCLOCK\_EnableLowOSC

- 函數

```
unsigned int DrvCLOCK_EnableLowOSC(E_CLOCK_SOURCE uSource · uint delay)
```

- 函數功能

開啟低速晶震頻率·並設置CPU時鐘源是內部晶震(LSRC)或者外部晶震(LSXT)及設置等待晶震達到穩定所需時間;

寄存器0x40300[6]=1. 0x40300[2]=1

- 輸入參數

uSource [in] : 輸入範圍 0~1  
0: 內部晶震LSRC  
1: 外部晶震LSXT

Delay[in] : 等待晶震穩定的時間設置·需要參考當前的運算速度來設置  
設定範圍 : 0x0~0xffffffff

外部低速晶震穩定時間 : 32768HZ 約1.3s

內部低速晶震LPO穩定時間: 約510us

- 包含標頭檔

Peripheral\_lib/DrvCLOCK.h

- 函數返回值

0 : 設置成功  
其他 : 設置失敗

- 函數用法

```
/*開啟外部低速晶振LSXT並設置穩定時間為130000*/  
DrvCLOCK_EnableLowOSC(E_EXTERNAL,130000);
```

### 3.3.6. DrvCLOCK\_CloseELOSC

- 函數

```
void DrvCLOCK_CloseELOSC()
```

- 函數功能

關閉外部低速晶震;但是需要先喚醒內部晶震(LPO)並切換至內部晶震(LPO)。  
寄存器0x40300[2]=0。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvCLOCK.h

- 函數返回值

無

- 函數用法

```
/*先開啟內部低速晶振,再關閉外部低速晶振*/  
DrvCLOCK_EnableLowOSC(E_INTERNAL,130000); //開啟內部低速晶震  
DrvCLOCK_CloseELOSC(); //關閉外部低速晶震
```

### 3.3.7. DrvCLOCK\_SelectMCUClock

- 函數

```
unsigned int DrvCLOCK_SelectMCUClock(uSource,uDiv)
```

- 函數功能

設置CPU的時鐘源為高速頻率(HS\_CK)或低速頻率(LS\_CK)·設置時鐘分頻數值;  
設置寄存器0x40308[0]與0x40308[1]。

- 輸入參數

uSource [in] : CPU時鐘源選擇  
0 : 高速頻率(HS\_CK)  
1 : 低速頻率(LS\_CK)  
uDiv [in] : CPU時鐘源分頻設置  
0 : ÷1  
1 : ÷2

- 包含標頭檔

Peripheral\_lib/DrvCLOCK.h

- 函數返回值

0 : 設置成功  
其他 : 設置失敗

● **函數用法**

```
/* 設置CPU時鐘源為高速頻率(HS_CK)並且2分頻 */  
DrvCLOCK_SelectMCUClock(0,1); //設置MCU的高速頻率源為HS_CK,且設置2分頻
```

### 3.3.8. DrvCLOCK\_TrimHAO

● **函數**

```
unsigned int DrvCLOCK_TrimHAO(uTrim)
```

● **函數功能**

校正內部晶震(HAO);  
設置寄存器0x40304[7:0]的值。

● **輸入參數**

uTrim [in] : 頻率校正值, 輸入範圍 : 0~0xff

● **包含標頭檔**

Peripheral\_lib/DrvCLOCK.h

● **函數返回值**

0 : 設置成功  
其他 : 設置失敗

● **函數用法**

```
/*寫入0x80在校正內部晶震控制暫存器 */  
DrvCLOCK_TrimHAO(0x80); // 設置 0x40304[7:0]=0x80
```

### 3.3.9. DrvCLOCK\_CalibrateHAO

● **函數**

```
void DrvCLOCK_CalibrateHAO(short int uMHZ)
```

● **函數功能**

按照晶片出廠時HAO校正值來校正內部晶震(HAO);使用時注意要與選定的HAO頻率對應 ;  
設置寄存器0x40304[7:0]的值。

● **輸入參數**

uMHZ [in] : 待校正值的HAO頻率模式選擇  
0 : 校正 2MHZ ; 1 : 校正 4MHZ ; 2 : 校正 10MHZ ; 3 : 校正 16MHZ ;

● **包含標頭檔**

Peripheral\_lib/DrvCLOCK.h

● **函數返回值**

無

● **函數用法**

```
/*校正內部晶震(HAO)4MHZ */  
DrvCLOCK_SelectIHOSC(1); //setting HAO=4MHZ;  
DrvCLOCK_CalibrateHAO(1); //calibrate 4MHZ ;
```

### 3.3.10. DrvCLOCK\_SelectOHS\_HS

- 函數

```
unsigned int DrvCLOCK_SelectOHS_HS(unsigned int uMode)
```

- 函數功能

外部高頻晶振模式(HSXT)選擇, HSXT可選擇是大於4MHZ, 或是小於4MHZ ;  
設置寄存器0x40300[7]的值。

- 輸入參數

uMode [in] : 外部高頻晶振(HSXT)模式選擇. 輸入範圍 : 0~1  
0 : HSXT<4MHz ; 1 : HSXT>4MHz ;

- 包含標頭檔

```
Peripheral_lib/DrvCLOCK.h
```

- 函數返回值

0 : 設置成功  
1 : 設置失敗

- 函數用法

```
/*選擇外部晶震(HSXT)>4MHZ */  
DrvCLOCK_SelectOHS_HS(1); //Select HSXT > 4MHZ;
```

### 3.3.11. DrvCLOCK\_EnableENHAO

- 函數

```
void DrvCLOCK_EnableENHAO (void)
```

- 函數功能

開啟內部高速震盪器  
設置寄存器0x40300[0]=1

- 輸入參數

無

- 包含標頭檔

```
Peripheral_lib/DrvCLOCK.h
```

- 函數返回值

無

- 函數用法

```
/*開啟內部高速震盪器 */  
DrvCLOCK_EnableENHAO(); //ENHAO=1b
```



### 3.3.12. DrvCLOCK\_SelectIHOSC\_CalHAO

- 函數

unsigned int DrvCLOCK\_SelectIHOSC\_CalHAO(unsigned int uMode)

- 函數功能

設置內部晶震HAO的頻率模式，並且按照晶片出廠時HAO校正值來校正內部晶震(HAO)  
設置寄存器0x40300[4:3]、0x40304[7:0]的值。

- 輸入參數

uMode [in] : HAO頻率值，輸入範圍 : 0~3

0 : 2MHz, 0x40300[4:3]=00b

1 : 4MHz, 0x40300[4:3]=01b

2 : 10MHz, 0x40300[4:3]=10b

3 : 16MHz, 0x40300[4:3]=11b

- 包含標頭檔

Peripheral\_lib/DrvCLOCK.h

- 函數返回值

0 : 設置成功 1 : 設置失敗

- 函數用法

*/\*選擇HAO=4MHz，並且校正內部晶震(HAO)4MHZ=4.147MHz \*/*

DrvCLOCK\_SelectIHOSC\_CalHAO(1);

## 4. 定時計數器 TIMER/WDT

### 4.1. 函數簡介

該部分函數描述看門狗(WDT)/定時計數器 A(Timer A)/ 定時計數器 B(Timer B) /定時計數器 B2(Timer B2)/定時計數器 C( Timer C)的功能控制，包含：

- 看門狗(WDT)的配置控制、啟動控制、中斷控制
- 定時計數器 A(Timer A)的配置控制、啟動控制、定時中斷控制
- 定時計數器 B(Timer B)的配置控制、啟動控制、定時控制及 PWM 模式控制
- 定時計數器 B2(Timer B2)的配置設置、啟動控制、定時控制機 PWM 模式控制
- 定時計數器 C( Timer C)的配置控制及 Capture 的控制

序號	函數名稱	功能描述
01	DrvWDT_Open	開啟看門狗(WDT)
02	DrvWDT_CounterRead	讀取看門狗計數值
03	DrvWDT_ClearWDT	清零看門狗計數值
04	DrvWDT_ResetEnable	WDT 中斷工作模式選擇為 Reset Mode
05	DrvTMA_Open	開啟定時計數器(Timer A)
06	DrvTMA_Close	關閉定時計數器(Timer A)
07	DrvTMA_CounterRead	讀取定時計數器(Timer A)計數值
08	DrvTMA_ClearTMA	清零定時計數器(Timer A)計數值
09	DrvTIMER_EnableInt	開啟計時器中斷 TA/TB/TC/WDT.
10	DrvTIMER_DisableInt	關閉計時器中斷 TA/TB/TC/WDT
11	DrvTIMER_GetIntFlag	讀取中斷要求標誌位元
12	DrvTIMER_ClearIntFlag	清除中斷要求標誌位元
13	DrvTMB_Open	開啟定時計數器(Timer B)
14	DrvTMBC_Clk_Source	設置定時計數器(Timer B/C)時鐘源
15	DrvTMBC_Clk_Disable	關閉定時計數器(Timer B/C)時鐘源
16	DrvTMB_ClearTMB	清零定時計數器(Timer B)計數值
17	DrvTMB_CounterRead	讀取定時計數器(Timer B)計數值
18	DrvTMB_Close	關閉定時計數器(Timer B)
19	DrvPWM0_Open	開啟 PWM 功能及 PWM0 模式
20	DrvPWM1_Open	開啟 PWM 功能及 PWM1 模式
21	DrvPWM_CountCondition	設置 PWM0/PWM1 占空比設置
22	DrvPWM0_Close	關閉 PWM0 模式

23	DrvPWM1_Close	關閉 PWM1 模式
24	DrvCAPTURE1_Open	開啟捕捉比較器 1
25	DrvCAPTURE2_Open	開啟捕捉比較器 2
26	DrvCAPTURE1_Read	讀取捕捉比較器 1 計數值
27	DrvCAPTURE2_Read	讀取捕捉比較器 2 計數值
28	DrvCAPTURE_IPort	設置捕捉比較器信號輸入 IO 口
29	DrvTMB_TCI1Edge	TMB TCI1 輸入埠觸發模式控制
30	DrvTMB_CPI1Input	TMB CPI1 模式下輸入源控制
31	DrvTMB2_Open	開啟定時計數器(Tmier B2)
32	DrvTMB2_Close	關閉定時計數器(Timer B2)
33	DrvTMB2_Clk_Source	設置定時計數器(Timer B2)時鐘源
34	DrvTMB2_Clk_Disable	關閉定時計數器(Timer B2)時鐘源
35	DrvTMB2_ClearTMB	清零定時計數器(Timer B2)計數值
36	DrvTMB2_CounterRead	讀取定時計數器(Timer B2)計數值
37	DrvPWM2_Open	開啟 PWM 功能及 PWM2 模式
38	DrvPWM3_Open	開啟 PWM 功能及 PWM3 模式
39	DrvTMB2PWM_CountCondition	設置 PWM2/PWM3 占空比設置
40	DrvPWM2_Close	關閉 PWM2 模式
41	DrvPWM3_Close	關閉 PWM3 模式
42	DrvTMB2_CPI3Input	TMB2 CPI3 模式下輸入源控制
43	DrvTMB2_TCI3Edge	TMB2 TCI3 輸入埠觸發模式控制

## 4.2. 內部定義常量

### E\_WDT\_MODE

識別字	設定值	功能意義
E_IRQ	0x0	IRQ mode
E_RST	0x1	RESET mode

### E\_WDT\_PRE\_SCALER

識別字	設定值	功能意義
E_PRE_SCALER_D2	0x0	WDT_CK / 2
E_PRE_SCALER_D8	0x1	WDT_CK / 8
E_PRE_SCALER_D32	0x2	WDT_CK / 32
E_PRE_SCALER_D128	0x3	WDT_CK / 128
E_PRE_SCALER_D512	0x4	WDT_CK / 512
E_PRE_SCALER_D2048	0x5	WDT_CK / 2048
E_PRE_SCALER_D8192	0x6	WDT_CK / 8192
E_PRE_SCALER_D32768	0x7	WDT_CK / 32768

### E\_TIMER\_CHANNEL

識別字	設定值	功能意義
E_TMA	0x0	計時器 A
E_TMB	0x1	計時器 B
E_TMC0	0x2	計時器 C
E_TMC1	0x3	計時器 C
E_WDT	0x4	看門狗 WDT
E_TMB2	0x5	計時器 B2

### E\_TMB\_MODE

識別字	設定值	功能意義
E_TMB_MODE0	0x0	16-bit 遞增計數器TBR[15:0]，步長為1；
E_TMB_MODE1	0x1	16-bit 遞增/遞減計數器TBR[15:0]，步長為1；
E_TMB_MODE2	0x2	兩個8-bit的遞增計數器TBR[15:8]/TBR[7:0]，兩個計數器獨立同時計數，步長為1。
E_TMB_MODE3	0x3	兩個8-bit遞增計數器TBR[15:8]/TBR[7:0]，計數器步長為1，計數器TBR[7:0]計數溢出後計數器TBR[15:0]才會自動加1。

### E\_TRIGGER\_SOURCE

識別字	設定值	功能意義
E_TMB_NORMAL	0x0	總是啟用 ( Always Enable ) 連續計數方式

E_TMB_CMP_HIGH	0x1	比較器(CMP)高電位觸發
E_TMB_OP_HIGH	0x2	運放(OP)高電位觸發
E_TMB_GPIO_HIGH	0x3	Timer C的輸出CPI1 高電位觸發

**E\_DRVTIMER\_CLOCK\_SOURCE**

識別字	數值	函數功能
E_HS_CK	0	TMA 時鐘源為HS_CK
E_LS_CK	1	TMA 時鐘源為 LS_CK

**E\_CAPTURE\_SOURCE**

識別字	數值	函數功能
E_TMC_CMPO	0x0	比較器輸出
E_TMC_OPOD	0x1	運算放大器數位輸出
E_TMC_LSCK	0x2	低頻時鐘源
E_TMC_TCI0	0x3	捕捉比較器1 I/O輸入
E_TMC_TCI1	0x0	捕捉比較器2 I/O輸入
E_TMC_ASTC0	0X1	捕捉比較器2 的輸入源與捕捉比較器1一致

### 4.3. 函數說明

#### 4.3.1. DrvWDT\_Open

- 函數

uint32\_t DrvWDT\_Open (E\_WDT\_MODE eMode , E\_WDT\_PRE\_SCALER eWDTpreScaler)

- 函數功能

使能看門狗(WDT) · 設置看門狗(WDT)工作模式 · 設置時鐘分頻來設定計數溢出值 ;  
設置寄存器0x40108[2:0] / 0x40108[4]=1 。

- 輸入參數

eMode [in] : 看門狗工作模式選擇

0 : 定時中斷模式

1 : 復位元模式

eWDTpreScaler [in] : 看門狗時鐘源分頻設置

0 : WDT\_CK / 2

1 : WDT\_CK / 8

2 : WDT\_CK / 32

3 : WDT\_CK / 128

4 : WDT\_CK / 512

5 : WDT\_CK / 2048

6 : WDT\_CK / 8192

7 : WDT\_CK / 32768

- 包含標頭檔

Peripheral\_lib/DrvTIMER.h

- 函數返回值

0 : 設置成功

1 : 設置失敗

- 函數用法

/\* 設置看門狗(WDT)為 IRQ mode 及 CLK / 32 \*/

DrvWDT\_Open(E\_IRQ , E\_PRE\_SCALER\_D32);

#### 4.3.2. DrvWDT\_CounterRead

- 函數

uint32\_t DrvWDT\_CounterRead (void)

- 函數功能

讀取看門狗(WDT)計數寄存器的值 ; 讀取寄存器0x40108[30:15] 。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvTIMER.h

- 函數返回值

看門狗計數值。

- 函數用法

```
/* 讀取看門狗(WDT)計數寄存器的值 */
```

```
unsigned int data ; data=DrvWDT_CounterRead();
```

### 4.3.3. DrvWDT\_ClearWDT

- 函數

```
void DrvWDT_ClearWDT (void)
```

- 函數功能

清零看門狗(WDT)計數寄存器值，設置寄存器0x40108[5]=1，設置後該位元自動置0。

寄存器0x40108[30:15]清除為0。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvTIMER.h

- 函數返回值

無

- 函數用法

```
/* 清零看門狗 */
```

```
DrvWDT_ClearWDT();
```

### 4.3.4. DrvWDT\_ResetEnable

- 函數

```
void DrvWDT_ResetEnable(void)
```

- 函數功能

WDT中斷工作模式選擇為Reset Mode，設置寄存器0x40108[6]=1b。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvTIMER.h

- 函數返回值

無

- 函數用法

/\* WDT中斷工作模式選擇為Reset Mode \*/

DrvWDT\_ResetEnable();

### 4.3.5. DrvTMA\_Open

- 函數

unsigned int DrvTMA\_Open (eTMAOV, E\_DRVTIMER\_CLOCK\_SOURCE uclk)

- 函數功能

使能定時計數器A(Timer A) · 設置定時計數器A(Timer A)時鐘源 · 設定計數溢出值 ;

設置寄存器0x40C00[5]=1b · 0x40C00[3:0] · 設置寄存器0x40308[3] · 0x40308[2]=1b 。

- 輸入參數

eTMAOV [in] : 定時計數器A(Timer A) 計數溢出值設置 : .

0 : taclk/2

1 : taclk/4

2 : taclk/8

3 : taclk/16

4 : taclk/32

5 : taclk/64

6 : taclk/128

7 : taclk/256

8 : taclk/512

9 : taclk/1024

10 : taclk/2048

11 : taclk/4096

12 : taclk/8192

13 : taclk/16384

14 : taclk/32768

15: taclk/65536

uclk[in] : 定時計數器A(Timer A)時鐘源設置.

0: Closed

1: HS\_CK

2 :HS\_CB

3 :LS\_CK

- 包含標頭檔

Peripheral\_lib/DrvTIMER.h

- 函數返回值

0 : 設置成功

其他 : 設置失敗



- 函數用法

```
/* 使能定時計數器A(Timer A) · 且計數溢出值為taclk/8. */  
DrvTMA_Open(2,0);
```

### 4.3.6. DrvTMA\_Close

- 函數

```
void DrvTMA_Close (void)
```

- 函數功能

關閉定時計數器A(Timer A);  
設置寄存器0x40C00[5]=0b 。

- 輸入參數

無

- 包含標頭檔

```
Peripheral_lib/DrvTIMER.h
```

- 函數返回值

無

- 函數用法

```
/* 關閉定時計數器A(Timer A) */  
DrvTMA_Close();
```

### 4.3.7. DrvTMA\_CounterRead

- 函數

```
unsigned int DrvTMA_CounterRead (void)
```

- 函數功能

讀取定時計數器A(Timer A)計數寄存器的值TAR;  
讀取寄存器0x40C00[15:0] 。

- 輸入參數

無

- 包含標頭檔

```
Peripheral_lib/DrvTIMER.h
```

- 函數返回值

定時計數器A(Timer A)計數值.

- 函數用法

```
/*讀取定時計數器A(Timer A)計數值 */  
unsigned short TMA_counter; TMA_counter =DrvTMA_CounterRead();
```

### 4.3.8. DrvTMA\_ClearTMA

- 函數

void DrvTMA\_ClearTMA (void)

- 函數功能

清零定時計數器A(Timer A)計數寄存器TMAR;

設置寄存器0x40C00[4]=1, 設置後該位元自動置0。寄存器0x40C00[15:0]清除為0

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvTIMER.h

- 函數返回值

無

- 函數用法

```
/* 清零定時計數器A(Timer A)計數寄存器 */
```

```
DrvTMA_ClearTMA();
```

### 4.3.9. DrvTIMER\_EnableInt

- 函數

unsigned int DrvTIMER\_EnableInt (E\_TIMER\_CHANNEL ch)

- 函數功能

使能WDT/Timer A/Timer B/Timer B2/Timer C 中斷功能 ;

設置寄存器0x40004[20:16]的對應中斷使能位=1。設置寄存器0x4001C[17] TimerB2的中斷使能位=1。

- 輸入參數

ch [in] : 中斷源設置. 輸入範圍 : 0~5

0 : 定時計數器 A    1 : 定時計數器B    2 : 定時計數器C的C0中斷

3 : 定時計數器C的C1中斷    4 : 看門狗    5 : 定時計數器B2

- 包含標頭檔

Peripheral\_lib/DrvTIMER.h

- 函數返回值

0 : 設置成功

其他 : 設置失敗

- 函數用法

```
/*使能定時計數器A(Timer A) 中斷 */
```

```
DrvTIMER_EnableInt(E_TMA);
```

### 4.3.10. DrvTIMER\_DisableInt

- 函數

unsigned int DrvTIMER\_DisableInt (E\_TIMER\_CHANNEL ch)

- 函數功能

關閉WDT/Timer A/Timer B/Timer B2/Timer C 中斷功能；

設置寄存器0x40004[20:16]的對應模組中斷使能位元=0，設置寄存器0x4001C[17]TimerB2的中斷使能位=0。

- 輸入參數

ch [in]：中斷源選擇。輸入範圍：0~5

0：定時計數器 A    1：定時計數器B    2：定時計數器C的C0中斷

3：定時計數器C的C1中斷    4：看門狗    5：定時計數器B2

- 包含標頭檔

Peripheral\_lib/DrvTIMER.h

- 函數返回值

0：設置成功

其他：設置失敗

- 函數用法

/\* 關閉定時計數器A(Timer A) 中斷向量\*/

DrvTIMER\_DisableInt(E\_TMA);

### 4.3.11. DrvTIMER\_GetIntFlag

- 函數

unsigned int DrvTIMER\_GetIntFlag (E\_TIMER\_CHANNEL ch)

- 函數功能

讀取WDT/Timer A/Timer B/Timer C/ Timer B2 中斷要求標誌位元；

讀取寄存器0x40004[4:0]對應模組中斷要求標誌位元，讀取寄存器0x4001C[1] Timer B2 中斷要求標誌位元。

- 輸入參數

ch [in]：中斷源選擇。輸入範圍：0~5

0：定時計數器 A    1：定時計數器B    2：定時計數器C的C0中斷

3：定時計數器C的C1中斷    4：看門狗    5：定時計數器B2

- 包含標頭檔

Peripheral\_lib/DrvTIMER.h

- 函數返回值

0: 無中斷要求

1: 有中斷要求

- 函數用法

/\*讀取定時計數器A(Timer A) 中斷要求標誌位元\*/

unsigned char flag ; flag=DrvTIMER\_GetIntFlag(E\_TMA);

### 4.3.12. DrvTIMER\_ClearIntFlag

- 函數

unsigned int DrvTIMER\_ClearIntFlag (E\_TIMER\_CHANNEL ch)

- 函數功能

清除WDT/Timer A/Timer B/Timer C/Timer B2 中斷要求標誌位元；

設置寄存器0x40004[4:0]對應模組功能中斷標誌位元=0，寄存器0x4001C[1]Timer B2中斷要求標誌位元=0。

- 輸入參數

ch [in]：中斷源設置。輸入範圍：0~5

0：定時計數器 A    1：定時計數器B    2：定時計數器C的C0中斷

3：定時計數器C的C1中斷    4：看門狗    5：定時計數器B2

- 包含標頭檔

Peripheral\_lib/DrvTIMER.h

- 函數返回值

0：設置成功

其他：設置失敗

- 函數用法

/\* 清除定時計數器A(Timer A) 中斷要求標誌位元\*/

DrvTIMER\_ClearIntFlag(E\_TMA);

### 4.3.13. DrvTMB\_Open

- 函數

Unsigned int DrvTMB\_Open (E\_TMB\_MODE eTMBmode, E\_TRIGGER\_SOURCE eTriSource, eTMBOV)

- 函數功能

使能定時計數器B(Timer B)，設置定時計數器B(Timer B)寄存器計數模式，設置定時計數器B(Timer B)計數觸發源，設置定時計數器B(Timer B)計數溢出值；支援比較器、捕捉、計數和定時功能；

設置寄存器0x40C0C[15:0]、設置寄存器0x40C04[3:0] / 0x40C04[5]=1b。

- 輸入參數

eTMBmode [in]：代表定時計數器B(Timer B)計數模式。

0：計數寄存器(TMBR)是遞增計數模式，在每一個TBCLK的上升沿加1。當TMBR >TBC0時，在下一個TBCLK的上升沿TMBR變為0且TMBIF被置1，然後TMBR又重新遞增計數。

1：計數寄存器(TMBR)是遞增遞減計數模式；作為遞增模式，每一個 TBCLK 上升沿 TMBR 自動加 1，當 TMBR=TBC0 時，TMBR 變為遞減模式，且 TMBR 在每一個 TBCLK 上升沿自動減 1，當 TMBR 遞減為 0 時，中斷標誌位元(TMBIF)被置 1，然後 TMBR 又重新開始遞增計數。

2：計數寄存器(TMBR)分為兩個 8-bitPWM 模式，兩個獨立的遞增計數器，兩個計數器都是在 TBCLK 的上升沿自動加 1；當 TMBR[15:8]=TBC0[15:8]時 TMBR[15:0]=0，TMBR[7:0]=TBC0[7:0]時，TMBR[7:0]=0；

當 TMBR[15:8]=TBC0[15:8]時，在 TBCLK 的下一個上升沿時 TMBR[15:8]=0，但 TMBIF 保持為 0；在 TMBR[7:0]=TBC0[7:0]時，在 TBCLK 的下一個上升沿 TMBR[7:0]=0，且中斷標誌位元(TMBIF)被置 1。

3: 計數寄存器(TMBR)作為步進模式。TMBR分為兩個8-bit遞增計數器，在TBCLK的每個上升沿自動加1，TMBR[15:8]計數上限受控於TBC0[15:8]，TMBR[7:0]計數上限受控於TBC0[7:0]；當TMBR[7:0]=TBC0[7:0]時，在TBCLK的下一個上升沿時TMBR[7:0]=0，且TMBR[15:8]自動加1，中斷標誌位元TMBIF被置1。

eTriSource [in]：表示Timer B 計數觸發源選擇。

0: 總是啟用 ( Always Enable ) 連續計數方式

1: 比較器(CMP)高電位觸發

2: 運算放大器(OP)數位輸出高電位觸發

3: 定時計數器 C(Timer C) 輸出高電位觸發 ( CPI1)

eTMAOV [in]：計數溢出值設置，設定範圍是0~0xffff

#### ● 包含標頭檔

Peripheral\_lib/DrvTIMER.h

#### ● 函數返回值

0：設置成功

其他：設置失敗

#### ● 函數用法

/\* 開啟定時計數器B(TMB)，設置模式0，計數溢出值為0xffff，OP 高電位觸發 \*/

DrvTMB\_Open(E\_TMB\_MODE0, E\_TMB\_OP\_HIGH, 0xffff);

### 4.3.14. DrvTMBC\_Clk\_Source

#### ● 函數

unsigned int DrvTMBC\_Clk\_Source (E\_DRVTIMER\_CLOCK\_SOURCE uclk, uPerScale)

#### ● 函數功能

定時計數器B/C 時鐘源設置，及時鐘源分頻設置；

設置寄存器0x40308[7:6], 0x40308[5:4]

#### ● 輸入參數

uclk[in]：定時計數器 B/C 時鐘源。輸入範圍：0~3

0: closed

1: HS\_CK

2: HS\_CB

3: LS\_CK

uPerScale[in]：定時計數器B/C 時鐘分頻設置。輸入範圍：0~3

0: ÷1

1: ÷2

2: ÷4

3: ÷8

- 包含標頭檔

Peripheral\_lib/DrvTIMER.h

- 函數返回值

0 : 設置成功

其他 : 設置失敗

- 函數用法

/\* 設置定時計數器B 時鐘源為HS\_CK, 分頻為 2. \*/

DrvTMBC\_Clk\_Source(1,1);

### 4.3.15. DrvTMBC\_Clk\_Disable

- 函數

void DrvTMBC\_Clk\_Disable (void)

- 函數功能

關閉定時計數器B/C 時鐘源;

設置寄存器0x40308[7:6]=00b。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvTIMER.h

- 函數返回值

無

- 函數用法

/\* 關閉定時計數器B/C 時鐘源\*/

DrvTMBC\_Clk\_Disable();

### 4.3.16. DrvTMB\_ClearTMB

- 函數

void DrvTMB\_ClearTMB (void)

- 函數功能

清除定時計數器B(Timer B)的計數寄存器;

設置寄存器0x40C04[4]=1,清零後該位自動置0, 寄存器0x40C08[15:0]清除為0。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvTIMER.h

- 函數返回值

無

- 函數用法

```
/* 清除定時計數器B(Timer B)的計數寄存器. */  
DrvTMB_ClearTMB();
```

### 4.3.17. DrvTMB\_CounterRead

- 函數

```
unsigned int DrvTMB_CounterRead (void)
```

- 函數功能

讀取定時計數器B(Timer B)的計數寄存器的值；讀取寄存器0x40C08[15:0]。

- 輸入參數

無

- 包含標頭檔

```
Peripheral_lib/DrvTIMER.h
```

- 函數返回值

Timer B 計數值

- 函數用法

```
/* 讀取定時計數器B(Timer B)的計數值 */  
unsigned short TMB_counter; TMB_counter =DrvTMB_CounterRead();
```

### 4.3.18. DrvTMB\_Close

- 函數

```
void DrvTMB_Close (void)
```

- 函數功能

關閉定時計數器B(Timer B)的功能；設置寄存器0x40C04[5]=0。

- 輸入參數

無

- 包含標頭檔

```
Peripheral_lib/DrvTIMER.h
```

- 函數返回值

無

- 函數用法

```
/*關閉定時計數器B(Timer B)*/  
DrvTMB_Close();
```

### 4.3.19. DrvPWM0\_Open

- 函數

unsigned int DrvPWM0\_Open (uPWM\_Mode , uInv, uOutputPin)

- 函數功能

使能PWM0功能，及設置PWM0的工作模式、輸出波形反相設置與輸出IO設置；  
設置寄存器0x40C04[18:16] / 0x40C04[19]、寄存器0x40840[4:2] / 0x40840[0]=1b。

- 輸入參數

uPWM\_Mode [in] : PWM 工作模式設置

0: PWM A	1: PWM B
2: PWM C	3: PWM D
4 : PWME	5 : PWM F
6 : PWM G	7 : PWM G

uInv[in] : PWM輸出PWM波形相位控制

0 : 輸出波形反相  
1 : 輸出波形正常

uOutputPin[in] : PWM輸出IO 設置

0 : Rsv  
1 : Rsv  
2 : Port 2.0 =PWMO0, Port 2.1 =PWMO1  
3 : Port 2.4 =PWMO0, Port 2.5 =PWMO1  
4 : Port 8.0 =PWMO0, Port 8.1 =PWMO1  
5 : Port 8.4 =PWMO0, Port 8.5 =PWMO1  
6 : Port 9.0 =PWMO0, Port 9.1 =PWMO1  
7 : Port 9.4 =PWMO0, Port 9.5 =PWMO1

- 包含標頭檔

Peripheral\_lib/DrvTIMER.h

- 函數返回值

0 : 設置成功  
其他 : 設置失敗

- 函數用法

```
/*使能PWM0且工作模式是PWMA，反相輸出，PT2.0輸出 */  
DrvPWM0_Open(0, 0, 2);
```

### 4.3.20. DrvPWM1\_Open

- 函數

unsigned int DrvPWM1\_Open (uPWM\_Mode , uInv, uOutputPin)

- 函數功能

使能PWM1功能，及設置PWM1的工作模式、輸出波形反相設置及輸出IO設置；  
設置寄存器0x40C04[23:20]、寄存器0x40840[4:2] / 0x40840[1]=1b。



- 輸入參數

uPWM\_Mode [in] : PWM工作模式設置

0: PWM A            1: PWM B

2: PWM C            3: PWM D

4 : PWME            5 : PWM F

6 : PWM G            7 : PWM G

uInv[in] : PWM輸出波形相位控制

0 : 輸出波形反相

1 : 輸出波形正常

uOutputPin[in] : PWM輸出IO口控制

0 : Rsv

1 : Rsv

2 : Port 2.0 =PWMO0, Port 2.1 =PWMO1

3 : Port 2.4 =PWMO0, Port 2.5 =PWMO1

4 : Port 8.0 =PWMO0, Port 8.1 =PWMO1

5 : Port 8.4 =PWMO0, Port 8.5 =PWMO1

6 : Port 9.0 =PWMO0, Port 9.1 =PWMO1

7 : Port 9.4 =PWMO0, Port 9.5 =PWMO1

- 包含標頭檔

Peripheral\_lib/DrvTIMER.h

- 函數返回值

0 : 設置成功

其他 : 設置失敗

- 函數用法

*/\*使能PWM1，且設置工作模式是PWMA，反相輸出，PT2.1輸出\*/*

```
DrvPWM1_Open(0, 0, 2);
```

### 4.3.21. DrvPWM\_CountCondition

- 函數

```
void DrvPWM_CountCondition (uTBC2 , uTBC1)
```

- 函數功能

PWM0/PWM1占空比設置，寫入計數寄存器(TBC2, TBC1);

設置寄存器0x40C10[15:0](TBC1) / 0x40C10[31:16](TBC2)

- 輸入參數

uTBC1[in] : PWM0占空比設置

TBC1 : 設定範圍0~0xFFFF

uTBC2[in] : PWM1占空比設置

TBC2 : 設定範圍0~0xFFFF

- 包含標頭檔

Peripheral\_lib/DrvTIMER.h

- 函數返回值

無

- 函數用法

/\* 設置TBC1, TBC2 值為0x4000 \*/

DrvPWM\_CountCondition(0x4000,0x4000);

### 4.3.22. DrvPWM0\_Close

- 函數

void DrvPWM0\_Close (void)

- 函數功能

關閉PWM0輸出;

設置寄存器0x40840[0]=0.

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvTIMER.h

- 函數返回值

無

- 函數用法

/\*PWM0輸出關閉 \*/

DrvPWM0\_Close();

### 4.3.23. DrvPWM1\_Close

- 函數

void DrvPWM1\_Close (void)

- 函數功能

關閉PWM1輸出 ;

設置寄存器0x40840[1]=0

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvTIMER.h

- 函數返回值

無

- 函數用法

```
/*PWM1輸出關閉*/
```

```
DrvPWM1_Close();
```

#### 4.3.24. DrvCAPTURE1\_Open

- 函數

```
unsigned int DrvCapture1_Open (CAPTURE_SOURCE uChannel , uDivider, uEdge)
```

- 函數功能

開啟信號捕捉比較器Capture1，捕捉比較器輸入信號源設置、信號除頻器設置及捕捉信號觸發邊沿設置。

設置寄存器0x40C14[21:20] / 0x40C14[19:16] / 0x40C14[1] / 0x40C14[0]=1。

- 輸入參數

uChannel [in] : 捕捉器Capture1輸入信號源設置. 輸入範圍 :0~3

0 : Rsv

1 : 運算放大器輸出(OPOD)

2 : 低速時鐘源(LS\_CK)

3 : IO口輸入(TCI1)

uDivider [in] : 輸入信號除頻設置. 輸入範圍 :0~15

0: ÷1            8: ÷256

1: ÷2            9: ÷512

2: ÷4            10: ÷1024

3: ÷8            11: ÷2048

4: ÷16           12: ÷4096

5: ÷32           13: ÷8192

6: ÷64           14: ÷16384

7: ÷128          15: ÷32768

uEdge [in] : 捕捉信號觸發邊沿設置

0 : 上升沿觸發

1 : 下降沿觸發

- 包含標頭檔

```
Peripheral_lib/DrvTIMER.h
```

- 函數返回值

0 : 設置成功

其他 : 設置失敗

- 函數用法

```
/* 使能捕捉器capture1, 輸入信號選擇TCI1, 信號除頻為2048 · 上升沿觸發模式 */
```

```
DrvCapture1_Open(3, 11, 0);
```

#### 4.3.25. DrvCAPTURE2\_Open

- 函數

unsigned int DrvCapture2\_Open (CAPTURE\_SOURCE uChannel, uEdge)

- 函數功能

使能信號捕捉比較器Capture2, 設置捕捉信號輸入源及捕捉信號觸發邊沿.

設置寄存器0x40C14[22] / 0x40C14[2] / 0x40C14[0]=1。

- 輸入參數

uChannel [in] : Capture 2 捕捉信號輸入源設置

0: 信號輸入源 TC12 來自 GPIO

1: 與 Capture1 一樣的信號輸入源

uEdge [in] : 捕捉信號觸發邊沿設置

0: 上升沿觸發

1: 下降沿觸發

- 包含標頭檔

Peripheral\_lib/DrvTIMER.h

- 函數返回值

0 : 設置成功

其他 : 設置失敗

- 函數用法

```
/* 使能捕捉器capture2, 輸入信號選擇TC11, 信號除頻為2048·上升沿觸發模式 */
```

```
DrvCapture2_Open(1, 0);
```

### 4.3.26. DrvCAPTURE1\_Read

- 函數

unsigned int DrvCapture1\_Read (void)

- 函數功能

讀取捕捉比較器Capture1的計數值;

讀取寄存器0x40C18[15:0]值

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvTIMER.h

- 函數返回值

Capture1的計數值TCR0(0~0xffff)

- 函數用法

```
/* 讀取捕捉比較器Capture1 計數值*/
```

```
unsigned short tcounter; tcounter=DrvCapture1_Read();
```

### 4.3.27. DrvCAPTURE2\_Read

- 函數

unsigned int DrvCapture2\_Read (void)

- 函數功能

讀取捕捉比較器Capture2 計數值;

讀取寄存器0x40C18[31:16]值

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvTIMER.h

- 函數返回值

Capture2 計數值TCR1(0~0xffff)

- 函數用法

```
/* 讀取捕捉比較器Capture2 計數值 */
```

```
unsigned short tcounter; tcounter=DrvCapture2_Read();
```

### 4.3.28. DrvCAPTURE\_IPort

- 函數

unsigned int DrvCapture\_Iport (uInputPin)

- 函數功能

設置信號捕捉比較器的信號輸入IO口;

設置寄存器0x40840[7:5]。

- 輸入參數

uInputPin[in] :

0 : Rsv

1 : Rsv

2 : Rsv

3 : Rsv

4 : Port 2.0 =TCI1, Port 2.1 =TCI2, Port 8.1 =TCI3

5 : Port 2.2 =TCI1, Port 2.3 =TCI2, Port 8.3 =TCI3

6 : Port 2.4 =TCI1, Port 2.5 =TCI2, Port 8.5 =TCI3

7 : Port 2.6 =TCI1, Port 2.7 =TCI2, Port 8.7 =TCI3

- 包含標頭檔

Peripheral\_lib/DrvTIMER.h

- 函數返回值

0 : 設置成功

其他 : 設置失敗

- 函數用法

```
/* 捕捉比較器Capture輸入IO設置Port 2.0=TCI1, Port2.1=TCI2 */  
DrvCapture_lport(4);
```

### 4.3.29. DrvTMB\_TCI1Edge

- 函數

unsigned char DrvTMB\_TCI1Edge(unsigned int uedge)

- 函數功能

設置TMB TCI1 IO輸入源的觸發邊沿.

設置寄存器0x40C14[23] 。

- 輸入參數

uedge [in] :TMB TCI1 IO 口觸發邊沿設置

0: 電平觸發

1: 上升沿觸發

- 包含標頭檔

Peripheral\_lib/DrvTIMER.h

- 函數返回值

0 : 設置成功

1 : 設置失敗

- 函數用法

```
/* 設置TCI1上升沿觸發模式 */
```

```
DrvTMB_CPI1Input(3); //選擇 TCI1 作為 CPI1 模式的輸入源
```

```
DrvTMB_TCI1Edge(1); //設置 TCI1 IO 口上升沿觸發 ;
```

### 4.3.30. DrvTMB\_CPI1Input

- 函數

unsigned char DrvTMB\_CPI1Input(unsigned int usource)

- 函數功能

設置TMB CPI1 模式下信號輸入源.

設置寄存器0x40C14[21:20] 。

- 輸入參數

usource [in] :TMB 的CPI1模式下輸入源設置

0: Rsv

1: R2R 運算放大器輸出

2: LS\_CK

3: IO 口輸入

- 包含標頭檔

Peripheral\_lib/DrvTIMER.h

- 函數返回值

- 0：設置成功
- 1：設置失敗

- 函數用法

/\* 設置TMB的CPI1 模式下輸入源為TCI1來自GPIO口的輸入 \*/  
DrvTMB\_CPI1Input(3); //TMB 的 CPI1 模式下輸入源為 TCI1 來自 GPIO 口的輸入。

### 4.3.31. DrvTMB2\_Open

- 函數

unsigned int DrvTMB2\_Open (E\_TMB\_MODE eTMBmode, E\_TRIGGER\_SOURCE eTriSource, eTMBOV)

- 函數功能

使能定時計數器B2(Timer B2) · 設置定時計數器B2(Timer B2)寄存器計數模式 · 設置定時計數器B2(Timer B2)計數觸發源 · 設置定時計數器B2(Timer B2)計數溢出值；支援比較器、捕捉、計數和定時功能；設置寄存器0x40C2C[15:0]、寄存器0x40C24[3:0] / 0x40C24[5]=1b。

- 輸入參數

eTMBmode[in]：代表定時計數器B2(Timer B2)計數模式。

0: 計數寄存器(TMB2R)是遞增計數模式，在每一個TB2CLK的上升沿加1。當TMB2R >TB2C0時，在下一個TB2CLK的上升沿TMB2R變為0且TMB2IF被置1，然後TMB2R又重新遞增計數。

1: 計數寄存器(TMB2R)是遞增遞減計數模式；作為遞增模式，每一個TB2CLK上升沿TMB2R自動加1，當TMB2R =TB2C0時，TMB2R變為遞減模式，且TMB2R在每一個TB2CLK上升沿自動減1，當TMB2R遞減為0時，中斷標誌位元(TMB2IF)被置1，然後TMB2R又重新開始遞增計數。

2: 計數寄存器(TMB2R)分為兩個8-bitPWM 模式，兩個獨立的遞增計數器，兩個計數器都是在TB2CLK的上升沿自動加1；當TMB2R [15:8]=TB2C0[15:8]時TMB2R [15:0]=0，TMB2R [7:0]=TB2C0[7:0]時，TMB2R [7:0]=0；當TMB2R [15:8]=TB2C0[15:8]時，在TB2CLK的下一個上升沿時TMB2R [15:8]=0，但TMB2IF保持為0；在TMB2R [7:0]=TB2C0[7:0]時，在TB2CLK的下一個上升沿TMB2R [7:0]=0，且中斷標誌位元(TMB2IF)被置1。

3: 計數寄存器(TMB2R)作為步進模式。TMB2R分為兩個8-bit遞增計數器，在TB2CLK的每個上升沿自動加1，TMB2R [15:8]計數上限受控於TB2C0[15:8]，TMB2R [7:0]計數上限受控於TB2C0[7:0]；當TMB2R [7:0]=TB2C0[7:0]時，在TB2CLK的下一個上升沿時TMB2R[7:0]=0，且TMB2R[15:8]自動加1，中斷標誌位元TMB2IF被置1。

eTriSource[in]：表示Timer B2 計數觸發源選擇。

- 0: 總是啟用 ( Always Enable ) 連續計數方式
- 1: 比較器(CMP)高電位觸發
- 2: 運算放大器(OP)數位輸出高電位觸發
- 3: 定時計數器 C(Timer C) 輸出高電位觸發 ( CPI1)

eTMAOV[in]：計數溢出值設置，設定範圍是0~0xffff

- 包含標頭檔

Peripheral\_lib/DrvTIMER.h

- 函數返回值

0 : 設置成功  
其他 : 設置失敗

- 函數用法

```
/* 開啟定時計數器B2(TMB2) · 設置模式0 · 計數溢出值為0xffff · OP 高電位觸發 */  
DrvTMB2_Open(E_TMB_MODE0, E_TMB_OP_HIGH, 0xffff);
```

### 4.3.32. DrvTMB2\_Close

- 函數

void DrvTMB2\_Close (void)

- 函數功能

關閉定時計數器B2(Timer B2)的功能 ;  
設置寄存器0x40C24[5]=0。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvTIMER.h

- 函數返回值

無

- 函數用法

```
/*關閉定時計數器B2(Timer B2)*/  
DrvTMB2_Close();
```

### 4.3.33. DrvTMB2\_Clk\_Source

- 函數

unsigned int DrvTMB2\_Clk\_Source (E\_DRVTIMER\_CLOCK\_SOURCE uclk, uPerScale)

- 函數功能

定時計數器B2 時鐘源設置 · 及時鐘源分頻設置 ;  
設置寄存器0x40314[7:6] / 0x40314[5:4]

- 輸入參數

uclk[in] :定時計數器B2 時鐘源

0: closed

1: HS\_CK

2 :HS\_CB

3: LS\_CK

uPerScale [in] : 定時計數器B2 時鐘分頻設置

0: ÷1

1: ÷2



2: ÷4      3: ÷8

- 包含標頭檔

Peripheral\_lib/DrvTIMER.h

- 函數返回值

0 : 設置成功

其他 : 設置失敗

- 函數用法

/\* 設置定時計數器B2 時鐘源為HS\_CK, 分頻為 2. \*/

```
DrvTMB2_Clk_Source(1,1);
```

### 4.3.34. DrvTMB2\_Clk\_Disable

- 函數

void DrvTMB2\_Clk\_Disable (void)

- 函數功能

關閉定時計數器B2 時鐘源;

設置寄存器0x40314[7:6]=00b。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvTIMER.h

- 函數返回值

無

- 函數用法

/\* 關閉定時計數器B2 時鐘源\*/

```
DrvTMB2_Clk_Disable();
```

### 4.3.35. DrvTMB2\_ClearTMB

- 函數

void DrvTMB2\_ClearTMB (void)

- 函數功能

清除定時計數器B2(Timer B2)的計數寄存器;

設置寄存器0x40C24[4]=1,清零後該位自動置0, 寄存器0x40C28[15:0]清除為0。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvTIMER.h

- 函數返回值

無

- 函數用法

/\* 清除定時計數器B2(Timer B2)的計數寄存器. \*/

DrvTMB2\_ClearTMB();

### 4.3.36. DrvTMB2\_CounterRead

- 函數

unsigned int DrvTMB2\_CounterRead (void)

- 函數功能

讀取定時計數器B2(Timer B2)的計數寄存器的值；

讀取寄存器0x40C28 [15:0]。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvTIMER.h

- 函數返回值

Timer B2 計數值

- 函數用法

/\* 讀取定時計數器B2(Timer B2)的計數值 \*/

unsigned short tcounter; tcounter=DrvTMB2\_CounterRead();

### 4.3.37. DrvPWM2\_Open

- 函數

unsigned int DrvPWM2\_Open (uPWM\_Mode , uInv, uOutputPin)

- 函數功能

使能PWM2功能，及設置PWM2的工作模式、輸出波形反相設置與輸出IO設置；

設置寄存器0x40C24[18:16] / 0x40C24[19]、寄存器0x40848[4:2] / 0x40848[0]。

- 輸入參數

uPWM\_Mode [in] :PWM2工作模式設置

0: PWM A                    1: PWM B

2: PWM C                    3: PWM D

4 : PWM E                    5 : PWM F

6 : PWM G                    7 : PWM G

uInv[in] : PWM2輸出PWM波形相位控制

0 : 輸出波形反相

1 : 輸出波形正常

uOutputPin[in] : PWM輸出IO 設置

0 : Rsv

1 : Rsv

2 : Port 2.2 =PWMO2, Port 2.3 =PWMO3

3 : Port 2.6 =PWMO2, Port 2.7 =PWMO3

4 : Port 8.2 =PWMO2, Port 8.3 =PWMO3

5 : Port 8.6 =PWMO2, Port 8.7 =PWMO3

6 : Port 9.2 =PWMO2, Port 9.3 =PWMO3

7 : Rsv

- 包含標頭檔

Peripheral\_lib/DrvTIMER.h

- 函數返回值

0 : 設置成功

其他 : 設置失敗

- 函數用法

```
/*使能PWM2且工作模式是PWMA · 反相輸出 · PT2.2輸出 */
```

```
DrvPWM2_Open(0, 0, 2);
```

### 4.3.38. DrvPWM3\_Open

- 函數

unsigned int DrvPWM3\_Open (uPWM\_Mode , uInv, uOutputPin)

- 函數功能

使能PWM3功能 · 及設置PWM3的工作模式、輸出波形反相設置及輸出IO設置 ;

設置寄存器0x40C24[23:20]、寄存器0x40848[4:1]。

- 輸入參數

uPWM\_Mode [in] : PWM3 工作模式設置

0: PWM A            1: PWM B

2: PWM C            3: PWM D

4 : PWM E           5 : PWM F

6 : PWM G           7 : PWM G

uInv[in] : PWM3 輸出波形相位控制

0 : 輸出波形反相

1 : 輸出波形正常

uOutputPin[in] : PWM3 輸出IO口控制

0 : Rsv

1 : Rsv

2 : Port 2.2 =PWMO2, Port 2.3 =PWMO3

3 : Port 2.6 =PWMO2, Port 2.7 =PWMO3

- 4 : Port 8.2 =PWMO2, Port 8.3 =PWMO3
- 5 : Port 8.6 =PWMO2, Port 8.7 =PWMO3
- 6 : Port 9.2 =PWMO2, Port 9.3 =PWMO3
- 7 : Rsv

- 包含標頭檔

Peripheral\_lib/DrvTIMER.h

- 函數返回值

0 : 設置成功

其他 : 設置失敗

- 函數用法

*/\*使能PWM3，且設置工作模式是PWMA，反相輸出，PT2.3輸出\*/*

DrvPWM3\_Open(0, 0, 2);

### 4.3.39. DrvTMB2PWM\_CountCondition

- 函數

void DrvTMB2PWM\_CountCondition (uTBC2 , uTBC1)

- 函數功能

PWM2/PWM3占空比設置，寫入計數寄存器(TBC2, TBC1);  
設置寄存器0x40C30[15:0](TB2C1) / 0x40C30[31:16](TB2C2)

- 輸入參數

uTBC1 [in] : PWM2占空比設置

TB2C1 : 設定範圍0~0xFFFF

uTBC2 [in] : PWM3占空比設置

TB2C2 : 設定範圍0~0xFFFF

- 包含標頭檔

Peripheral\_lib/DrvTIMER.h

- 函數返回值

無

- 函數用法

*/\* 設置TBC1, TBC2 值為0x4000 \*/*

DrvTMB2PWM\_CountCondition(0x4000,0x4000);

### 4.3.40. DrvPWM2\_Close

- 函數

void DrvPWM2\_Close (void)

- 函數功能

關閉PWM2輸出;

設置寄存器0x40848[0]=0.

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvTIMER.h

- 函數返回值

無

- 函數用法

```
/*PWM2輸出關閉 */
```

```
DrvPWM2_Close();
```

#### 4.3.41. DrvPWM3\_Close

- 函數

```
void DrvPWM3_Close (void)
```

- 函數功能

關閉PWM3輸出；

設置寄存器0x40848[1]=0b

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvTIMER.h

- 函數返回值

無

- 函數用法

```
/*PWM3輸出關閉*/
```

```
DrvPWM3_Close();
```

#### 4.3.42. DrvTMB2\_CPI3Input

- 函數

```
unsigned char DrvTMB2_CPI3Input(unsigned int usource)
```

- 函數功能

設置TMB2 CPI3 模式下信號輸入源.

設置寄存器0x40C34[21:20] 。

- 輸入參數

usource [in] : TMB2 的CPI3模式下輸入源設置

0: Rsv

1: R2R 運算放大器輸出

- 2: LS\_CK
- 3: IO 口輸入通過 TCI3

- **包含標頭檔**

Peripheral\_lib/DrvTIMER.h

- **函數返回值**

- 0 : 設置成功
- 1 : 設置失敗

- **函數用法**

/\* 設置TMB2的CPI3 模式下輸入源為TCI3 \*/

DrvTMB2\_CPI3Input(3); //TMB2 的 CPI3 模式下輸入源為 TCI3。

### 4.3.43. DrvTMB2\_TCI3Edge

- **函數**

unsigned char DrvTMB2\_TCI3Edge(unsigned int uedge)

- **函數功能**

設置TMB2 TCI3 IO輸入源的觸發邊沿。

設置寄存器0x40C34[23]。

- **輸入參數**

uedge [in] : TMB2 TCI3 IO 口觸發邊沿設置

- 0: 電平觸發
- 1: 上升沿觸發

- **包含標頭檔**

Peripheral\_lib/DrvTIMER.h

- **函數返回值**

- 0 : 設置成功
- 1 : 設置失敗

- **函數用法**

/\* 設置TCI3上升沿觸發模式 \*/

DrvTMB2\_CPI3Input(3); //選擇 TCI3 作為 CPI3 模式的輸入源

DrvTMB2\_TCI3Edge(1); //設置 TCI3 IO 口上升沿觸發 ;

## 5. 晶片 IO 口 GPIO

### 5.1. 函數簡介

該部分函數描述 GPIO 的工作模式控制，包含：

- GPIO 口的工作模式控制
- GPIO 口的上拉控制
- GPIO 口的外部中斷功能控制
- GPIO 口狀態及採樣頻率控制

序號	函數名稱	功能描述
01	DrvGPIO_Open	設置GPIO PT2~3 操作模式
02	DrvGPIO_SetBit	設置GPIO pin 為1
03	DrvGPIO_ClrBit	設置GPIO pin 為0.
04	DrvGPIO_GetBit	獲取GPIO pin的值
05	DrvGPIO_SetPortBits	設置GPIO port 值
06	DrvGPIO_ClrPortBits	清除輸出GPIO port 的值
07	DrvGPIO_GetPortBits	獲取輸入GPIO port 的值
08	DrvGPIO_IntTrigger	設置GPIO 口中斷觸發源模式
09	DrvGPIO_ClkGenerator	設置GPIO 採樣時鐘源
10	DrvGPIO_ClearIntFlag	清除外部中斷標誌位元
11	DrvGPIO_GetIntFlag	讀取外部中斷標誌位元
12	DrvGPIO_Close	關閉GPIO 任何引腳的工作模式
13	DrvGPIO_LCDIOOpen	設置GPIO PT6~13操作模式
14	DrvGPIO_LCDIOClose	關閉GPIO PT6~13相應操作模式
15	DrvGPIO_LCDIOSetPorts	設置GPIO PT6~13對應引腳為1
16	DrvGPIO_LCDIOClrPorts	設置GPIO PT6~13對應引腳為0
17	DrvGPIO_LCDIOSetBit	設置GPIO PT6~13某一引腳為1 (位操作)
18	DrvGPIO_LCDIOClrBit	設置GPIO PT6~13某一引腳為0 (位操作)
19	DrvGPIO_LCDIOGetBit	獲取GPIO PT6~13某一引腳輸入值 (位操作)
20	DrvGPIO_EnableAnalogPin	關閉IO數位功能，開啟類比功能
21	DrvGPIO_PT2_EnableINPUT	使能PT2口對應引腳輸入模式功能
22	DrvGPIO_PT2_DisableINPUT	關閉PT2口對應引腳輸入模式功能
23	DrvGPIO_PT2_EnablePullHigh	使能PT2口對應引腳上拉電阻功能
24	DrvGPIO_PT2_DisablePullHigh	關閉PT2口對應引腳上拉電阻功能
25	DrvGPIO_PT2_EnableOUTPUT	使能PT2口對應引腳輸出模式功能

26	DrvGPIO_PT2_DisableOUTPUT	關閉PT2口對應引腳輸出模式功能
27	DrvGPIO_PT2_EnableINT	使能PT2口對應引腳外部中斷功能
28	DrvGPIO_PT2_DisableINT	關閉PT2口對應引腳外部中斷功能
29	DrvGPIO_PT2_IntTriggerPorts	設置PT2外部中斷觸發邊沿
30	DrvGPIO_PT2_IntTriggerBit	設置PT2口的某一位IO pin的外部中斷觸發沿
31	DrvGPIO_PT2_GetIntFlag	清除PT2外部中斷要求標誌位元
32	DrvGPIO_PT2_ClearIntFlag	讀取PT2外部中斷要求標誌位元
33	DrvGPIO_PT2_GetPortBits	讀取PT2口輸入狀態值
34	DrvGPIO_PT2_SetPortBits	設置PT2口對應引腳輸出1
35	DrvGPIO_PT2_ClrPortBits	設置PT2口對應引腳輸出0
36	DrvGPIO_PT3_EnableINPUT	使能PT3口對應引腳輸入模式功能
37	DrvGPIO_PT3_DisableINPUT	關閉PT3口對應引腳輸入模式功能
38	DrvGPIO_PT3_EnablePullHigh	使能PT3口對應引腳上拉電阻功能
39	DrvGPIO_PT3_DisablePullHigh	關閉PT3口對應引腳上拉電阻功能
40	DrvGPIO_PT3_EnableOUTPUT	使能PT3口對應引腳輸出模式功能
41	DrvGPIO_PT3_DisableOUTPUT	關閉PT3口對應引腳輸出模式功能
42	DrvGPIO_PT3_GetPortBits	讀取PT3口輸入狀態值
43	DrvGPIO_PT3_SetPortBits	設置PT3口對應引腳輸出1
44	DrvGPIO_PT3_ClrPortBits	設置PT3口對應引腳輸出0
45	DrvGPIO_PT6_EnableINPUT	使能PT6口對應引腳輸入模式功能
46	DrvGPIO_PT6_DisableINPUT	關閉PT6口對應引腳輸入模式功能
47	DrvGPIO_PT6_EnableOUTPUT	使能PT6口對應引腳輸出模式功能
48	DrvGPIO_PT6_DisableOUTPUT	關閉PT6口對應引腳輸出模式功能
49	DrvGPIO_PT6_GetPortBits	讀取PT6口輸入狀態值
50	DrvGPIO_PT6_SetPortBits	設置PT6口對應引腳輸出1
51	DrvGPIO_PT6_ClrPortBits	設置PT6口對應引腳輸出0
52	DrvGPIO_PT7_EnableINPUT	使能PT7口對應引腳輸入模式功能
53	DrvGPIO_PT7_DisableINPUT	關閉PT7口對應引腳輸入模式功能
54	DrvGPIO_PT7_EnableOUTPUT	使能PT7口對應引腳輸出模式功能
55	DrvGPIO_PT7_DisableOUTPUT	關閉PT7口對應引腳輸出模式功能
56	DrvGPIO_PT7_GetPortBits	讀取PT7口輸入狀態值
57	DrvGPIO_PT7_SetPortBits	設置PT7口對應引腳輸出1
58	DrvGPIO_PT7_ClrPortBits	設置PT7口對應引腳輸出0
59	DrvGPIO_PT8_EnableINPUT	使能PT8口對應引腳輸入模式功能
60	DrvGPIO_PT8_DisableINPUT	關閉PT8口對應引腳輸入模式功能
61	DrvGPIO_PT8_EnableOUTPUT	使能PT8口對應引腳輸出模式功能
62	DrvGPIO_PT8_DisableOUTPUT	關閉PT8口對應引腳輸出模式功能
63	DrvGPIO_PT8_GetPortBits	讀取PT8口輸入狀態值



64	DrvGPIO_PT8_SetPortBits	設置PT8口對應引腳輸出1
65	DrvGPIO_PT8_ClrPortBits	設置PT8口對應引腳輸出0
66	DrvGPIO_PT9_EnableINPUT	使能PT9口對應引腳輸入模式功能
67	DrvGPIO_PT9_DisableINPUT	關閉PT9口對應引腳輸入模式功能
68	DrvGPIO_PT9_EnableOUTPUT	使能PT9口對應引腳輸出模式功能
69	DrvGPIO_PT9_DisableOUTPUT	關閉PT9口對應引腳輸出模式功能
70	DrvGPIO_PT9_GetPortBits	讀取PT9口輸入狀態值
71	DrvGPIO_PT9_SetPortBits	設置PT9口對應引腳輸出1
72	DrvGPIO_PT9_ClrPortBits	設置PT9口對應引腳輸出0
73	DrvGPIO_PT13_EnableINPUT	使能PT13口對應引腳輸入模式功能
74	DrvGPIO_PT13_DisableINPUT	關閉PT13口對應引腳輸入模式功能
75	DrvGPIO_PT13_EnableOUTPUT	使能PT13口對應引腳輸出模式功能
76	DrvGPIO_PT13_DisableOUTPUT	關閉PT13口對應引腳輸出模式功能
77	DrvGPIO_PT13_GetPortBits	讀取PT13口輸入狀態值
78	DrvGPIO_PT13_SetPortBits	設置PT13口對應引腳輸出1
79	DrvGPIO_PT13_ClrPortBits	設置PT13口對應引腳輸出0
80	DrvGPIO_PortIDIF	讀取PT2 外部中斷條件旗標

## 5.2. 內部定義常量

### E\_DRVGPIOPORT

識別字	數值	功能意義
E_PT0	0	定義PT0
E_PT1	1	定義PT1
E_PT2	2	定義PT2
E_PT3	3	定義PT3
E_PT6	0	定義PT6
E_PT7	1	定義PT7
E_PT8	2	定義PT8
E_PT9	3	定義PT9
E_PT13	4	定義PT13
E_COM54	5	定義COM5/COM4

### E\_DRVGPIO\_IO

識別字	數值	功能意義
E_IO_INPIT	0	設置GPIO作為輸入模式
E_IO_OUTPUT	1	設置GPIO作為輸出模式
E_IO_PullHigh	2	使能上拉電阻功能
E_IO_IntEnable	3	使能外部中斷功能

### E\_DRVGPIO\_IntTriMethod

識別字	數值	功能意義
E_DisableGPIOInt	0	關閉GPIO中斷功能
E_P_Edge	1	上升沿觸發
E_N_Edge	2	下降沿觸發
E_Chang_Level	3	電平變化觸發
E_LLTri	4	低電平觸發
E_LHTri	5	高電平觸發
E_LLTri	6	低低電平觸發
E_LHTri	7	高高電平觸發

### E\_DRVGPIO\_CLOCK\_SOURCE

識別字	數值	功能意義
E_HS_CK	0	設置IO 採樣時鐘源為HS_CK
E_LS_CK	1	設置IO 採樣時鐘源為LS_CK

## 5.3. 函數說明

### 5.3.1. DrvGPIO\_Open

- 函數

int32\_t DrvGPIO\_Open ( E\_DRVGPIO\_PORT port, int32\_t i32Bit, E\_DRVGPIO\_IO mode )

- 函數功能

設置GPIO PT2~PT3任何一位元IO引腳的工作模式，可選工作模式有輸入/輸出/外部中斷/電阻上拉。

PT2寄存器 0x40810[23:16] / 0x40810[7:0] / 0x40814[23:16] / 0x40014[23:16]

PT3寄存器 0x40820[23:16] / 0x40820[7:0] / 0x40824[23:16] / 0x40010[23:16]

- 輸入參數

port [in] :代表GPIO port. 設置值有效範圍：2~3.

1 : Rsv 2 : PT2

3 : PT3 4 : Reserved.

i32Bit [in] :代表 GPIO 任何一位IO口引腳，對應位的值為1表示打開對應IO引腳工作模式，為0時不做設置  
設置值範圍：0~255.

mode [in] :代表GPIO每一位IO口的工作模式. 設置值範圍：0~3.

0 : 輸入模式 1 : 輸出模式

2 : 內部上拉 3 : 使能外部中斷

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

0 : 設置成功

其他 : 設置失敗

- 函數用法

/\* 設置PT2.0 作為輸出模式及 PT2.1 作為輸入模式\*/

DrvGPIO\_Open(E\_PT2, 0x01, E\_IO\_OUTPUT); //PT2.0打開輸出模式

DrvGPIO\_Open(E\_PT2, 0x02, E\_IO\_INPUT); //PT2.1打開輸入模式

### 5.3.2. DrvGPIO\_SetBit

- 函數

unsigned int DrvGPIO\_SetBit (E\_DRVGPIO\_PORT uport, unsigned int i32Bit)

- 函數功能

設置PT2~PT3對應的IO口輸出1.

設置GPIO的輸出狀態寄存器0x40814[7:0] / 0x40824[7:0]

- 輸入參數

uport [in] :代表GPIO port. 設置值有效範圍 : 2~3.

1 : Rsv 2 : PT2

3 : PT3 4 : Rsv.

i32Bit [in] : 代表GPIO的每一位IO口. 設置值範圍 : 0~7.

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

0 : 設置成功

其他 : 設置失敗

- 函數用法

```
/* 設置PT2.0 作為輸出模式*/
```

```
DrvGPIO_Open (E_PT2, 1, E_IO_OUTPUT);
```

```
/* 設定PT2.0輸出1 */
```

```
DrvGPIO_SetBit(E_PT2,0);
```

### 5.3.3. DrvGPIO\_ClrBit

- 函數

unsigned int DrvGPIO\_ClrBit (E\_DRVGPIO\_PORT uport, unsigned int i32Bit)

- 函數功能

設置PT2~PT3對應任何一位的IO口輸出狀態為0.

清零GPIO的輸出狀態寄存器0x40814[7:0] / 0x40824[7:0]

- 輸入參數

uport [in] : 代表GPIO port. 設置值有效範圍 : 2~3.

1 : Rsv 2 : PT2

3 : PT3 4 : Rsv.

i32Bit [in] : 代表GPIO的每一位IO口. 設置值範圍 : 0~7.

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

0 : 設置成功

0xff000000 : 設置失敗

- 函數用法

```
/* 設定PT2.0輸出0 */
```

```
DrvGPIO_ClrBit(E_PT2,0);
```

### 5.3.4. DrvGPIO\_GetBit

- 函數

uint8\_t DrvGPIO\_GetBit (E\_DRVGPIO\_PORT port, uint8\_t u32Bit)

- 函數功能

讀取GPIO PT2~PT3任何一位IO口的輸入狀態值。

讀取GPIO輸入狀態寄存器0x40818[7:0] / 0x40828[7:0]

- 輸入參數

port [in] :代表GPIO port. 設置值有效範圍 : 2~3.

1 : Rsv 2 : PT2

3 : PT3 4 : Rsv.

u32Bit [in] : 代表GPIO port任何一位IO口. 設置值範圍 : 0、1、2、3、4、5、6、7.

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

0/1 : IO pin的輸入狀態

0xff000000 : 讀取失敗

- 函數用法

```
uint32_t i32Bit;
```

```
/* 設置PT2.1作為輸入模式，並讀取PT2.1的輸入狀態值*/
```

```
DrvGPIO_Open(E_PT2, 0x02, E_IO_INPUT);
```

```
DrvGPIO_Open(E_PT2, 0x02, E_IO_PullHigh);
```

```
i32Bit = DrvGPIO_GetBit(E_PT2,1); // Read 0x40818[1]
```

### 5.3.5. DrvGPIO\_SetPortBits

- 函數

unsigned int DrvGPIO\_SetPortBits (E\_DRVGPIO\_PORT uport, unsigned int ui32Data)

- 函數功能

設置GPIO PT2~PT3 對應IO口的輸出狀態。

設置GPIO的輸出狀態寄存器0x40814[7:0] / 0x40824[7:0]

- 輸入參數

uport [in] : 代表GPIO port. 設置值有效範圍 : 2~3.

1 : Rsv 2 : PT2

3 : PT3 4 : Rsv.

i32Data [in] : 設置對應位IO口，對應位為1則會被置1，對應位為0則會被置0. 設置值範圍 : 0x00~0xFF.

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

0 : 設置成功

其他 : 設置失敗

- 函數用法

```
/* 設定PT2.1、PT2.4為1，所以設定參數0x12 */  
DrvGPIO_SetPortBits(E_PT2, 0x12); //set 0x40814[1][4]
```

### 5.3.6. DrvGPIO\_ClrPortBits

- 函數

unsigned int DrvGPIO\_ClrPortBits (E\_DRVGPIO\_PORT uport, unsigned int ui32Data)

- 函數功能

清除GPIO PT2~PT3 對應位IO口輸出狀態值。  
清零GPIO的輸出狀態寄存器0x40814[7:0] / 0x40824[7:0]

- 輸入參數

uport [in]: 代表GPIO port. 設置值有效範圍 : 2~3.

1 : Rsv    2 : PT2  
3 : PT3    4 : Rsv.

i32Data [in]: 代表對應位的IO口，對應位為1才會被置0. 設置值範圍 : 0x00~0xFF.

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

0 : 設置成功  
其他 : 設置失敗

- 函數用法

```
/* 清除PT2.1/PT2.4的輸出為0，設定輸入參數0x12 */  
DrvGPIO_ClrPortBits(E_PT2, 0x12); //Clear 0x40814[1][4]
```

### 5.3.7. DrvGPIO\_GetPortBits

- 函數

uint32\_t DrvGPIO\_GetPortBits (E\_DRVGPIO\_PORT port)

- 函數功能

讀取GPIO PT2~PT3輸入狀態值. 讀取GPIO的輸出狀態寄存器0x40818[7:0] / 0x40828[7:0]

- 輸入參數

port [in]: 代表GPIO port. 設置值有效範圍 : 2~3.

1 : Rsv    2 : PT2  
3 : PT3    4 : Rsv.

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

0 ~ 0xFF : 讀取成功，待讀取GPIO PORT的輸入狀態值:

0xff000000 : 讀取失敗

- 函數用法

/\*讀取PT2/PT3的輸入狀態值\*/

```
uint32_t i32Port;
```

```
i32Port = DrvGPIO_GetPortBits(E_PT2); //Read 0x40818[7:0]
```

```
i32Port = DrvGPIO_GetPortBits(E_PT3); // Read 0x40828[7:0]
```

### 5.3.8. DrvGPIO\_IntTrigger

- 函數

int32\_t DrvGPIO\_IntTrigger ( E\_DRVGPIO\_PORT port, uint32\_t u32Bit, E\_DRVGPIO\_TriMethod mode )

- 函數功能

使能GPIO PT3~PT2的外部中斷觸發沿並設置外部中斷的觸發沿模式.

設置GPIO寄存器0x4082C[31:0] / 0x4081C[31:0]

- 輸入參數

port [in] : 代表GPIO port. 設置值範圍 : 2~3.

2 : PT2 3 : PT3

u32Bit [in] : 代表GPIO port的每一位IO 口, 對應位為1表示該位IO被設置, 輸入0x0時, 觸發功能無效.

設置值範圍 : 0x00~0xFF.

mode [in] : IO口的中斷觸發模式選擇. 設置值範圍 : 0~7

0 : 關閉IO 外部中斷觸發      1 : 上升沿觸發

2 : 下降沿觸發                3 : 電平變化觸發

4 : 低電平觸發                5 : 高電平觸發

6 : 低電平觸發                7 : 高電平觸發.

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

0 : 設置成功

其他 : 設置失敗

- 函數用法

/\* 設置PT2.0中斷觸發模式為下降沿觸發\*/

```
DrvGPIO_ClkGenerator(0,1);                                //設置IO採樣頻率
```

```
DrvGPIO_Open(E_PT2,0x01,E_IO_IntEnable);    //使能IO外部中斷, PT2.0. 0x40014[16]=1b=PT20IE=1b
```

```
DrvGPIO_IntTrigger(E_PT2,0x01,E_N_Edge);    //設置中斷觸發模式, PT2.0. 0x4081C[2:0]=010
```

### 5.3.9. DrvGPIO\_ClkGenerator

- 函數

uint32\_t DrvGPIO\_ClkGenerator ( E\_DRVGPIO\_CLK\_SOURCE uClk, uint32\_t uDivider)

- 函數功能

設置IO採樣頻率時鐘源及時鐘分頻值。

設置寄存器0x4030C[20:16]

- 輸入參數

uClk [in]：代表GPIO採樣頻率時鐘源。設置值範圍：0~1。

0：高速時鐘源（HS\_CK）

1：低速時鐘源（LS\_CK）

uDivider [in]：IO 口的時鐘源分頻值。設置值範圍：0~15。

0: off	8: ÷128
1: ÷1	9: ÷256
2: ÷2	10: ÷512
3: ÷4	11: ÷1024
4: ÷8	12: ÷2048
5: ÷16	13: ÷4096
6: ÷32	14: ÷8192
7: ÷64	15: ÷16384

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

0：設置成功

其他：設置失敗

- 函數用法

/\* 設置IO 採樣頻率時鐘源為高速時鐘(HS\_CK)及clk/2。\*/

DrvGPIO\_ClkGenerator(E\_HS\_CK,2); //0x4030C[20]=0b,[19:16]=0010

### 5.3.10. DrvGPIO\_ClearIntFlag

- 函數

unsigned int DrvGPIO\_ClearIntFlag (E\_DRVGPIOPORT port, uint32\_t u32Bit)

- 函數功能

清除GPIO PT2~PT3外部中斷標誌位元；

清零中斷寄存器0x40010[7:0] / 0x40014[7:0]。

- 輸入參數

port [in]：代表GPIO。設置值範圍：2~3

2：PT2 3：PT3

u32Bit [in]：

代表GPIO port的每一位IO，對應位為1的才會被清零。設置值範圍：0x00~0xFF；

設定值的每一位對應一位IO pin，對應位為1的IO口的標誌位元就被清零。

- 包含標頭檔



Peripheral\_lib/DrvGPIO.h

- 函數返回值

GPIO 外部中斷標誌位元的當前狀態值

- 函數用法

```
/* 清零 PT2.2 interrupt flag */  
DrvGPIO_ClearIntFlag(E_PT2,0x04); //0x40014[3]=0b  
/*清零 PT2.3 interrupt flag*/  
DrvGPIO_ClearIntFlag(E_PT2,0x08); //0x40014[7]=0b
```

### 5.3.11. DrvGPIO\_GetIntFlag

- 函數

unsigned int DrvGPIO\_GetIntFlag(E\_DRVGPIOPORT port)

- 函數功能

讀取對應GPIO PT2~PT3的中斷標誌位元·返回寄存器的值·返回值的對應位為1表示該位的IO 口發生中斷·若為0,則表示沒有中斷產生。

讀取中斷寄存器0x40010[7:0] / 0x40014[7:0]的值。

- 輸入參數

port [in]: 代表GPIO port. 設置值範圍 : 2~3

2 : PT2 3 : PT3

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

返回值是 GPIO 的中斷標誌位元值: 0x00 ~ 0Xff

- 函數用法

```
/* 讀取 PT2/PT3 外部中斷標誌位元 */  
unsigned char flag ;  
flag=DrvGPIO_GetIntFlag(E_PT2); //read 0x40014[7:0]  
flag=DrvGPIO_GetIntFlag(E_PT3); //read 0x40010[7:0]
```

### 5.3.12. DrvGPIO\_Close

- 函數

int32\_t DrvGPIO\_Close ( E\_DRVGPIOPORT port, int32\_t i32Bit, E\_DRVGPIOMODE mode )

- 函數功能

關閉GPIO PT2~PT3任何一位IO引腳的工作模式·可選工作模式有輸入/輸出/外部中斷/電阻上拉。

設置PT2寄存器0x40810[23:16] / 0x40810[7:0] / 0x40814[23:16] / 0x40014[23:16]

PT3寄存器0x40820[23:16] / 0x40820[7:0] / 0x40824[23:16] / 0x40010[23:16]

- 輸入參數

port [in]: 代表GPIO port. 設置值有效範圍 : 2~4.

1 : Rsv 2 : PT2

3 : PT3    4 : Rsv.

i32Bit [in] : 代表 GPIO 任何一位IO 口引腳，對應位的值為1表示關閉對應IO引腳工作模式，為0時不做設置  
設置值範圍 : 0x00~0xFF.

mode [in] : 代表GPIO每一位IO口的工作模式. 設置值範圍 : 0~3.

0 : 輸入模式    1 : 輸出模式  
2 : 內部上拉    3 : 使能外部中斷

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

0 : 設置成功  
其他 : 設置失敗

- 函數用法

/\* 關閉PT2.0 作為輸出模式及 PT2.1 作為輸入模式\*/

DrvGPIO\_Close(E\_PT2, 0x01, E\_IO\_OUTPUT); //關閉PT2.0打開輸出模式

DrvGPIO\_Close(E\_PT2, 0x02, E\_IO\_INPUT); // 關閉PT2.1打開輸入模式

### 5.3.13. DrvGPIO\_LCDIOOpen

- 函數

unsigned char DrvGPIO\_LCDIOOpen ( E\_DRVGPIO\_PORT port, int32\_t i32Bit, E\_DRVGPIO\_IO mode )

- 函數功能

設置GPIO PT6~PT13任何一位IO引腳的工作模式，可選工作模式有輸入/輸出。

設置寄存器

PT6寄存器0x40850[19:18][3:2] / 0x40854[19:18][3:2] / 0x40858[19:18][3:2] / 0x4085C[19:18][3:2]

PT7寄存器0x40860[19:18][3:2] / 0x40864[19:18][3:2] / 0x40868[19:18][3:2] / 0x4086C[19:18][3:2]

PT8寄存器0x40870[19:18][3:2] / 0x40874[19:18][3:2] / 0x40878[19:18][3:2] / 0x4087C[19:18][3:2]

PT9寄存器0x40880[19:18][3:2] / 0x40884[19:18][3:2] / 0x40888[19:18][3:2] / 0x4088C[19:18][3:2]

PT13寄存器0x408C0[19:18][3:2] / 0x408C4[19:18][3:2] / 0x408C8[19:18][3:2]

- 輸入參數

port [in] : 代表GPIO port. 設置值範圍 : 0~4

0 : PT6    1 : PT7

2 : PT8    3 : PT9    4 : PT13

i32Bit [in] :代表GPIO 任何一位IO 口引腳，對應位的值為1表示打開對應IO引腳工作模式，為0時不做設置；  
設置值範圍 : 0x00~0xFF

mode [in] : 代表GPIO每一位IO口的工作模式. 設置值範圍 : 0~1.

0 : 輸入模式    1 : 輸出模式

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- **函數返回值**

0 : 設置成功  
其他 : 設置失敗

- **函數用法**

/\* 設置PT6.0作為輸出模式及PT6.1作為輸入模式\*/

DrvGPIO\_LCDIOOpen(E\_PT6,0x01,E\_IO\_OUTPUT); //PT6.0打開輸出模式 · 0x40850[3]=1b

DrvGPIO\_LCDIOOpen(E\_PT6,0x02,E\_IO\_INPUT); //PT6.1打開輸入模式, 0x40850[18]=1b

### 5.3.14. DrvGPIO\_LCDIOClose

- **函數**

unsigned char DrvGPIO\_LCDIOClose ( E\_DRVGPIO\_PORT port, int32\_t i32Bit, E\_DRVGPIO\_IO mode )

- **函數功能**

關閉GPIO PT6~PT13任何一位IO引腳的工作模式 · 可選工作模式有輸入/輸出。

設置寄存器

PT6寄存器0x40850[19:18][3:2] / 0x40854[19:18][3:2] / 0x40858[19:18][3:2] / 0x4085C[19:18][3:2]

PT7寄存器0x40860[19:18][3:2] / 0x40864[19:18][3:2] / 0x40868[19:18][3:2] / 0x4086C[19:18][3:2]

PT8寄存器0x40870[19:18][3:2] / 0x40874[19:18][3:2] / 0x40878[19:18][3:2] / 0x4087C[19:18][3:2]

PT9寄存器0x40880[19:18][3:2] / 0x40884[19:18][3:2] / 0x40888[19:18][3:2] / 0x4088C[19:18][3:2]

PT13寄存器0x408C0[19:18][3:2] / 0x408C4[19:18][3:2] / 0x408C8[19:18][3:2]

- **輸入參數**

port [in] : 代表GPIO port. 設置值範圍 : 0~4

0 : PT6    1 : PT7

2 : PT8    3 : PT9    4 : PT13

i32Bit [in] :代表 GPIO 任何一位IO 口引腳 · 對應位的值為1表示打開對應IO引腳工作模式 · 為0時不做設置 ;

設置值範圍 : 0x00~0xFF.

mode [in] : 代表GPIO 每一位IO 口的工作模式. 設置值範圍 : 0~1.

0 : 輸入模式    1 : 輸出模式

- **包含標頭檔**

Peripheral\_lib/DrvGPIO.h

- **函數返回值**

0 : 設置成功  
其他 : 設置失敗

- **函數用法**

/\* 關閉PT6.0 作為輸出模式及 PT6.1 作為輸入模式\*/

DrvGPIO\_LCDIOClose(E\_PT6, 0x01, E\_IO\_OUTPUT); //關閉PT6.0打開輸出模式, 0x40850[3]=0b

DrvGPIO\_LCDIOClose(E\_PT6, 0x02, E\_IO\_INPUT); //關閉PT6.1打開輸入模式, 0x40850[18]=0b

### 5.3.15. DrvGPIO\_LCDIOSetPorts

- 函數

unsigned char DrvGPIO\_LCDIOSetPorts (E\_DRVGPIO\_PORT uport, unsigned int ui32Data)

- 函數功能

設置GPIO PT6~PT13對應IO口的輸出狀態為1.

PT6寄存器0x40850[17][1] / 0x40854[17][1] / 0x40858[17][1] / 0x4085C[17][1]

PT7寄存器0x40860[17][1] / 0x40864[17][1] / 0x40868[17][1] / 0x4086C[17][1]

PT8寄存器0x40870[17][1] / 0x40874[17][1] / 0x40878[17][1] / 0x4087C[17][1]

PT9寄存器0x40880[17][1] / 0x40884[17][1] / 0x40888[17][1] / 0x4088C[17][1]

PT13寄存器0x408C0[17][1] / 0x408C4[17][1] / 0x408C8[17][1]

- 輸入參數

uport [in]: 代表GPIO port. 設置值範圍 : 0~4

0 : PT6    1 : PT7

2 : PT8    3 : PT9    4 : PT13

i32Data [in]: 設置對應位IO口·對應位為1才被置1. 設置值範圍 : 0x00~0xFF.

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

0 : 設置成功

其他 : 設置失敗

- 函數用法

/\* 設定PT6.1、PT6.4為1·所以設定參數0x12 \*/

DrvGPIO\_LCDIOSetPorts(E\_PT6, 0x12); // 0x40850[17]=1b, 0x40858[17]=1b

### 5.3.16. DrvGPIO\_LCDIOClrPorts

- 函數

unsigned char DrvGPIO\_LCDIOClrPorts (E\_DRVGPIO\_PORT uport, unsigned int ui32Data)

- 函數功能

設置GPIO PT6~PT13對應IO口的輸出狀態為0

PT6寄存器0x40850[17][1] / 0x40854[17][1] / 0x40858[17][1] / 0x4085C[17][1]

PT7寄存器0x40860[17][1] / 0x40864[17][1] / 0x40868[17][1] / 0x4086C[17][1]

PT8寄存器0x40870[17][1] / 0x40874[17][1] / 0x40878[17][1] / 0x4087C[17][1]

PT9寄存器0x40880[17][1] / 0x40884[17][1] / 0x40888[17][1] / 0x4088C[17][1]

PT13寄存器0x408C0[17][1] / 0x408C4[17][1] / 0x408C8[17][1]

- 輸入參數

uport [in]: 代表GPIO port. 設置值範圍 : 0~4

0 : PT6    1 : PT7

2 : PT8    3 : PT9    4 : PT13

i32Data [in]: 設置對應位IO口，對應位為1才被清零。設置值範圍：0x00~0xFF。

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

0 : 設置成功

其他：設置失敗

- 函數用法

/\* 清除設定PT6.1、PT6.4，所以設定參數0x12\*/

DrvGPIO\_LCDIOClrPorts(E\_PT6,0x12); // 0x40850[17]=0b, 0x40858[1]=0b

### 5.3.17. DrvGPIO\_LCDIOSetBit

- 函數

unsigned char DrvGPIO\_LCDIOSetBit (E\_DRVGPIO\_PORT uport, unsigned int i32Bit)

- 函數功能

設置GPIO PT6~PT13對應的IO口輸出1。

PT6寄存器0x40850[17][1] / 0x40854[17][1] / 0x40858[17][1] / 0x4085C[17][1]

PT7寄存器0x40860[17][1] / 0x40864[17][1] / 0x40868[17][1] / 0x4086C[17][1]

PT8寄存器0x40870[17][1] / 0x40874[17][1] / 0x40878[17][1] / 0x4087C[17][1]

PT9寄存器0x40880[17][1] / 0x40884[17][1] / 0x40888[17][1] / 0x4088C[17][1]

PT13寄存器0x408C0[17][1] / 0x408C4[17][1] / 0x408C8[17][1]

- 輸入參數

uport [in]: 代表GPIO port。設置值範圍：0~4

0 : PT6    1 : PT7

2 : PT8    3 : PT9    4 : PT13

i32Bit [in]: 代表GPIO的每一位IO口。設置值範圍：0~7。

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

0 : 設置成功

其他：設置失敗

- 函數用法

/\* 設置PT6.0 作為輸出模式\*/

DrvGPIO\_LCDIOOpen(E\_PT6,1, E\_IO\_OUTPUT); //設置PT6.0作為輸出模式, 0x40850[3]=1b

/\* 設定PT6.0輸出1 \*/

DrvGPIO\_LCDIOSetBit(E\_PT6,0); //設定PT6.0輸出1, 0x40850[1]=1b

### 5.3.18. DrvGPIO\_LCDIOClrBit

- 函數

unsigned char DrvGPIO\_LCDIOClrBit (E\_DRVGPIO\_PORT uport, unsigned int i32Bit)

- 函數功能

設置PT6~PT9對應任何一位的IO口輸出狀態為 0。

PT6寄存器0x40850[17][1] / 0x40854[17][1] / 0x40858[17][1] / 0x4085C[17][1]

PT7寄存器0x40860[17][1] / 0x40864[17][1] / 0x40868[17][1] / 0x4086C[17][1]

PT8寄存器0x40870[17][1] / 0x40874[17][1] / 0x40878[17][1] / 0x4087C[17][1]

PT9寄存器0x40880[17][1] / 0x40884[17][1] / 0x40888[17][1] / 0x4088C[17][1]

PT13寄存器0x408C0[17][1] / 0x408C4[17][1] / 0x408C8[17][1]

- 輸入參數

uport [in] :代表GPIO port. 設置值範圍 : 0~4

0 : PT6    1 : PT7

2 : PT8    3 : PT9    4 : PT13

i32Bit [in] :代表GPIO的每一位IO口. 設置值範圍 : 0~7..

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

0 : 設置成功

1 : 設置失敗

- 函數用法

/\* 設定PT6.0輸出0 \*/

DrvGPIO\_LCDIOClrBit(E\_PT6,0); //設定PT6.0輸出0, 0x40850[1]=0b

### 5.3.19. DrvGPIO\_LCDIOGetBit

- 函數

unsigned int DrvGPIO\_LCDIOGetBit (E\_DRVGPIO\_PORT port, uint8\_t u32Bit)

- 函數功能

讀取GPIO PT6~PT13任何一位IO口的輸入狀態值。

PT6寄存器0x40850[16][0] / 0x40854[16][0] / 0x40858[16][0] / 0x4085C[16][0]

PT7寄存器0x40860[16][0] / 0x40864[16][0] / 0x40868[16][0] / 0x4086C[16][0]

PT8寄存器0x40870[16][0] / 0x40874[16][0] / 0x40878[16][0] / 0x4087C[16][0]

PT9寄存器0x40880[16][0] / 0x40884[16][0] / 0x40888[16][0] / 0x4088C[16][0]

PT13寄存器0x408C0[16][0] / 0x408C4[16][0] / 0x408C8[16][0]

- 輸入參數

uport [in] :代表GPIO port. 設置值範圍 : 0~4

0 : PT6    1 : PT7

2 : PT8    3 : PT9    4 : PT13

u32Bit [in] :代表GPIO port任何一位IO口. 設置值範圍 : 0、1、2、3、4、5、6、7.

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

0/1 : IO pin的輸入狀態

0xff : 讀取失敗

- 函數用法

/\* 讀取PT6.0~PT6.7的輸入狀態值\*/

uint32\_t i32Bit;

i32Bit = DrvGPIO\_LCDIOGetBit(E\_PT6,0); //讀取PT6.0的輸入狀態值, read 0x40850[0]

i32Bit = DrvGPIO\_LCDIOGetBit(E\_PT6,1); //讀取PT6.1的輸入狀態值, read 0x40850[16]

i32Bit= DrvGPIO\_LCDIOGetBit(E\_PT6,2); //讀取PT6.2的輸入狀態值, read 0x40854[0]

i32Bit= DrvGPIO\_LCDIOGetBit(E\_PT6,3); //讀取PT6.3的輸入狀態值, read 0x40854[16]

i32Bit= DrvGPIO\_LCDIOGetBit(E\_PT6,4); //讀取PT6.4的輸入狀態值, read 0x40858[0]

i32Bit= DrvGPIO\_LCDIOGetBit(E\_PT6,5); //讀取PT6.5的輸入狀態值, read 0x40858[16]

i32Bit= DrvGPIO\_LCDIOGetBit(E\_PT6,6); //讀取PT6.6的輸入狀態值, read 0x4085C[0]

i32Bit= DrvGPIO\_LCDIOGetBit(E\_PT6,7); //讀取PT6.6的輸入狀態值, read 0x4085C[16]

### 5.3.20. DrvGPIO\_EnableAnalogPin

- 函數

unsigned char DrvGPIO\_EnableAnalogPin(short port,unsigned int i32Bit)

- 函數功能

關閉GPIO任何一位IO引腳的數位工作模式, 如輸入/輸出/電阻上拉/中斷觸發沿, 開啟IO類比工作模式。

PT2寄存器 0x40810[23:16] / 0x40810[7:0] / 0x40814[23:16] /0x4081C[23:0]

PT3寄存器 0x40820[23:16] / 0x40820[7:0] / 0x40824[23:16] /0x4082C[23:0]

- 輸入參數

port [in] : 代表GPIO port. 設置值範圍 : 2~3.

1 : Rsv    2 : PT2

3 : PT3

u32Bit [in] :代表GPIO任何一位IO口引腳, 對應位的值為1表示關閉對應IO引腳工作模式, 為0時不做設置  
設置值範圍 : 0x00~0xFF.

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

0 : 設置成功

1 : 設置失敗

- 函數用法

```
/* 關閉PT3.1/PT3.3/PT3.5/PT3.7數位功能*/  
DrvGPIO_Open(E_PT3,0xAA,E_IO_INPUT);  
DrvGPIO_Open(E_PT3,0x55,E_IO_OUTPUT);  
DrvGPIO_Open(E_PT3,0xAA,E_IO_PullHigh);  
DrvGPIO_IntTrigger(E_PT3,0xAA,E_N_Edge);  
DrvGPIO_EnableAnalogPin(E_PT3,0xAA);
```

### 5.3.21. DrvGPIO\_PT2\_EnableINPUT

- 函數

void DrvGPIO\_PT2\_EnableINPUT(short int ubit)

- 函數功能

使能GPIO PT2任何一位IO引腳的輸入模式  
設置PT2寄存器0x40814[23:16]

- 輸入參數

ubit [in]: 代表GPIO任何一位IO口引腳，對應位的值為1表示打開對應IO引腳輸入模式，為0時不做設置；  
設置值範圍：0x00~0xff；

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

```
/* 設置PT2.0/PT2.1作為輸入模式*/  
DrvGPIO_PT2_EnableINPUT(0x01|0x02); //PT2.0/PT2.1打開輸入模式
```

### 5.3.22. DrvGPIO\_PT2\_DisableINPUT

- 函數

void DrvGPIO\_PT2\_DisableINPUT(short int ubit)

- 函數功能

關閉GPIO PT2任何一位IO引腳的輸入模式  
設置PT2寄存器0x40814[23:16]

- 輸入參數

ubit [in]: 代表GPIO任何一位IO口引腳，對應位的值為1表示關閉對應IO引腳輸入模式，為0時不做設置；  
設置值範圍：0x00~0xff;

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h



- 函數返回值

無

- 函數用法

*/\* 關閉PT2.0/PT2.1作為輸入模式\*/*

`DrvGPIO_PT2_DisableINPUT(0x01|0x02); //PT2.0/PT2.1關閉輸入模式`

### 5.3.23. DrvGPIO\_PT2\_EnablePullHigh

- 函數

`void DrvGPIO_PT2_EnablePullHigh(short int ubit)`

- 函數功能

使能GPIO PT2任何一位IO引腳的上拉電阻

設置PT2寄存器0x40810[23:16]

- 輸入參數

ubit [in] : 代表GPIO任何一位IO口引腳，對應位的值為1表示打開對應IO引腳上拉電阻，為0時不做設置；

設置值範圍：0x00~0xff；

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

*/\* 使能PT2.0/PT2.1 上拉電阻\*/*

`DrvGPIO_PT2_EnablePullHigh(0x01|0x02); //PT2.0/PT2.1打開上拉電阻`

### 5.3.24. DrvGPIO\_PT2\_DisablePullHigh

- 函數

`void DrvGPIO_PT2_DisablePullHigh(short int ubit)`

- 函數功能

關閉GPIO PT2任何一位IO引腳的上拉電阻

設置PT2寄存器0x40810[23:16]

- 輸入參數

ubit [in] : 代表GPIO任何一位IO口引腳，對應位的值為1表示關閉對應IO引腳上拉電阻，為0時不做設置；

設置值範圍：0x00~0xff；

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

```
/* 關閉PT2.0/PT2.1上拉電阻*/
```

```
DrvGPIO_PT2_DisablePullHigh(0x01|0x02); //PT2.0/PT2.1關閉上拉電阻
```

### 5.3.25. DrvGPIO\_PT2\_EnableOUTPUT

- 函數

```
void DrvGPIO_PT2_EnableOUTPUT(short int ubit)
```

- 函數功能

使能GPIO PT2任何一位IO引腳的輸出模式

設置PT2寄存器0x40810[7:0]

- 輸入參數

ubit [in] :代表 GPIO 任何一位IO 口引腳，對應位的值為1表示打開對應IO引腳輸出模式，為0時不做設置；

設置值範圍：0x00~0xff

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

```
/* 使能PT2.0/PT2.1 輸出模式*/
```

```
DrvGPIO_PT2_EnableOUTPUT(0x01|0x02); //PT2.0/PT2.1打開輸出模式
```

### 5.3.26. DrvGPIO\_PT2\_DisableOUTPUT

- 函數

```
void DrvGPIO_PT2_DisableOUTPUT(short int ubit)
```

- 函數功能

關閉GPIO PT2任何一位IO引腳的輸出模式

設置PT2寄存器0x40810[7:0]

- 輸入參數

ubit [in] :代表GPIO任何一位IO 口引腳，對應位的值為1表示關閉對應IO引腳輸出模式，為0時不做設置；

設置值範圍：0x00~0xff

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

```
/* 關閉PT2.0/PT2.1 輸出模式*/
```

```
DrvGPIO_PT2_DisableOUTPUT(0x01|0x02); //PT2.0/PT2.1關閉輸出模式
```

### 5.3.27. DrvGPIO\_PT2\_EnableINT

- 函數

void DrvGPIO\_PT2\_EnableINT(short int ubit)

- 函數功能

使能GPIO PT2任何一位IO引腳的外部中斷功能。

設置PT2寄存器0x40014[23:16]

- 輸入參數

ubit [in]: 代表GPIO任何一位IO 口引腳，對應位的值為1表示打開對應IO引腳外部中斷功能，為0時不做設置；

設置值範圍：0x00~0xff

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

```
/* 使能PT2.0/PT2.1 外部中斷功能*/
```

```
DrvGPIO_PT2_EnableINT(0x01|0x02); //PT2.0/PT2.1打開外部中斷功能
```

### 5.3.28. DrvGPIO\_PT2\_DisableINT

- 函數

void DrvGPIO\_PT2\_DisableINT(short int ubit)

- 函數功能

關閉GPIO PT2任何一位IO引腳的外部中斷功能。

設置PT2寄存器0x40014[23:16]

- 輸入參數

ubit [in]: 代表GPIO任何一位IO口引腳，對應位的值為1表示關閉對應IO引腳外部中斷功能，為0時不做設置

設置值範圍：0x00~0xff

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

```
/* 關閉PT2.0/PT2.1 外部中斷功能*/
```

```
DrvGPIO_PT2_DisableINT(0x01|0x02); //PT2.0/PT2.1關閉外部中斷功能
```

### 5.3.29. DrvGPIO\_PT2\_IntTriggerPorts

- 函數

`void DrvGPIO_PT2_IntTriggerPorts(uint32_t i32Bit, uint32_t mode)`

- 函數功能

使能GPIO PT2的外部中斷觸發沿並設置外部中斷的觸發沿模式。

設置GPIO寄存器0x4081C[31:0]

- 輸入參數

`u32Bit [in]` :代表GPIO port的每一位IO口，對應位為1表示該位IO被設置，輸入0時，觸發功能無效。

設置值範圍：0x00~0xff

`mode [in]` : IO口的中斷觸發模式選擇。設置值範圍：0~7

0：關閉IO 外部中斷觸發	1：上升沿觸發	2：下降沿觸發	3：電平變化觸發
4：低電平觸發	5：高電平觸發	6：低電平觸發	7：高電平觸發。

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

*/\* 設置PT2.0中斷觸發模式為下降沿觸發\*/*

```
DrvGPIO_ClkGenerator(0,1); //設置IO采样频率
```

```
DrvGPIO_PT2_EnableINT(0x1); //使能IO外部中斷, PT2.0
```

```
DrvGPIO_PT2_IntTriggerPorts(0x1,E_N_Edge); //設置中斷觸發模式, PT2.0
```

### 5.3.30. DrvGPIO\_PT2\_IntTriggerBit

- 函數

`void DrvGPIO_PT2_IntTriggerBit(uint32_t i32Bit, uint32_t mode)`

- 函數功能

使能GPIO PT2被選中引腳的外部中斷觸發沿並設置外部中斷的觸發沿模式。

設置GPIO寄存器0x4081C[31:0]

- 輸入參數

`u32Bit [in]` : 代表GPIO port的bit7~bit0，選中的引腳才被設置。設置值範圍：0~7。

`mode [in]` : IO口的中斷觸發模式選擇。設置值範圍：0~7

0：關閉IO 外部中斷觸發	1：上升沿觸發	2：下降沿觸發	3：電平變化觸發
4：低電平觸發	5：高電平觸發	6：低電平觸發	7：高電平觸發。

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

```
/* 設置PT2.0中斷觸發模式為下降沿觸發*/  
DrvGPIO_PT2_IntTriggerBit(0, E_N_Edge); //設置中斷觸發模式, PT2.0  
DrvGPIO_PT2_IntTriggerBit(1, E_N_Edge); //設置中斷觸發模式, PT2.1
```

### 5.3.31. DrvGPIO\_PT2\_GetIntFlag

- 函數

```
unsigned char DrvGPIO_PT2_GetIntFlag(void)
```

- 函數功能

讀取對應GPIO PT2的中斷標誌位元，返回寄存器的值，返回值的對應位為1表示該位的IO口發生中斷，若為0，則表示沒有中斷產生。

讀取中斷寄存器0x40014[7:0]的值。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

返回值是 PT2 的中斷標誌位元值: 0x00 ~ 0xFF

- 函數用法

```
/* 讀取 PT2 外部中斷標誌位元 */  
unsigned char flag ;  
flag=DrvGPIO_PT2_GetIntFlag(); //read 0x40014[7:0]
```

### 5.3.32. DrvGPIO\_PT2\_ClearIntFlag

- 函數

```
void DrvGPIO_PT2_ClearIntFlag(short int uint32)
```

- 函數功能

清除GPIO PT2外部中斷標誌位元；

清零中斷寄存器0x40014[7:0]。

- 輸入參數

u32Bit [in]：代表GPIO port的每一位IO，對應位為1的才會被清零。設置值範圍 :: 0x00~0xFF；  
設定值的每一位對應一位IO pin，對應位為1的IO 口的標誌位元就被清零。

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

```
/* 清零 PT2.2 interrupt flag */  
DrvGPIO_PT2_ClearIntFlag(0x04);
```

```
/*清零 PT2.3 interrupt flag*/  
DrvGPIO_PT2_ClearIntFlag(0x08);
```

### 5.3.33. DrvGPIO\_PT2\_GetPortBits

- 函數

unsigned char DrvGPIO\_PT2\_GetPortBits (void)

- 函數功能

讀取GPIO PT2輸入狀態值. 讀取GPIO的輸入狀態寄存器0x40818[7:0]

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

0x00~ 0xFF : 待讀取GPIO PORT的輸入狀態值:

- 函數用法

```
/*讀取PT2的輸入狀態值*/
```

```
uint32_t i32Port;
```

```
i32Port = DrvGPIO_PT2_GetPortBits();
```

### 5.3.34. DrvGPIO\_PT2\_SetPortBits

- 函數

void DrvGPIO\_PT2\_SetPortBits (unsigned char ui32Data)

- 函數功能

設置GPIO PT2對應IO口的輸出狀態.

設置GPIO的輸出狀態寄存器0x40814[7:0]

- 輸入參數

i32Data [in] : bit7~bit0對應每一位IO PIN · 對應位為1則會被置1 · 對應位為0則會被置0. 設定值範圍:  
0x00~0xFF.

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

```
/* 設定PT2.2、PT2.4為1 · 所以設定參數0x14 */
```

```
DrvGPIO_PT2_SetPortBits(0x14);
```

### 5.3.35. DrvGPIO\_PT2\_ClrPortBits

- 函數

void DrvGPIO\_PT2\_ClrPortBits (unsigned int ui32Data)

- 函數功能

清除GPIO PT2 對應位IO口輸出狀態值.

清零GPIO的輸出狀態寄存器0x40814[7:0]

- 輸入參數

i32Data [in] : bit7~bit0對應每一位IO PIN · 對應位為1輸出才被置0. 設定值範圍: 0x00~0xFF.

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

/\* 清除PT2.1/PT2.4的輸出為0 · 設定輸入參數0x12 \*/

DrvGPIO\_PT2\_ClrPortBits(0x12);

### 5.3.36. DrvGPIO\_PT3\_EnableINPUT

- 函數

void DrvGPIO\_PT3\_EnableINPUT(short int ubit)

- 函數功能

使能GPIO PT3任何一位IO引腳的輸入模式

設置PT3寄存器0x40824[23:16]

- 輸入參數

ubit [in] : 代表 GPIO 任何一位IO 口引腳 · 對應位的值為1表示打開對應IO引腳輸入模式 · 為0時不做設置 ;

設置值範圍 : 0x00~0xFF ;

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

/\* 設置PT3.0/PT3.1 作為輸入模式\*/

DrvGPIO\_PT3\_EnableINPUT(0x01|0x02); //PT3.0/PT3.1打開輸入模式

### 5.3.37. DrvGPIO\_PT3\_DisableINPUT

- 函數

void DrvGPIO\_PT3\_DisableINPUT(short int ubit)

- 函數功能

關閉GPIO PT3任何一位IO引腳的輸入模式

設置PT3寄存器0x40824[23:16]

- 輸入參數

ubit [in]: 代表 GPIO 任何一位IO 口引腳, 對應位的值為1表示關閉對應IO引腳輸入模式, 為0時不做設置;

設置值範圍: 0x00~0xFF;

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

```
/* 關閉PT3.0/PT3.1 作為輸入模式*/
```

```
DrvGPIO_PT3_DisableINPUT(0x01|0x02); //PT3.0/PT3.1關閉輸入模式
```

### 5.3.38. DrvGPIO\_PT3\_EnablePullHigh

- 函數

void DrvGPIO\_PT3\_EnablePullHigh(short int ubit)

- 函數功能

使能GPIO PT3任何一位IO引腳的上拉電阻

設置PT3寄存器0x40820[23:16]

- 輸入參數

ubit [in]: 代表 GPIO 任何一位IO 口引腳, 對應位的值為1表示打開對應IO引腳上拉電阻, 為0時不做設置;

設置值範圍: 0x00~0xFF;

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

```
/* 使能PT3.0/PT3.1 上拉電阻*/
```

```
DrvGPIO_PT3_EnablePullHigh(0x01|0x02); //PT3.0/PT3.1打開上拉電阻
```

### 5.3.39. DrvGPIO\_PT3\_DisablePullHigh

- 函數

void DrvGPIO\_PT3\_DisablePullHigh(short int ubit)

- 函數功能



關閉GPIO PT3任何一位IO引腳的上拉電阻

設置PT3寄存器0x40820[23:16]

- **輸入參數**

ubit [in]: 代表 GPIO 任何一位IO 口引腳, 對應位的值為1表示關閉對應IO引腳上拉電阻, 為0時不做設置;

設置值範圍: 0x00~0xFF;

- **包含標頭檔**

Peripheral\_lib/DrvGPIO.h

- **函數返回值**

無

- **函數用法**

```
/* 關閉PT3.0/PT3.1 上拉電阻*/
```

```
DrvGPIO_PT3_DisablePullHigh(0x01|0x02); //PT3.0/PT3.1關閉上拉電阻
```

### 5.3.40. DrvGPIO\_PT3\_EnableOUTPUT

- **函數**

```
void DrvGPIO_PT3_EnableOUTPUT(short int ubit)
```

- **函數功能**

使能GPIO PT3任何一位IO引腳的輸出模式

設置PT3寄存器0x40820[7:0]

- **輸入參數**

ubit [in]: 代表GPIO 任何一位IO 口引腳, 對應位的值為1表示打開對應IO引腳輸出模式, 為0時不做設置;

設置值範圍: 0x00~0xFF;

- **包含標頭檔**

Peripheral\_lib/DrvGPIO.h

- **函數返回值**

無

- **函數用法**

```
/* 使能PT3.0/PT3.1 輸出模式*/
```

```
DrvGPIO_PT3_EnableOUTPUT(0x01|0x02); //PT3.0/PT3.1打開輸出模式
```

### 5.3.41. DrvGPIO\_PT3\_DisableOUTPUT

- **函數**

```
void DrvGPIO_PT3_DisableOUTPUT(short int ubit)
```

- **函數功能**

關閉GPIO PT3任何一位IO引腳的輸出模式

設置PT3寄存器0x40820[7:0]

- **輸入參數**

ubit [in] : 代表 GPIO 任何一位IO 口引腳，對應位的值為1表示關閉對應IO引腳輸出模式，為0時不做設置；  
設置值範圍：0x00~0xFF；

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

```
/* 關閉PT3.0/PT3.1 輸出模式*/
```

```
DrvGPIO_PT3_DisableOUTPUT(0x01|0x02); //PT3.0/PT3.1關閉輸出模式
```

### 5.3.42. DrvGPIO\_PT3\_GetPortBits

- 函數

```
unsigned char DrvGPIO_PT3_GetPortBits(void)
```

- 函數功能

讀取GPIO PT3輸入狀態值。讀取GPIO的輸入狀態寄存器0x40828[7:0]

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

0 ~ 0xFF：待讀取GPIO PORT的輸入狀態值：

- 函數用法

```
/*讀取PT3的輸入狀態值*/
```

```
uint32_t i32Port;
```

```
i32Port = DrvGPIO_PT3_GetPortBits();
```

### 5.3.43. DrvGPIO\_PT3\_SetPortBits

- 函數

```
void DrvGPIO_PT3_SetPortBits (unsigned char ui32Data)
```

- 函數功能

設置GPIO PT3對應IO口的輸出狀態。

設置GPIO的輸出狀態寄存器0x40824[7:0]

- 輸入參數

i32Data [in] : bit7~bit0對應每一位IO PIN，對應位為1則會被置1，對應位為0則會被置0。設定值範圍：  
0x00~0xFF

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

/\* 設定PT3.2、PT3.4為1，所以設定參數0x14 \*/

DrvGPIO\_PT3\_SetPortBits(0x14);

### 5.3.44. DrvGPIO\_PT3\_ClrPortBits

- 函數

void DrvGPIO\_PT3\_ClrPortBits (unsigned int ui32Data)

- 函數功能

清除GPIO PT3 對應位IO口輸出狀態值.

清零GPIO的輸出狀態寄存器0x40824[7:0]

- 輸入參數

i32Data [in] : bit7~bit0對應每一位IO PIN，對應位為1輸出才被置0. 設定範圍：0x00~0xFF.

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

/\* 清除PT3.1/PT3.4的輸出為0，設定輸入參數 0x12 \*/

DrvGPIO\_PT3\_ClrPortBits(0x12);

### 5.3.45. DrvGPIO\_PT6\_EnableINPUT

- 函數

void DrvGPIO\_PT6\_EnableINPUT(short int ubit)

- 函數功能

使能GPIO PT6任何一位IO引腳的輸入模式

設置PT6寄存器0x40850[18][2] / 0x40854[18][2] / 0x40858[18][2] / 0x4085C[18][2]

- 輸入參數

ubit [in] 代表 GPIO 任何一位IO 口引腳，對應位的值為1表示打開對應IO引腳輸入模式，為0時不做設置；

設置值範圍：0x00~0xFF；

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

/\* 設置PT6.0/PT6.1 作為輸入模式\*/

DrvGPIO\_PT6\_EnableINPUT(0x01|0x02); //PT6.0/PT6.1打開輸入模式

### 5.3.46. DrvGPIO\_PT6\_DisableINPUT

- 函數

void DrvGPIO\_PT6\_DisableINPUT(short int ubit)

- 函數功能

關閉GPIO PT6任何一位IO引腳的輸入模式

設置PT6寄存器0x40850[18][2] / 0x40854[18][2] / 0x40858[18][2] / 0x4085C[18][2]

- 輸入參數

ubit [in]: 代表 GPIO 任何一位IO 口引腳, 對應位的值為1表示關閉對應IO引腳輸入模式, 為0時不做設置;

設置值範圍: 0x00~0xFF;

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

/\* 關閉PT6.0/PT6.1 作為輸入模式\*/

DrvGPIO\_PT6\_DisableINPUT(0x01|0x02); //PT6.0/PT6.1關閉輸入模式

### 5.3.47. DrvGPIO\_PT6\_EnableOUTPUT

- 函數

void DrvGPIO\_PT6\_EnableOUTPUT(short int ubit)

- 函數功能

使能GPIO PT6任何一位IO引腳的輸出模式

設置PT6寄存器0x40850[19][3] / 0x40854[19][3] / 0x40858[19][3] / 0x4085C[19][3]

- 輸入參數

ubit [in]: 代表 GPIO 任何一位IO 口引腳, 對應位的值為1表示打開對應IO引腳輸出模式, 為0時不做設置;

設置值範圍: 0x00~0xFF;

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

/\* 使能PT6.0/PT6.1 輸出模式\*/

DrvGPIO\_PT6\_EnableOUTPUT(0x01|0x02); //PT6.0/PT6.1打開輸出模式

### 5.3.48. DrvGPIO\_PT6\_DisableOUTPUT

- 函數

void DrvGPIO\_PT6\_DisableOUTPUT(short int ubit)

- 函數功能

關閉GPIO PT6任何一位IO引腳的輸出模式

設置PT6寄存器0x40850[19][3] / 0x40854[19][3] / 0x40858[19][3] / 0x4085C[19][3]

- 輸入參數

ubit [in]: 代表 GPIO 任何一位IO 口引腳, 對應位的值為1表示關閉對應IO引腳輸出模式, 為0時不做設置;

設置值範圍: 0x00~0xFF;

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

/\* 關閉PT6.0/PT6.1 輸出模式\*/

DrvGPIO\_PT6\_DisableOUTPUT(0x01|0x02); //PT6.0/PT6.1關閉輸出模式

### 5.3.49. DrvGPIO\_PT6\_GetPortBits

- 函數

unsigned char DrvGPIO\_PT6\_GetPortBits(void)

- 函數功能

讀取GPIO PT6輸入狀態值.

讀取PT6寄存器0x40850[16][0] / 0x40854[16][0] / 0x40858[16][0] / 0x4085C[16][0]

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

0x00 ~ 0xFF: 待讀取GPIO PORT的輸入狀態值:

- 函數用法

/\*讀取PT6的輸入狀態值\*/

uint32\_t i32Port;

i32Port = DrvGPIO\_PT6\_GetPortBits();

### 5.3.50. DrvGPIO\_PT6\_SetPortBits

- 函數

void DrvGPIO\_PT6\_SetPortBits (unsigned char ui32Data)

- 函數功能

設置GPIO PT6對應IO口的輸出狀態.

設置PT6寄存器0x40850[17][1] / 0x40854[17][1] / 0x40858[17][1] / 0x4085C[17][1]

- 輸入參數

i32Data [in] : bit7~bit0對應每一位IO PIN，對應位為1則會被置1，對應位為0則會被置0. 設定值範圍：  
0x00~0xFF.

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

/\* 設定PT6.2、PT6.4為1，所以設定參數0x14 \*/

DrvGPIO\_PT6\_SetPortBits(0x14);

### 5.3.51. DrvGPIO\_PT6\_ClrPortBits

- 函數

void DrvGPIO\_PT6\_ClrPortBits (unsigned int ui32Data)

- 函數功能

清除GPIO PT6 對應位IO口輸出狀態值.

清零PT6寄存器0x40850[17][1] / 0x40854[17][1] / 0x40858[17][1] / 0x4085C[17][1]

- 輸入參數

i32Data [in] : bit7~bit0對應每一位IO PIN，對應位為1輸出才被置0. 設定範圍：0x00~0xFF.

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

/\* 清除PT6.1/PT6.4的輸出為0，設定輸入參數 0x12 \*/

DrvGPIO\_PT6\_ClrPortBits(0x12);

### 5.3.52. DrvGPIO\_PT7\_EnableINPUT

- 函數

void DrvGPIO\_PT7\_EnableINPUT(short int ubit)

- 函數功能

使能GPIO PT7任何一位IO引腳的輸入模式

設置PT7寄存器0x40860[18][2] / 0x40864[18][2] / 0x40868[18][2] / 0x4086C[18][2]

- 輸入參數

ubit [in]: 代表 GPIO任何一位IO 口引腳，對應位的值為1表示打開對應IO引腳輸入模式，為0時不做設置；

設置值範圍：0x00~0xFF；

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

```
/* 設置PT7.0/PT7.1 作為輸入模式*/
```

```
DrvGPIO_PT7_EnableINPUT(0x01|0x02); //PT7.0/PT7.1打開輸入模式
```

### 5.3.53. DrvGPIO\_PT7\_DisableINPUT

- 函數

void DrvGPIO\_PT7\_DisableINPUT(short int ubit)

- 函數功能

關閉GPIO PT7任何一位IO引腳的輸入模式

設置PT7寄存器0x40860[18][2] / 0x40864[18][2] / 0x40868[18][2] / 0x4086C[18][2]

- 輸入參數

ubit [in] 代表 GPIO 任何一位IO 口引腳，對應位的值為1表示關閉對應IO引腳輸入模式，為0時不做設置；

設置值範圍：0x00~0xFF；

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

```
/* 關閉PT7.0/PT7.1 作為輸入模式*/
```

```
DrvGPIO_PT7_DisableINPUT(0x01|0x02); //PT7.0/PT7.1關閉輸入模式
```

### 5.3.54. DrvGPIO\_PT7\_EnableOUTPUT

- 函數

void DrvGPIO\_PT7\_EnableOUTPUT(short int ubit)

- 函數功能

使能GPIO PT7任何一位IO引腳的輸出模式

設置PT7寄存器0x40860[19][3]/ 0x40864[19][3]0x40868[19][3] / 0x4086C[19][3]

- 輸入參數

ubit [in]: 代表 GPIO任何一位IO 口引腳，對應位的值為1表示打開對應IO引腳輸出模式，為0時不做設置；

設置值範圍：0x00~0xff；

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

/\* 使能PT7.0/PT7.1 輸出模式\*/

DrvGPIO\_PT7\_EnableOUTPUT(0x01|0x02); //PT7.0/PT7.1打開輸出模式

### 5.3.55. DrvGPIO\_PT7\_DisableOUTPUT

- 函數

void DrvGPIO\_PT7\_DisableOUTPUT(short int ubit)

- 函數功能

關閉GPIO PT7任何一位IO引腳的輸出模式

設置PT7寄存器0x40860[19][3]/ 0x40864[19][3]0x40868[19][3] / 0x4086C[19][3]

- 輸入參數

ubit [in]: 代表 GPIO 任何一位IO 口引腳，對應位的值為1表示關閉對應IO引腳輸出模式，為0時不做設置；

設置值範圍：0x00~0xFF；

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

/\* 關閉PT7.0/PT7.1 輸出模式\*/

DrvGPIO\_PT7\_DisableOUTPUT(0x01|0x02); //PT7.0/PT7.1關閉輸出模式

### 5.3.56. DrvGPIO\_PT7\_GetPortBits

- 函數

unsigned char DrvGPIO\_PT7\_GetPortBits(void)

- 函數功能

讀取GPIO PT7輸入狀態值。



讀取PT7寄存器0x40860[16][0]/ 0x40864[16][0]/ 0x40868[16][0] / 0x4086C[16][0]

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

0 ~ 0xFF : 待讀取GPIO PORT的輸入狀態值:

- 函數用法

/\*讀取PT7的輸入狀態值\*/

```
uint32_t i32Port;
```

```
i32Port = DrvGPIO_PT7_GetPortBits();
```

### 5.3.57. DrvGPIO\_PT7\_SetPortBits

- 函數

```
void DrvGPIO_PT7_SetPortBits (unsigned char ui32Data)
```

- 函數功能

設置GPIO PT7對應IO口的輸出狀態.

設置PT7寄存器0x40860[17][1]/ 0x40864[17][1]/ 0x40868[17][1] / 0x4086C[17][1]

- 輸入參數

i32Data [in] : bit7~bit0對應每一位IO PIN，對應位為1則會被置1，對應位為0則會被置0. 設定值範圍：  
0x00~0xFF.

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

/\* 設定PT7.2、PT7.4為1，所以設定參數0x14 \*/

```
DrvGPIO_PT7_SetPortBits(0x14);
```

### 5.3.58. DrvGPIO\_PT7\_ClrPortBits

- 函數

```
void DrvGPIO_PT7_ClrPortBits(unsigned int ui32Data)
```

- 函數功能

清除GPIO PT7 對應位IO口輸出狀態值.

清零PT7寄存器0x40860[17][1] / 0x40864[17][1] / 0x40868[17][1] / 0x4086C[17][1]

- 輸入參數

i32Data [in] : bit7~bit0對應每一位IO PIN，對應位為1輸出才被置0。設定範圍：0x00~0xFF。

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

/\* 清除PT7.1/PT7.4的輸出為0，設定輸入參數 0x12 \*/

DrvGPIO\_PT7\_ClrPortBits(0x12);

### 5.3.59. DrvGPIO\_PT8\_EnableINPUT

- 函數

void DrvGPIO\_PT8\_EnableINPUT(short int ubit)

- 函數功能

使能GPIO PT8任何一位IO引腳的輸入模式

設置PT8寄存器0x40870[18][2] / 0x40874[18][2] / 0x40878[18][2] / 0x4087C[18][2]

- 輸入參數

ubit [in] : 代表 GPIO 任何一位IO 口引腳，對應位的值為1表示打開對應IO引腳輸入模式，為0時不做設置；

設置值範圍：0x00~0xFF；

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

/\* 設置PT8.0/PT8.1 作為輸入模式\*/

DrvGPIO\_PT8\_EnableINPUT(0x01|0x02); //PT8.0/PT8.1打開輸入模式

### 5.3.60. DrvGPIO\_PT8\_DisableINPUT

- 函數

void DrvGPIO\_PT8\_DisableINPUT(short int ubit)

- 函數功能

關閉GPIO PT8任何一位IO引腳的輸入模式

設置PT7寄存器0x40870[18][2] / 0x40874[18][2] / 0x40878[18][2] / 0x4087C[18][2]

- 輸入參數

ubit [in] : 代表 GPIO 任何一位IO 口引腳，對應位的值為1表示關閉對應IO引腳輸入模式，為0時不做設置；

設置值範圍：0x00~0xFF；

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

/\* 關閉PT8.0/PT8.1 作為輸入模式\*/

DrvGPIO\_PT8\_DisableINPUT(0x01|0x02); //PT8.0/PT8.1關閉輸入模式

### 5.3.61. DrvGPIO\_PT8\_EnableOUTPUT

- 函數

void DrvGPIO\_PT8\_EnableOUTPUT(short int ubit)

- 函數功能

使能GPIO PT8任何一位IO引腳的輸出模式

設置PT8寄存器0x40870[19][3]/ 0x40874[19][3] / 0x40878[19][3] / 0x4087C[19][3]

- 輸入參數

ubit [in]: 代表 GPIO 任何一位IO 口引腳·對應位的值為1表示打開對應IO引腳輸出模式·為0時不做設置;  
設置值範圍: 0x00~0xff;

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

/\* 使能PT8.0/PT8.1 輸出模式\*/

DrvGPIO\_PT8\_EnableOUTPUT(0x01|0x02); //PT8.0/PT8.1打開輸出模式

### 5.3.62. DrvGPIO\_PT8\_DisableOUTPUT

- 函數

void DrvGPIO\_PT8\_DisableOUTPUT(short int ubit)

- 函數功能

關閉GPIO PT8任何一位IO引腳的輸出模式

設置PT8寄存器0x40870[19][3] / 0x40874[19][3] / 0x40878[19][3] / 0x4087C[19][3]

- 輸入參數

ubit [in]: 代表GPIO 任何一位IO 口引腳·對應位的值為1表示關閉對應IO引腳輸出模式·為0時不做設置;  
設置值範圍: 0x00~0xff;

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

```
/* 關閉PT8.0/PT8.1 輸出模式*/  
DrvGPIO_PT8_DisableOUTPUT(0x01|0x02); //PT8.0/PT8.1關閉輸出模式
```

### 5.3.63. DrvGPIO\_PT8\_GetPortBits

- 函數

unsigned char DrvGPIO\_PT8\_GetPortBits (void)

- 函數功能

讀取GPIO PT8輸入狀態值.

讀取PT8寄存器0x40870[16][0] / 0x40874[16][0] / 0x40878[16][0] / 0x4087C[16][0]

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

0 ~ 0xFF : 待讀取GPIO PORT的輸入狀態值:

- 函數用法

```
/*讀取PT8的輸入狀態值*/
```

```
uint32_t i32Port;
```

```
i32Port = DrvGPIO_PT8_GetPortBits();
```

### 5.3.64. DrvGPIO\_PT8\_SetPortBits

- 函數

void DrvGPIO\_PT8\_SetPortBits (unsigned char ui32Data)

- 函數功能

設置GPIO PT8對應IO口的輸出狀態.

設置PT8寄存器0x40870[17][1] / 0x40874[17][1] / 0x40878[17][1] / 0x4087C[17][1]

- 輸入參數

i32Data [in] : bit7~bit0對應每一位IO PIN，對應位為1則會被置1，對應位為0則會被置0. 設定值範圍：  
0x00~0xFF.

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

```
/* 設定PT8.2、PT8.4為1，所以設定參數0x14 */
```

```
DrvGPIO_PT8_SetPortBits(0x14);
```

### 5.3.65. DrvGPIO\_PT8\_ClrPortBits

- 函數

```
void DrvGPIO_PT8_ClrPortBits (unsigned int ui32Data)
```

- 函數功能

清除GPIO PT8 對應位IO口輸出狀態值。

清零PT8寄存器0x40870[17][1] / 0x40874[17][1] / 0x40878[17][1] / 0x4087C[17][1]

- 輸入參數

i32Data [in] : bit7~bit0對應每一位IO PIN · 對應位為1輸出才被置0. 設定範圍是: 0x00~0xFF.

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

```
/* 清除PT8.1/PT8.4的輸出為0 · 設定輸入參數 0x12 */
```

```
DrvGPIO_PT8_ClrPortBits(0x12);
```

### 5.3.66. DrvGPIO\_PT9\_EnableINPUT

- 函數

```
void DrvGPIO_PT9_EnableINPUT(short int ubit)
```

- 函數功能

使能GPIO PT9任何一位IO引腳的輸入模式

設置PT9寄存器0x40880[18][2] / 0x40884[18][2] / 0x40888[18][2] / 0x4088C[18][2]

- 輸入參數

ubit [in] :代表 GPIO 任何一位IO口引腳 · 對應位的值為1表示打開對應IO引腳輸入模式 · 為0時不做設置 ;

設置值範圍 : 0x00~0xFF;

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

```
/* 設置PT9.0/PT9.1 作為輸入模式*/
```

```
DrvGPIO_PT9_EnableINPUT(0x01|0x02); //PT9.0/PT9.1打開輸入模式
```

### 5.3.67. DrvGPIO\_PT9\_DisableINPUT

- 函數

void DrvGPIO\_PT9\_DisableINPUT(short int ubit)

- 函數功能

關閉GPIO PT9任何一位IO引腳的輸入模式

設置PT9寄存器0x40880[18][2] / 0x40884[18][2] / 0x40888[18][2] / 0x4088C[18][2]

- 輸入參數

ubit [in]: 代表GPIO 任何一位IO 口引腳，對應位的值為1表示關閉對應IO引腳輸入模式，為0時不做設置；

設置值範圍：0x00~0xFF；

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

```
/* 關閉PT9.0/PT9.1 作為輸入模式*/
```

```
DrvGPIO_PT9_DisableINPUT(0x01|0x02); //PT9.0/PT9.1關閉輸入模式
```

### 5.3.68. DrvGPIO\_PT9\_EnableOUTPUT

- 函數

void DrvGPIO\_PT9\_EnableOUTPUT(short int ubit)

- 函數功能

使能GPIO PT9任何一位IO引腳的輸出模式

設置PT9寄存器0x40880[19][3] / 0x40884[19][3] / 0x40888[19][3] / 0x4088C[19][3]

- 輸入參數

ubit [in]: 代表 GPIO 任何一位IO 口引腳，對應位的值為1表示打開對應IO引腳輸出模式，為0時不做設置；

設置值範圍：0x00~0xFF；

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

```
/* 使能PT9.0/PT9.1 輸出模式*/
```

```
DrvGPIO_PT9_EnableOUTPUT(0x01|0x02); //PT9.0/PT9.1打開輸出模式
```

### 5.3.69. DrvGPIO\_PT9\_DisableOUTPUT

- 函數

void DrvGPIO\_PT9\_DisableOUTPUT(short int ubit)

- 函數功能

關閉GPIO PT9任何一位IO引腳的輸出模式

設置PT9寄存器0x40880[19][3]/ 0x40884[19][3]0x40888[19][3] / 0x4088C[19][3]

- 輸入參數

ubit [in]: 代表 GPIO 任何一位IO 口引腳, 對應位的值為1表示關閉對應IO引腳輸出模式, 為0時不做設置;

設置值範圍: 0x00~0xFF;

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

```
/* 關閉PT9.0/PT9.1 輸出模式*/
```

```
DrvGPIO_PT9_DisableOUTPUT(0x01|0x02); //PT9.0/PT9.1關閉輸出模式
```

### 5.3.70. DrvGPIO\_PT9\_GetPortBits

- 函數

unsigned char DrvGPIO\_PT9\_GetPortBits(void)

- 函數功能

讀取GPIO PT9輸入狀態值.

讀取PT9寄存器0x40880[16][0] / 0x40884[16][0] / 0x40888[16][0] / 0x4088C[16][0]

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

0 ~ 0xFF: 待讀取GPIO PORT的輸入狀態值:

- 函數用法

```
/*讀取PT9的輸入狀態值*/
```

```
uint32_t i32Port;
```

```
i32Port = DrvGPIO_PT9_GetPortBits();
```

### 5.3.71. DrvGPIO\_PT9\_SetPortBits

- 函數

void DrvGPIO\_PT9\_SetPortBits (unsigned char ui32Data)

- 函數功能

設置GPIO PT9對應IO口的輸出狀態.

設置PT9寄存器0x40880[17][1]/ 0x40884[17][1]/ 0x40888[17][1] / 0x4088C[17][1]

- 輸入參數

i32Data [in] : bit7~bit0對應每一位IO PIN，對應位為1則會被置1，對應位為0則會被置0。設定值範圍：  
0x00~0xFF.

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

/\* 設定PT9.2、PT9.4為1，所以設定參數0x14 \*/

DrvGPIO\_PT9\_SetPortBits(0x14);

### 5.3.72. DrvGPIO\_PT9\_ClrPortBits

- 函數

void DrvGPIO\_PT9\_ClrPortBits (unsigned int ui32Data)

- 函數功能

清除GPIO PT9 對應位IO口輸出狀態值。

清零PT9寄存器0x40880[17][1] / 0x40884[17][1] / 0x40888[17][1] / 0x4088C[17][1]

- 輸入參數

i32Data [in] : bit7~bit0對應每一位IO PIN，對應位為1輸出才被置0。設定範圍：0x00~0xFF.

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

/\* 清除PT9.1/PT9.4的輸出為0，設定輸入參數 0x12 \*/

DrvGPIO\_PT9\_ClrPortBits(0x12);

### 5.3.73. DrvGPIO\_PT13\_EnableINPUT

- 函數

void DrvGPIO\_PT13\_EnableINPUT(short int ubit)

- 函數功能

使能GPIO PT13任何一位IO引腳的輸入模式

設置PT13寄存器0x408C0[18][2] / 0x408C4[18][2] / 0x408C8[18][2]

- 輸入參數

ubit [in] :代表 GPIO 任何一位IO口引腳，對應位的值為1表示打開對應IO引腳輸入模式，為0時不做設置；

設置值範圍：0x00~0xFF;

- 包含標頭檔



Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

/\* 設置PT13.0/PT13.1 作為輸入模式\*/

DrvGPIO\_PT13\_EnableINPUT(0x01|0x02); //PT13.0/PT13.1打開輸入模式

### 5.3.74. DrvGPIO\_PT13\_DisableINPUT

- 函數

void DrvGPIO\_PT13\_DisableINPUT(short int ubit)

- 函數功能

關閉GPIO PT13任何一位IO引腳的輸入模式

設置PT9寄存器0x408C0[18][2] / 0x408C4[18][2] / 0x408C8[18][2]

- 輸入參數

ubit [in]: 代表GPIO 任何一位IO 口引腳，對應位的值為1表示關閉對應IO引腳輸入模式，為0時不做設置；

設置值範圍：0x00~0xFF；

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

/\* 關閉PT13.0/PT13.1 作為輸入模式\*/

DrvGPIO\_PT13\_DisableINPUT(0x01|0x02); //PT13.0/PT13.1關閉輸入模式

### 5.3.75. DrvGPIO\_PT13\_EnableOUTPUT

- 函數

void DrvGPIO\_PT13\_EnableOUTPUT(short int ubit)

- 函數功能

使能GPIO PT13任何一位IO引腳的輸出模式

設置PT13寄存器0x408C0[19][3] / 0x408C4[19][3] / 0x408C8[19][3]

- 輸入參數

ubit [in]: 代表 GPIO 任何一位IO 口引腳，對應位的值為1表示打開對應IO引腳輸出模式，為0時不做設置；

設置值範圍：0x00~0xFF；

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

```
/* 使能PT13.0/PT13.1 輸出模式*/
```

```
DrvGPIO_PT13_EnableOUTPUT(0x01|0x02); //PT13.0/PT13.1打開輸出模式
```

### 5.3.76. DrvGPIO\_PT13\_DisableOUTPUT

- 函數

```
void DrvGPIO_PT13_DisableOUTPUT(short int ubit)
```

- 函數功能

關閉GPIO PT13任何一位IO引腳的輸出模式

設置PT13寄存器0x408C0[19][3]/ 0x408C4[19][3]0x408C8[19][3]

- 輸入參數

ubit [in]：代表 GPIO 任何一位IO 口引腳，對應位的值為1表示關閉對應IO引腳輸出模式，為0時不做設置；  
設置值範圍：0x00~0xFF；

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

```
/* 關閉PT13.0/PT13.1 輸出模式*/
```

```
DrvGPIO_PT13_DisableOUTPUT(0x01|0x02); //PT13.0/PT13.1關閉輸出模式
```

### 5.3.77. DrvGPIO\_PT13\_GetPortBits

- 函數

```
unsigned char DrvGPIO_PT13_GetPortBits(void)
```

- 函數功能

讀取GPIO PT13輸入狀態值。

讀取PT13寄存器0x408C0[16][0] / 0x408C4[16][0] / 0x408C8[16][0]

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

0 ~ 0xFF：待讀取GPIO PORT的輸入狀態值：

- 函數用法

```
/*讀取PT13的輸入狀態值*/
```

```
uint32_t i32Port;
```

```
i32Port = DrvGPIO_PT13_GetPortBits();
```

### 5.3.78. DrvGPIO\_PT13\_SetPortBits

- 函數

void DrvGPIO\_PT13\_SetPortBits (unsigned char ui32Data)

- 函數功能

設置GPIO PT13對應IO口的輸出狀態.

設置PT13寄存器0x408C0[17][1]/ 0x408C4[17][1]/ 0x408C8[17][1]

- 輸入參數

i32Data [in] : bit7~bit0對應每一位IO PIN，對應位為1則會被置1，對應位為0則會被置0. 設定值範圍：  
0x00~0xFF.

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

/\* 設定PT13.2、PT13.4為1，所以設定參數0x14 \*/

DrvGPIO\_PT13\_SetPortBits(0x14);

### 5.3.79. DrvGPIO\_PT13\_ClrPortBits

- 函數

void DrvGPIO\_PT13\_ClrPortBits (unsigned int ui32Data)

- 函數功能

清除GPIO PT13 對應位IO口輸出狀態值.

清零PT13寄存器0x408C0[17][1] / 0x408C4[17][1] / 0x408C8[17][1]

- 輸入參數

i32Data [in] : bit7~bit0對應每一位IO PIN，對應位為1輸出才被置0. 設定範圍：0x00~0xFF.

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

無

- 函數用法

/\* 清除PT13.1/PT13.4的輸出為0，設定輸入參數 0x12 \*/

DrvGPIO\_PT13\_ClrPortBits(0x12);

### 5.3.80. DrvGPIO\_PortIDIF

- 函數

unsigned int DrvGPIO\_PortIDIF (uint32\_t port)

- 函數功能

讀取PT2/PT3對應位IO口作為外部中斷輸入口時，輸入狀態中斷條件旗標值。該條件旗標值取決中斷出發沿的觸發方式。使用外部中斷喚醒低功耗模式時，在進入低功耗模式前，可判斷該中斷條件旗標的值，確定按鍵是否處於中斷觸發方式的起始狀態。

讀取PT2寄存器0x4081C[31:24] / PT3寄存器0x4082C[31:24]。

- 輸入參數

port [in]：設定範圍是2~3

2: PT2 3: PT3.

- 包含標頭檔

Peripheral\_lib/DrvGPIO.h

- 函數返回值

返回值是0~0XFF，對應IO PORT的8位IO PIN的中斷條件旗標值。

- 函數用法

/\* 設定PT2.2作為外部中斷輸入，下降沿觸發，讀取PT2.2中斷條件旗標\*/

```
int i32Bit;
DrvGPIO_IntTrigger(E_PT2,0x04,E_N_Edge);
i32Bit= DrvGPIO_PortIDIF(E_PT2); //read 0x4081C[31:24]
```

## 6. 模數轉換器 ADC

### 6.1. 函數簡介

該部分函數描述ADC 系統的控制，包含：

- ADC的信號輸入埠與參考輸入埠的配置與切換
- ADC放大倍數的設置
- ADC中斷配置
- ADC轉換值的讀取

序號	函數名稱	功能描述
01	DrvADC_PInputChannel	ADC正端信號輸入源設置
02	DrvADC_NInputChannel	ADC負端信號輸入源設置
03	DrvADC_SetADCInputChannel	ADC正負端信號輸入源設置
04	DrvADC_InputSwitch	ADC輸入端短路開關控制
05	DrvADC_RefInputShort	ADC參考電壓輸入端短路開關控制
06	DrvADC_ADGain	ADC輸入信號放大倍數ADGain設置
07	DrvADC_DCoffset	ADC輸入零點平移(DC offset)設置
08	DrvADC_RefVoltage	ADC參考電壓輸入設置
09	DrvADC_FullRefRange	ADC參考電壓放大倍數設置
10	DrvADC_OSR	ADC轉換輸出率OSR設置
11	DrvADC_ClkEnable	開啟ADC時鐘源
12	DrvADC_ClkDisable	關閉ADC時鐘源
13	DrvADC_CombFilter	梳狀濾波器開啟控制
14	DrvADC_EnableInt	ADC中斷開啟
15	DrvADC_DisableInt	ADC中斷關閉
16	DrvADC_ReadIntFlag	讀取ADC中斷標誌位元
17	DrvADC_Enable	開啟ADC
18	DrvADC_Disable	關閉ADC
19	DrvADC_GetConversionData	讀取ADC的A/D轉換值

## 6.2. 內部定義常量

### E\_ADC\_INPUT\_CHANNEL

識別字	數值	函數功能
OP_OP	0	信號輸入端
OP_ON	1	信號輸入端
ADC_Input_AIO2	2	信號輸入端
ADC_Input_AIO3	3	信號輸入端
ADC_Input_AIO4	4	信號輸入端
ADC_Input_AIO5	5	信號輸入端
ADC_Input_AIO6	6	信號輸入端
ADC_Input_AIO7	7	信號輸入端
TPS0_TPS1	8	信號輸入端
TPS1_TPS0	9	信號輸入端
REFO_I	10	信號輸入端
VDDA_VSS	11	信號輸入端
R2ROPO	12	信號輸入端
DAOI	13	信號輸入端
ADC_Input_AIO8	14	信號輸入端
VDD3V5_VSS	15	信號輸入端

### E\_ADC\_REFV

識別字	數值	函數功能
External	0	外部輸入源
Internal	1	使能緩衝器並使用內部源

### E\_ADC\_PGA & E\_ADC\_ADGN

識別字	數值	函數功能	識別字	數值	函數功能
ADC_PGA_Disable	0	Disable PGA	ADC_ADGN_1	0	ADGN=1
ADC_PGA_8	1	PGA=8	ADC_ADGN_2	1	ADGN=2
ADC_PGA_16	3	PGA=16	ADC_ADGN_RESER	2	Reserve
ADC_PGA_32	7	PGA=32	ADC_ADGN_4	3	ADGN=4

### E\_ADC\_SIGNAL\_SHORT

識別字	數值	函數功能
OPEN	0	ADC信號輸入短路開關斷開
SHORT	1	ADC信號輸入短路開關閉合

### E\_ADC\_VRPS\_REF\_VOLTAGE

識別字	數值	函數功能
VDDA	0	參考電壓正端輸入為 VDDA
AIO2	1	參考電壓正端輸入為 AIO2
AIO4	2	參考電壓正端輸入為 AIO4
REF_BUFFER_OUT	3	參考電壓正端輸入為 REFO_I

**E\_ADC\_VRNS\_REF\_VOLTAGE**

識別字	數值	函數功能
VSSA	0	參考電壓負端輸入為VSSA
AIO3	1	參考電壓負端輸入為 AIO3
AIO5	2	參考電壓負端輸入為 AIO5
REF_BUFFER_OUT	3	參考電壓負端輸入為 REFO_I

## 6.3. 函數說明

### 6.3.1. DrvADC\_PInputChannel

- 函數

unsigned int DrvADC\_PInputChannel (E\_ADC\_INPUT\_Channel uINP);

- 函數功能

設置ADC 輸入信號正向輸入端；設置寄存器0x41104[7:4].

- 輸入參數

uINP [in]：代表ADC的正向輸入埠選擇。設定值範圍：0~15

0 : OP\_OP,            1 : OP\_ON,  
2 : AIO2,            3 : AIO3,  
4 : AIO4,            5 : AIO5,  
6 : AIO6,            7 : AIO7,  
8 : TS0,            9 : TS1;  
10 : REFO\_I,        11 : VDDA  
12 : R2ROPO,       13 : DAOI  
14 : AIO8,           15 : VDD3V/5

- 包含標頭檔

Peripheral\_lib/DrvADC.h

- 函數返回值

0 : 設置成功  
其他：設置失敗

- 函數用法

```
/* 設定ADC正向輸入端為AIO2*/  
DrvADC_PInputChannel(ADC_Input_AIO2);
```

### 6.3.2. DrvADC\_NInputChannel

- 函數

unsigned int DrvADC\_NInputChannel (E\_ADC\_INPUT\_Channel uINN);

- 函數功能

設置ADC 輸入信號負向輸入端，設置寄存器0x41104[3:0].

- 輸入參數

uINN [in] :代表ADC負端輸入選擇埠。設定範圍值：0~15



0 : OP\_OP,        1 : OP\_ON,  
2 : AIO2,        3 : AIO3,  
4 : AIO4,        5 : AIO5,  
6 : AIO6,        7 : AIO7,  
8 : TS1,         9 : TS0;  
10 : REFO\_I,     11 : VDDA  
12 : R2ROPO,    13 : DAOI  
14 : AIO8,       15 : VDD3V/5

● **包含標頭檔**

Peripheral\_lib/DrvADC.h

● **函數返回值**

0 : 設置成功  
其他 : 設置失敗

● **函數用法**

```
/* 設定ADC負向輸入端為AIO3*/  
DrvADC_NInputChannel(ADC_Input_AIO3);
```

### 6.3.3. DrvADC\_SetADCInputChannel

● **函數**

```
unsigned int DrvADC_SetADCInputChannel (  
                  E_ADC_INPUT_Channel uINP,  
                  E_ADC_INPUT_Channel uINN );
```

● **函數功能**

設置ADC輸入信號的正向、負向輸入埠，設置寄存器0x41104[7:4] / 0x41104[3:0].

● **輸入參數**

uINP [in]: 代表ADC的正向輸入選擇埠。設定值範圍 : 0~15

0 : OP\_OP,        1 : OP\_ON,  
2 : AIO2,        3 : AIO3,  
4 : AIO4,        5 : AIO5,  
6 : AIO6,        7 : AIO7,  
8 : TS0,         9 : TS1;  
10 : REFO\_I,     11 : VDDA  
12 : R2ROPO,    13 : DAOI  
14 : AIO8,       15 : VDD3V/5

uINN [in]: 代表ADC負端輸入選擇埠。設定範圍值 : 0~15

0 : OP\_OP,        1 : OP\_ON,

2 : AIO2,            3 : AIO3,  
4 : AIO4,            5 : AIO5,  
6 : AIO6,            7 : AIO7,  
8 : TS1,             9 : TS0;  
10 : REFO\_I,        11 : VDDA  
12 : R2ROPO,        13 : DA0I  
14 : AIO8,           15 : VDD3V/5

● **包含標頭檔**

Peripheral\_lib/DrvADC.h

● **函數返回值**

0 : 設置成功  
其他 : 設置失敗

● **函數用法**

```
/* 設定ADC正向輸入端為AIO2 · 負向輸入端為AIO3*/  
DrvADC_SetADCInputChannel(ADC_Input_AIO2, ADC_Input_AIO3);
```

### 6.3.4. DrvADC\_InputSwitch

● **函數**

unsigned int DrvADC\_InputSwitch (uVISHR)

● **函數功能**

ADC信號輸入端短路開關控制. 設置寄存器0x41100[21]

● **輸入參數**

uVISHR[in] : ADC信號輸入端短路開關控制. 設定值範圍 : 0~1  
0: 短路開關斷開  
1: 短路開關閉合

● **包含標頭檔**

Peripheral\_lib/DrvADC.h

● **函數返回值**

0 : 設置成功  
其他 : 設置失敗

● **函數用法**

```
/* ADC 輸入端短路開關閉合*/  
DrvADC_InputSwitch(1);
```

### 6.3.5. DrvADC\_RefInputShort

- 函數

unsigned int DrvADC\_RefInputShort(E\_ADC\_SIGNAL\_SHORT uVrshr);

- 函數功能

ADC參考電壓輸入端短路開關控制。設置寄存器0x41100[20].

- 輸入參數

uVrshr [in] : ADC參考電壓輸入端短路開關控制。設定值範圍 : 0~1

0 : ADC 參考電壓輸入端短路開關斷開

1 : ADC 參考電壓輸入端短路開關閉合

- 包含標頭檔

Peripheral\_lib/DrvADC.h

- 函數返回值

0 : 設置失敗

其他 : 設置失敗

- 函數用法

/\* 設置ADC 參考電壓輸入端短路開關閉合 \*/

DrvADC\_RefInputShort(SHORT);

### 6.3.6. DrvADC\_ADGain

- 函數

unsigned int DrvADC\_ADGain(uADgain);

- 函數功能

配置ADC 輸入信號內部放大倍數控制器ADGIN ; 設置寄存器0x41104[18:16].

- 輸入參數

uADgain [in] : 代表ADC內部放大倍數控制器ADGN. 設定值範圍 : 0~7

0: 放大倍數為 1

1: 放大倍數為 2

2 : 放大倍數為 保留

3: 放大倍數為 4

4: 放大倍數為 保留

5: 放大倍數為 保留

6: 放大倍數為 保留

7: 放大倍數為 8

- 包含標頭檔

Peripheral\_lib/DrvADC.h

- 函數返回值

0 : 設置成功

其他：設置失敗

- 函數用法

```
/*設置ADGN=2 */
```

```
DrvADC_ADGain(1);
```

### 6.3.7. DrvADC\_DCOffset

- 函數

```
unsigned int DrvADC_DCOffset (uDCOffset);
```

- 函數功能

設置ADC 輸入信號的零點平移(DC offset);設置寄存器0x41104[27:24]。

- 輸入參數

uDCoffice [in]：代表ADC 零點平移DCSET，VREF=REFP-REFN。設定值範圍：0~15

0 : 0 VREF

1 : +1/8 VREF

2 : +1/4 VREF

3 : +3/8 VREF

4 : +1/2 VREF

5 : +5/8 VREF

6 : +3/4 VREF

7 : +7/8 VREF

8 : 0 VREF

9 : -1/8 VREF

10 : -1/4 VREF

11 : -3/8 VREF

12 : -1/2 VREF

13 : -5/8 VREF

14 : -3/4 VREF

15 : -7/8 VREF

- 包含標頭檔

```
Peripheral_lib/DrvADC.h
```

- 函數返回值

0：設置成功

其他：設置失敗

- 函數用法

```
/* 設置零點平移(DCSET)為+1/8 VREF. */
```

```
DrvADC_DCOffset(1);
```

### 6.3.8. DrvADC\_RefVoltage

- 函數

```
unsigned int DrvADC_RefVoltage (  
    E_ADC_VRPS_REF_VOLTAGE uVrps,  
    E_ADC_VRNS_REF_VOLTAGE uVrns  
);
```

- 函數功能

設置ADC參考電壓輸入埠,參考電壓(VREF)=VRPS-VRNS ;  
設置寄存器0x41100[19:18]及0x41100[17:16].

- 輸入參數

uVrps [in] : 代表 ADC 參考電壓正向輸入端 VRPS. 設定值範圍 : 0~3

0: 參考電壓正向輸入來自 VDDA

1: 參考電壓正向輸入來自 AIO2

2: 參考電壓正向輸入來自 AIO4

3: 參考電壓正向輸入來自 REFO\_I

uVrns [in] : 代表 ADC 參考電壓的負向輸入端 VRNS. 設定值範圍 : 0~3

0: 參考電壓負向輸入來自 VSSA

1: 參考電壓負向輸入來自 AIO3

2: 參考電壓負向輸入來自 AIO5

3: 參考電壓負向輸入來自 REFO\_I

- 包含標頭檔

Peripheral\_lib/DrvADC.h

- 函數返回值

0 : 設置成功

其他 : 設置失敗

- 函數用法

/\* 設置ADC參考輸入電壓(VRPS=AIO2, VRNS=AIO3) \*/

DrvADC\_RefVoltage(AIO2, AIO3);

### 6.3.9. DrvADC\_FullRefRange

- 函數

```
unsigned int DrvADC_FullRefRange(uFullRange);
```

- 函數功能

設置ADC 輸入參考電壓(VREF)的放大倍數;設置寄存器0x41104[19]。

- 輸入參數

uFullRange [in] : 設置 ADC 輸入參考電壓(VREF)的放大數 · VREF=VRPS-VRNS. 設定值範圍 : 0~1

0: 1 輸入參考電壓VREF\*1

1: 1/2 輸入參考電壓VREF\*1/2

- 包含標頭檔

Peripheral\_lib/DrvADC.h

- 函數返回值

0 : 設置成功

其他 : 設置失敗

- 函數用法

```
/*設置ADC輸入參考電壓VREF*1 */
```

```
DrvADC_FullRefRange(0);
```

### 6.3.10. DrvADC\_OSR

- 函數

```
unsigned int DrvADC_OSR (uADCOSR);
```

- 函數功能

配置ADC 轉換值的輸出頻率(OSR) · 設置寄存器0x41100[5:2]。

- 輸入參數

uADCOSR [in] : 表示ADC 轉換值輸出率(OSR)除頻器設置(以下輸出率是以時鐘源為1MHz計算).

設定值範圍 : 0~10

0 : ÷32768 · 資料輸出率是31sps

1 : ÷16384 · 資料輸出率是61sps

2 : ÷8192 · 資料輸出率是122sps

3 : ÷4096 · 資料輸出率是244sps

4 : ÷2048 · 資料輸出率是488sps

5 : ÷1024 · 資料輸出率是977sps

6 : ÷512 · 資料輸出率是1953sps

7 : ÷256 · 資料輸出率是3906sps

8 : ÷128 · 資料輸出率是7813sps

9 : ÷64 · 資料輸出率是15625sps

10 : Reserved

- 包含標頭檔

Peripheral\_lib/DrvADC.h

- 函數返回值

0 : 設置成功

其他 : 設置失敗

- 函數用法

```
/* 設置輸出率(OSR)為8192, 122sps */  
DrvADC_OSR(2);
```

### 6.3.11. DrvADC\_ClkEnable

- 函數

```
unsigned int DrvADC_ClkEnable(uADCD);
```

- 函數功能

使能ADC時鐘源，並設置時鐘源分頻值和ADC 時鐘源相位調整；設置寄存器0x4030C[6:4].

- 輸入參數

uADCD [in]: 表示ADC 時鐘源分頻器. 設定值範圍 : 2~7

- 1 : Reserved
- 2 : HS\_CK/4
- 3 : HS\_CK/8
- 4 : HS\_CK/16
- 5 : HS\_CK/32
- 6 : HS\_CK/64
- 7 : HS\_CK/128

- 包含標頭檔

```
Peripheral_lib/DrvADC.h
```

- 函數返回值

- 0 : 設置成功
- 其他 : 設置失敗

- 函數用法

```
/*設置ADC 時鐘頻率分頻HS_CK/4 */  
DrvADC_ClkEnable(2);
```

### 6.3.12. DrvADC\_ClkDisable

- 函數

```
void DrvADC_ClkDisable(void);
```

- 函數功能

關閉ADC時鐘源；設置寄存器0x4030C[6:4]=000b.

- 輸入參數

無

- 包含標頭檔

```
Peripheral_lib/DrvADC.h
```

- **函數返回值**

0 : 設置成功  
其他 : 設置失敗

- **函數用法**

```
/* 關閉ADC時鐘源 */  
DrvADC_ClkDisable();
```

### 6.3.13. DrvADC\_CombFilter

- **函數**

```
unsigned int DrvADC_CombFilter(uCFRST);
```

- **函數功能**

梳狀濾波器使能控制，設置該位可以自動丟棄前3筆無效ADC 資料；設置寄存器0x41100[1]。

- **輸入參數**

uCFRST [in] : 梳狀濾波器使能控制. 設定值範圍 : 0~1  
0: 復位(RESET)  
1: 開啟(ON)

- **包含標頭檔**

Peripheral\_lib/DrvADC.h

- **函數返回值**

0 : 設置成功  
其他 : 設置失敗

- **函數用法**

```
/* 使能梳狀濾波器並配置自動丟棄前3筆無效資料. */  
DrvADC_CombFilter(0); //濾波器復位  
DrvADC_CombFilter(1); //使能濾波器
```

### 6.3.14. DrvADC\_EnableInt

- **函數**

```
void DrvADC_EnableInt (void)
```

- **函數功能**

開啟ADC中斷向量，ADC為中斷向量HW2；設置寄存器0x40008[16]=1b.

- **輸入參數**

無

- **包含標頭檔**

Peripheral\_lib/DrvADC.h



- 函數返回值

無

- 函數用法

```
/* 使能ADC 中斷 */
```

```
DrvADC_EnableInt();
```

### 6.3.15. DrvADC\_DisableInt

- 函數

```
void DrvADC_DisableInt (void)
```

- 函數功能

關閉ADC 中斷向量；設置寄存器0x40008[16]=0b.

- 輸入參數

無

- 包含標頭檔

```
Peripheral_lib/DrvADC.h
```

- 函數返回值

無

- 函數用法

```
/* 關閉ADC中斷向量 */
```

```
DrvADC_DisableInt();
```

### 6.3.16. DrvADC\_ReadIntFlag

- 函數

```
unsigned int DrvADC_ReadIntFlag (void)
```

- 函數功能

讀取ADC中斷標誌位元(ADCIF).讀取寄存器0x40008[0]值

- 輸入參數

無

- 包含標頭檔

```
Peripheral_lib/DrvADC.h
```

- 函數返回值

0：中斷標誌位元值是0，表示無中斷產生

1：中斷標誌位元值是1，表示有中斷產生

>1: 無效返回值

- 函數用法

```
/* 讀取ADC 中斷標誌位元*/  
flag = DrvADC_ReadIntFlag(); //讀取ADC中斷要求標誌位元
```

### 6.3.17. DrvADC\_Enable

- 函數

```
void DrvADC_Enable(void)
```

- 函數功能

開啟ADC功能；設置寄存器0x41100[0]=1b.

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvADC.h

- 函數返回值

無

- 函數用法

```
/* 使能ADC */  
DrvADC_Enable();
```

### 6.3.18. DrvADC\_Disable

- 函數

```
void DrvADC_Disable(void)
```

- 函數功能

關閉ADC功能；設置寄存器0x41100[0]=0b

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvADC.h

- 函數返回值

無

- 函數用法

```
/* 關閉ADC功能 */  
DrvADC_Disable();
```

### 6.3.19. DrvADC\_GetConversionData

- 函數

int DrvADC\_GetConversionData (void);

- 函數功能

讀取A/D 轉換值，資料是帶符號的。讀取寄存器0x41108[31 :0]。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvADC.h

- 函數返回值

返回A/D轉換值。

- 函數用法

/\*讀取ADC 轉換值\*/

int adc\_data ;

adc\_data=DrvADC\_GetConversionDate();

## 7. SPI32 串列通訊

### 7.1. 函數簡介

該部分函數描述 SPI 功能的控制，包括：

- SPI 功能的開啟控制
- SPI 的工作模式及參數的配置
- SPI 的中斷向量的控制
- SPI 狀態控制
- SPI 的資料的收發

序號	函數名稱	功能描述
01	DrvSPI32_Open	開啟SPI 功能
02	DrvSPI32_Close	關閉SPI 功能
03	DrvSPI32_IsBusy	查詢SPI 匯流排繁忙狀態
04	DrvSPI32_SetClockFreq	配置SPI 時鐘頻率
05	DrvSPI32_IsRxBufferFull	查詢接收暫存器狀態位元
06	DrvSPI32_IsTxBufferFull	查詢發送暫存器狀態位元
07	DrvSPI32_EnableRxInt	開啟SPI 接收中斷功能
08	DrvSPI32_EnableTxInt	開啟SPI 發送中斷功能
09	DrvSPI32_DisableRxInt	關閉SPI 接收中斷功能
10	DrvSPI32_DisableTxInt	關閉SPI 發送中斷功能
11	DrvSPI32_GetRxIntFlag	讀取SPI 接收中斷標誌位元
12	DrvSPI32_GetTxIntFlag	讀取SPI 發送中斷標誌位元
13	DrvSPI32_ClrIntRxFlag	清除SPI 接收中斷標誌位元
14	DrvSPI32_ClrIntTxFlag	清除SPI 發送中斷標誌位元
15	DrvSPI32_Read	讀取SPI 接收暫存器的資料
16	DrvSPI32_Write	寫入資料至發送暫存器
17	DrvSPI32_Enable	開啟SPI 功能
18	DrvSPI32_BitLength	設置數據的長度
19	DrvSPI32_GetDCFlag	讀取SPI 資料丟失狀態位元
20	DrvSPI32_IsABFlag	讀取SPI 接收到的資料長度小的狀態
21	DrvSPI32_IsOVFlag	檢查SPI 接收到資料是否過長
22	DrvSPI32_IsRxFlag	檢查接收暫存器資料是否更新
23	DrvSPI32_SetEndian	設定資料發送是從MSB或LSB開始發送
24	DrvSPI32_SetCSO	SPI 時序源極性選擇位設置

---

25	DrvSPI32_DisableIO	關閉IO口複用為SPI通訊口的功能
26	DrvSPI32_EnableIO	開啟及選擇IO口複用為SPI通訊口功能

## 7.2. 內部定義常量

### E\_DRVSPi\_MODE

識別字	數值	函數功能
E_DRVSPi_MASTER1	0	4-wire 主動模式
E_DRVSPi_MASTER2	1	3-wire 主動模式
E_DRVSPi_MASTER3	2	TI 方式主動模式
E_DRVSPi_SLAVE1	3	4-wire 被動模式
E_DRVSPi_SLAVE2	4	3-wire 被動模式
E_DRVSPi_SLAVE3	5	TI 方式被動模式

### E\_DRVSPi\_TRANS\_TYPE

識別字	數值	函數功能
E_DRVSPi_TYPE0	0	SPI 發送模式0
E_DRVSPi_TYPE1	1	SPI 發送模式1
E_DRVSPi_TYPE2	2	SPI 發送模式2
E_DRVSPi_TYPE3	3	SPI 發送模式3

### E\_DRVSPi\_ENDIAN

識別字	數值	函數功能
E_DRVSPi_LSB_FIRST	1	從低8bit(LSB)開始發送
E_DRVSPi_MSB_FIRST	0	從高8bit(MSB)開始發送

### E\_DRVSPi\_CS

識別字	數值	函數功能
E_DRVSPi_CSLow	0	CS0 low
E_DRVSPi_CSHigh	1	CS0 high

## 7.3. 函數說明

### 7.3.1. DrvSPI32\_Open

- 函數

```
unsigned int DrvSPI32_Open(  
    E_DRVSPI_MODE uMode,  
    E_DRVSPI_TRANS_TYPE uType,  
    uOutputPin,  
    uClkDiv  
);
```

- 函數功能

函數開啟SPI功能，設置SPI工作是主動模式或者被動模式，設置SPI匯流排時序及通訊IO；  
設置寄存器

0x4030C[2:0],0x4030C[3]=1b, 0x40844[4]=1b, 0x40844[7:5],0x40F00[3:0],0x40f04[16:17]

uMode : 0x40f00[0]=1b, 0x40f00[1]=xb, 0x40f04[16:17]=0xb. uMode : 0~5

uType : 0x40f00[3:2]=xxb. uType : 0~3

uOutputPin : 0x40844[4]=1b, 0x40844[7:5]=xxxb. uOutputPin : 0~7

uClkDiv : 0x4030C[2:0]=xxxb, 0x4030C[3]=1b. uClkDiv : 0~7

- 輸入參數

uMode [in] : 工作模式設置，設置範圍 : 0~5

0 : 4-wire通訊介面的主動模式.

1 : 3-wire通訊介面的主動模式.

2 : TI 模式介面的主動模式.

3 : 4-wire通訊介面的被動模式.

4 : 3-wire通訊介面的被動模式.

5 : TI 模式介面的被動模式.

uType [in] 傳輸類型，如通訊匯流排時序，設置範圍是 : 0~3.

0: 抓取資料在第一個時鐘沿，時鐘源低電平為空閒狀態.(CPHA=0 CPOL=0)

1: 抓取資料在第一個時鐘沿，時鐘源高電平為空閒狀態.(CPHA=0 CPOL=1)

2: 抓取資料在第二個時鐘沿，時鐘源低電平為空閒狀態.(CPHA=1 CPOL=0)

3: 抓取資料在第二個時鐘沿，時鐘源高電平為空閒狀態.(CPHA=1 CPOL=1)

uOutputPin [in] : SPI通訊IO 口設置，設置範圍是 : 0~7

0 : - (Rsv)

1 : - (Rsv)

2 : Port2.0 =CS, Port2.1 =CK, Port2.2 = DI, Port2.3 =DO

3 : Port2.4 =CS, Port2.5 =CK, Port2.6 = DI, Port2.7 =DO

4 : Port8.0 =CS, Port8.1 =CK, Port8.2 = DI, Port8.3 =DO  
5 : Port8.4 =CS, Port8.5 =CK, Port8.6 = DI, Port8.7 =DO  
6 : Port9.0 =CS, Port9.1 =CK, Port9.2 = DI, Port9.3 =DO  
7 : - (Rsv)

uClkDiv [in] : SPI時鐘源分頻器設置 · 設置範圍是 : 0~7

0 : ÷1

1 : ÷2

2 : ÷4

3 : ÷8

4 : ÷32

5 : ÷128

6 : ÷512

7 : ÷2048

- 包含標頭檔

Peripheral\_lib/DrvSPI32.h

- 函數返回值

0 : 設置成功

其他 : 設置失敗

- 函數用法

/\*使能SPI主動模式 · 時鐘源分頻CLOCK/512 · 設置傳送類型1 · 通訊IO 口設置: Port2.0 =CS, Port2.1 =CK, Port2.2 = DI, Port2.3 =DO\*/

DrvSPI32\_Open(E\_DRVSPI\_MASTER1, E\_DRVSPI\_TYPE1, 2,6);

### 7.3.2. DrvSPI32\_Close

- 函數

void DrvSPI32\_Close (void);

- 函數功能

關閉SPI功能 · 關閉SPI的時鐘、IO等功能；

設置寄存器0x40F00[0]=0, 0x4030C[3]=0,0x40844[4]=0 .

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvSPI32.h

- 函數返回值

無

- 函數用法

/\* 關閉SPI \*/

DrvSPI32\_Close();



### 7.3.3. DrvSPI32\_IsBusy

- 函數

```
unsigned int DrvSPI32_IsBusy( void );
```

- 函數功能

查詢SPI匯流排上是否繁忙狀態.

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvSPI32.h

- 函數返回值

1 : SPI匯流排繁忙

0 : SPI匯流排空閒.

- 函數用法

```
/* 檢查匯流排繁忙狀態*/
```

```
unsigned char  flag;
```

```
flag=DrvSPI32_IsBusy (); //read 0x40f00[19]
```

### 7.3.4. DrvSPI32\_SetClockFreq

- 函數

```
unsigned int DrvSPI32_SetClockFreq(unsigned int uCPUDV, unsigned int uTMRDV );
```

- 函數功能

配置MCU時鐘分頻器及SPI時鐘分頻器且使能SPI時鐘源。主動模式下輸出時鐘頻率可程式設計設置；設置寄存器0x40308[1], 0x4030C[2:0].

- 輸入參數

eCPUDV [in] : MCU 時鐘分頻器設置. 設定值範圍 : 0~1

0 : ÷1

1 : ÷2

eTMRDV [in] : SPI 時鐘分頻器設置. 設定值範圍 : 0~7

0 : Reserved

1 : ÷2

2 : ÷4

3 : ÷8

4 : ÷32

5 : ÷128

6 : ÷512

7 : ÷2048

- 包含標頭檔

Peripheral\_lib/DrvSPI32.h

- 函數返回值

無

- 函數用法

/\* 設置MCU時鐘分頻器/2, SPI頻率是APCK/512 \*/

DrvSPI32\_SetClockFreq(1, 6);

### 7.3.5. DrvSPI32\_IsRxBufferFull

- 函數

unsigned int DrvSPI32\_IsRxBufferFull(void );

- 函數功能

查詢接收暫存器滿狀態位元(RXBF) ( 只用於資料接收 ); 讀取寄存器0x40F00[16]。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvSPI32.h

- 函數返回值

1: 接收已完成 · 接收暫存器已滿.

0: 接收未完成 · 接收暫存器為空.

- 函數用法

/\* 查詢接收暫存器狀態\*/

unsigned char flag;

flag = DrvSPI32\_IsRxBufferFull();

### 7.3.6. DrvSPI32\_IsTxBufferFull

- 函數

```
unsigned int DrvSPI32_IsTxBufferFull(void );
```

- 函數功能

查詢發送暫存器滿的狀態(TXBF) ( 只用於資料發送 ) 讀取寄存器0x40F00[17]。

- 輸入參數

無

- 包含標頭檔

```
Peripheral_lib/DrvSPI32.h
```

- 函數返回值

1: 發送未完成，發送暫存器還有資料。

0: 發送已完成，發送暫存器為空。

- 函數用法

```
/* 查詢發送暫存器的狀態 */
```

```
unsigned char flag; flag =DrvSPI32_IsTxBufferFull();
```

### 7.3.7. DrvSPI32\_EnableRxInt

- 函數

```
void DrvSPI32_EnableRxInt(void);
```

- 函數功能

使能SPI接收中斷，屬於中斷向量HW0；設置寄存器0x40000[16]=1b。

- 輸入參數

無

- 包含標頭檔

```
Peripheral_lib/DrvSPI32.h
```

- 函數返回值

無

- 函數用法

```
/* SPI 接收中斷使能*/
```

```
DrvSPI32_EnableRxInt();
```

### 7.3.8. DrvSPI32\_EnableTxInt

- 函數

```
void DrvSPI32_EnableTxInt(void);
```

- 函數功能

使能SPI 發送中斷，屬於中斷向量HW0；設置寄存器0x40000[17]=1b。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvSPI32.h

- 函數返回值

無

- 函數用法

```
/*SPI 發送中斷使能*/
```

```
DrvSPI32_EnableTxInt();
```

### 7.3.9. DrvSPI32\_DisableRxInt

- 函數

```
void DrvSPI32_DisableRxInt(void);
```

- 函數功能

關閉SPI 接收中斷功能，設置寄存器0x40000[16]=0b

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvSPI32.h

- 函數返回值

無

- 函數用法

```
/*關閉SPI 接收中斷 */
```

```
DrvSPI32_DisableRxInt();
```

### 7.3.10. DrvSPI32\_DisableTxInt

- 函數

```
void DrvSPI32_DisableTxInt(void);
```

- 函數功能

關閉SPI 發送中斷，設置寄存器0x40000[17]=0b。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvSPI32.h

- 函數返回值

無

- 函數用法

```
/* 關閉SPI 發送中斷 */  
DrvSPI32_DisableTxInt();
```

### 7.3.11. DrvSPI32\_GetRxIntFlag

- 函數

```
unsigned int DrvSPI32_GetRxIntFlag ();
```

- 函數功能

讀取SPI 接收中斷要求標誌位元(SRXIF)；讀取寄存器0x40000[0]。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvSPI32.h

- 函數返回值

1: 中斷標誌位元為1，有中斷要求  
0: 中斷標誌位元為0，無中斷要求

- 函數用法

```
/*讀取SPI接收中斷要求標誌位元*/  
unsigned char flag ;  
flag=DrvSPI32_GetRxIntFlag();
```

### 7.3.12. DrvSPI32\_GetTxIntFlag

- 函數

```
unsigned int DrvSPI32_GetTxIntFlag ();
```

- 函數功能

讀取SPI 發送中斷要求標誌位元(STXIF)；讀取寄存器0x40000[1]。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvSPI32.h

- 函數返回值

1: 中斷標誌位元為1，有中斷要求

0: 中斷標誌位元為0，無中斷要求

- 函數用法

```
/* 讀取SPI 發送中斷要求標誌位元.*/  
unsigned char flag ;  
flag=DrvSPI32_GetTxIntFlag();
```

### 7.3.13. DrvSPI32\_ClrIntRxFlag

- 函數

```
void DrvSPI32_ClrIntRxFlag ();
```

- 函數功能

清除SPI 接收中斷要求標誌位元(SRXIF)；設置寄存器0x40000[0]=0b.

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvSPI32.h

- 函數返回值

無

- 函數用法

```
/*清除SPI 接收中斷要求標誌位元*/  
DrvSPI32_ClrIntRxFlag();
```

### 7.3.14. DrvSPI32\_ClrIntTxFlag

- 函數

```
void DrvSPI32_ClrIntTxFlag ();
```

- 函數功能

清除SPI 發送中斷要求標誌位元 ( STXIF ) ,設置寄存器0x40000[1]=0b .

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvSPI32.h

- 函數返回值

無

- 函數用法

```
/* 清除SPI發送中斷要求標誌位元*/
```

```
DrvSPI32_ClrIntTxFlag();
```

### 7.3.15. DrvSPI32\_Read

- 函數

```
unsigned int DrvSPI32_Read();
```

- 函數功能

讀取SPI 資料接收暫存器 ; 讀取寄存器0x40F08[31:0] . .

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvSPI32.h

- 函數返回值

返回值是SPI 接收暫存器的值

- 函數用法

```
/* 讀取接收暫存器的值 */
```

```
/*資料接收方式LSB First 8bit 數據*/
```

```
unsigned int data ; data=DrvSPI32_Read(>>24;
```

```
/*資料接收方式MSB 8bit 數據*/
```

```
unsigned int data ; data=DrvSPI32_Read();
```

### 7.3.16. DrvSPI32\_Write

- 函數

void DrvSPI32\_Write (unsigned int uData );

- 函數功能

寫入待發送資料至發送暫存器並發送；寫入寄存器0x40F0C[31:0].

- 輸入參數

uData [in]：待發送資料. 輸入範圍：0~0xFFFFFFFF。

- 包含標頭檔

Peripheral\_lib/DrvSPI32.h

- 函數返回值

無

- 函數用法

/\*資料傳送方式MSB First 8bit 發送0x55\*/

DrvSPI32\_Write(0x55<<24);

/\*資料傳送方式LSB First 8bit 發送0x55\*/

DrvSPI32\_Write(0x55);

### 7.3.17. DrvSPI32\_Enable

- 函數

void DrvSPI32\_Enable (void);

- 函數功能

使能SPI 功能；設置寄存器0x40F00[0]=1b。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvSPI32.h

- 函數返回值

無

- 函數用法

/\* 開啟SPI \*/

DrvSPI32\_Enable();



### 7.3.18. DrvSPI32\_BitLength

- 函數

```
void DrvSPI32_BitLength (unsigned int uData);
```

- 函數功能

設置SPI 發送資料的長度；設置寄存器0x40F04[4:0]。

- 輸入參數

uData[in]：設置SPI 發送資料的長度。設定值範圍是：0x04~0x20

- 包含標頭檔

Peripheral\_lib/DrvSPI32.h

- 函數返回值

無

- 函數用法

```
/* 設定SPI發送資料的長度為8bit*/
```

```
DrvSPI32_BitLength(0x8);
```

```
/* 設定SPI發送資料的長度為32bit*/
```

```
DrvSPI32_BitLength(0x20);
```

### 7.3.19. DrvSPI32\_GetDCFlag

- 函數

```
unsigned int DrvSPI32_GetDCFlag(void);
```

- 函數功能

讀取SPI 資料丟失狀態位元(DCF) · 讀取寄存器0x40F00[18]。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvSPI32.h

- 函數返回值

0: 正常.

1: 接收暫存器已滿 · 讀取接收暫存器可以清零該位.

- 函數用法

```
/* 讀取資料丟失狀態位元 DCF */
```

```
unsigned char flag ;
```

```
flag=DrvSPI32_GetDCFlag();
```

### 7.3.20. DrvSPI32\_IsABFlag

- 函數

unsigned int DrvSPI32\_IsABFlag(void);

- 函數功能

讀取SPI 接收到的資料長度是否缺少的狀態位元(ABF)；讀取寄存器0x40F00[20]的值。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvSPI32.h

- 函數返回值

0: 正常.

1: SPI接收到的資料長度比設置的資料長度少

- 函數用法

/\* 讀取資料長度標誌位元ABF \*/

unsigned char flag ;

flag=DrvSPI32\_IsABFlag();

### 7.3.21. DrvSPI32\_IsOVFlag

- 函數

unsigned int DrvSPI32\_IsOVFlag(void);

- 函數功能

讀取接收到的資料長度是否比設定值長的狀態位元(VOF)；讀取寄存器0x40F00[21]的值。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvSPI32.h

- 函數返回值

0: 正常

1: SPI接收到的資料長度比設定的資料長度大

- 函數用法

/\*讀取接收到資料長度過長標誌位元OVF\*/

unsigned char flag ;

flag=DrvSPI32\_IsOVFlag();

### 7.3.22. DrvSPI32\_IsRxFlag

- 函數

unsigned int DrvSPI32\_IsRxFlag(void);

- 函數功能

讀取SPI 資料接收暫存器的資料更新標誌位元(RXF) , 確定是否讀取接收暫存器 ; 讀取寄存器0x40F00[22] ..

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvSPI32.h

- 函數返回值

0: 正常.

1: SPI 接收暫存器有資料在更新 , 不能讀取接收暫存器。

- 函數用法

*/\*讀取SPI 接收暫存器資料更新標誌位元RxF \*/*

unsigned char flag ;

flag=DrvSPI32\_IsRxFlag();

### 7.3.23. DrvSPI32\_SetEndian

- 函數

void DrvSPI32\_SetEndian(E\_DRVSPI\_ENDIAN eEndian);

- 函數功能

設置SPI 是從高8位還是低8位元資料開始發送 ; 設置寄存器0x40F04[18] .

- 輸入參數

eEndian [in] : 輸入範圍 : 0~1

1 : 低8位(LSB) 開始發送

0 : 高8位(MSB) 開始發送

- 包含標頭檔

Peripheral\_lib/DrvSPI32.h

- 函數返回值

無

- 函數用法

*/\*設置SPI 從低 8位元資料開始發送 \*/*

DrvSPI32\_SetEndian(E\_DRVSPI\_LSB\_FIRST);

### 7.3.24. DrvSPI32\_SetCSO

- 函數

```
void DrvSPI32_SetCSO(E_DRVSPI_CS eCS);
```

- 函數功能

SPI 時序源極性選擇位設置 · 設置寄存器0x40F04[20] .

- 輸入參數

eCS [in]: 輸入範圍 : 0~1

0 : 時序源低電平有效 ( CSO low )

1 : 時序源高電平有效 ( CSO high )

- 包含標頭檔

Peripheral\_lib/DrvSPI32.h

- 函數返回值

無

- 函數用法

```
/* 設置低電平有效 */
```

```
DrvSPI32_SetCSO(E_DRVSPI_CSLow);
```

### 7.3.25. DrvSPI32\_DisableIO

- 函數

```
void DrvSPI32_DisableIO(void);
```

- 函數功能

關閉 SPI 通訊口 · 設置寄存器0x40844[4]=0; .

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvSPI32.h

- 函數返回值

無

- 函數用法

```
/*關閉SPI 通訊口 */
```

```
DrvSPI32_DisableIO();
```

### 7.3.26. DrvSPI32\_EnableIO

- 函數

```
unsigned char DrvSPI32_EnableIO(uint32_t uOutputPin);
```

- 函數功能

開啟SPI 通訊口，設置寄存器0x40844[7:5] / 0x40844[4]=1; .

- 輸入參數

uOutputPin [in] : SPI通訊IO 口設置. 輸入範圍 : 0~7

0 : - (Rsv)

1 : - (Rsv)

2 : Port2.0 =CS, Port2.1 =CK, Port2.2 = DI, Port2.3 =DO

3 : Port2.4 =CS, Port2.5 =CK, Port2.6 = DI, Port2.7 =DO

4 : Port8.0 =CS, Port8.1 =CK, Port8.2 = DI, Port8.3 =DO

5 : Port8.4 =CS, Port8.5 =CK, Port8.6 = DI, Port8.7 =DO

6 : Port9.0 =CS, Port9.1 =CK, Port9.2 = DI, Port9.3 =DO

7 : - (Rsv)

- 包含標頭檔

```
Peripheral_lib/DrvSPI32.h
```

- 函數返回值

0 : 設置成功

1 : 設置失敗

- 函數用法

```
/*開啟SPI 通訊口並選擇PT2.0~PT2.3*/
```

```
DrvSPI32_EnableIO(2);
```

## 8. 非同步串列通訊 UART

### 8.1. 函數簡介

該部分函數描述對 UART 功能的控制，包含：

- UART 功能的啟動與關閉
- UART 功能的配置包括發送速率、時鐘源、資料格式等
- UART 資料的發送與接收
- UART 中斷向量控制
- UART 收發錯誤控制
- UART2 功能的啟動與關閉
- UART2 功能的配置包括發送速率、時鐘源、資料格式等
- UART2 資料的發送與接收
- UART2 中斷向量控制
- UART2 收發錯誤控制

序號	函數名稱	功能描述
01	DrvUART_Open	開啟UART1 功能並配置相關參數
02	DrvUART_Close	關閉UART1 功能
03	DrvUART_EnableInt	使能UART1 中斷向量
04	DrvUART_GetTxFlag	讀取UART1TX中斷標誌位元
05	DrvUART_GetRxFlag	讀取UART1RX中斷標誌位元
06	DrvUART_ClrTxFlag	清除UART1TX中斷標誌位元
07	DrvUART_ClrRxFlag	清除UART1RX中斷標誌位元
08	DrvUART_Read	讀取UART1接收到的資料
09	DrvUART_ClrABDOVF	清除UART1自動串列傳輸速率翻轉狀態 檢測位元
10	DrvUART_Write	UART1寫入資料並發送
11	DrvUART_EnableWakeUp	開啟UART1喚醒功能
12	DrvUART_GetPERR	讀取UART1校驗錯誤狀態位元
13	DrvUART_GetFERR	讀取UART1幀錯誤狀態為
14	DrvUART_GetOERR	讀取UART1溢出錯誤狀態位元
15	DrvUART_GetABDOVF	讀取UART1自動串列傳輸速率翻轉狀態 檢測位元
16	DrvUART_Enable_AutoBaudrate	開啟UART1自動串列傳輸速率功能
17	DrvUART_Disable_AutoBaudrate	關閉UART1自動串列傳輸速率功能
18	DrvUART_CheckTRMT	讀取UART1發送寄存器狀態位元

19	DrvUART_ClkEnable	開啟UART1時鐘源並設置時鐘源
20	DrvUART_ClkDisable	關閉UART1時鐘源
21	DrvUART_Enable	開啟UART1 功能
22	DrvUART_ConfigIO	設置IO複用為UART1通訊口並選擇IO口
23	DrvUART_TRStatus	讀取UART1發送與接收的狀態
24	DrvUART_IntType	設置UART1的TX/RX中斷觸發方式
25	DrvUART_DisableWakeUp	關閉UART1喚醒功能
26	DrvUART_GetNERR	讀取UART1雜訊偵測標誌位元
27	DrvUART_ClrPERR	清除UART1校驗錯誤狀態位元
28	DrvUART_ClrFERR	清除UART1幀錯誤狀態為
29	DrvUART_ClrOERR	清除UART1溢出錯誤狀態位元
30	DrvUART_ClrNERR	清除UART1雜訊偵測標誌位元
31	DrvUART2_Open	開啟UART2 功能並配置相關參數
32	DrvUART2_Enable	開啟UART2 功能
33	DrvUART2_Close	關閉UART2 功能
34	DrvUART2_EnableInt	使能UART2 中斷向量
35	DrvUART2_IntType	設置UART2的TX/RX中斷觸發方式
36	DrvUART2_GetTxFlag	讀取UART2 TX中斷標誌位元
37	DrvUART2_GetRxFlag	讀取UART1 RX中斷標誌位元
38	DrvUART2_ClrTxFlag	清除UART1 TX中斷標誌位元
39	DrvUART2_ClrRxFlag	清除UART1 RX中斷標誌位元
40	DrvUART2_Read	讀取UART2接收到的資料
41	DrvUART2_Write	UART2寫入資料並發送
42	DrvUART2_EnableWakeUp	開啟UART2喚醒功能
43	DrvUART2_DisableWakeUp	關閉UART2喚醒功能
44	DrvUART2_Enable_AutoBaudrate	開啟UART2自動串列傳輸速率功能
45	DrvUART2_Disable_AutoBaudrate	關閉UART2自動串列傳輸速率功能
46	DrvUART2_GetPERR	讀取UART2校驗錯誤狀態位元
47	DrvUART2_GetFERR	讀取UART2幀錯誤狀態為
48	DrvUART2_GetOERR	讀取UART2溢出錯誤狀態位元
49	DrvUART2_GetNERR	清除UART2雜訊偵測標誌位元
50	DrvUART2_ClrPERR	清除UART2校驗錯誤狀態位元
51	DrvUART2_ClrFERR	清除UART2幀錯誤狀態為
52	DrvUART2_ClrOERR	清除UART2溢出錯誤狀態位元
53	DrvUART2_ClrNERR	清除UART2雜訊偵測標誌位元
54	DrvUART2_GetABDOVF	讀取UART2自動串列傳輸速率翻轉狀態 檢測位元
55	DrvUART2_ClrABDOVF	清除UART2自動串列傳輸速率翻轉狀態

		檢測位元
56	DrvUART2_TRStatus	讀取UART2發送與接收的狀態
57	DrvUART2_CheckTRMT	讀取UART2發送寄存器狀態位元
58	DrvUART2_ClkEnable	開啟UART2時鐘源並設置時鐘源
59	DrvUART2_ClkDisable	關閉UART2時鐘源
60	DrvUART2_ConfigIO	設置IO複用為UART2通訊口並選擇IO口



## 8.2. 內部定義常量

### E\_DATABITS\_SETTINGS

識別字	數值	函數功能
DRVUART_DATABITS_6	0x0	數據長度為6 bits
DRVUART_DATABITS_7	0x1	數據長度為7 bits.
DRVUART_DATABITS_8	0x2	數據長度為8 bits
DRVUART_DATABITS_9	0x3	數據長度為9 bits.

### E\_STOPBITS\_SETTINGS

識別字	數值	函數功能
DRVUART_STOPBITS_05	0x0	數據長度為0.5 bits
DRVUART_STOPBITS_1	0x1	數據長度為1 bits.
DRVUART_STOPBITS_15	0x2	數據長度為1.5 bits
DRVUART_STOPBITS_2	0x3	數據長度為2 bits.

### E\_PARITY\_SETTINGS

識別字	數值	函數功能
DRVUART_PARITY_NONE	0x0	無同位校驗
DRVUART_PARITY_ODD	0x1	使能奇同位校驗
DRVUART_PARITY_EVEN	0x2	使能偶同位校驗

### E\_BAUD\_RATE\_SETTINGS

識別字	數值	函數功能
B1200	0x0	Baud rate=1200
B2400	0x1	Baud rate=2400
B4800	0x2	Baud rate=4800
B9600	0x3	Baud rate=9600
B14400	0x4	Baud rate=14400
B19200	0x5	Baud rate=19200
B38400	0x6	Baud rate=38400

### E\_UART\_ERROR\_MESSAGE

識別字	數值	函數功能
E_UART_ERR_CLOCK	0x2	時鐘源輸入錯誤
E_UART_ERR_BAUDRATE	0x3	串列傳輸速率輸入錯誤
E_UART_ERR_PARITY	0x4	校驗方式輸入錯誤
E_UART_ERR_DATABIT	0x5	資料長度輸入錯誤
E_UART_ERR_STOPBIT	0x6	停止位元長度設置錯誤
E_UART_ERR_OUTPIN	0x7	輸出 IO 設置輸入錯誤

## 8.3. 函數說明

### 8.3.1. DrvUART\_Open

#### ● 函數

```
unsigned int DrvUART_Open (  
    unsigned int uClock  
    E_RAUD_RATE_SETTINGS uBaudRate ,  
    E_PARITY_SETTINGS uParity,  
    E_DATABITS_SETTINGS uDataBits,  
    unsigned int uStopBits,  
    unsigned int uOutputPin  
);
```

#### ● 函數功能

設置UART的工作頻率源(除了晶震源為外部晶震(HSXT)或內部晶震(HSRC), UART除頻設置也會影響到實際UART的工作頻率源)並根據寫入的串列傳輸速率值自動計算出串列傳輸速率寄存器0x40E08[15:0]的值;設置UART1的資料校驗模式、資料的位元元數、停止位元及TX/RX的通訊用IO口。

設置寄存器0x40E00[7:4] · 0x40E00[2]=1, 0x40E00[0]=1 / 0x40E04[1:0];

寄存器0x40E08[15:0]; 設置IO口寄存器0x40844[3:0].

#### ● 輸入參數

uClock[in]: 設置UART工作頻率源, 輸入值為URCK 的頻率大小, URCK是由高速晶振頻率(外部高速HSXT或者內部高速頻率HSRC) 經過UACD[3:0]分頻得到, 若UACD=1 · 則URCK=HSXT(或HSRC), 若UACD=2 · 則URCK=HSXT/2(或HSRC/2)依此類推, 以kHz作為單位計算; 輸入範圍 : 1000~20000

uBaudRate [in]: UART1通訊資料串列傳輸速率.

uParity [in]: 校驗模式, 分別為無校驗/奇數同位檢查/偶校驗. 設定值範圍 : 0~2

0: 無校驗

1: 偶校驗

2: 奇數同位檢查

UDataBits[in]: 資料位元數設置. 設定範圍 : 0~3

0: 6 bit 數據.

1: 7 bit 數據.

2: 8 bit 數據.

3: 9 bit 數據

uStopBits[in]: 停止位的長度設置. 設定範圍 : 0~3

0: 0.5 Bit            1: 1 Bit

2: 1.5 Bit           3: 2 Bit

uOutputPin [in]: 通訊線TX/RX IO口設置. 設定範圍 : 0~7

0: Rsv

- 1 : Rsv
- 2 : Port 2.0 =TX, Port 2.1 =RX
- 3 : Port 2.4 =TX, Port 2.5 =RX
- 4 : Port 8.0 =TX, Port 8.1 =RX
- 5 : Port 8.4 =TX, Port 8.5 =RX
- 6 : Port 9.0 =TX, Port 9.1 =RX
- 7 : Port 9.4 =TX, Port 9.5 =RX

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

- 0 : 設置成功.
- 2 : 時鐘設置錯誤
- 3 : 串列傳輸速率設置錯誤
- 4 : 校驗位元設置錯誤
- 5 : 資料長度設置錯誤
- 6 : 停止位元長度設置錯誤
- 7 : 通訊IO設置錯誤

- 函數用法

/\* 設置UART1 4MHZ/115200bps, 8 位元數據 , 且無校驗 · 停止位為1 · 通訊口為PT2.0/PT2.1\*/  
DrvUART\_Open (4147,115200, DRVUART\_PARITY\_NONE ,DRVUART\_DATABITS\_8,1,2);  
Note: 因為 UART1 的工作頻率源為 4.147MHz,, 所以輸入頻率為 4147, 單位為 kHz.

### 8.3.2. DrvUART\_Close

- 函數

void DrvUART\_Close (void );

- 函數功能

關閉UART1功能 ; 清零寄存器0x40E00[2]=0 / 0x40E00[0]=0;

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

無

- 函數用法

/\* 關閉UART1 \*/  
DrvUART\_Close();

### 8.3.3. DrvUART\_EnableInt

- 函數

unsigned int DrvUART\_EnableInt(unsigned int uTXIE, unsigned int uRXIE);

- 函數功能

UART1的發送(TX)或接收(RX)中斷向量控制. UART屬於中斷向量HW0;設置寄存器0x40000[19:18]。

- 輸入參數

uTXIE [in] : UART1 發送(TX)中斷控制. 設定範圍 : 0~1

0 : 關閉中斷

1 : 使能中斷

uRXIE [in] : UART1 接收(RX)中斷控制. 設定範圍 : 0~1

0 : 關閉中斷

1 : 使能中斷

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

0 : 設置成功

其他 : 設置失敗

- 函數用法

*/\* 使能UART1發送及接收中斷 \*/*

DrvUART\_EnableInt(1,1);

### 8.3.4. DrvUART\_GetTxFlag

- 函數

unsigned int DrvUART\_GetTxFlag (void);

- 函數功能

讀取UART1的發送中斷標誌位元(UTXIF)值 · 讀取寄存器0x40000[3]的值。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

1: 有中斷產生

0: 無中斷產生

- 函數用法

*/\* 讀取UART1發送中斷標誌位元. \*/*

unsigned char flag ; flag =DrvUART\_GetTxFlag();

### 8.3.5. DrvUART\_GetRxFlag

- 函數

unsigned int DrvUART\_GetRxFlag (void);

- 函數功能

讀取UART1接收中斷標誌位元URXIF值，讀取寄存器0x40000[2]的值。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

1：有中斷要求

0：無中斷要求

- 函數用法

```
/* 讀取UART1接收中斷標誌位元. */
```

```
unsigned char flag ; flag=DrvUART_GetRxFlag();
```

### 8.3.6. DrvUART\_ClrTxFlag

- 函數

void DrvUART\_ClrTxFlag (void);

- 函數功能

清除UART1發送中斷標誌位元UTXIF 值，清零寄存器0x40000[3]

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

無

- 函數用法

```
/* 清除UART1發送中斷標誌位元. */
```

```
DrvUART_ClrTxFlag();
```

### 8.3.7. DrvUART\_ClrRxFlag

- 函數

void DrvUART\_ClrRxFlag (void);

- 函數功能

清除UART1接收中斷標誌位元URXIF值，清零寄存器0x40000[2]

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

無

- 函數用法

```
/* 清除UART1接收中斷標誌位元. */
```

```
DrvUART_ClrRxFlag();
```

### 8.3.8. DrvUART\_Read

- 函數

```
unsigned int DrvUART_Read(void);
```

- 函數功能

讀取UART1接收到的資料，讀取寄存器0x40E0C[8:0]的值

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

返回接收暫存器的值。

- 函數用法

```
/* 讀取UART1接收到的資料. */
```

```
unsigned int rx_data ; rx_data=DrvUART_Read();
```

### 8.3.9. DrvUART\_ClrABDOVF

- 函數

```
unsigned int DrvUART_ClrABDOVF(void)
```

- 函數功能

接清除UART1的自動串列傳輸速率偵測錯誤標誌位元，清零寄存器0x40E04[4].

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

無

- 函數用法

```
/* 清除UART1的自動串列傳輸速率偵測錯誤標誌位元*/
```

```
DrvUART_ClrABDOVF();
```

### 8.3.10. DrvUART\_Write

- 函數

```
void DrvUART_Write(unsigned int uData);
```

- 函數功能

寫入數值至UART1的發送暫存器(TX data)並等待發送，寫入待發送的值至寄存器0x40E0C[24:16]。

- 輸入參數

uData [in]：待發送的資料

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

無

- 函數用法

```
/*UART1發送0x55 */
```

```
DrvUART_Write(0x55);
```

### 8.3.11. DrvUART\_EnableWakeUp

- 函數

```
void DrvUART_EnableWakeUp(void);
```

- 函數功能

使能UART1的喚醒功能，同樣啟動接收喚醒功能只要接收中斷打開；

設置寄存器0x40E04[2]=1。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

無

- 函數用法

```
/* 使能UART1喚醒功能 */
```

```
DrvUART_EnableWakeUp();
```

### 8.3.12. DrvUART\_GetPERR

- 函數

```
unsigned int DrvUART_GetPERR(void);
```

- 函數功能

讀取UART1校驗錯誤標誌位元(PERR) · 讀取寄存器0x40E00[20]的值。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

1：有校驗錯誤

0：無校驗錯誤

- 函數用法

```
/* 讀取UART1校驗錯誤標誌位元. */
```

```
unsigned char flag; flag=DrvUART_GetPERR();
```

### 8.3.13. DrvUART\_GetFERR

- 函數

```
unsigned int DrvUART_GetFERR(void);
```

- 函數功能

讀取UART1幀錯誤標誌位元(FERR) · 讀取寄存器0x40E00[21]的值。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

1：有幀錯誤

0：無幀錯誤

- 函數用法

```
/* 讀取UART1幀錯誤標誌位元. */
```

```
unsigned char flag ; flag=DrvUART_GetFERR();
```

### 8.3.14. DrvUART\_GetOERR

- 函數

```
unsigned int DrvUART_GetOERR(void);
```

- 函數功能

讀取UART1溢出錯誤標誌位元(OERR) · 讀取寄存器0x40E00[23]的值。

- 輸入參數



無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

1：有溢出錯誤

0：無溢出錯誤

- 函數用法

/\* 讀取UART1溢出錯誤標誌位元. \*/

```
unsigned char flag ; flag=DrvUART_GetOERR();
```

### 8.3.15. DrvUART\_GetABDOVF

- 函數

```
unsigned int DrvUART_GetABDOVF(void);
```

- 函數功能

讀取UART1自動串列傳輸速率發生器翻轉狀態檢測標誌位元(RxABDF) · 讀取寄存器0x40E04[4]值。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

1：在自動串列傳輸速率檢測模式下串列傳輸速率發生器發生翻轉

0：沒有串列傳輸速率發生器發生翻轉

- 函數用法

/\* 讀取UART1串列傳輸速率發生器翻轉標誌位元RXABDF. \*/

```
unsigned char flag ; flag=DrvUART_GetABDOVF();
```

### 8.3.16. DrvUART\_Enable\_AutoBaudrate

- 函數

```
void DrvUART_Enable_AutoBaudrate (void);
```

- 函數功能

使能UART1自動串列傳輸速率功能 · 設置寄存器0x40E04[3]=1.

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

無

- 函數用法

```
/* 使能UART1自動串列傳輸速率功能 */  
DrvUART_Enable_AutoBaudrate();
```

### 8.3.17. DrvUART\_Disable\_AutoBaudrate

- 函數

```
void DrvUART_Disable_AutoBaudrate (void);
```

- 函數功能

關閉UART1自動串列傳輸速率功能，設置寄存器0x40E04[3]=0。

- 輸入參數

無

- 包含標頭檔

```
Peripheral_lib/DrvUART.h
```

- 函數返回值

無

- 函數用法

```
/* 關閉UART1自動串列傳輸速率功能 */  
DrvUART_Disable_AutoBaudrate();
```

### 8.3.18. DrvUART\_CheckTRMT

- 函數

```
Unsigned int DrvUART_CheckTRMT
```

- 函數功能

讀取UART1發送狀態位元(TXBF)，讀取寄存器0x40E00[18]值

- 輸入參數

無

- 包含標頭檔

```
Peripheral_lib/DrvUART.h
```

- 函數返回值

返回發送狀態位元TXBF的值;

- 函數用法

```
/* 讀取UART1發送狀態位元值並設置查詢方式發送資料*/  
DrvUART_Write(data);  
While(DrvUART_CheckTRMT()) ;//等待TRMT=0
```

### 8.3.19. DrvUART\_ClkEnable

- 函數

unsigned int DrvUART\_ClkEnable(unsigned int uclk,unsigned int uprescale) ;

- 函數功能

使能UART1的時鐘源並選擇時鐘源及設置時鐘源的分頻值  
設置寄存器0x40308[21:16] 。

- 輸入參數

uclk[in] : EUART 時鐘源設置. 輸入範圍 : 0~1

0 : 外部晶震高速時鐘

1 : 內部晶震高速時鐘

uprescale[in] : 時鐘源分頻器. 輸入範圍 : 0~15

0	EUART CLOCK SOURCE/1	8	Rsv
1	EUART CLOCK SOURCE/2	9	Rsv
2	EUART CLOCK SOURCE/4	10	Rsv
3	EUART CLOCK SOURCE/8	11	Rsv
4	EUART CLOCK SOURCE/16	12	Rsv
5	EUART CLOCK SOURCE/32	13	Rsv
6	EUART CLOCK SOURCE/64	14	Rsv
7	EUART CLOCK SOURCE/128	15	Rsv

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

0 : 設置成功

1 : 設置失敗

- 函數用法

/\* 設置UART1時鐘源為外部時鐘且分頻clk/1 \*/  
DrvUART\_ClkEnable(0,0) ;

### 8.3.20. DrvUART\_ClkDisable

- 函數

Void DrvUART\_ClkDisable(void) ;

- 函數功能

關閉UART1時鐘源,設置寄存器0x40308[20]=0 。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

無

- 函數用法

*/\*關閉UART1時鐘源\*/*

DrvUART\_ClkDisable();

### 8.3.21. DrvUART\_Enable

- 函數

Void DrvUART\_Enable(void);

- 函數功能

使能UART1功能 ,設置寄存器0x40E00[2]=1/ 0x40E00[0]=1 。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

無

- 函數用法

*/\*使能UART1功能\*/*

DrvUART\_Enable();

### 8.3.22. DrvUART\_ConfigIO

- 函數

unsigned char DrvUART\_ConfigIO(unsigned char ioen,unsigned int uOutputPin);

- 函數功能

設置IO口複用為UART1通訊口 ,及選擇IO口 ,設置寄存器0x40844[3:0] 。

- 輸入參數

ioen[in] :IO 口複用功能使能控制

0 : 關閉IO 複用功能

1 : 開啟IO 複用功能

uoutputPin[in] :選擇通訊IO 口

0 : Rsv

1 : Rsv

2 : Port 2.0 =TX, Port 2.1 =RX

3 : Port 2.4 =TX, Port 2.5 =RX

4 : Port 8.0 =TX, Port 8.1 =RX

5 : Port 8.4 =TX, Port 8.5 =RX

6 : Port 9.0 =TX, Port 9.1 =RX

7 : Port 9.4 =TX, Port 9.5 =RX

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

0 : 設置成功

其他 : 設置失敗

- 函數用法

/\*開啟IO複用為UART1通訊口，並選擇PT2.0/PT2.1\*/

DrvUART\_ConfigIO(1,2);

### 8.3.23. DrvUART\_TRStatus

- 函數

unsigned int DrvUART\_TRStatus(unsigned int uMode)

- 函數功能

讀取UART1發送與接收的狀態，讀取寄存器0x40E00[19:16]值

- 輸入參數

uMode[in]:

0 : RXBF; 1 : RXBUSY; 2 : TXBF; 3 : TXBUSY

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

返回發送及接收的狀態值

TXBUSY : 0 idle; 1 Busy

TXBF : 0 empty; 1 full

RXBUSY: 0 idle; 1 Busy

RXBF : 0 empty; 1 full

- 函數用法

/\* 讀取UART1發送狀態位元值並實現查詢方式發送資料\*/

DrvUART\_Write(data);

While(DrvUART\_TRStatus(2)); //等待TXBF=0

### 8.3.24. DrvUART\_IntType

- 函數

unsigned int DrvUART\_IntType(unsigned int uTXIT, unsigned int uRXIT)

- 函數功能

設置UART1發送與接收的中斷觸發方式，讀取寄存器0x40E00[1]/ 0x40E00[3]

- 輸入參數

uTXIT [in]: UART1發送中斷觸發方式設置

0 當TX 發送暫存器為空時發生中斷要求，寫入資料後中斷標誌位元消失;

- 1 當TX 發送完一筆資料後發生中斷要求。
- uRXIT[in] : UART1接收中斷觸發方式設置
- 0 當RX 接收暫存器有資料時發生中斷要求，讀取資料後中斷標誌位元消失；
- 1 當RX 接收完一筆資料後發生中斷要求。

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

返回發送及接收的狀態值

0 設置成功； 1 設置失敗

- 函數用法

/\* 讀取UART1的發送中斷觸發方式當暫存器為空時產生，接收中斷方式為接收暫存器有資料時發送中斷\*/  
DrvUART\_IntType(0,0) ;

### 8.3.25. DrvUART\_DisableWakeUp

- 函數

void DrvUART\_DisableWakeUp(void);

- 函數功能

關閉UART1的喚醒功能；  
設置寄存器0x40E04[2]=0。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

無

- 函數用法

/\* 關閉UART1喚醒功能 \*/  
DrvUART\_DisableWakeUp();

### 8.3.26. DrvUART\_GetNERR

- 函數

unsigned int DrvUART\_GetNERR(void);

- 函數功能

讀取UART1的雜訊偵測標誌位元(NERR)，讀取寄存器0x40E00[22]的值。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

1：雜訊偵測

0：正常

- 函數用法

/\* 讀取UART1雜訊偵測標誌位元. \*/

```
unsigned char flag ; flag=DrvUART_GetNERR();
```

### 8.3.27. DrvUART\_ClrPERR

- 函數

```
void DrvUART_ClrPERR(void);
```

- 函數功能

清除UART1校驗錯誤標誌位元(PERR) · 寄存器0x40E00[20]=0。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

無

- 函數用法

/\* 清除UART1校驗錯誤標誌位元. \*/

```
DrvUART_ClrPERR();
```

### 8.3.28. DrvUART\_ClrFERR

- 函數

```
void DrvUART_ClrFERR(void);
```

- 函數功能

清除UART1幀錯誤標誌位元(FERR) · 寄存器0x40E00[21]=0。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

無

- 函數用法

/\*清除UART1幀錯誤標誌位元. \*/

DrvUART\_ClrFERR();

### 8.3.29. DrvUART\_ClrOERR

- 函數

void DrvUART\_ClrOERR(void);

- 函數功能

清除UART1溢出錯誤標誌位元(OERR) · 寄存器0x40E00[23]=0。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

無

- 函數用法

/\* 清除UART1溢出錯誤標誌位元. \*/

DrvUART\_ClrOERR();

### 8.3.30. DrvUART\_ClrNERR

- 函數

void DrvUART\_ClrNERR(void);

- 函數功能

清除UART1雜訊偵測標誌位元(NERR) · 寄存器0x40E00[22]=0。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

無

- 函數用法

/\* 清除UART1雜訊偵測標誌位元. \*/

DrvUART\_ClrNERR();

### 8.3.31. DrvUART2\_Open

- 函數

```
unsigned int DrvUART2_Open (  
    unsigned int  uClock  
    E_RAUD_RATE_SETTINGS  uBaudRate ,  
    E_PARITY_SETTINGS      uParity,
```



```
E_DATABITS_SETTINGS    uDataBits,  
unsigned int           uStopBits,  
unsigned int           uOutputPin  
);
```

● **函數功能**

設置UART2的工作頻率源 (除了晶震源為外部晶震(HSXT)或內部晶震(HSRC), UART2除頻設置也會影響到實際UART2的工作頻率源)並根據寫入的串列傳輸速率值自動計算出串列傳輸速率寄存器0x40E18[15:0]的值; 設置UART2的資料校驗模式、資料的位元元數、停止位及TX/RX的通訊用IO口。

設置寄存器0x40E10[7:4] · 0x40E10[2]=1, 0x40E10[0]=1 / 0x40E14[1:0];

寄存器0x40E18[15:0]; 設置IO口寄存器0x4084C[3:0].

● **輸入參數**

uClock[in] : uClock[in] : 設置UART2工作頻率源, 輸入值為UR2CK 的頻率大小, UR2CK是由高速晶振頻率(外部高速HSXT或者內部高速頻率HSRC) 經過UA2CD[3:0]分頻得到, 若UA2CD=1 · 則UR2CK=HSXT(或HSRC), 若UA2CD=2 · 則UR2CK=HSXT/2(或HSRC/2) 依此類推, 以kHz作為單位計算; 輸入範圍 1000~20000

uBaudRate [in] : UART2通訊資料串列傳輸速率

uParity [in] : UART2校驗模式, 分別為無校驗/奇數同位檢查/偶校驗, 設定值範圍: 0~2

0 : 無校驗

1 : 偶校驗

2 : 奇數同位檢查

uDataBits[in] : UART2 資料位元數設置, 設定範圍是: 0~3

0 : 6 bit 數據.

1 : 7 bit 數據.

2 : 8 bit 數據.

3 : 9 bit 數據

uStopBits[in] : UART2 停止位的長度設置, 設定範圍是: 0~3

0 : 0.5 Bit            1: 1 Bit

2 : 1.5 Bit           3 : 2 Bit

uOutputPin [in] : 通訊線TX/RX IO口設置, 設定範圍是: 0~7

0 : Rsv

1 : Rsv

2 : Port 2.2 =TX2, Port 2.3 =RX2

3 : Port 2.6 =TX2, Port 2.7 =RX2

4 : Port 8.2 =TX2, Port 8.3 =RX2

5 : Port 8.6 =TX2, Port 8.7 =RX2

6 : Port 9.2 =TX2, Port 9.3 =RX2

7 : Rsv

● **包含標頭檔**

Peripheral\_lib/DrvUART.h

- 函數返回值

- 0：設置成功.
- 2：時鐘設置錯誤
- 3：串列傳輸速率設置錯誤
- 4：校驗位元設置錯誤
- 5：資料長度設置錯誤
- 6：停止位元長度設置錯誤
- 7：通訊IO設置錯誤

- 函數用法

```
/* 設置UART2 4MHZ/115200bps, 8 位元數據 , 且無校驗 · 停止位為1 · 通訊口為PT2.2/PT2.3*/  
DrvUART2_Open(4147,115200, DRVUART_PARITY_NONE ,DRVUART_DATABITS_8,1,2);  
Note: 因為 UART2 的工作頻率源為 4.147MHz., 所以輸入頻率為 4147, 單位為 kHz..
```

### 8.3.32. DrvUART2\_Enable

- 函數

```
Void DrvUART2_Enable(void);
```

- 函數功能

使能UART2功能 ,設置寄存器0x40E10[2]=1/ 0x40E10[0]=1 。

- 輸入參數

無

- 包含標頭檔

```
Peripheral_lib/DrvUART.h
```

- 函數返回值

無

- 函數用法

```
/*使能UART2功能*/  
DrvUART2_Enable();
```

### 8.3.33. DrvUART2\_Close

- 函數

```
void DrvUART2_Close (void );
```

- 函數功能

關閉UART2功能 ; 清零寄存器0x40E10[2]=0/0x40E10[0]=0;

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

無

- 函數用法

```
/* 關閉UART2 */
```

```
DrvUART2_Close();
```

### 8.3.34. DrvUART2\_EnableInt

- 函數

```
unsigned int DrvUART2_EnableInt(unsigned int uTXIE, unsigned int uRXIE);
```

- 函數功能

UART2的發送(TX)或接收(RX)中斷向量控制. UART2屬於中斷向量HW7;設置寄存器0x40018[19:18]。

- 輸入參數

uTXIE [in] : UART2 發送(TX) 中斷控制

0 : 關閉中斷

1 : 使能中斷

uRXIE [in] : UART2 接收(RX) 中斷控制

0 : 關閉中斷

1 : 使能中斷

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

0 : 設置成功

其他 : 設置失敗

- 函數用法

```
/* 使能UART2發送及接收中斷 */
```

```
DrvUART2_IntType(0,0); //設置UART2中斷觸發方式
```

```
DrvUART2_EnableInt(1,1);//使能UART2的TX/RX中斷向量
```

### 8.3.35. DrvUART2\_IntType

- 函數

```
Unsigned int DrvUART2_IntType(unsigned int uTXIT, unsigned int uRXIT)
```

- 函數功能

設置UART2發送與接收的中斷觸發方式 · 讀取寄存器0x40E10[1]/ 0x40E10[3]

- 輸入參數

uTXIT [in] : UART2發送中斷觸犯方式設置. 輸入範圍 : 0~1

0：當TX 發送暫存器為空時發生中斷要求，寫入資料後中斷標誌位元消失；

1：當TX 發送完一筆資料後發生中斷要求。

uRXIT[in]：UART2接收中斷觸發方式設置。輸入範圍：0~1

0：當RX 接收暫存器有資料時發生中斷要求，讀取資料後中斷標誌位元消失；

1：當RX 接收完一筆資料後發生中斷要求。

- **包含標頭檔**

Peripheral\_lib/DrvUART.h

- **函數返回值**

返回發送及接收的狀態值

0 設置成功； 1 設置失敗

- **函數用法**

/\* 讀取UART2的發送中斷觸發方式當暫存器為空時產生，接收中斷方式為接收暫存器有資料時發送中斷\*/  
DrvUART\_IntType(0,0)；

### **8.3.36. DrvUART2\_GetTxFlag**

- **函數**

unsigned int DrvUART2\_GetTxFlag (void);

- **函數功能**

讀取UART2的發送中斷標誌位元(UTXIF)值，讀取寄存器0x40018[3]的值。

- **輸入參數**

無

- **包含標頭檔**

Peripheral\_lib/DrvUART.h

- **函數返回值**

1: 有中斷產生

0: 無中斷產生

- **函數用法**

/\* 讀取UART2發送中斷標誌位元. \*/

DrvUART2\_GetTxFlag();

### **8.3.37. DrvUART2\_GetRxFlag**

- **函數**

unsigned int DrvUART2\_GetRxFlag (void);

- **函數功能**

讀取UART2接收中斷標誌位元URXIF值，讀取寄存器0x40018[2]的值。

- **輸入參數**

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

1：有中斷要求

0：無中斷要求

- 函數用法

*/\* 讀取UART2接收中斷標誌位元. \*/*

```
unsigned char flag ; flag=DrvUART2_GetRxFlag();
```

### 8.3.38. DrvUART2\_ClrTxFlag

- 函數

void DrvUART2\_ClrTxFlag (void);

- 函數功能

清除UART2發送中斷標誌位元UTXIF 值，清零寄存器0x40018[3]

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

無

- 函數用法

*/\* 清除UART2發送中斷標誌位元. \*/*

```
DrvUART2_ClrTxFlag();
```

### 8.3.39. DrvUART2\_ClrRxFlag

- 函數

void DrvUART2\_ClrRxFlag (void);

- 函數功能

清除UART2接收中斷標誌位元URXIF值，清零寄存器0x40018[2]

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

無

- 函數用法

```
/* 清除UART2接收中斷標誌位元. */  
DrvUART2_ClrRxFlag();
```

### 8.3.40. DrvUART2\_Read

- 函數

```
unsigned int DrvUART2_Read(void);
```

- 函數功能

讀取UART2接收到的資料，讀取寄存器0x40E1C[8:0]的值

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

返回接收暫存器的值。

- 函數用法

```
/* 讀取UART2接收到的資料. */  
unsigned int rx_data ; rx_data=DrvUART2_Read();
```

### 8.3.41. DrvUART2\_Write

- 函數

```
void DrvUART2_Write(unsigned int uData);
```

- 函數功能

寫入數值至UART2的發送暫存器(TXREG)並等待發送，寫入待發送的值至寄存器0x40E1C[24:16]。

- 輸入參數

uData [in]：待發送的資料

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

無

- 函數用法

```
/*UART2發送0x55 */  
DrvUART2_Write(0x55);
```

### 8.3.42. DrvUART2\_EnableWakeUp

- 函數

```
void DrvUART2_EnableWakeUp(void);
```

- 函數功能

使能UART2的喚醒功能，同樣啟動接收喚醒功能只要接收中斷打開；  
設置寄存器0x40E14[2]=1。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

無

- 函數用法

```
/* 使能UART2喚醒功能 */  
DrvUART2_EnableWakeUp();
```

### 8.3.43. DrvUART2\_DisableWakeUp

- 函數

```
void DrvUART2_DisableWakeUp(void);
```

- 函數功能

關閉UART2的喚醒功能；  
設置寄存器0x40E14[2]=0。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

無

- 函數用法

```
/* 關閉UART2喚醒功能 */  
DrvUART2_DisableWakeUp();
```

### 8.3.44. DrvUART2\_Enable\_AutoBaudrate

- 函數

```
void DrvUART2_Enable_AutoBaudrate (void);
```

- 函數功能

使能UART2自動串列傳輸速率功能，設置寄存器0x40E14[3]=1。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

無

- 函數用法

```
/* 使能UART2自動串列傳輸速率功能 */  
DrvUART2_Enable_AutoBaudrate();
```

### 8.3.45. DrvUART2\_Disable\_AutoBaudrate

- 函數

```
void DrvUART2_Disable_AutoBaudrate (void);
```

- 函數功能

關閉UART2自動串列傳輸速率功能，設置寄存器0x40E14[3]=0。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

無

- 函數用法

```
/* 關閉UART2自動串列傳輸速率功能 */  
DrvUART2_Disable_AutoBaudrate();
```

### 8.3.46. DrvUART2\_GetPERR

- 函數

```
unsigned int DrvUART2_GetPERR(void);
```

- 函數功能

讀取UART2校驗錯誤標誌位元(PERR)，讀取寄存器0x40E10[20]的值。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

1：有校驗錯誤

0：無校驗錯誤



- 函數用法

```
/* 讀取UART2校驗錯誤標誌位元. */  
unsigned char flag; flag=DrvUART2_GetPERR();
```

### 8.3.47. DrvUART2\_GetFERR

- 函數

```
unsigned int DrvUART2_GetFERR(void);
```

- 函數功能

讀取UART2幀錯誤標誌位元(FERR) · 讀取寄存器0x40E10[21]的值。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

1：有幀錯誤

0：無幀錯誤

- 函數用法

```
/* 讀取UART2幀錯誤標誌位元. */  
unsigned char flag ; flag=DrvUART2_GetFERR();
```

### 8.3.48. DrvUART2\_GetOERR

- 函數

```
unsigned int DrvUART2_GetOERR(void);
```

- 函數功能

讀取UART2溢出錯誤標誌位元(OERR) · 讀取寄存器0x40E10[23]的值。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

1：有溢出錯誤

0：無溢出錯誤

- 函數用法

```
/* 讀取UART2溢出錯誤標誌位元. */  
unsigned char flag ; flag=DrvUART2_GetOERR();
```

### 8.3.49. DrvUART2\_GetNERR

- 函數

```
unsigned int DrvUART2_GetNERR(void);
```

- 函數功能

讀取UART2的雜訊偵測標誌位元(NERR) · 讀取寄存器0x40E10[22]的值。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

1 : 雜訊偵測

0 : 正常

- 函數用法

```
/* 讀取UART2雜訊偵測標誌位元. */
```

```
unsigned char flag ; flag=DrvUART2_GetNERR();
```

### 8.3.50. DrvUART2\_ClrPERR

- 函數

```
void DrvUART2_ClrPERR(void);
```

- 函數功能

清除UART2校驗錯誤標誌位元(PERR) · 寄存器0x40E10[20]=0。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

無

- 函數用法

```
/* 清除UART2校驗錯誤標誌位元. */
```

```
DrvUART2_ClrPERR();
```

### 8.3.51. DrvUART2\_ClrFERR

- 函數

```
void DrvUART2_ClrFERR(void);
```

- 函數功能

清除UART2幀錯誤標誌位元(FERR) · 寄存器0x40E10[21]=0。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

無

- 函數用法

```
/*清除UART2幀錯誤標誌位元.*/
```

```
DrvUART2_ClrFERR();
```

### 8.3.52. DrvUART2\_ClrOERR

- 函數

```
void DrvUART2_ClrOERR(void);
```

- 函數功能

清除UART2溢出錯誤標誌位元(OERR) · 寄存器0x40E10[23]=0。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

無

- 函數用法

```
/* 清除UART2溢出錯誤標誌位元.*/
```

```
DrvUART2_ClrOERR();
```

### 8.3.53. DrvUART2\_ClrNERR

- 函數

```
void DrvUART2_ClrNERR(void);
```

- 函數功能

清除UART2雜訊偵測標誌位元(NERR) · 寄存器0x40E10[22]=0。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

無

- 函數用法

```
/* 清除UART2雜訊偵測標誌位元. */
```

```
DrvUART2_ClrNERR();
```

### 8.3.54. DrvUART2\_GetABDOVF

- 函數

```
unsigned int DrvUART2_GetABDOVF(void);
```

- 函數功能

讀取UART2自動串列傳輸速率發生器翻轉狀態檢測標誌位元(RXABDF) · 讀取寄存器0x40E14[4]值。

- 輸入參數

無

- 包含標頭檔

```
Peripheral_lib/DrvUART.h
```

- 函數返回值

1：在自動串列傳輸速率檢測模式下串列傳輸速率發生器發生翻轉

0：沒有串列傳輸速率發生器發生翻轉

- 函數用法

```
/* 讀取UART2串列傳輸速率發生器翻轉標誌位元RXABDF. */
```

```
unsigned char flag ; flag=DrvUART2_GetABDOVF();
```

### 8.3.55. DrvUART2\_ClrABDOVF

- 函數

```
unsigned int DrvUART2_ClrABDOVF(void)
```

- 函數功能

接清除UART2的自動串列傳輸速率偵測錯誤標誌位元 · 清零寄存器0x40E14[4].

- 輸入參數

無

- 包含標頭檔

```
Peripheral_lib/DrvUART.h
```

- 函數返回值

無

- 函數用法

```
/* 清除UART2的自動串列傳輸速率偵測錯誤標誌位元*/
```

```
DrvUART2_ClrABDOVF();
```

### 8.3.56. DrvUART2\_TRStatus

- 函數

Unsigned int DrvUART2\_TRStatus(unsigned int uMode)

- 函數功能

讀取UART2發送與接收的狀態 · 讀取寄存器0x40E10[19 :16]值

- 輸入參數

uMode[in] : 輸入範圍 : 0~3

0 : RXBF; 1 : RXBUSY; 2 : TXBF; 3 : TXBUSY

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

返回發送及接收的狀態值

TXBUSY : 0 idle; 1 Busy

TXBF : 0 empty; 1 full

RXBUSY: 0 idle; 1 Busy

RXBF : 0 empty; 1 full

- 函數用法

```
/* 讀取UART2發送狀態位元值並實現查詢方式發送資料*/  
DrvUART2_Write(data);  
While(DrvUART2_TRStatus(2)); //等待TXBF=0
```

### 8.3.57. DrvUART2\_CheckTRMT

- 函數

Unsigned int DrvUART2\_CheckTRMT

- 函數功能

讀取UART2發送狀態位元(TXBF) · 讀取寄存器0x40E10[18]值

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

返回發送狀態位元TXBF的值;

- 函數用法

```
/* 讀取UART2發送狀態位元值並設置查詢方式發送資料*/  
DrvUART2_Write(data);  
While(DrvUART2_CheckTRMT()); //等待TRMT=0
```

### 8.3.58. DrvUART2\_ClkEnable

- 函數

unsigned int DrvUART2\_ClkEnable(unsigned int uclk,unsigned int uprescale) ;

- 函數功能

使能UART2的時鐘源並選擇時鐘源及設置時鐘源的分頻值  
設置寄存器0x40310[21:20]/ 0x40310[18 :16] 。

- 輸入參數

uclk[in] : EUART2 時鐘源設置

0 : 外部晶震高速時鐘

1 : 內部晶震高速時鐘

uprescale[in] : 時鐘源分頻器

0000 EUART2 CLOCK SOURCE/1

0001 EUART2 CLOCK SOURCE/2

0010 EUART2 CLOCK SOURCE/4

0011 EUART2 CLOCK SOURCE/8

0100 EUART2 CLOCK SOURCE/16

0101 EUART2 CLOCK SOURCE/32

0110 EUART2 CLOCK SOURCE/64

0111 EUART2 CLOCK SOURCE/128

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

0 : 設置成功

1 : 設置失敗

- 函數用法

/\* 設置UART2時鐘源為外部時鐘且分頻clk/1 \*/

DrvUART2\_ClkEnable(0,0) ;

### 8.3.59. DrvUART2\_ClkDisable

- 函數

Void DrvUART2\_ClkDisable(void) ;

- 函數功能

關閉UART2時鐘源,設置寄存器0x40310[20]=0 。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

無

- 函數用法

*/\*關閉UART2時鐘源\*/*

DrvUART2\_ClkDisable();

### 8.3.60. DrvUART2\_ConfigIO

- 函數

unsigned char DrvUART2\_ConfigIO(unsigned char ioen,unsigned int uOutputPin) ;

- 函數功能

設置IO口複用為UART2通訊口，及選擇IO口，設置寄存器0x4084C[3 :0]。

- 輸入參數

ioen[in] :IO 口複用功能使能控制

0 : 關閉IO 複用功能

1 : 開啟IO 複用功能

uoutputPin[in] :選擇通訊IO 口

0 : Rsv

1 : Rsv

2 : Port 2.2 =TX2, Port 2.3 =RX2

3 : Port 2.6 =TX2, Port 2.7 =RX2

4 : Port 8.2 =TX2, Port 8.3 =RX2

5 : Port 8.6 =TX2, Port 8.7 =RX2

6 : Port 9.2 =TX2, Port 9.3 =RX2

7 : Rsv

- 包含標頭檔

Peripheral\_lib/DrvUART.h

- 函數返回值

0 : 設置成功

其他 : 設置失敗

- 函數用法

*/\*開啟IO複用為UART2通訊口，並選擇PT2.2/PT2.3\*/*

DrvUART2\_ConfigIO(1,2);

## 9. 儀表放大器 IA

### 9.1. 函數簡介

該部分函數描述IA功能的控制，包含：

- IA 模組功能的啟動與關閉
- IA 配置控制

序號	函數名稱	功能描述
01	DrvIA_SetIInputChannel	設置儀表放大器正端輸入通道與負端輸入通道
02	DrvIA_PInputChannel	設置儀表放大器正端輸入通道
03	DrvIA_NInputChannel	設置儀表放大器負端輸入通道
04	DrvIA_IAGain	設置儀表放大器放大倍率
05	DrvIA_IACHM	設置儀表放大器的chopper模式
06	DrvIA_IAIS	設置儀表放大器的輸入短路開關
07	DrvIA_ENIA	儀表放大器的功能開啟控制



## 9.2. 內部定義常量

### E\_IA\_INPUT\_CHANNEL

識別字	數值	函數功能
IA_Input_AIO0	0x0	設置IA輸入源
IA_Input_AIO1	0x1	設置IA輸入源
IA_Input_REFO	0x2	設置IA輸入源
IA_Input_HighZ	0x3	設置IA輸入源

### E\_IA\_IAGain

識別字	數值	函數功能
IA_IAGain_4	0x0	設置IA放大倍數
IA_IAGain_8	0x1	設置IA放大倍數
IA_IAGain_16	0x2	設置IA放大倍數
IA_IAGain_32	0x3	設置IA放大倍數

## 9.3. 函數說明

### 9.3.1. DrvIA\_SetIInputChannel

- 函數

unsigned int DrvIA\_SetIInputChannel(unsigned int uINP,unsigned int uINN)

- 函數功能

設置儀表放大器正端輸入通道與負端輸入通道。

設置寄存器0x41600[24:26] / 0x41600[16:18]

- 輸入參數

uINP [in] : 儀表放大器正端輸入

0 : AIO0

1 : AIO1

2 : REFO

3 : 高阻抗

uINN [in] : 儀表放大器負端輸入

0 : AIO0

1 : AIO1

2 : REFO

3 : 高阻抗

- 包含標頭檔

Peripheral\_lib/DrvIA.h

- 函數返回值

0 : 設置成功

其他 : 設置失敗

- 函數用法

/\*設置儀表放大器正端輸入為AIO1, 負端輸入為AIO0 \*/

DrvIA\_SetIInputChannel(IA\_Input\_AIO1,IA\_Input\_AIO0);

### 9.3.2. DrvIA\_PInputChannel

- 函數

unsigned int DrvIA\_PInputChannel(unsigned int uINP)

- 函數功能

設置儀表放大器正端輸入通道。

設置寄存器0x41600[24:26]

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvIA.h

- 函數返回值

0：設置成功

其他：設置失敗

- 函數用法

/\* 設置儀表放大器正端輸入通道為AIO1 \*/

DrvIA\_PInputChannel(IA\_Input\_AIO1);

### 9.3.3. DrvIA\_NInputChannel

- 函數

unsigned int DrvIA\_NInputChannel(unsigned int uINN)

- 函數功能

設置儀表放大器負端輸入通道。

設置寄存器0x41600[16:18]

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvIA.h

- 函數返回值

0：設置成功

其他：設置失敗

- 函數用法

/\* 設置儀表放大器負端輸入通道為AIO0 \*/

DrvIA\_NInputChannel(IA\_Input\_AIO0);

### 9.3.4. DrvIA\_IAGain

- 函數

unsigned int DrvIA\_IAGain(unsigned int uIAGain)

- 函數功能

設置儀表放大器放大倍率。設置寄存器0x41600[8:9]。

- 輸入參數

uIAGain [in]：儀表放大器放大倍率選擇：

0：放大4倍

1：放大8倍

2：放大16倍

3：放大32倍

- 包含標頭檔

Peripheral\_lib/DrvIA.h

- 函數返回值

0：設置成功

其他：設置失敗

- 函數用法

/\* 設置儀表放大器放大倍率為4倍 \*/

```
DrvIA_IAGain(IA_IAGain_4);
```

### 9.3.5. DrvIA\_IACHM

- 函數

unsigned int DrvIA\_IACHM(unsigned int uIACHM)

- 函數功能

設置儀表放大器的chopper模式。設置寄存器0x41600[5:4]

- 輸入參數

uIACHM [in]：儀表放大器的chopper模式選擇：

0：NoChopper

1：IndividualInputstage

2：Inputstage

3：Both

- 包含標頭檔

Peripheral\_lib/DrvIA.h

- 函數返回值

0：設置成功

其他：設置失敗

- 函數用法

/\* 設置儀表放大器的chopper模式為No chopper. \*/

```
DrvIA_IACHM(IA_IACHM_NoChopper);
```

### 9.3.6. DrvIA\_IAIS

- 函數

unsigned int DrvIA\_IAIS(unsigned int uAIS)

- 函數功能

設置儀表放大器的輸入短路開關，設置寄存器0x41600[1]

- 輸入參數

uAIS [in]：輸入端短路開關控制。

0：短路開關斷開

1：短路開關閉合

- 包含標頭檔

Peripheral\_lib/DrvIA.h

- 函數返回值

0：設置成功

其他：設置失敗

- 函數用法

```
/* 設置儀表放大器的輸入短路為開(open)*/
```

```
DrvIA_IAIS(0);
```

### 9.3.7. DrvIA\_ENIA

- 函數

unsigned int DrvIA\_ENIA(unsigned int uENIA)

- 函數功能

儀表放大器的功能開啟控制，設置寄存器0x41600[0]

- 輸入參數

uENIA [in]：功能開啟控制

0：關閉

1：開啟

- 包含標頭檔

Peripheral\_lib/DrvIA.h

- 函數返回值

0：設置成功

其他：設置失敗

- 函數用法

```
DrvIA_ENIA(0); //關閉儀表放大器
```

```
DrvIA_ENIA(1); //開啟儀表放大器
```

## 10. 低雜訊運算放大器 OPAMP

### 10.1. 功能簡介

該函數部分描述低雜訊運算放大器 OPAMP 功能的操作

- OPAMP 功能的開關
- OPAMP 輸入埠及輸出埠的控制
- OPAMP 中斷的設置及開關
- OPAMP 輸出信號的反相輸出及濾波控制

序號	函數名稱	函數功能
01	DrvOP_Open	開啟OPAMP功能
02	DrvOP_Close	關閉OPAMP功能
03	DrvOP_PInput	OPAMP正端輸入設置
04	DrvOP_NInput	OPAMP負端輸入設置
05	DrvOP_OPOoutEnable	開啟OPAMP輸出
06	DrvOP_OPOoutDisable	關閉OPAMP輸出
07	DrvOP_OoutputFilter	OPAMP數位濾波輸出設置
08	DrvOP_OutputPinEnable	開啟OPAMP 數位輸出IO pin.
09	DrvOP_OutputPinDisable	關閉OPAMP 數位輸出IO pin
10	DrvOP_OutputInverse	OPAMP數位輸出反相設置
11	DrvOP_OutputWithCHPCK	CPCLK多功能器設置
12	DrvOP_EnableInt	開啟OPAMP中斷
13	DrvOP_DisableInt	關閉OPAMP中斷
14	DrvOP_ReadIntFlag	讀取OPAMP中斷標誌位元
15	DrvOP_ClearIntFlag	清除OPAMP中斷標誌位元
16	DrvOP_Feedback	OPAMP回饋電路設置
17	DrvOP_OPDEN	OPAMP數字輸出功能控制

## 10.2. 內部定義常量

### E\_OUTPUT\_PIN

識別字	數值	功能定義
E_OPO1	0x0	PT3.0作為 OPAMP 數位輸出IO口
E_OPO2	0x1	PT3.1作為 OPAMP 數位輸出IO口

### E\_OPN\_PPIN

識別字	數值	功能定義	識別字	數值	功能定義
E_OPP_AIO2	0x1	AIO2 作為OPAMP 輸入口	E_OPN_AIO3	0x1	AIO3 作為OPAMP 輸入口
E_OPP_AIO4	0x2	AIO4 作為OPAMP 輸入口	E_OPN_AIO5	0x2	AIO5 作為OPAMP 輸入口
E_OPP_DAO	0x4	DAO 作為OPAMP 輸入口	E_OPN_DAO	0x4	DAO 作為OPAMP 輸入口
E_OPP_REF_O_I	0x8	REFO_I 作為OPAMP 輸入口	E_OPN_OPOI	0x8	OPOI 作為OPAMP 輸入口
E_OPP_AIO5	0x10	AIO5 作為OPAMP 輸入口	E_OPN_OPO	0x10	OPO 作為OPAMP 輸入口
E_OPP_AIO6	0x20	AIO6 作為OPAMP 輸入口	E_OPN_OPC	0x20	OPC 作為OPAMP 輸入口
E_OPP_AIO7	0x40	AIO7 作為OPAMP 輸入口	E_OPN_AIO2	0x40	AIO2 作為OPAMP 輸入口
			E_OPN_AIO8	0x80	AIO8 作為OPAMP 輸入口

## 10.3. 函數說明

### 10.3.1. DrvOP\_Open

- 函數

void DrvOP\_Open ( void)

- 函數功能

開啟運算放大器(OPAMP)功能；設置寄存器0x41900[0]=1.

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvOP.h

- 函數返回值

無

- 函數用法

```
/* 使能運算放大器(OPAMP) */  
DrvOP_Open();
```

### 10.3.2. DrvOP\_Close

- 函數

void DrvOP\_Close ( void)

- 函數功能

關閉運算放大器(OPAMP)功能，設置寄存器0x41900[0]=0。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvOP.h

- 函數返回值

無

- 函數用法

```
/* 關閉運算放大器(OPAMP) */  
DrvOP_Close();
```



### 10.3.3. DrvOP\_PInput

- 函數

unsigned int DrvOP\_PInput (uOPPS)

- 函數功能

運算放大器(OPAMP)的正向輸入端設置。

設置寄存器0x41904[22:16]即OPPS[22:16]，每一位對應一路通道，且可以同時設置多路通道有效。

- 輸入參數

uOPPS[6:0]：運算放大器OPAMP 正向輸入端選擇，輸入範圍0x00~0x7F，為0時關閉所有通道。

uOPPS[6]：運算放大器(OPAMP)正向輸入通道6

0：關閉通道，高阻抗狀態

1：開啟通道 AIO7

uOPPS[5]：運算放大器(OPAMP)正向輸入通道5

0：關閉通道，高阻抗狀態

1：開啟通道 AIO6

uOPPS[4]：運算放大器(OPAMP)正向輸入通道4

0：關閉通道，高阻抗狀態

1：開啟通道 AIO5

uOPPS[3]：運算放大器(OPAMP)正向輸入通道3

0：關閉通道，高阻抗狀態

1：開啟通道 REFO\_I

uOPPS[2]：運算放大器(OPAMP)正向輸入通道2

0：關閉通道，高阻抗狀態

1：開啟通道 DAO

uOPPS[1]：運算放大器(OPAMP)正向輸入通道1

0：關閉通道，高阻抗狀態

1：開啟通道 AIO4

uOPPS[0]：運算放大器(OPAMP)正向輸入通道0

0：關閉通道，高阻抗狀態

1：開啟通道 AIO2

- 包含標頭檔

Peripheral\_lib/DrvOP.h

- 函數返回值

0：設置成功

其他：設置失敗

- 函數用法

/\* 設置運算放大器(OPAMP)正向輸入端AIO2與AIO4. \*/

DrvOP\_PInput(0x1|0x2);

### 10.3.4. DrvOP\_NInput

- 函數

unsigned int DrvOP\_NInput (uOPNS)

- 函數功能

運算放大器(OPAMP)負向端輸入埠選擇;

設置寄存器0x41904[7:0]即OPNS[7:0]，且每一位對應一路輸入通道，且可以同時設置多路通道有效。

- 輸入參數

uOPNS[7:0]：運算放大器(OPAMP)負向輸入端選擇，輸入範圍0x00~0xFF,為0時關閉所有通道。

uOPNS[7]：運算放大器(OPAMP)負向輸入通道 7

0：關閉通道，高阻抗狀態

1：開啟通道 AIO8

uOPNS[6]：運算放大器(OPAMP)負向輸入通道 6

0：關閉通道，高阻抗狀態

1：開啟通道 AIO2

uOPNS[5]：運算放大器(OPAMP)負向輸入通道 5

0：關閉通道，高阻抗狀態

1：開啟通道 OPC: 內部連接10pF 電容

uOPNS[4]：運算放大器(OPAMP)負向輸入通道 4

0：關閉通道，高阻抗狀態

1：開啟通道 OPO: 運算放大器OPAMP 內部輸出

uOPNS[3]：運算放大器(OPAMP)負向輸入通道 3

0：關閉通道，高阻抗狀態

1：開啟通道 OPOI: 運算放大器OPAMP 外部輸出

uOPNS[2]：運算放大器(OPAMP)負向輸入通道 2

0：關閉通道，高阻抗狀態

1：開啟通道 DAO DAC的輸出

uOPNS[1]：運算放大器(OPAMP)負向輸入通道 1

0：關閉通道，高阻抗狀態

1：開啟通道 AI5

uOPNS[0]：運算放大器(OPAMP)負向輸入通道 0

0：關閉通道，高阻抗狀態

1：開啟通道 AI3

- 包含標頭檔

Peripheral\_lib/DrvOP.h

- 函數返回值

0：設置成功

其他：設置失敗

- 函數用法

/\*開啟運算放大器(OPAMP)負向輸入端通道AIO3與AIO5.\*/

```
DrvOP_NInput(0x1|0x2);
```

### 10.3.5. DrvOP\_OPOoutEnable

- 函數

```
void DrvOP_OPOoutEnable(void)
```

- 函數功能

打開運算放大器(OPAMP)模擬輸出功能 · 寄存器0x41900[1]=1。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvOP.h

- 函數返回值

無

- 函數用法

```
/* 打開運算放大器(OPAMP)模式輸出*/
```

```
DrvOP_OPOoutEnable();
```

### 10.3.6. DrvOP\_OPOoutDisable

- 函數

```
void DrvOP_OPOoutDisable(void)
```

- 函數功能

運算放大器(OPAMP) 模擬輸出功能關閉 · 寄存器0x41900[1]=0.

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvOP.h

- 函數返回值

無

- 函數用法

```
/* 關閉運算放大器(OPAMP)模擬輸出 */
```

```
DrvOP_OPOoutDisable();
```

### 10.3.7. DrvOP\_OuputFilter

- 函數

```
unsigned int DrvOP_OuputFilter(uFilter)
```

- **函數功能**

運算放大器(OPAMP)數位輸出濾波控制·控制數字輸出是否經過2s延時濾波;  
設置寄存器0x41900[3]。

- **輸入參數**

uFilter[in]

0 : 無濾波

1 : 經過2us delay 濾波

- **包含標頭檔**

Peripheral\_lib/DrvOP.h

- **函數返回值**

0 : 設置成功

其他 : 設置失敗

- **函數用法**

```
/*設置運算放大器(OPAMP) 2us 延時濾波.*/
```

```
DrvOP_OuputFilter(1);
```

### **10.3.8. DrvOP\_OutputPinEnable**

- **函數**

```
unsigned int DrvOP_OutputPinEnable (E_OUTPUT_PIN uPin)
```

- **函數功能**

使能運算放大器(OPAMP)數位輸出埠·並選擇OPAMP輸出IO口 ;  
寄存器0x41900[2]=1,且設置寄存器0x40840[19:18]。

- **輸入參數**

uPin [in]

0 : PT3.0

1 : PT3.1

- **包含標頭檔**

Peripheral\_lib/DrvOP.h

- **函數返回值**

0 : 設置成功

其他 : 設置失敗

- **函數用法**

```
/* 使能運算放大器(OPAMP)數位輸出· IO= PT3.0*/
```

```
DrvOP_OutputPinEnable(0);
```

### **10.3.9. DrvOP\_OutputPinDisable**

- 函數

void DrvOP\_OutputPinDisable (void)

- 函數功能

關閉運算放大器(OPAMP)數位輸出功能及OPAMP輸出IO口功能；  
寄存器0x41900[2]=0, 寄存器0x40840[18]=0。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvOP.h

- 函數返回值

無

- 函數用法

```
/* 關閉運算放大器(OPAMP)數位輸出*/  
DrvOP_OutputPinDisable();
```

### 10.3.10. DrvOP\_OutputInverse

- 函數

unsigned int DrvOP\_OutputInverse(uInv)

- 函數功能

運算放大器(OPAMP)數位輸出信號輸出反相控制，設置寄存器0x41900[5].

- 輸入參數

uInv [in]

0：正常輸出

1：輸出反相

- 包含標頭檔

Peripheral\_lib/DrvOP.h

- 函數返回值

0：設置成功

其他：設置失敗

- 函數用法

```
/* 使能運算放大器(OPAMP)數位輸出反相*/  
DrvOP_OutputInverse(1);
```

### 10.3.11. DrvOP\_OutputWithCHPCK

- 函數

unsigned int DrvOP\_OutputWithCHPCK (uCHPCK)

- 函數功能

運算放大器(OPAMP)數位輸出信號OPO1/OPO2 是否經過CHPCK多功能器設置。  
設置寄存器0x41900[6],且在使能該功能前需要打開ADC clock ,才能正確啟動該功能。

- 輸入參數

uCHPCK [in]

0 : 不帶 CHPCK 多功能器, OPO1/OPO2 輸出等於 OPOD

1 : 帶 CHPCK 多功能器 r, OPO1/OPO2 輸出一個基於 CHPCK 的高頻信號

- 包含標頭檔

Peripheral\_lib/DrvOP.h

- 函數返回值

0 : 設置成功

其他 : 設置失敗

- 函數用法

```
/* 設置運算放大器(OPAMP)數位輸出帶CHPCK */  
DrvADC_ClkEnable(0,0); //開啟ADC 時鐘源  
DrvOP_OutputWithCHPCK(1); //使能CHPCK 多功能器
```

### 10.3.12. DrvOP\_EnableInt

- 函數

void DrvOP\_EnableInt (void)

- 函數功能

使能運算放大器(OPAMP)中斷向量 , 運算放大器(OPAMP)處於中斷向量HW3 ;  
設置寄存器0x4000c[16]=1.

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvOP.h

- 函數返回值

無

- 函數用法

```
/* 使能運算放大器(OPAMP)中斷 */  
DrvOP_EnableInt();
```

### 10.3.13. DrvOP\_DisableInt

- 函數

void DrvOP\_DisableInt (void)

- 函數功能

關閉運算放大器(OPAMP)中斷向量 , 清零寄存器0x4000c[16]=0;

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvOP.h

- 函數返回值

無

- 函數用法

```
/* 關閉運算放大器(OPAMP)中斷 */  
DrvOP_DisableInt();
```

### 10.3.14. DrvOP\_ReadIntFlag

- 函數

unsigned int DrvOP\_ReadIntFlag (void)

- 函數功能

讀取運算放大器(OPAMP)中斷標誌位元OPOIF ; 讀取寄存器0x4000c[0]的值。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvOP.h

- 函數返回值

0 : 運算放大器(OPAMP)中斷標誌位元為0 , 無中斷要求

1 : 運算放大器(OPAMP)中斷標誌位元為1 , 有中斷要求

>1: 無效返回值

- 函數用法

```
/* 讀取運算放大器(OPAMP)中斷標誌位元 */  
unsigned char flag ; flag=DrvOP_ReadIntFlag();
```

### 10.3.15. DrvOP\_ClearIntFlag

- 函數

void DrvOP\_ClearIntFlag (void)

- 函數功能

清除運算放大器(OPAMP)中斷要求標誌位元OPOIF。清零寄存器0x4000c[0] =0.

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvOP.h

- 函數返回值

無

- 函數用法

```
/* 清除運算放大器(OPAMP)中斷要求標誌位元 */  
DrvOP_ClearIntFlag();
```

### 10.3.16. DrvOP\_Feedback

- 函數

unsigned int DrvOP\_Feedback(uFeedback)

- 函數功能

運算放大器(OPAMP)回饋電路或採樣電容連接設置。設置寄存器0x41900[4]。

- 輸入參數

uFeedback [in]

0：電容作為集成電容器。下端連接至 OPOI

1：電容作為採樣電容。下端連接至 VSSA

- 包含標頭檔

Peripheral\_lib/DrvOP.h

- 函數返回值

0：設置成功

其他：設置失敗

- 函數用法

```
/*運算放大器(OPAMP)回饋電容連接到 OPOI */  
DrvOP_Feedback(0);
```



### 10.3.17. DrvOP\_OPDEN

- 函數

unsigned char DrvOP\_OPDEN(uOPDEN)

- 函數功能

OPAMP數字輸出功能控制，寫入寄存器0x41900[2]。

- 輸入參數

uOPDEN [in]：OPAMP數字輸出功能控制，輸入範圍：0~1

0：關閉

1：開啟

- 包含標頭檔

Peripheral\_lib/ DrvOP.h

- 函數返回值

0：設置成功

1：設置失敗

- 函數用法

/\* 開啟OPAMP數字輸出功能控制\*/

DrvOP\_OPDEN(1);

## 11. 電源管理 PMU

### 11.1. 函數簡介

該部分函數描述電源管理系統的控制，包含：

- VDDA 電壓的控制
- 帶隙(BANDGAP)參考電壓的控制
- Charge Pump升壓電路控制
- REFO 電壓的控制
- Low Power模式及ADC analog ground的控制

序號	函數名稱	功能描述
01	DrvPMU_VDDA_Voltage	VDDA電壓值設置
02	DrvPMU_VDDA_LDO_Ctrl	VDDA LDO 使能控制
03	DrvPMU_BandgapEnable	帶隙參考電壓開啟控制
04	DrvPMU_BandgapDisable	帶隙參考電壓關閉控制
05	DrvPMU_REFO_Enable	類比參考電壓REFO開啟控制
06	DrvPMU_REFO_Disable	類比參考電壓REFO關閉控制
07	DrvPMU_AnalogGround	ADC模擬共地端控制
08	DrvPMU_LDO_LowPower	VDD LDO 低功耗控制
09	DrvPMU_EnableENLVD	致能 LVD 低電壓偵測
10	DrvPMU_DisableENLVD	關閉 LVD 低電壓偵測
11	DrvPMU_SetLVDVS	設置 LVDVS, LVD 正端電壓源
12	DrvPMU_SetLVD12	設置 LVD12, LVD 負端電壓源
13	DrvPMU_SetLVDS	設置 LVDS 正端電壓值
14	DrvPMU_GetLVDO	讀取 LVDO 暫存器

## 11.2. 內部定義常量

### E\_VDDA\_OUTPUT\_VOLTAGE

識別字	數值	函數功能
E_VDDA2_4	0x0	設置VDDA=2.4V
E_VDDA2_6	0x1	設置VDDA=2.6V
E_VDDA2_9	0x2	設置VDDA=2.9V
E_VDDA3_2	0x3	設置VDDA=3.2V

### E\_VDDA\_LDO\_ENABLE\_CONTROL

識別字	數值	函數功能
E_HighZ	0x0	設置VDDA=0v
E_VDD3V	0x1	設置VDDA=VDD3V
E_PullDown	0x2	設置VDDA=0v
E_LDO	0x3	設置VDDA=2.4~3.3V可調

## 11.3. 函數說明

### 11.3.1. DrvPMU\_VDDA\_Voltage

- 函數

unsigned int DrvPMU\_VDDA\_Voltage(E\_VDDA\_OUTPUT\_VOLTAGE uVoltage)

- 函數功能

設置VDDA輸出電壓值，設置寄存器0x40400[19:18].

- 輸入參數

uVoltage [in] :VDDA電壓選擇. 輸入範圍 : 0~3

0 : 2.4V

1 : 2.6V

2 : 2.9V

3 : 3.2V

- 包含標頭檔

Peripheral\_lib/DrvPMU.h

- 函數返回值

0 : 設置成功

其他 : 設置失敗

- 函數用法

```
/* 設置VDDA =2.6V. */
```

```
DrvPMU_VDDA_Voltage(E_VDDA2_6);
```

### 11.3.2. DrvPMU\_VDDA\_LDO\_Ctrl

- 函數

unsigned int DrvPMU\_VDDA\_LDO\_Ctrl(E\_VDDA\_LDO\_ENABLE\_CONTROL uCtrl)

- 函數功能

設置VDDA穩壓電壓輸入源；該功能影響到VDDA輸出電壓，所以配合VDDA設置一起使用；  
設置寄存器0x40400[17:16] .

- 輸入參數

uCtrl [in] : VDDA穩壓電壓輸入源選擇. 輸入範圍 : 0~3

0 : 高阻態 ( High Z ) ,VDDA=0

1 : VDD3V · VDDA=VDD3V

2 : 弱下拉(Weak pull down) · VDDA=0

3 : 可調穩壓(LDO),此模式VDDA才能可調。

- 包含標頭檔

Peripheral\_lib/DrvPMU.h

- 函數返回值

0 : 設置成功  
其他 : 設置失敗

- 函數用法

```
/* 設置VDDA LDO 使能.且設置VDDA=2.7V*/  
DrvPMU_VDDA_LDO_Ctrl(E_LDO);  
DrvPMU_VDDA_Voltage(E_VDDA2_7);
```

### 11.3.3. DrvPMU\_BandgapEnable

- 函數

```
void DrvPMU_BandgapEnable(void)
```

- 函數功能

使能帶隙(Bandgap)參考電壓 · 設置寄存器0x40400[4]=1 .

- 輸入參數

無

- 包含標頭檔

```
Peripheral_lib/DrvPMU.h
```

- 函數返回值

無

- 函數用法

```
/* 使能帶隙參考電壓*/  
DrvPMU_BandgapEnable();
```

### 11.3.4. DrvPMU\_BandgapDisable

- 函數

void DrvPMU\_BandgapDisable(void)

- 函數功能

關閉帶隙(Bandgap)參考電壓功能，設置寄存器0x40400[4]=0。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvPMU.h

- 函數返回值

無

- 函數用法

```
/* 關閉帶隙參考電壓. */
```

```
DrvPMU_BandgapDisable();
```

### 11.3.5. DrvPMU\_REFO\_Enable

- 函數

void DrvPMU\_REFO\_Enable(void)

- 函數功能

類比參考電壓REFO使能控制，輸出1.2v電壓,但是需要先開啟帶隙參考電壓；設置寄存器0x40400[1] =1。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvPMU.h

- 函數返回值

無

- 函數用法

```
/* 使能類比參考電壓REFO. */
```

```
DrvPMU_BandgapEnable(); //開啟帶隙參考電壓
```

```
DrvPMU_REFO_Enable(); //開啟類比參考電壓REFO
```

### 11.3.6. DrvPMU\_REFO\_Disable

- 函數

void DrvPMU\_REFO\_Disable(void)

- 函數功能

類比參考電壓REFO關閉控制，關閉1.2v電壓輸出；設置寄存器0x40400[1] =0。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvPMU.h

- 函數返回值

無

- 函數用法

```
/*關閉類比參考電壓REFO.*/
```

```
DrvPMU_REFO_Disable();
```

### 11.3.7. DrvPMU\_AnalogGround

- 函數

```
unsigned int DrvPMU_AnalogGround(uAG)
```

- 函數功能

ADC模擬地輸入源選擇，設置寄存器0x40400[3]。

- 輸入參數

uAG [in]

0：外部

1：使能buffer及使用內部（配合ADC一起使用）

- 包含標頭檔

Peripheral\_lib/DrvPMU.h

- 函數返回值

0：設置成功

其他：設置失敗

- 函數用法

```
/* 設置ADC 模擬地輸入來源外部 */
```

```
DrvPMU_AnalogGround(0);
```

### 11.3.8. DrvPMU\_LDO\_LowPower

- 函數

unsigned int DrvPMU\_LDO\_LowPower(uLP)

- 函數功能

VDD LDO 低功耗控制，設置寄存器0x40400[0]

- 輸入參數

uLP [in]

0：正常功耗（從sleep模式喚醒後需要設置0）

1：低功耗

- 包含標頭檔

Peripheral\_lib/DrvPMU.h

- 函數返回值

0：設置成功

其他：設置失敗

- 函數用法

/\* 使能 LDO 低功耗模式 \*/

DrvPMU\_LDO\_LowPower(1);

### 11.3.9. DrvPMU\_EnableENLVD

- 函數

void DrvPMU\_EnableENLVD (void)

- 函數功能

致能LVD低電壓偵測；設置寄存器0x40408[0] =1

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvPMU.h

- 函數返回值

無

- 函數用法

/\* 致能LVD低電壓偵測\*/

DrvPMU\_EnableENLVD();

### 11.3.10. DrvPMU\_DisableENLVD

- 函數

void DrvPMU\_DisableENLVD (void)

- 函數功能



關閉LVD低電壓偵測；設置寄存器0x40408[0] =0

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvPMU.h

- 函數返回值

無

- 函數用法

/\* 關閉LVD低電壓偵測\*/

DrvPMU\_DisableENLVD();

### 11.3.11. DrvPMU\_SetLVDVS

- 函數

void DrvPMU\_SetLVDVS(unsigned char uLVDVS)

- 函數功能

設置LVDVS, LVD正端電壓源；設置寄存器0x40408[1]

- 輸入參數

uLVDVS [in]：輸入範圍：0~1

0：VDD3V

1：VLCD

- 包含標頭檔

Peripheral\_lib/DrvPMU.h

- 函數返回值

無

- 函數用法

/\* 設置LVD正端電壓源為VDD3V \*/

DrvPMU\_SetLVDVS(0);

### 11.3.12. DrvPMU\_SetLVD12

- 函數

void DrvPMU\_SetLVD12(unsigned char uLVD12)

- 函數功能

設置LVD12, LVD負端電壓源；設置寄存器0x40408[2]

- 輸入參數

uLVD12 [in]：輸入範圍：0~1

0：V12\_BOR

1：V12\_BGR

- 包含標頭檔

Peripheral\_lib/DrvPMU.h

- 函數返回值

無

- 函數用法

/\* 設置LVD負端電壓源為V12\_BGR \*/

DrvPMU\_SetLVDVS(1);

### 11.3.13. DrvPMU\_SetLVDS

- 函數

void DrvPMU\_SetLVDS(unsigned int uLVDS)

- 函數功能

設置LVDS正端電壓值；設置寄存器0x40408[7 :4]

- 輸入參數

uLVDS [in] : 輸入範圍 : 0~15

0 : 外部輸入電壓LVDIN.

1 : 2.0V

2 : 2.1V

3 : 2.2V

4 : 2.3V

5 : 2.4V

6 : 2.5V

7 : 2.6V

8 : 2.7V

9 : 2.8V

10 : 2.9V

11 : 3.0V

12 : 3.1V

13 : 3.2V

14 : 3.3V

15 : 3.4V

- 包含標頭檔

Peripheral\_lib/DrvPMU.h

- 函數返回值

無

- 函數用法

/\* 設置LVDS正端電壓值為2.0V \*/

DrvPMU\_SetLVDS(1);

### 11.3.14. DrvPMU\_GetLVDO

- 函數

unsigned int DrvPMU\_GetLVDO(void)

- 函數功能

讀取LVDO暫存器，讀取寄存器0x40408[16]的值。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvPMU.h

- 函數返回值

0 : 當負端電壓 > 正端電壓時, LVDO=0

1 : 當正端電壓 > 負端電壓時, LVDO=1

- 函數用法

/\* 讀取LVDO暫存器 \*/

DrvPMU\_GetLVDO();

unsigned char flag ; flag= DrvPMU\_GetLVDO();

## 12. 數模轉換器 DAC

### 12.1. 函數功能簡介

該部分函數介紹 DAC 功能的設置

--DAC 功能的啟動與關閉

--DAC 輸入埠的設置

--DAC 的輸出口設置

--DAC 輸出電壓的設置

序號	函數名稱	功能描述
01	DrvDAC_Open	開啟DAC並配置相關參數
02	DrvDAC_Close	關閉DAC及DAC IO口輸出功能
03	DrvDAC_Enable	開啟DAC
04	DrvDAC_Disable	關閉DAC
05	DrvDAC_EnableOutput	開啟DAC輸出
06	DrvDAC_DisableOutput	關閉DAC輸出
07	DrvDAC_PInput	DAC正端參考輸入設置
08	DrvDAC_NInput	DAC負端參考輸入設置
09	DrvDAC_DABIT	D/A轉換輸出電壓比例設置
10	DrvDAC_DALH	12bit resistance ladder(DAC)內部輸出控制

## 12.2. 內部定義常量

### E\_DAC\_INPUT

識別字	數值	函數功能
E_DAC_PVDDA	0x0	正端信號輸入端
E_DAC_PVDD18	0x1	正端信號輸入端
E_DAC_PREFO_I	0x2	正端信號輸入端
E_DAC_POPO	0x3	正端信號輸入端
E_DAC_PAIO4	0x4	正端信號輸入端
E_DAC_PAIO5	0x5	正端信號輸入端
E_DAC_PAIO6	0x6	正端信號輸入端
E_DAC_PAIO7	0x7	正端信號輸入端
E_DAC_NVSSA	0x0	負端信號輸入端
E_DAC_NREFO_I	0x1	負端信號輸入端
E_DAC_NOPO	0x2	負端信號輸入端
E_DAC_NAIO7	0x3	負端信號輸入端

## 12.3. 函數說明

### 12.3.1. DrvDAC\_Open

- 函數

unsigned int DrvDAC\_Open(E\_DAC\_INPUT uPininput ,E\_DAC\_INPUT uNinput, uDAO)

- 函數功能

使能DAC及設置DAC正向、負向參考電壓輸入埠，並且設置DAC輸出分壓的初始比例值；  
設置寄存器0x41700[0]=1b / 0x41700[1]=1b / 0x41700[19:16] / 0x41700[26:24] / 0x41704[11:0].

- 輸入參數

uPininput [in] : DAC 正向參考輸入端選擇：

- 0 : VDDA
- 1 : VDD18
- 2 : REFO\_I
- 3 : OPO
- 4 : AIO4
- 5 : AIO5
- 6 : AIO6
- 7 : AIO7

uNinput [in] : DAC 負向參考輸入端選擇：.

- 0 : VSSA
- 1 : REFO\_I
- 2 : OPO
- 3 : AIO7

uDAO [in] : DAO[11:0] 輸出電壓值的分壓比例設置DAO/4095. 輸入範圍：0~4095

- 包含標頭檔

Peripheral\_lib/DrvDAC.h

- 函數返回值

- 0 : 設置成功
- 其他：設置失敗

- 函數用法

```
/* 使能DAC, 設定正向輸入端為AIO6, 負向輸入端為VSSA ,DAO=6 */  
DrvDAC_Open(E_DAC_PAIO6, E_DAC_NVSSA ,6 );
```

### 12.3.2. DrvDAC\_Close

- 函數

void DrvDAC\_Close(void)

- 函數功能

關閉DAC及關閉DAC電壓輸出，設置寄存器0x41700[1:0]=00b;

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvDAC.h

- 函數返回值

無

- 函數用法

```
/* 關閉 DAC及輸出功能*/
```

```
DrvDAC_Close();
```

### 12.3.3. DrvDAC\_Enable

- 函數

void DrvDAC\_Enable(void)

- 函數功能

開啟DAC, 設置寄存器0x41700[0]=1.

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvDAC.h

- 函數返回值

無

- 函數用法

```
/* 開啟 DAC */
```

```
DrvDAC_Enable();
```

### 12.3.4. DrvDAC\_Disable

- 函數

void DrvDAC\_Disable(void)

- 函數功能

關閉DAC，設置寄存器0x41700[0]=0;

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvDAC.h

- 函數返回值

無

- 函數用法

/\* 關閉 DAC \*/

DrvDAC\_Disable();

### 12.3.5. DrvDAC\_EnableOutput

- 函數

void DrvDAC\_EnableOutput(void)

- 函數功能

使能DAC 輸出，設置寄存器0x41700[1]=1;

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvDAC.h

- 函數返回值

無

- 函數用法

/\* 使能DAC輸出 \*/

DrvDAC\_EnableOutput();

### 12.3.6. DrvDAC\_DisableOutput

- 函數

void DrvDAC\_DisableOutput (void)

- 函數功能

關閉DAC 輸出，設置寄存器0x41700[1]=0.

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvDAC.h

- 函數返回值

無

- 函數用法

/\*關閉 DAC 輸出 \*/



```
DrvDAC_DisableOutput();
```

### 12.3.7. DrvDAC\_PInput

- 函數

```
unsigned int DrvDAC_Pinput(E_DAC_INPUT uPinput)
```

- 函數功能

DAC正向輸入端選擇設置，設置寄存器0x41700[19:16].

- 輸入參數

uPinput [in]: DAC 正向輸入端選擇. 輸入範圍 : 0~7

0 : VDDA

1 : VDD18

2 : REFO\_I

3 : OPO

4 : AIO4

5 : AIO5

6 : AIO6

7 : AIO7

- 包含標頭檔

Peripheral\_lib/DrvDAC.h

- 函數返回值

0 : 設置成功

其他 : 設置失敗

- 函數用法

```
/* 設置DAC正向輸入端為VDDA. */
```

```
DrvDAC_Pinput(0);
```

### 12.3.8. DrvDAC\_NInput

- 函數

```
unsigned int DrvDAC_Ninput(E_DAC_INPUT uNinput)
```

- 函數功能

DAC 負向輸入端選擇設置，設置寄存器0x41700[26:24]

- 輸入參數

uNinput [in]: DAC 負向輸入端選擇. 輸入範圍 : 0~3

0 : VSSA

1 : REFO\_I

2 : OPO

3 : AIO7

- 包含標頭檔

Peripheral\_lib/DrvDAC.h

- 函數返回值

0 : 設置成功

其他 : 設置失敗

- 函數用法

/\* 設定 DAC 負向輸入端為VSSA. \*/

DrvDAC\_Ninput(E\_DAC\_VSSA);

### 12.3.9. DrvDAC\_DABIT

- 函數

unsigned int DrvDAC\_DABIT(uDABIT)

- 函數功能

DAO[11:0] 輸出電壓值的分壓比例設置即DAO/4095 · 寫入寄存器0x41704[11:0]。

- 輸入參數

uDABIT [in] : 寫入DAO[11:0] · D/A轉換輸出電壓比例值設置uDABIT/4095. 輸入範圍: 0~4095

- 包含標頭檔

Peripheral\_lib/DrvDAC.h

- 函數返回值

0 : 設置成功

其他 : 設置失敗

- 函數用法

/\* DAO [11:0] =5 \*/

DrvDAC\_DABIT(5);

### 12.3.10. DrvDAC\_DALH

- 函數

unsigned char DrvDAC\_DALH(uDALH)

- 函數功能

12bit resistance ladder(DAC)內部輸出控制 · 寫入寄存器0x41700[2]。

- 輸入參數

uDALH [in] : . 輸入範圍: 0~1

0 : 關閉

1 : 開啟

- 包含標頭檔

Peripheral\_lib/DrvDAC.h

- 函數返回值

0 : 設置成功

1 : 設置失敗

- 函數用法

/\* 開啟12bit resistance ladder(DAC)內部輸出控制 \*/

DrvDAC\_DALH(1);

## 13. 即時時鐘 RTC

### 13.1. 函數簡介

函數描述對 RTC 系統的控制包含：

- RTC 時鐘源的設置
- RTC 的啟動與關閉
- RTC 的時間格式設置及當前時間的寫入與讀取操作
- RTC 的鬧鐘功能的設置

序號	函數名稱	功能描述
01	DrvRTC_SetFrequencyCompensation	設置 RTC 頻率補償值
02	DrvRTC_WriteEnable	寫入寄存器解鎖碼，解鎖寄存器寫入操作
03	DrvRTC_WriteDisable	清除寄存器解鎖碼，寄存器無法寫入
04	DrvRTC_AlarmEnable	開啟RTC鬧鐘功能
05	DrvRTC_AlarmDisable	關閉RTC鬧鐘功能
06	DrvRTC_PeriodicTimeEnable	開啟RTC定時喚醒功能及設置定時喚醒時間
07	DrvRTC_PeriodicTimeDisable	關閉RTC定時喚醒功能
08	DrvRTC_Enable	開啟RTC功能
09	DrvRTC_Disable	關閉RTC功能
10	DrvRTC_HourFormat	設置時間格式為12小時制或24小時制
11	DrvRTC_ReadState	讀取RTC的狀態位元
12	DrvRTC_ClearState	清除RTC的狀態位元
13	DrvRTC_EnableInt	開啟RTC中斷功能
14	DrvRTC_DisableInt	關閉RTC中斷功能
15	DrvRTC_ReadIntFlag	讀取RTC中斷標誌位元
16	DrvRTC_ClearIntFlag	清除RTC中斷標誌位元
17	DrvRTC_Write	設定'當前/鬧鐘'的時間和日期
18	DrvRTC_Read	讀取'當前/鬧鐘'的時間和日期
19	DrvRTC_ClkConfig	RTC時鐘源的設置控制
20	DrvRTC_EnableWUEn	開啟RTC喚醒功能
21	DrvRTC_DisableWUEn	關閉RTC喚醒功能

### 13.2. 內部定義常量

#### E\_DRVRTC\_CLOCK\_SOURCE

識別字	數值	功能意義
E_EXT_CK	0	RTC時鐘源由外部低頻時鐘提供
E_INT_CK	1	RTC時鐘源有內部低頻時鐘提供

#### E\_DRVRTC\_TICK

識別字	數值	功能意義
E_DRVRTC_1_128_SEC	0	定時喚醒除頻 1/128
E_DRVRTC_1_64_SEC	1	定時喚醒除頻 1/64
E_DRVRTC_1_32_SEC	2	定時喚醒除頻 1/32
E_DRVRTC_1_16_SEC	3	定時喚醒除頻 1/16
E_DRVRTC_1_8_SEC	4	定時喚醒除頻 1/8
E_DRVRTC_1_4_SEC	5	定時喚醒除頻 1/4
E_DRVRTC_1_2_SEC	6	定時喚醒除頻 1/2
E_DRVRTC_1_SEC	7	定時喚醒除頻 1

#### E\_DRVRTC\_HOUR\_FORMAT

識別字	數值	功能意義
E_DRVRTC_HOUR_12	1	12小時制
E_DRVRTC_HOUR_24	0	24小時制

#### E\_DRVRTC\_TIME\_SELECT

識別字	數值	功能意義
DRVRTC_CURRENT_TIME	0	選擇'當前時間'選項
DRVRTC_ALARM_TIME	1	選擇'鬧鐘時間'選項

#### E\_DRVRTC\_FLAG

識別字	數值	功能意義
E_DRVRTC_ALARM_FLAG	0	鬧鐘標誌位元
E_DRVRTC_PERIODIC_FLAG	1	定時時間標誌位元
E_DRVRTC_CLEAR_ALL	2	鬧鐘標誌位元和定時時間標誌位元

### 13.3. 函數說明

注意：需要先使能 RTC clock，再寫入解鎖碼(對寄存器 0X41A00[23:20]寫 0110b)，然後才能正確寫入寄存器

#### 13.3.1. DrvRTC\_SetFrequencyCompensation

- 函數

```
unsigned int DrvRTC_SetFrequencyCompensation(  
    unsigned int uFrequencyCom );
```

- 函數功能

設置RTC時鐘頻率補償值，設置寄存器0x41a04[22:16]。

- 輸入參數

uFrequencyCom [in]：設置RTC時鐘頻率補償值，設定範圍是：0~0x7f

01111111 : +126 ppm

01111110 : +124 ppm

|:

0000001 : +2 ppm

0000000 : +0 ppm

1000000 : - 0 ppm

1000001 : - 2 ppm

|:

11111110 : -124 ppm

11111111 : -126 ppm

- 包含標頭檔

Peripheral\_lib/DrvRTC.h

- 函數返回值

0：設置成功

其他：設置失敗

- 函數用法

```
/*設置頻率補償為 -2 PPM */
```

```
DrvRTC_SetFrequencyCompensation(0x41);
```

#### 13.3.2. DrvRTC\_WriteEnable

- 函數

```
void DrvRTC_WriteEnable(void);
```

- 函數功能

寫入解鎖碼恢復RTC寄存器寫入操作。對寄存器0x41A00[23:20]寫入0110b

注意：必須要寫入解鎖碼才能對RTC的其它寄存器進行寫入

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvRTC.h

- 函數返回值

無

- 函數用法

/\* RTC寄存器解鎖，解鎖後才能寫入操作 \*/

DrvRTC\_WriteEnable();

### 13.3.3. DrvRTC\_WriteDisable

- 函數

void DrvRTC\_WriteDisable(void);

- 函數功能

清除RTC解鎖碼，重新鎖住寄存器不允許寫入，對寄存器0x41A00[23:20]寫入0000b

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvRTC.h

- 函數返回值

無

- 函數用法

/\* 鎖住RTC寄存器，不允許寫入操作\*/

DrvRTC\_WriteDisable();

### 13.3.4. DrvRTC\_AlarmEnable

- 函數

void DrvRTC\_AlarmEnable (void);

- 函數功能

使能鬧鐘(Alarm)功能；設置寄存器0x41A00[3]=1.

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvRTC.h

- 函數返回值

無

- 函數用法

```
/* 使能 RTC 鬧鐘(Alarm)功能*/  
DrvRTC_AlarmEnable();
```

### 13.3.5. DrvRTC\_AlarmDisable

- 函數

```
void DrvRTC_AlarmDisable (void);
```

- 函數功能

關閉鬧鐘(Alarm)功能；設置寄存器0x41A00[3]=0.

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvRTC.h

- 函數返回值

無

- 函數用法

```
/*關閉鬧鐘(Alarm)功能*/  
DrvRTC_AlarmDisable();
```

### 13.3.6. DrvRTC\_PeriodicTimeEnable

- 函數

```
unsigned int DrvRTC_PeriodicTimeEnable (E_DRVRTC_TICK uPeriodicTimer);
```

- 函數功能

使能定時喚醒(Periodic Time)功能並設置定時喚醒的時間；  
設置寄存器0x41A04[2:0] 及 寄存器0x41A00[5]=1,0x41A00[4]=1.

- 輸入參數

uPeriodicTimer[in]：定時喚醒除頻器設置

0: 1/128

1: 1/64

2: 1/32

3: 1/16

4: 1/8

5: 1/4

6: 1/2

7: 1



- 包含標頭檔

Peripheral\_lib/DrvRTC.h

- 函數返回值

0 : 設置成功

其他 : 設置失敗

- 函數用法

/\* 使能RTC定時喚醒功能 · 設置定時喚醒時間為1/16second\*/

DrvRTC\_PeriodicTimeEnable(3);

### 13.3.7. DrvRTC\_PeriodicTimeDisable

- 函數

void DrvRTC\_PeriodicTimeDisable (void);

- 函數功能

關閉定時喚醒(Periodic Time)功能 · 設置寄存器0x41A00[5]=0 / 0x41A00[4]=0。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvRTC.h

- 函數返回值

無

- 函數用法

/\* 關閉定時喚醒(Periodic time)功能\*/

DrvRTC\_PeriodicTimeDisable();

### 13.3.8. DrvRTC\_Enable

- 函數

void DrvRTC\_Enable (void);

- 函數功能

使能RTC 功能 · 設置寄存器0x41A00[0]=1。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvRTC.h

- 函數返回值

無

- 函數用法

```
/* 使能 RTC 功能*/  
DrvRTC_Enable();
```

### 13.3.9. DrvRTC\_Disable

- 函數

```
void DrvRTC_Disable (void);
```

- 函數功能

關閉RTC功能，設置寄存器0x41A00[0]=0。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvRTC.h

- 函數返回值

無

- 函數用法

```
/* 關閉RTC功能 */  
DrvRTC_Disable();
```

### 13.3.10. DrvRTC\_HourFormat

- 函數

```
unsigned int DrvRTC_HourFormat(E_DRVRTC_HOUR_FORMAT uHourFormat);
```

- 函數功能

設置小時格式為12小時制或24小時制，設置寄存器0x41A00[2]。

- 輸入參數

uHourFormat[in]：小時格式設置

0：24小時制

1：12小時制

- 包含標頭檔

Peripheral\_lib/DrvRTC.h

- 函數返回值

0：設置成功

其他：設置失敗

- 函數用法

```
/* 設置12小時制 */  
DrvRTC_HourFormat(1);
```

### 13.3.11. DrvRTC\_ReadState

- 函數

unsigned int DrvRTC\_ReadState(void);

- 函數功能

讀取RTC狀態標誌位元，讀取寄存器0x41A00[19:16]值

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvRTC.h

- 函數返回值

返回值有效範圍是0x0~0xf，每一位元值對應一個標誌位元狀態值

Bit 0 : RTC 鬧鐘標誌位元TAF

Bit 1 : RTC 喚醒功能標誌位元WUF

Bit 2 : RTC 定時喚醒標誌位元PTF

Bit 3 : RTC 閏年標誌位元LPYF

- 函數用法

/\* 查詢RTC鬧鐘狀態位元 \*/

Sample code 1 :

```
If (DrvRTC_ReadState()&0x1)
    //RTC alarm triggered
else
```

```
    // RTC Wakeup triggered
```

Sample code 2 :

```
flag = DrvRTC_ReadState();
```

### 13.3.12. DrvRTC\_ClearState

- 函數

unsigned int DrvRTC\_ClearState(E\_DRVRTC\_FLAG uFlag);

- 函數功能

清除RTC狀態標誌位元，清零寄存器0x41A00[19:16]值

- 輸入參數

uFlag[in] 待清除狀態位元選擇

0 : 清除鬧鐘標誌位元TAF

1 : 清除定時喚醒標誌位元PTF

2: 清除鬧鐘 ( TAF ) 和定時 ( PTF ) 標誌位元

- 包含標頭檔

Peripheral\_lib/DrvRTC.h

- 函數返回值

0 : 設置成功

其他 : 設置失敗

- 函數用法

```
/* 清除TAF/ PTF標誌位元 */
```

```
DrvRTC_ClearState(2);
```

### 13.3.13. DrvRTC\_EnableInt

- 函數

```
void DrvRTC_EnableInt(void)
```

- 函數功能

使能RTC中斷。每次進入中斷都需要清除定時喚醒標誌位元 ( PTF ) 下次才能正常進入中斷 ;

設置寄存器0x40004[21]=1.

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvRTC.h

- 函數返回值

無

- 函數用法

```
/* 使能RTC中斷 */
```

```
DrvRTC_EnableInt();
```

### 13.3.14. DrvRTC\_DisableInt

- 函數

```
void DrvRTC_DisableInt(void)
```

- 函數功能

關閉RTC 中斷, 設置寄存器0x40004[21]=0.

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvRTC.h

- 函數返回值

無

- 函數用法

```
/* 關閉RTC中斷 */
```

```
DrvRTC_DisableInt();
```

### 13.3.15. DrvRTC\_ReadIntFlag

- 函數

```
unsigned int DrvRTC_ReadIntFlag(void)
```

- 函數功能

讀取RTC中斷要求標誌位元RTCIF，讀取寄存器0x40004[5]的值。

- 輸入參數

無

- 包含標頭檔

```
Peripheral_lib/DrvRTC.h
```

- 函數返回值

0：無中斷要求

1：有中斷要求

- 函數用法

```
/* 讀取RTC中斷標誌位元*/
```

```
unsigned char flag ; flag=DrvRTC_ReadIntFlag();
```

### 13.3.16. DrvRTC\_ClearIntFlag

- 函數

```
void DrvRTC_ClearIntFlag(void)
```

- 函數功能

清除RTC中斷要求標誌位元RTCIF； 寄存器0x40004[5]=0

- 輸入參數

無

- 包含標頭檔

```
Peripheral_lib/DrvRTC.h
```

- 函數返回值

無

- 函數用法

```
/* 清除RTC中斷要求標誌位元 */
```

```
DrvRTC_ClearIntFlag();
```

### 13.3.17. DrvRTC\_Write

- 函數

```
unsigned int DrvRTC_Write (
```

```
E_DRVRTC_TIME_SELECT eTime, S_DRVRTC_TIME_DATA_T *sPt );
```

- **函數功能**

設置RTC的當前 ( 或鬧鐘 ) 的時間和日期;

設置寄存器0x41A08/0x41A0C/0x41A10/0x41A14/0x41A18/0x41A1C

- **輸入參數**

eTime [in] : 指向'當前時間/日期'或'鬧鐘時間/日期'.

0: 當前時間/日期

1: 鬧鐘時間/日期

\*sPt [in] : 表示要設置的時間/日期, 該變數為一個結構體, 設置內容為 :

u8cClockDisplay    DRVRTC\_CLOCK\_12 / DRVRTC\_CLOCK\_24

u8cAmPm            DRVRTC\_AM / DRVRTC\_PM

u32cSecond        Second 數值

u32cMinute        Minute 數值

u32cHour           Hour 數值(12小時制的輸入範圍 : 0~11 ; 24小時制輸入範圍 : 0~23)

u32cDayOfWeek    Day of week

u32cDay            Day 數值

u32cMonth         Month 數值

u32Year            Year 數值

- **包含標頭檔**

Peripheral\_lib/DrvRTC.h

- **函數返回值**

0 : 設置成功

其他 : 設置失敗.

- **函數用法**

```
/* 更新當前時間'秒數'為0 */
```

```
S_DRVRTC_TIME_DATA_T sCurTime;
```

```
DrvRTC_Read(DRVRTC_ALARM_TIME, &sCurTime);
```

```
sCurTime.u32cSecond = 0;
```

```
DrvRTC_Write(DRVRTC_ALARM_TIME, &sCurTime);
```

### **13.3.18. DrvRTC\_Read**

- **函數**

```
unsigned int DrvRTC_Read (
```

```
E_DRVRTC_TIME_SELECT eTime,
```

```
S_DRVRTC_TIME_DATA_T *sPt );
```

- **函數功能**

讀取RTC的'當前時間/日期'或'鬧鐘的時間/日期'的資料

讀取寄存器0x41A08/0x41A0C/0x41A10/0x41A14/0x41A18/0x41A1C

● **輸入參數**

eTime [in] : 指向'當前時間/日期'或'鬧鐘的時間/日期'選項 :

0 : 當前時間/日期(Current time)

1 : 鬧鐘時間/日期(Alarm time)

\*sPt [in] : 表示要設置的時間/日期, 該變數為一個結構體, 設置內容為 :

u8cClockDisplay    DRVRTC\_CLOCK\_12 / DRVRTC\_CLOCK\_24

u8cAmPm            DRVRTC\_AM / DRVRTC\_PM

u32cSecond        Second 數值

u32cMinute        Minute 數值

u32cHour           Hour 數值

u32cDayOfWeek    Day of week

u32cDay            Day 數值

u32cMonth         Month 數值

u32Year            Year 數值

● **包含標頭檔**

Peripheral\_lib/DrvRTC.h

● **函數返回值**

0 : 設置成功

其他 : 設置失敗.

● **函數用法**

/\* 獲取當前的時間和日期 \*/

S\_DRVRTC\_TIME\_DATA\_T sCurTime;

DrvRTC\_Read(DRVRTC\_CURRENT\_TIME, &sCurTime);

### 13.3.19. DrvRTC\_ClkConfig

● **函數**

unsigned char DrvRTC\_ClkConfig(unsigned char uclken);

● **函數功能**

RTC 時鐘源使能控制, 設置寄存器0x40308[22:23] 。

● **輸入參數**

uclken [in] : RTC時脈源選擇, 輸入範圍 : 0~3.

0 : 關閉RTC的時鐘源

1 : 關閉RTC的時鐘源

2 : LSXT(LSXT需先Enable, 否則視為Disable)

3 : LPO

● **包含標頭檔**

Peripheral\_lib/DrvRTC.h

● **函數返回值**

1：設置失敗

0：設置成功

● **函數用法**

```
/*使能RTC 時鐘源=LPO*/
```

```
DrvRTC_ClkConfig(3);
```

### 13.3.20. DrvRTC\_EnableWUEn

● **函數**

```
void DrvRTC_EnableWUEn (void)
```

● **函數功能**

開啟RTC喚醒功能, WUEn=1b

設置寄存器0x41A00[4]=1

● **輸入參數**

無

● **包含標頭檔**

Peripheral\_lib/DrvRTC.h

● **函數返回值**

無

● **函數用法**

```
/* 開啟RTC喚醒功能 */
```

```
DrvRTC_EnableWUEn();
```

### 13.3.21. DrvRTC\_DisableWUEn

● **函數**

```
void DrvRTC_DisableWUEn (void)
```

● **函數功能**

關閉RTC喚醒功能, WUEn=0b

設置寄存器0x41A00[4]=0

● **輸入參數**

無

● **包含標頭檔**

Peripheral\_lib/DrvRTC.h

● **函數返回值**

無

● **函數用法**



---

/\* 關閉RTC喚醒功能 \*/

DrvRTC\_DisableWUEn();

## 14. IIC 串列通訊 I2C

### 14.1. 函數簡介

該部分函數描述 IIC 通訊模組的控制，包含：

- IIC 啟動控制
- IIC 工作模式控制
- IIC 通訊及收發資料配置控制
- IIC 中斷向量控制

序號	函數名稱	功能描述
01	DrvI2C_Open	開啟 I2C 及配置 I2C 匯流排時鐘
02	DrvI2C_Close	關閉I2C
03	DrvI2C_SlaveSet	開啟I2C從機模式，設置從機位址及GC使能控制
04	DrvI2C_SetIOPin	開啟並設置I2C通訊IO 口
05	DrvI2C_WriteData	發送1byte資料
06	DrvI2C_Write3ByteData	發送3byte資料
07	DrvI2C_ReadData	讀取接收暫存器的資料
08	DrvI2C_Ctrl	設置I2C控制位：STA、STO、AA、SI，控制起始信號、停止信號及應答信號
09	DrvI2C_EnableInt	開啟I2C中斷功能
10	DrvI2C_DisableInt	關閉I2C中斷功能
11	DrvI2C_ReadIntFlag	讀取I2C中斷要求標誌位元
12	DrvI2C_ClearIntFlag	清除I2C中斷要求標誌位元
13	DrvI2C_ClearEIRQ	清除錯誤中斷標誌位元
14	DrvI2C_ClearIRQ	清除I2C器件準備好標誌位元
15	DrvI2C_GetStatusFlag	讀取I2C狀態位元
16	DrvI2C_TimeOutEnable	開啟超時重定功能，設置時鐘分頻及超時控制值
17	DrvI2C_TimeOutDisable	關閉超時復位功能
18	DrvI2C_STSP	設置I2C發送起始信號或停止信號
19	DrvI2C_MGetACK	查詢從機回饋的應答信號ACK/NACK
20	DrvI2C_DisableIOPin	關閉IO口複用作為I2C通訊口功能
21	DrvI2C_EnableSEn	開啟I2C Slave模式功能
22	DrvI2C_DisableSEn	關閉I2C Slave模式功能
23	DrvI2C_EnableI2CEn	開啟I2C

## 14.2. 內部定義常量

### E\_DRVI2C\_Status

識別字	數值	功能意義
E_DRVI2C_ARBITRATION_FLAG	0	仲裁漏失標誌位元
E_DRVI2C_GENERAL_CALL_FLAG	1	全呼標誌位元
E_DRVI2C_ACKNOWLEDGE_FLAG	2	應答信號狀態標誌位元
E_DRVI2C_DATA_FIELD_FLAG	3	資料標誌位元
E_DRVI2C_RW_STATE_FLAG	4	讀/寫狀態標誌位元
E_DRVI2C_RS_FLAG	5	接收停止或重新開始標誌位元
E_DRVI2C_SLAVE_ACTIVE_FLAG	6	從機模式有效標誌位元
E_DRVI2C_MASTER_ACTIVE_FLAG	7	主機模式有效標誌位元

### E\_DRVI2C\_TIMEOUT\_PRESCALE

識別字	數值	功能意義
E_DRVI2C_I2CLK_DIV_1	0	I2C CLK/1
E_DRVI2C_I2CLK_DIV_2	1	I2C CLK/2
E_DRVI2C_I2CLK_DIV_4	2	I2C CLK/4
E_DRVI2C_I2CLK_DIV_8	3	I2C CLK/8
E_DRVI2C_I2CLK_DIV_16	4	I2C CLK/16
E_DRVI2C_I2CLK_DIV_32	5	I2C CLK/32
E_DRVI2C_I2CLK_DIV_64	6	I2C CLK/64
E_DRVI2C_I2CLK_DIV_128	7	I2C CLK/128

### E\_DRVI2C\_TIMEOUT\_LIMIT

識別字	數值	功能意義
E_DRVI2C_CLKPSX1	0	1 * CLKps Cycle
E_DRVI2C_CLKPSX2	1	2 * CLKps Cycle
E_DRVI2C_CLKPSX3	2	3 * CLKps Cycle
E_DRVI2C_CLKPSX4	3	4 * CLKps Cycle
E_DRVI2C_CLKPSX5	4	5 * CLKps Cycle
E_DRVI2C_CLKPSX6	5	6 * CLKps Cycle
E_DRVI2C_CLKPSX7	6	7 * CLKps Cycle
E_DRVI2C_CLKPSX8	7	8 * CLKps Cycle
E_DRVI2C_CLKPSX9	8	9 * CLKps Cycle
E_DRVI2C_CLKPSX10	9	10 * CLKps Cycle
E_DRVI2C_CLKPSX11	10	11 * CLKps Cycle
E_DRVI2C_CLKPSX12	11	12 * CLKps Cycle
E_DRVI2C_CLKPSX13	12	13 * CLKps Cycle
E_DRVI2C_CLKPSX14	13	14 * CLKps Cycle
E_DRVI2C_CLKPSX15	14	15 * CLKps Cycle
E_DRVI2C_CLKPSX16	15	16 * CLKps Cycle

### E\_DRVI2C\_INTERRUPT

識別字	數值	功能意義
E_DRVI2C_INT	1	開啟I2C 中斷功能

E_DRVI2C_ERROR_INT	2	開啟I2C 錯誤中斷功能
E_DRVI2C_INT_ALL	3	開啟I2C 中斷及錯誤中斷功能
E_DRVI2C_SLAVE_BIT		
識別字	數值	功能意義
E_DRVI2C_SLAVE_7BIT	0	從機7bit 位址碼模式
E_DRVI2C_SLAVE_10BIT	1	從機10bit 位址碼模式

### 14.3. 函數說明

注意：只有使能 I2C 後才能對 I2C 的其它寄存器做設置。

#### 14.3.1. DrvI2C\_Open

- 函數

unsigned int DrvI2C\_Open (uint32\_t u32CRG);

- 函數功能

使能I2C功能，並設置I2C匯流排串列傳輸速率；  
設置寄存器0x41000[0]=1,串列傳輸速率寫入寄存器0x41008[23:16]。  
注意：只有使能I2C後才能對I2C其他寄存器設置。

- 輸入參數

u32CRG [in]:  
匯流排串列傳輸速率設置值CRG，設置範圍 0~0xff。  
資料匯流排串列傳輸速率 = (I2CLK/(4\*(CRG+1)))

- 包含標頭檔

Peripheral\_lib/DrvI2C.h

- 函數返回值

0：設置成功  
其他：設置失敗

- 函數用法

```
/* 使能I2C，設置CRG =100 */  
DrvI2C_Open(100);
```

#### 14.3.2. DrvI2C\_Close

- 函數

void DrvI2C\_Close (void);

- 函數功能

關閉I2C功能，設置寄存器0x41000[0]=0。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvI2C.h

- 函數返回值

無

- 函數用法

```
/* 關閉I2C */  
DrvI2C_Close();
```

### 14.3.3. DrvI2C\_SlaveSet

- 函數

```
unsigned int DrvI2C_SlaveSet(  
uint32_t uSlaveAddr,  
E_DRVI2C_SLAVE_BIT uAddrBit,  
uint8_t uSlave3Byte,  
uint8_t GC_Flag);
```

- 函數功能

使能I2C從機模式，並設置從機位址碼及位址碼模式，從機3byte資料發送模式設置，全呼模式GC設置。；  
設置寄存器0x41004[7] / 0x41004[5] / 0x41000[2] / 0x4100C[7:0]

- 輸入參數

uSlaveAddr[in]：從機地址碼

7bit :0~0x7f，主要輸入值為偶數，如0x00/0x02/0x0C等。

10bit: 0~0x3ff，主要輸入值為偶數，即保持bit[0]為0，如0x30C/0x3CC等。

uAddrBit[in]：從機位址碼模式

0: 從機地址碼為7bit

1: 從機地址碼為10bit

uSlave3Byte[in]：從機3byte資料發送模式

0: 正常資料發送模式

1: 從機發送3byte資料模式

GC\_Flag[in]：全呼模式設置

0: 正常呼叫模式

1: 使能全線廣播模式

- 包含標頭檔

Peripheral\_lib/DrvI2C.h

- 函數返回值

0：設置成功

其他：設置失敗

- 函數用法

```
/* 使能從機模式，從機位址碼為0x30，正常資料模式，正常呼叫模式*/
```

```
DrvI2C_SlaveSet(0x30,0,0,0);
```

### 14.3.4. DrvI2C\_SetIOPin

- 函數

unsigned char DrvI2C\_SetIOPin(unsigned int upin);

- 函數功能

設置I2C通訊IO口，設置寄存器0x40844[19:16]。

- 輸入參數

upin[in]：通訊IO選擇

0 : Rsv

1 : Rsv

2 : Rsv

3 : Rsv

4 : SCL=PT2.0;SDA=PT2.1

5 : SCL=PT2.2;SDA=PT2.3

6 : SCL=PT2.4;SDA=PT2.5

7 : SCL=PT2.6;SDA=PT2.7

- 包含標頭檔

Peripheral\_lib/DrvI2C.h

- 函數返回值

無

- 函數用法

/\* 使能通訊口 PT2.0 and PT2.1 \*/

DrvI2C\_SetIOPin(4);

### 14.3.5. DrvI2C\_WriteData

- 函數

void DrvI2C\_WriteData(uint8\_t uData);

- 函數功能

寫入待發送的資料，寫入寄存器0x41014[7:0].

- 輸入參數

uData [IN]: 待發送的資料. 輸入範圍：0~0xFF

- 包含標頭檔

Peripheral\_lib/DrvI2C.h

- 函數返回值

無

- 函數用法

/\*寫入資料0x55至發送暫存器 \*/

DrvI2C\_WriteData(0x55);

### 14.3.6. DrvI2C\_Write3ByteData

- 函數

```
void DrvI2C_Write3ByteData(uint8_t uData1,uData2,uData3);
```

- 函數功能

寫入3byte待發送資料，連續發送3byte。

- 輸入參數

uData1, uData2, uData3 [IN]：待發送的資料。輸入範圍：0~0xFF

- 包含標頭檔

Peripheral\_lib/DrvI2C.h

- 函數返回值

無

- 函數用法

```
/* 寫入3byte 數據0x11 0x22 0x33 準備發送 */
```

```
DrvI2C_Write3ByteData(0x11,0x22,0x33);
```

### 14.3.7. DrvI2C\_ReadData

- 函數

```
unsigned char DrvI2C_ReadData(void);
```

- 函數功能

讀取接收暫存器接收到的資料並控制主機返回應答或非應答信號。

讀取寄存器0x41010[7:0]的值。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvI2C.h

- 函數返回值

1Byte 暫存器接收到的資料

- 函數用法

```
/* 讀取數據*/
```

```
unsigned int data; data=DrvI2C_ReadData();
```

### 14.3.8. DrvI2C\_Ctrl

- 函數

```
void DrvI2C_Ctrl(uint8_t start, uint8_t stop, uint8_t intFlag, uint8_t ack);
```

- 函數功能

設置I2C控制位包括STA, STO, AA, SI，控制start信號、stop信號、I2C器件準備狀態標誌位元及應答信號。

設置寄存器0x41004[3:0]



- **輸入參數**

start [in] : 起始信號控制位元

1 : I2C產生start信號

0 : I2C正常空間。

stop [in] : 停止信號控制位元

1 : I2C生產停止信號

0 : I2C正常空間。

intFlag [in] : I2C器件狀態控制標誌位元

1 : 產生中斷，接收到到9個clock後器件回應，此時SCL會被器件強行拉低，直到IRQFlag清零後釋放SCL

0 : 正常，寫入0，將會清除I2C器件狀態控制旗標，使I2C往一個狀態執行

ack [in] :

1 : ACK應答回復

0 : 未回復ACK或回復NACK信號

- **包含標頭檔**

Peripheral\_lib/DrvI2C.h

- **函數返回值**

無

- **函數用法**

DrvI2C\_Ctrl(0, 0, 1, 0); /\* 清除I2C器件狀態控制標誌位元IRQFlag \*/

DrvI2C\_Ctrl(1, 0, 0, 0); /\* 設置I2C 發送起始信號 \*/

### 14.3.9. DrvI2C\_EnableInt

- **函數**

void DrvI2C\_EnableInt(E\_DRVI2C\_INTERRUPT uINT)

- **函數功能**

使能I2C中斷，包括收發中斷及錯誤中斷，屬於中斷向量HW0.

設置寄存器0x40000[21:20]

- **輸入參數**

uINT[IN] : 中斷項選擇

0 : I2C 收發中斷使能

1 : I2C 錯誤中斷使能

2 : I2C 收發中斷及錯誤中斷使能

- **包含標頭檔**

Peripheral\_lib/DrvI2C.h

- **函數返回值**

無

- **函數用法**

/\*I2C 收發中斷使能\*/

DrvI2C\_EnableInt(1);

### 14.3.10. DrvI2C\_DisableInt

- 函數

void DrvI2C\_DisableInt(E\_DRVI2C\_INTERRUPT uINT)

- 函數功能

關閉I2C中斷控制，包括收發中斷和錯誤中斷；  
清零寄存器0x40000[21:20]

- 輸入參數

uINT[IN]：中斷項選擇

0：關閉I2C收發中斷

1：關閉I2C錯誤中斷

2：關閉I2C收發中斷及錯誤中斷

- 包含標頭檔

Peripheral\_lib/DrvI2C.h

- 函數返回值

無

- 函數用法

```
/* 關閉I2C收發中斷 */
```

```
DrvI2C_DisableInt(1);
```

### 14.3.11. DrvI2C\_ReadIntFlag

- 函數

E\_DRVI2C\_INTERRUPT DrvI2C\_ReadIntFlag(void)

- 函數功能

讀取I2C中斷標誌位元I2CEIF/I2CIF。讀取寄存器0x40000[5:4]的值

- 輸入參數

None

- 包含標頭檔

Peripheral\_lib/DrvI2C.h

- 函數返回值

0：I2C 無中斷要求

1：I2C 收發中斷要求標誌位元I2CIF

2：I2C 錯誤中斷要求標誌位元I2CEIF

3：I2C 有收發中斷要求標誌位元I2CIF和錯誤中斷要求標誌位元I2CEIF

- 函數用法

```
/* Read the I2C Interrupt flag */
```

```
uint32_t temp;
```

```
temp=DrvRTC_ReadIntFlag();
```

### 14.3.12. DrvI2C\_ClearIntFlag

- 函數

```
void DrvI2C_ClearIntFlag(E_DRVI2C_INTERRUPT uINT)
```

- 函數功能

清除I2C中斷標誌位元I2CEIF/I2CIF。清除寄存器0x40000[5:4]的值

- 輸入參數

uINT[IN] :

0 : 清除I2C中斷標誌位元I2CIF

1 : 清除I2C中斷標誌位元I2CEIF

2 : 清除I2C中斷標誌位元I2CEIF/I2CIF

- 包含標頭檔

Peripheral\_lib/DrvI2C.h

- 函數返回值

無

- 函數用法

```
/*清除I2C中斷標誌位元I2CIF */
```

```
DrvI2C_ClearIntFlag(0);
```

### 14.3.13. DrvI2C\_ClearEIRQ

- 函數

```
void DrvI2C_ClearEIRQ(void)
```

- 函數功能

清除錯誤中斷旗標EIRQFlag。只有先清零超時標誌位元(TOFLAG)才能清零EIRQFlag。寫入0就可清零；只有清除EIRQFlag才能清除中斷要求標誌位元(I2CEIF)。

設置寄存器0x41004[4]=0。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvI2C.h

- 函數返回值

無

- 函數用法

```
/*清除錯誤中斷旗標EIRQ */
```

```
DrvI2C_ClearEIRQ();
```

### 14.3.14. DrvI2C\_ClearIRQ

- 函數

void DrvI2C\_ClearIRQ(void)

- 函數功能

清除I2C器件狀態控制標誌位元IRQFlag，IRQFlag從1變為0，使I2C執行下一個狀態。

清零寄存器0x41004[1]=0。

無論作為主機模式還是從機模式，只要接收到9個clock後，IRQFlag就被置1，此時SCL就會被拉低直到IRQFlag被清零，SCL得到釋放，器件才能執行下一個狀態動作。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvI2C.h

- 函數返回值

無

- 函數用法

/\*清除I2C器件準備狀態標誌位元IRQFlag \*/

DrvI2C\_ClearIRQ();

### 14.3.15. DrvI2C\_GetStatusFlag

- 函數

unsigned char DrvI2C\_GetStatusFlag(void)

- 函數功能

獲取I2C狀態標誌位元，返回一個8位元的資料，讀取寄存器0x41004[23:16]的值。

- 輸入參數

- 包含標頭檔

Peripheral\_lib/DrvI2C.h

- 函數返回值

返回值對應位表示的專案設置

Bit 0: 仲裁漏失標誌而為 ARB

Bit 1: General Call Flag(GC)

Bit 2: ACK應答標誌位元 ( A/NA )

Bit 3: 資料標誌位元 ( DF )

Bit 4: 讀寫狀態標誌位元 ( R/W )

Bit 5: 接收停止或重新開始標誌 ( RX P/SR )

Bit 6: 從機模式有效標誌位元 ( SAct )

Bit 7: 主機模式有效標誌位元(MAct)

ARB: uStatus=0

0 : 正常

1 : 仲裁漏失

GC: uStatus=1

0 : 正常I

1 : 當前全呼模式

A/NA: uStatus=2

0 : NACK已經發送或接收.

1 : ACK已經接收或發送.

DF: uStatus=3

0 : 正常

1 : 資料被發送或接收.

R/W: uStatus=4

0 : 寫命令已被發送或接收

1 : 讀命令已被發送或接收.

RX P/Sr: uStatus=5

0 : 正常

1 : 接收停止或重新開始信號已被發送或接收.

SAct : uStatus=6

0 : 從機模式無效

1 : 從機模式有效

MAct : uStatus=7

0 : 主機模式無效

1 : 主機模式有效

#### ● 函數用法

/\* 讀取應答信號標誌位元 /

DrvRTC\_GetStatusFlag(2); //讀取應答信號ACK的狀態標誌位元

### 14.3.16. DrvI2C\_TimeOutEnable

#### ● 函數

```
unsigned char DrvI2C_TimeOutEnable(  
E_DRVI2C_TIMEOUT_PRESCALE uPreScale,  
E_DRVI2C_TIMEOUT_LIMIT uTimeOutLimit  
);
```

#### ● 函數功能

使能超時復位功能, 設置超時功能時鐘分頻及超時時間。

設置寄存器0x41000[1]=1。及寄存器0x41008[6:0]。

#### ● 輸入參數

uPreScale[in]: 時鐘分頻

0 I2C CLK/1

- 1 I2C CLK/2
- 2 I2C CLK/4
- 3 I2C CLK/8
- 4 I2C CLK/16
- 5 I2C CLK/32
- 6 I2C CLK/64
- 7 I2C CLK/128

uTimeOutLimit [in] : 超時時間設置

- 0 1 \* CLKps Cycle
- 1 2 \* CLKps Cycle
- 2 3 \* CLKps Cycle
- 3 4 \* CLKps Cycle
- 4 5 \* CLKps Cycle
- 5 6 \* CLKps Cycle
- 6 7 \* CLKps Cycle
- 7 8 \* CLKps Cycle
- 8 9 \* CLKps Cycle
- 9 10 \* CLKps Cycle
- 10 11 \* CLKps Cycle
- 11 12 \* CLKps Cycle
- 12 13 \* CLKps Cycle
- 13 14 \* CLKps Cycle
- 14 15 \* CLKps Cycle
- 15 16 \* CLKps Cycle

- 包含標頭檔

Peripheral\_lib/DrvI2C.h

- 函數返回值

0 : 設置成功

0xff : 設置失敗

- 函數用法

/\*開啟超時重定功能 · 設置頻率分頻器/ 32 · 及超時限制15 \* CLKps Cycle \*/

DrvI2C\_TimeOutEnable(5,14);

### 14.3.17. DrvI2C\_TimeOutDisable

- 函數

void DrvI2C\_TimeOutDisable(void)

- 函數功能

關閉超時復位功能，設置寄存器0x41000[1] =0。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvI2C.h

- 函數返回值

無

- 函數用法

```
/* 關閉超時復位功能 */  
DrvI2C_TimeOutDisable();
```

### 14.3.18. DrvI2C\_STSP

- 函數

```
void DrvI2C_STSP(unsigned char usignal);
```

- 函數功能

啟動I2C的起始信號或停止信號，設置寄存器0x41004[3:2]。

- 輸入參數

usignal[in]：信號模式設置

0 啟動起始信號

1 啟動停止信號

- 包含標頭檔

Peripheral\_lib/DrvI2C.h

- 函數返回值

無

- 函數用法

```
/* 啟動起始信號*/  
DrvI2C_STSP(0);
```

### 14.3.19. DrvI2C\_MGetACK

- 函數

```
unsigned char DrvI2C_MGetACK(unsigned int utime);
```

- 函數功能

在設定的時間內查詢從機回饋的應答信號，設置寄存器0x41004[1]

- 輸入參數

utime[in]：設定的查詢時間0~0xffff

- 包含標頭檔

Peripheral\_lib/DrvI2C.h

- 函數返回值

- 0 查詢到應答信號ACK標明發送成功
- 1 查詢到非應答信號NACK，標明發送失敗

- 函數用法

```
/* check the ACK during the 0xffff time*/  
Err_flag=DrvI2C_MGetACK(0xffff);
```

### 14.3.20. DrvI2C\_DisableIOPin

- 函數

```
void DrvI2C_DisableIOPin(void)
```

- 函數功能

關閉I2C通訊口功能，設置寄存器0x40844[16]=0

- 輸入參數

無

- 包含標頭檔

```
Peripheral_lib/DrvI2C.h
```

- 函數返回值

無

- 函數用法

```
/*關閉I2C的通訊IO口*/  
DrvI2C_DisableIOPin();
```

### 14.3.21. DrvI2C\_EnableSEn

- 函數

```
void DrvI2C_EnableSEn (void);
```

- 函數功能

開啟I2C Slave模式功能，設置寄存器0x41004[7]=1。

- 輸入參數

無

- 包含標頭檔

```
Peripheral_lib/DrvI2C.h
```

- 函數返回值

無

- 函數用法

```
/* 開啟I2C Slave模式功能 */  
DrvI2C_EnableSEn();
```



### 14.3.22. DrvI2C\_DisableSEn

- 函數

void DrvI2C\_DisableSEn (void);

- 函數功能

關閉I2C Slave模式功能. · 設置寄存器0x41004[7]=0。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvI2C.h

- 函數返回值

無

- 函數用法

/\* 關閉I2C Slave模式功能 \*/

DrvI2C\_DisableSEn();

### 14.3.23. DrvI2C\_EnableI2CEn

- 函數

void DrvI2C\_EnableI2CEn (void);

- 函數功能

開啟I2C功能. · 設置寄存器0x41000[0]=1。

- 輸入參數

無

- 包含標頭檔

Peripheral\_lib/DrvI2C.h

- 函數返回值

無

- 函數用法

/\* 開啟I2C功能 \*/

DrvI2C\_EnableI2CEn();

## 15. LCD 顯示驅動器

### 15.1. 函數簡介

該部分函數描述 LCD 驅動器相關設置

- LCD 驅動器的時鐘頻率設置
- LCD 驅動器的偏執電壓的設置
- LCD 驅動器 duty 設置
- LCD 驅動器 COM/SEG 口的工作模式設置。
- LCD 驅動器顯示資料的寫入

序號	函數名稱	功能描述
01	DrvLCD_EnableCLK	開啟LCD驅動器時鐘源
02	DrvLCD_DisplayMode	設置LCD驅動器顯示模式
03	DrvLCD_LcdDuty	設置LCD的duty
04	DrvLCD_LCDBuffer	設置LCD驅動器的輸入緩衝器
05	DrvLCD_SwpCOMSEG	設置COM線與SEG線的順序
06	DrvLCD_IOMode	設置LCD複用IO口的工作模式
07	DrvLCD_WriteData	寫入資料至LCD驅動器的資料緩衝器
08	DrvLCD_VLCDTrim	選擇晶片出廠時VLCD的校正參數
09	DrvLCD_VLCDMode	LCD驅動器偏置電壓的設置

**15.2. 內部常量定義**

E\_VLCD\_MODE

識別字	數值	功能意義
E_VLCD_DISABLE	0	關閉VLCD
E_VLCD_RTYPE	1	R_TYPE
E_VLCD33	2	VLCD=3.3V
E_VLCD30	3	VLCD=3.0V
E_VLCD27	4	VLCD=2.7V
E_VLCD24	5	VLCD=2.4V

E\_LCD\_DUTY

識別字	數值	功能意義
E_LCD_DUTY3	0	LCD 工作週期為1/3 duty
E_LCD_DUTY4	1	LCD 工作週期為1/4 duty
E_LCD_DUTY5	2	LCD 工作週期為1/5 duty
E_LCD_DUTY6	3	LCD 工作週期為1/6 duty

E\_LCD\_DISPLAY\_MDE

識別字	數值	功能意義
E_LCD_NORMAL	0	LCD顯示模式為正常顯示
E_LCD_NORMAL	1	不論輸入任何值・LCD都是全顯
E_LCD_NORMAL	2	不論輸入任何值・LCD都是全滅

### 15.3. 函數說明

#### 15.3.1. DrvLCD\_EnableCLK

● 函數

unsigned char DrvLCD\_EnableCLK(unsigned int uLCD1,unsigned int uLCD2,unsigned int usource)

● 函數功能

LCD時脈源的設置及LCDE/LCDO的時脈源除頻設置；設置寄存器0x40310[6:0]

● 輸入參數

uLCD1[in] LCD時鐘除頻器1(LCDO)設置

0 : ÷ 1;    1 : ÷ 3;    2 : ÷ 5;    3 : ÷ 7

4 : ÷ 9;    5 : ÷ 11;    6 : ÷ 13;    7 : ÷ 15

uLCD2[in] : LCD時鐘除頻器1(LCDE)設置, 輸入範圍 : 0~7

0 : Disable;    1 : ÷ 1;    2 : ÷ 2;    3 : ÷ 4

4 : ÷ 8;    5 : ÷ 16;    6 : ÷ 32;    7 : Disable

source[in] : LCD驅動器時鐘源設置, 輸入範圍 : 0~1

0 : LS\_CK(固定/8)

1 : HS\_CK(固定/64)

● 包含標頭檔

Peripheral\_lib/DrvLCD.h

● 函數返回值

0 設置成功

1 設置失敗

● 函數用法

/\*設置LCD的時鐘源為HS\_CK,且整體除頻為LCD1\*LCD2=5\*1;\*/

DrvLCD\_EnableCLK(2,1,1);

#### 15.3.2. DrvLCD\_DisplayMode

● 函數

unsigned char DrvLCD\_DisplayMode(unsigned int uDISMODE)

● 函數功能

LCD顯示模式設置；設置寄存器0x41B00[17:16]

● 輸入參數

uDISMODE[in] LCD顯示模式選擇. 輸入範圍 : 0~2

0 : 正常顯示模式

1 : 不論輸入任何值, 都是全顯模式

2 : 不論輸入任何值, 都是全滅模式

- 包含標頭檔

Peripheral\_lib/DrvLCD.h

- 函數返回值

0 設置成功

1 設置失敗

- 函數用法

/\*設置LCD為正常顯示模式 \*/

DrvLCD\_DisplayMode(0);

### 15.3.3. DrvLCD\_LcdDuty

- 函數

unsigned char DrvLCD\_LcdDuty(unsigned int uDUTY)

- 函數功能

LCD驅動器工作週期選擇；設置寄存器0x41B00[5:4]

- 輸入參數

uDUTY[in] : LCD工作週期選擇, 輸入範圍 : 0~3

0 : 1/3 Duty                    1 : 1/4 Duty

2 : 1/5 Duty                    3 : 1/6 Duty

- 包含標頭檔

Peripheral\_lib/DrvLCD.h

- 函數返回值

0 設置成功

1 設置失敗

- 函數用法

/\*設置LCD的工作週期為1/4 Duty \*/

DrvLCD\_LcdDuty(1);

### 15.3.4. DrvLCD\_LCDBuffer

- 函數

unsigned char DrvLCD\_LCDBuffer(unsigned int uBEN)

- 函數功能

VLCD輸入緩衝器控制；設置寄存器0x41B00[3]

- 輸入參數

uBEN[in] : VLCD輸入緩衝器控制, 輸入範圍 : 0~1

0 : 關閉

1 : 開啟

- 包含標頭檔

Peripheral\_lib/DrvLCD.h

- 函數返回值

0 設置成功

1 設置失敗

- 函數用法

/\*開啟VLCD的輸入緩衝器 \*/

DrvLCD\_LCDBuffer(1);

### 15.3.5. DrvLCD\_SwpCOMSEG

- 函數

unsigned char DrvLCD\_SwpCOMSEG(unsigned int uflip)

- 函數功能

LCDCOM Port Selection ; 設置寄存器0x41B00[7:6]

- 輸入參數

uflip[in] : 輸入範圍 : 0~3

0 : PT13.0~PT13.5 is COM Port

1 : PT6.0~PT6.5 is COM Port

2 : PT9.0~PT9.5 is COM Port

3 : PT8.2~PT8.7 is COM Port

- 包含標頭檔

Peripheral\_lib/DrvLCD.h

- 函數返回值

0 設置成功

1 設置失敗

- 函數用法

/\*設置PT13.0~PT13.5 is COM Port\*/

DrvLCD\_SwpCOMSEG(0);

### 15.3.6. DrvLCD\_IOMode

- 函數

unsigned char DrvLCD\_IOMode(unsigned int uport,unsigned int uIOMODE)

- 函數功能

設置LCD的複用IO口PT6~PT13及COM5/COM4的工作模式 ; 設置寄存器0x41B04[]/0x41B08[] ;

- 輸入參數

uport[in] IO口選擇, 輸入範圍 : 0~4

0 : PT6     1 : PT7     2 : PT8

3 : PT9      4 : PT13

uIOMODE[in] : 為8位數值，每一位數值對應一位IO引腳，數值範圍是0~0xFF；

uIOMODE的每1bit數值的定義如下：

0 : I/O模式

1 : LCD模式

● **包含標頭檔**

Peripheral\_lib/DrvLCD.h

● **函數返回值**

0 設置成功

1 設置失敗

● **函數用法**

/\*設置PT6為LCD模式\*/

DrvLCD\_IOMode(0, 0xFF);

### 15.3.7. DrvLCD\_WriteData

● **函數**

unsigned char DrvLCD\_WriteData(unsigned int uSEG, unsigned int data)

● **函數功能**

寫入資料到LCD資料緩衝器LCD0~LCD15；設置寄存器0x40850~0x408C8；

● **輸入參數**

uSEG[in] : LCD 資料緩衝器LCD0~LCD15，每個緩衝器都包含兩個SEG.

LCD0=SEG1:SEG0; //0x408C8

LCD1=SEG3:SEG2; //0x40850

LCD2=SEG5:SEG4; //0x40854

LCD3=SEG7:SEG6; //0x40858

LCD4=SEG9:SEG8; //0x4085C

LCD5=SEG11:SEG10; //0x40860

LCD6=SEG13:SEG12; //0x40864

LCD7=SEG15:SEG14; //0x40868

LCD8=SEG17:SEG16; //0x4086C

LCD9=SEG19:SEG18; //0x40870

LCD10=SEG21:SEG20; //0x40874

LCD11=SEG23:SEG22; //0x40878

LCD12=SEG25:SEG24; //0x4087C

LCD13=SEG27:SEG26; //0x40880

LCD14=SEG29:SEG28; //0x40884

LCD15=SEG31:SEG30; //0x40888

設置範圍：0~15，分別對應LCD0~LCD15

Data[in]：要寫入到LCD緩衝器的數值，且該資料只適應HYCON LCD的SEG的位置排列。

設置範圍：0~0xff;

注意：

客戶在使用時，請注意資料需要配置自己的LCD面板及SEG線的排列是否一致；

要寫入到LCD緩衝器的data數值，需要注意LCD工作週期選擇，並且數據格式也需要注意。

例如：LCD工作週期為1/6Duty，並且在LCD1寫入資料，對應到的數據格式data[5:0]=SEG3,

data[11:6]=SEG2

- 包含標頭檔

Peripheral\_lib/DrvLCD.h

- 函數返回值

0 設置成功

1 設置失敗

- 函數用法

/\* 設定LCD工作週期為1/6Duty，並且在LCD1寫入資料 \*/

DrvLCD\_LcdDuty (E\_LCD\_DUTY6);

DrvLCD\_WriteData(1,0x03F); //0x40850=0x003f0000

DrvLCD\_WriteData(1,0xFC0); //0x40850=0x0000003f

DrvLCD\_WriteData(1,0xFFF); //0x40850=0x003f003f

/\* 設定LCD工作週期為1/4Duty，並且在LCD1寫入資料 \*/

DrvLCD\_LcdDuty (E\_LCD\_DUTY4);

DrvLCD\_WriteData(1,0x03F); //0x40850=0x000f0003

DrvLCD\_WriteData(1,0xFF); //0x40850=0x000f000f

DrvLCD\_WriteData(1,0xF0); //0x40850=0x0000000f

### 15.3.8. DrvLCD\_VLCDTrim

- 函數

unsigned char DrvLCD\_VLCDTrim(short umode)

- 函數功能

選擇晶片出廠時VLCD的校正參數.設置寄存器0x41B00[2:0]/ 0x41B10[3:0] ；

- 輸入參數

umode[in]：待校正VLCD電壓模式選擇. 輸入範圍：1~5

1: VLCD~3.43V ; 2: VLCD~3.16V

3: VLCD~2.93V ; 4: VLCD~2.73V

5: VLCD~2.55V

- 包含標頭檔

Peripheral\_lib/DrvLCD.h

- 函數返回值

0 設置成功



1 設置失敗

● **函數用法**

/\*校正VLCD電壓為3.16V\*/

DrvLCD\_VLCDTrim(2);

### 15.3.9. DrvLCD\_VLCDMode

● **函數**

unsigned char DrvLCD\_VLCDMode(unsigned int uVLCDMODE)

● **函數功能**

LCD驅動器偏置電壓的設置；設置寄存器0x41B00[2:0]. 輸入範圍：0~7

● **輸入參數**

uVLCDMODE[in] LCD偏置電壓選擇

0：關閉VLCD偏置電壓

1：R\_TYPE模式 · Charge PUMP 關閉 · VLCD R開啟

2：3.3V · Charge PUMP 開啟 · VLCD R 關閉

3：3.0V · Charge PUMP 開啟 · VLCD R 關閉

4：2.7V · Charge PUMP 開啟 · VLCD R 關閉

5：2.4V · Charge PUMP 開啟 · VLCD R 關閉

6：VLCD Charge Pump 關閉, VLCD R關閉, VLCD緩衝器關閉

7：VLCD Charge Pump 關閉, VLCD R關閉, VLCD緩衝器關閉

● **包含標頭檔**

Peripheral\_lib/DrvLCD.h

● **函數返回值**

0 設置成功

1 設置失敗

● **函數用法**

/\*設置LCD的偏置電壓VLCD為3.0V \*/

DrvLCD\_VLCDMode(3);

## 16. Flash 讀寫

### 16.1. 函數簡介

該部分函數描述晶片 Flash 區域的讀寫操作，包含：

--Flash 的資料寫入與讀取

--Flash 的字/頁寫入、字/頁的讀取；

--Flash 的數據擦除

序號	函數名稱	功能描述
01	DrvFlash_Burn_Word	寫入一個字資料到flash
02	ROM_BurnPage	寫入連續一頁的32個字資料到flash
03	ReadWord	讀取flash的一個字的資料
04	ReadPage	讀取flash連續一頁32個字的資料
05	PageErase	擦除flash一頁連續的資料
06	SectorErase	擦除一個sector的資料
07	ROM_BurnWordonly	寫入一個字資料到flash(只有寫入功能不包含Erase功能)
08	ROM_BurnPageWriteonly	寫入連續一頁的32個字資料到flash(只有寫入功能不包含Erase功能)

## 16.2. 函數說明

注意1：在執行Flash燒錄與讀取程式指令之前，必須先執行SYS\_DisableGIE(); 關閉全域使能中斷，這可以避免程式運行異常的行為發生。

注意2：執行Flash燒錄指令，必須確保晶片工作電壓VDD3V高於2.7V，如果晶片電壓VDD3V低於2.7V，則可能會發生燒錄錯誤行為。

### 16.2.1. DrvFlash\_Burn\_Word

- 函數

int DrvFlash\_Burn\_Word(unsigned int addr,unsigned int DelayTime,unsigned int data);

- 函數功能

寫入一個word的資料至Flash的對應位址。

- 輸入參數

addr[in]: 待寫入資料的位址

16位的位址，Flash空間是從0x90000開始，所以該位址值只需寫16位值，輸入範圍0~0xffff，且位址的間隔步長為4；如要想0x9A880寫入一個word資料，函數參數只需填寫0xA880值就可以。

Delay time[in]: 燒錄延時

data [in]: 帶寫入的資料，輸入範圍0~0xffffffff

- 包含標頭檔

Peripheral\_lib/Drvflash.h

- 函數返回值

0 設置成功

- 函數用法

```
/* 寫入0xFF05到flash的0x90880位址 */
```

```
DrvFlash_Burn_Word(0x0880,0x2000,0xFF05);
```

注意：執行Flash燒錄指令，必須確保晶片工作電壓VDD3V高於2.7V，如果晶片電壓VDD3V低於2.7V，則可能會發生燒錄錯誤行為。

### 16.2.2. ROM\_BurnPage

- 函數

int ROM\_BurnPage(unsigned int key,unsigned int addr,unsigned int DelayTime,unsigned int \*data);

- 函數功能

一次性寫入一頁32個word的資料到Flash對應的連續位址。

- 輸入參數

addr[in]: 待寫入資料的首位址，16位的位址，Flash空間是從0x90000開始，所以該位址值只需寫16位值，輸入範圍0~0xffff，且位址的間隔步長為128 ( 32\*4 )，且不能進行跨頁寫入，因為每個page只有128byte，所以位址只能為0xuu00或0xuu80；如要想從0x9A880開始，函數參數只需填寫0xA880值就可以。

Delay time[in]：燒錄延時

data [in]：待寫入的資料，屬於指標類型，資料的長度為32word，每個資料的輸入範圍0~0xffffffff

- 包含標頭檔

Peripheral\_lib/Drvflash.h

- 函數返回值

0xFF 設置成功

- 函數用法

/\* 從位址0X90880開始連續一次寫入32word的資料 \*/

```
unsigned int *A[32]={0};
```

```
ROM_BurnPage(FLASH_KEY_A, 0x0880, 0x2000, A); // FLASH_KEY_A= 0x4B6579A8
```

注意：執行Flash燒錄指令，必須確保晶片工作電壓VDD3V高於2.7V，如果晶片電壓VDD3V低於2.7V，則可能會發生燒錄錯誤行為。

### 16.2.3. ReadWord

- 函數

```
int ReadWord(unsigned int addr);
```

- 函數功能

讀取Flash對應位址的一個word的資料。

- 輸入參數

addr[in]：待讀取資料的位址

16位的位址，Flash空間是從0x90000開始，所以該位址值只需寫16位值，輸入範圍0~0xffff，且位址的間隔步長為4；如要想0x9A880讀取一個word資料，函數參數只需填寫0xA880值就可以。

- 包含標頭檔

Peripheral\_lib/Drvflash.h

- 函數返回值

返回一個word的資料

- 函數用法

/\* 讀取Flash的0x90880位址的資料 \*/

```
Int flag; flag= ReadWord(0x0880);
```

### 16.2.4. ReadPage

- e函數

```
int ReadPage(unsigned int addr,int* data);
```

- u函數功能

一次性連續讀取flash對應連續位址的32個word的數據

- 數輸入參數

addr[in]：待讀取資料的首位址，16位的地址，Flash空間是從0x90000開始，所以該位址值只需寫16位值，輸入範圍0~0xffff，且地址的間隔步長為128 ( 32\*4 )，且不能進行跨頁讀取，因為每個page只有128byte，所以

位址只能為0xuu00或0xuu80；如要想從0x9A880開始，函數參數只需填寫0xA880值就可以。

data [in]：用於保存讀取到的資料，屬於指標類型，資料的長度為32word，每個資料的輸入範圍0~0xffffffff

●~包含標頭檔

Peripheral\_lib/DrvFlash.h

●函數返回值

0 設置成功

1 設置失敗

●資函數用法

/\* 讀取flash以0x90880位址開始的連續32個word的資料 \*/

unsigned int \*A[32]={0};

ReadPage(0x0880,A);

### 16.2.5. PageErase

●a函數

int PageErase (unsigned int key, unsigned int addr, unsigned int DelayTime);

●n函數功能

一次性清除flash中對應連續位址的32個word的資料

●輸入參數

addr[in]：待清除資料的首位址，16位的位址，Flash空間是從0x90000開始，所以該位址值只需寫16位值，輸入範圍0~0xffff，且位址的間隔步長為128 ( 32\*4 )，且不能進行跨頁寫入，因為每個page只有128byte，所以位址只能為0xuu00或0xuu80；如要想從0x9A880開始，函數參數只需填寫0xA880值就可以。

Delay time[in]：延時時間

●包含標頭檔

Peripheral\_lib/DrvFlash.h

●函數返回值

0xFF 設置成功

●函數用法

/\*清除從0x90880位址開始連續32個word資料\*/

PageErase(FLASH\_KEY\_A, 0x0880,0x2000); // FLASH\_KEY\_A= 0x4B6579A8

### 16.2.6. SectorErase

●函數

int SectorErase (unsigned int key, unsigned int addr, unsigned int DelayTime);

●函數功能

清除flash對應位址的一個sector的資料

●輸入參數

addr[in]: 待清除資料的首位址, 16位的位址, Flash空間是從0x90000開始, 所以該位址值只需寫16位值, 輸入範圍0~0xffff, 且一個sector包含32page, 所以位址的間隔步長為32\*128; 所以首位址是以頁來計算的, 且需要對齊, 位址為0xu000 (u為可變的值)。如要想從0x91000開始, 函數參數只需填寫0x1000值就可以。

Delay time[in]: 延時時間

- 包含標頭檔

Peripheral\_lib/DrvFlash.h

- 函數返回值

0xFF 設置成功

- 函數用法

/\* 從0x91000位址開始連續清除一個sector的資料\*/

```
SectorErase(FLASH_KEY_A, 0x1000,0x2000); //FLASH_KEY_A= 0x4B6579A8
```

## 16.2.7. ROM\_BurnWordonly

- 函數

int ROM\_BurnWordonly(unsigned int addr,unsigned int DelayTime,unsigned int data);

- 函數功能

寫入一個word的資料至Flash的對應位址。

- 輸入參數

addr[in]: 待寫入資料的位址

16位的位址, Flash空間是從0x90000開始, 所以該位址值只需寫16位值, 輸入範圍0~0xffff, 且位址的間隔步長為4; 如要想0x9A880寫入一個word資料, 函數參數只需填寫0xA880值就可以。

DelayTime [in]: 燒錄延時

data [in]: 帶寫入的資料, 輸入範圍0~0xffffffff

- 包含標頭檔

Peripheral\_lib/DrvFlash.h

- 函數返回值

0xFF 設置成功

- 函數用法

/\* 先清除從0x90880位址開始連續32個word資料, 再寫入0xFF05到flash的0x90880位址 \*/

```
PageErase(0x0880, 0x2000);
```

```
ROM_BurnWordonly (0x0880, 0x2000, 0xFF05);
```

注意1: 此函數不包含Erase指令, 執行此函數之前需要先做Flash Erase動作, 才能確保Flash資料正確被寫入。

注意2: 執行Flash燒錄指令, 必須確保晶片工作電壓VDD3V高於2.7V, 如果晶片電壓VDD3V低於2.7V, 則可能會發生燒錄錯誤行為。

### 16.2.8. ROM\_BurnPageWriteonly

- 函數

int ROM\_BurnPageWriteonly(unsigned int addr,unsigned int DelayTime,unsigned int\* data);

- 函數功能

一次性寫入一頁32個word的資料到Flash對應的連續位址。

- 輸入參數

addr[in]：待寫入資料的首位址，16位的位址，Flash空間是從0x90000開始，所以該位址值只需寫16位值，輸入範圍0~0xffff，且位址的間隔步長為128 ( 32\*4 )，且不能進行跨頁寫入，因為每個page只有128byte，所以位址只能為0xuu00或0xuu80；如要想從0x9A880開始，函數參數只需填寫0xA880值就可以。

Delay time[in]：燒錄延時

data [in]：待寫入的資料，屬於指標類型，資料的長度為32word，每個資料的輸入範圍0~0xffffffff

- 包含標頭檔

Peripheral\_lib/Drvflash.h

- 函數返回值

0xFF 設置成功

- 函數用法

/\* 先清除從0x90880位址開始連續32個word資料，再從位址0x90880開始連續一次寫入32word的資料 \*/

```
unsigned int *A[32]={0};
```

```
PageErase(0x0880, 0x2000);
```

```
ROM_BurnPageWriteonly (0x0880, 0x2000, A);
```

注意1：此函數不包含Erase指令，執行此函數之前需要先做Flash Erase動作，才能確保Flash資料正確被寫入。

注意2：執行Flash燒錄指令，必須確保晶片工作電壓VDD3V高於2.7V，如果晶片電壓VDD3V低於2.7V，則可能會發生燒錄錯誤行為。

**16.3. Flash 儲存空間結構**

1 page=128 Bytes

1 sector=32 pages=4096 Bytes

Sector & Page			
Sector	Page	Address	Range (Byte)
0	0	000000H	00007FH
	1	000080H	0000FFH
	...	...	...
	30	000F00H	000F7FH
	31	000F80H	000FFFH
1	32	001000H	00107FH
	33	001080H	0010FFH
	...	...	...
	63	001F80H	001FFFH
...	...	...	...
15	480	00F000H	00F07FH
	...	...	...
	511	00FF80H	00FFFFH



## 17. ACE 指令讀寫

### 17.1. 函數簡介

使用 ACE 指令可做 bit 與 half byte control, 減少程式空間與加強程式控制效率.

注意事項 : 要可以使用 ACE 指令, 需要先有#include "ace\_user.h"在 C 檔案內.

### 17.2. 函數說明

#### 17.2.1. ace\_mtar (Move to ACE Register)

- **c**函數

```
void ace_mtar(unsigned int Din, const unsigned ldx4);
```

- **o**函數功能

設定ACE Register的Bit Operation Base Address

- **i**輸入參數

Din : 寫入ACE Register的資料, 32 Bits變數或常數

Idx4: 指向將被Din覆寫的ACE Register Index, 4 Bits常數

- **數**包含標頭檔

Peripheral\_lib/ace\_user.h

- **e**函數返回值

無

- **數**函數用法

```
/*將ACE Register 0x08的Bit Operation Base Address 0設為0x040814*/
```

```
#define ACEpio2_2 (0x40814)
```

```
ace_mtar(ACEpio2_2,8);
```

說明 : ACE Register 有 0x08 和 0x09 兩個可設定使用.

使用 ACE 指令做控制存取之前, 需要先設定 ACE Register 在 0x08 or 0x09.

#### 17.2.2. ace\_mfar (Move from ACE Register)

- **c**函數

```
unsigned int ace_mfar(const unsigned ldx4)
```

- **函數**功能

將先前存在ACE Register的值讀取出來

- **值**輸入參數

Idx4: 指向想要讀出的ACE Register Index, 4 Bits常數

Return : 回讀的ACE Register值

● C包含標頭檔

Peripheral\_lib/ace\_user.h

● e函數返回值

回讀的ACE Register值

● C函數用法

/\*將ACE Register 0x08的Bit Operation Base Address 0設為0x040814, 並且讀回ACE Register\*/

```
#define ACEpio2_2 (0x40814)
```

```
unsigned int outputda;
```

```
ace_mtar(ACEpio2_2,8);
```

```
outputda=ace_mfar(8); //after read, outputda=0x40814
```

### 17.2.3. ace\_BitRd (Bus Bit Read)

● (函數)

```
unsigned int ace_BitRd(const unsigned IdxS1, const unsigned Adr12, const unsigned BSel3)
```

● 函數功能

讀取指定暫存器的bit資料

● 料輸入參數

IdxS1 : Bit Operation Base Address Selection, 1 Bit常數

0 : Bit Operation Base Address 0 (ACE Register 0x08)

1 : Bit Operation Base Address 1 (ACE Register 0x09)

Adr12 : Target Address = Base Address + Adr12 (12 Bits常數)

BSel3 : Target Bit Select, 3 Bits常數

Return : 回讀的1 Bit資料

● 料包含標頭檔

Peripheral\_lib/ace\_user.h

● e函數返回值

回讀的1 Bit資料

● 料函數用法

/\*讀回0x40814的Bit3\*/

```
#define ACEpio2_2 (0x40814)
```

```
unsigned int outputda;
```

```
ace_mtar(ACEpio2_2,8);
```

```
outputda=ace_mfar(8); //outputda=0x40814
```

```
outputda=ace_BitRd(0, 0, 3); //read 0x40814[3]
```

### 17.2.4. ace\_BitWt (Bus Bit Write)

● (函數)

```
void ace_BitWt(const unsigned IdxS1, const unsigned Adr16, const unsigned BSel3, const unsigned Bin)
```

● **函數功能**

寫入指定暫存器的bit資料, 可寫0b or 1b

● **輸入參數**

IdxS1 : Bit Operation Base Address Selection, 1 Bit常數

0 : Bit Operation Base Address 0 (ACE Register 0x08)

1 : Bit Operation Base Address 1 (ACE Register 0x09)

Adr16 : Target Address = Base Address + Adr16 (16 Bits常數)

BSel3 : Target Bit Select, 3 Bits常數

Bin : 想要寫入的資料, 1 Bit常數

● **數包含標頭檔**

Peripheral\_lib/ace\_user.h

● **函數返回值**

無

● **數函數用法**

```
/*寫入0x40814的Bit3為並且回讀*/  
#define ACEpio2_2 (0x40814)  
unsigned int outputda;  
ace_mtar(ACEpio2_2,8);  
outputda=ace_mfar(8); //outputda=0x40814  
ace_BitWt(0,0,3,1); // write ACEpio2_2=0x40814[3]=1b  
outputda=ace_BitRd(0, 0, 3); //read 0x40814[3]=1b  
ace_BitWt(0,0,3,0); // write ACEpio2_2=0x40814[3]=0b  
outputda=ace_BitRd(0, 0, 3); //read 0x40814[3]=0b
```

### 17.2.5. ace\_BitTg (Bus Bit Toggle)

● **函數**

```
void ace_BitTg(const unsigned IdxS1, const unsigned Adr16, const unsigned BSel3)
```

● **函數功能**

對指定暫存器的bit資料做Toggle動作

● **作輸入參數**

IdxS1 : Bit Operation Base Address Selection, 1 Bit常數

0 : Bit Operation Base Address 0 (ACE Register 0x08)

1 : Bit Operation Base Address 1 (ACE Register 0x09)

Adr16 : Target Address = Base Address + Adr16 (16 Bits常數)

BSel3 : Target Bit Select, 3 Bits常數

● **數包含標頭檔**

Peripheral\_lib/ace\_user.h

● **e** 函數返回值

無

● **數** 函數用法

```
/*對0x40814做Toggle變化並且回讀觀察*/
#define ACEpio2_2 (0x40814)
unsigned int outputda;
ace_mtar(ACEpio2_2,8);
outputda=ace_mfar(8); //outputda=0x40814
ace_BitWt(0,0,3,0); // write ACEpio2_2=0x40814[3]=0b
outputda=ace_BitRd(0, 0, 3); //read 0x40814[3]=0b
ace_BitTg(0,0,3); //Toggle 0x40814[3]. 0b ->1b
outputda=ace_BitRd(0, 0, 3); //read 0x40814[3]=1b
ace_BitTg(0,0,3); //Toggle 0x40814[3]. 1b ->0b
outputda=ace_BitRd(0, 0, 3); //read 0x40814[3]=0b
```

### 17.2.6. ace\_HByteWt (Bus Half Byte Write)

● **c** 函數

void ace\_HByteWt(const unsigned IdxS1, const unsigned Adr10, const unsigned HBS1, const unsigned HBM4, const unsigned HBD4)

● **函** 數功能

寫入指定暫存器的Half byte資料

● **料** 輸入參數

IdxS1 : Bit Operation Base Address Selection, 1 Bit常數  
0 : Bit Operation Base Address 0 (ACE Register 0x08)  
1 : Bit Operation Base Address 1 (ACE Register 0x09)  
Adr10 : Target Address = Base Address + Adr10 (10 Bits常數)  
HBS1 : 高低位元選擇, 1 Bit常數  
0 : Bit 3~0  
1 : Bit 7~4  
HBM4 : Bit Mask, 4 Bits常數  
HBD4 : Write Data, 4 Bits常數

● **數** 包含標頭檔

Peripheral\_lib/ace\_user.h

● **e** 函數返回值

無

● **數** 函數用法

```
/*先寫入0x40824的Bit3~Bit0=0x1111b與0x40824的Bit7~Bit4=0x1111b, 再寫入0x40824的  
Bit3~Bit0=0x0000b與0x40824的Bit7~Bit4=0x0000b */
```

```
#define ACEpio2_2 (0x40814)
unsigned int outputda;
ace_mtar(ACEpio2_2,8);
outputda=ace_mfar(8); //outputda=0x40814
ace_HByteWt(0,0x0010,0,0xf,0xf); //0x40824[3:0]=1111b
asm("nop");
ace_HByteWt(0,0x0010,1,0xf,0xf); //0x40824[7:4]=1111b
asm("nop");
ace_HByteWt(0,0x0010,0,0xf,0x0); //0x40824[3:0]=0000b
asm("nop");
ace_HByteWt(0,0x0010,1,0xf,0x0); //0x40824[7:4]=0000b
asm("nop");
```

### 17.2.7. ace\_RBitWt (Register Bit Write)

#### ● (函數)

unsigned int ace\_RBitWt(unsigned int Din, const unsigned BSel5, const unsigned Bin)

#### ● 函數功能

寫入指定暫存器的bit資料，可寫0b or 1b

#### ● b輸入參數

Din：想要寫入資料的變數

BSel5：Target Bit Select, 5 Bits常數

Bin：想要寫入的資料，1 Bit常數

Return：執行Bit Write後回傳的資料，變數

#### ● 數包含標頭檔

Peripheral\_lib/ace\_user.h

#### ● e函數返回值

執行Bit Write後回傳的資料，變數

#### ● 數函數用法

/\*寫入0x40814的Bit資料並且回傳變數資料\*/

```
unsigned int a=0x40814; //pio2_2
unsigned char *outputda0=(unsigned char*)a;
*outputda0=ace_RBitWt(*outputda0,0,1); //0x40814[0]=1b
*outputda0=ace_RBitWt(*outputda0,1,1); //0x40814[1]=1b
*outputda0=ace_RBitWt(*outputda0,2,1); //0x40814[2]=1b
*outputda0=ace_RBitWt(*outputda0,3,1); //0x40814[3]=1b
*outputda0=ace_RBitWt(*outputda0,4,1); //0x40814[4]=1b
*outputda0=ace_RBitWt(*outputda0,5,1); //0x40814[5]=1b
*outputda0=ace_RBitWt(*outputda0,6,1); //0x40814[6]=1b
```

```
*outputda0=ace_RBitWt(*outputda0,7,1); //0x40814[7]=1b
*outputda0=ace_RBitWt(*outputda0,0,0); //0x40814[0]=0b
*outputda0=ace_RBitWt(*outputda0,1,0); //0x40814[1]=0b
*outputda0=ace_RBitWt(*outputda0,2,0); //0x40814[2]=0b
*outputda0=ace_RBitWt(*outputda0,3,0); //0x40814[3]=0b
*outputda0=ace_RBitWt(*outputda0,4,0); //0x40814[4]=0b
*outputda0=ace_RBitWt(*outputda0,5,0); //0x40814[5]=0b
*outputda0=ace_RBitWt(*outputda0,6,0); //0x40814[6]=0b
*outputda0=ace_RBitWt(*outputda0,7,0); //0x40814[7]=0b
```

### 17.2.8. ace\_RBitTg (Register Bit Toggle)

#### ● (函數)

unsigned int ace\_RBitTg(unsigned int Din, const unsigned BSel5)

#### ● 函數功能

對指定暫存器的bit資料做Toggle動作

#### ● 作輸入參數

Din : 想要Toggle資料的變數

BSel5 : Target Bit Select, 5 Bits常數

Return : 執行Bit Toggle後回傳的資料, 變數

#### ● 數包含標頭檔

Peripheral\_lib/ace\_user.h

#### ● e函數返回值

執行Bit Toggle後回傳的資料, 變數

#### ● 數函數用法

/\*寫入0x40814的Bit0和Bit1 資料做Toggle變化\*/

```
unsigned int a=0x40814; //pio2_2
unsigned char *outputda0=(unsigned char*)a;
*outputda0=ace_RBitTg(*outputda0,0); //0x40814[0] to do toggle
*outputda0=ace_RBitTg(*outputda0,0); //0x40814[0] to do toggle
*outputda0=ace_RBitTg(*outputda0,1); //0x40814[1] to do toggle
*outputda0=ace_RBitTg(*outputda0,1); //0x40814[1] to do toggle
```

### 17.2.9. ace\_RBitXor (Register Bit Level XOR)

#### ● c函數

unsigned int ace\_RBitXor (unsigned int Din, const unsigned MSel3, unsigned int Dm)

#### ● 函數功能

執行Bit Level XOR. 通常ECC運算會使用到該功能

#### ● 算輸入參數

Din：執行Bit Level XOR的來源變數, 32 Bits變數

MSEL3：執行Bit Level XOR的模式選擇, 3 Bits常數

MSEL3	Mask	Description
0	55555555	將Din的Bit 0, 2, 4, 6, ..., 28, 30執行XOR
1	AAAAAAAA	將Din的Bit 1, 3, 5, 7, ..., 29, 31執行XOR
2	CCCCCCCC	將Din的Bit 2, 3, 6, 7, ..., 30, 31執行XOR
3	F0F0F0F0	
4	FF00FF00	
5	FFFF0000	將Din的Bit 16, 17, 18, 19, ..., 30, 31執行XOR
6	FFFFFFFF	將Din的所有位元執行XOR
7	Dm	根據Dm變數以1指定的位元執行XOR

Dm：自行指定XOR位元的Mask, 32 Bits變數

Return：執行Bit Level XOR的結果, 只有0或1

●有包含標頭檔

Peripheral\_lib/ace\_user.h

●e函數返回值

執行Bit Level XOR的結果, 只有0或1

●有函數用法

/\*對0x41F00做Xor, 並由變數d讀取返回值\*/

(\*volatile unsigned int \*)0x41F00)=0x0000000F;

unsigned int c=0x41F00,d=0;

unsigned int \*outputda3=(unsigned int\*)c;

```
d=ace_RBitXor(*outputda3,7,0x00); //MSEL3=7. if 0x41f00=0xf=1111, 取0000 do Xor, d=0
d=ace_RBitXor(*outputda3,7,0x01); //MSEL3=7. if 0x41f00=0xf=1111, 取0001 do Xor, d=1
d=ace_RBitXor(*outputda3,7,0x02); //MSEL3=7. if 0x41f00=0xf=1111, 取0010 do Xor, d=1
d=ace_RBitXor(*outputda3,7,0x03); //MSEL3=7. if 0x41f00=0xf=1111, 取0011 do Xor, d=0
d=ace_RBitXor(*outputda3,7,0x04); //MSEL3=7. if 0x41f00=0xf=1111, 取0100 do Xor, d=1
d=ace_RBitXor(*outputda3,7,0x05); //MSEL3=7. if 0x41f00=0xf=1111, 取0101 do Xor, d=0
d=ace_RBitXor(*outputda3,7,0x06); //MSEL3=7. if 0x41f00=0xf=1111, 取0110 do Xor, d=0
d=ace_RBitXor(*outputda3,7,0x07); //MSEL3=7. if 0x41f00=0xf=1111, 取0111 do Xor, d=1
d=ace_RBitXor(*outputda3,7,0x08); //MSEL3=7. if 0x41f00=0xf=1111, 取1000 do Xor, d=1
d=ace_RBitXor(*outputda3,7,0x09); //MSEL3=7. if 0x41f00=0xf=1111, 取1001 do Xor, d=0
```

## 18. Revision History

Version	Page	Revision Summary	The Date Of Revision
V01	ALL	First edition	2017/03/07
V02	CH2.2.9(SYS_EnableGIE)	Add HW9 Before modification: SYS_EnableGIE(4, 0x1FF); After modification : SYS_EnableGIE(4, 0x3FF);	2018/05/30
	CH 4.3.19(DrvPWM0_Open) /CH4.3.20(DrvPWM1_Open)	Before modification: 7 : Rsv After modification : 7 : Port 9.4 =PWMO0, Port 9.5 =PWMO1	
	CH5	Add PT13 control functions	
	CH 8.3.31(DrvUART2_Open) /CH8.3.60(DrvUART2_ConfigIO)	Before modification: 7 : Port 9.6 =TX2, Port 9.7 =RX2 After modification 7 : Rsv	
	CH6.3.10(DrvADC_OSR)	Modification ADC Data Output Rates	
	CH6.3.11(DrvADC_ClkEnable)	Before modification: 1 : HS_CK/2 After modification 1 : Reserved	
	CH7	Modify the functions : uint32_t DrvSPI32_IsRxBufferFull(void) uint32_t DrvSPI32_IsTxBufferFull(void)	
	CH9	Correction the DrvIA_PInputChannel and DrvIA_NInputChannel and DrvIA_SetIInputChannel reighster description	
	CH11	ADD LVD functions void DrvPMU_DisableENLVD (void); void DrvPMU_EnableENLVD (void); void DrvPMU_SetLVDVS(unsigned char uLVDVS); void DrvPMU_SetLVD12(unsigned char uLVD12); void DrvPMU_SetLVDS(unsigned int uLVDS); unsigned int DrvPMU_GetLVDO(void);	
	CH15	Modify the DrvLCD_IOMode Add note on DrvLCD_WriteData function	
	CH16	1. Add note on Flash burn function. VDD3V have to work more than 2.7V 2. add Flash burn function "ROM_BurnWordonly" and "ROM_BurnPageWriteonly"	
V03	CH3	Add function, DrvCLOCK_EnableENHAO and DrvCLOCK_SelectHOSC_CalHAO	
	CH11	Modify the E_VDDA_OUTPUT_VOLTAGE definition value.	
	CH13	Add functions, DrvRTC_EnableWUEn and DrvRTC_DisableWUEn	
	CH14	Add functions, DrvI2C_EnableSEn and DrvI2C_DisableSEn and DrvI2C_EnableI2Cen	
	CH16	Modify Flash Function, the return value.	