AW9523X_AW9527

Software Design Guide

1 AW9523X_AW9527 Overview

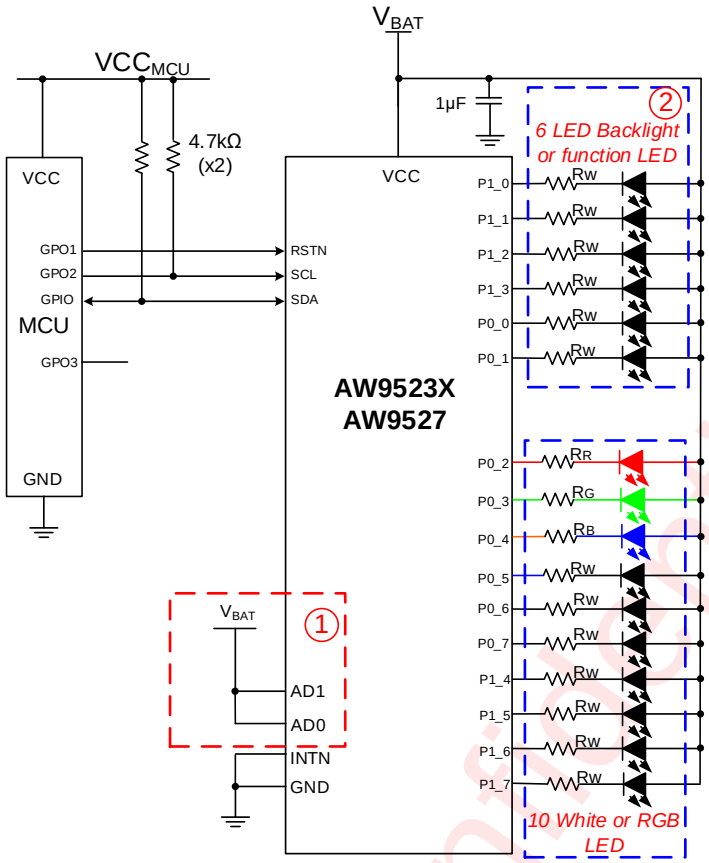
AW9523X series contains AW9523BTQR, AW9523CTQR, AW9523DTQR, AW9523QTQR four kinds of products.

AW9523X_AW9527 series chip has 16 multi-function ports, each port can be used as LED driver or GPIO. The 16 ports on the chip are divided into two groups, each group has eight ports labeled P0 and P1. The chip multifunctional port can be configured via I2C interface to work in GPIO mode or LED mode.

When the port is used as an LED driver, the chip uses constant current linear dimming to provide adjustable 256 brightness levels. The maximum output current of each port is 37mA.

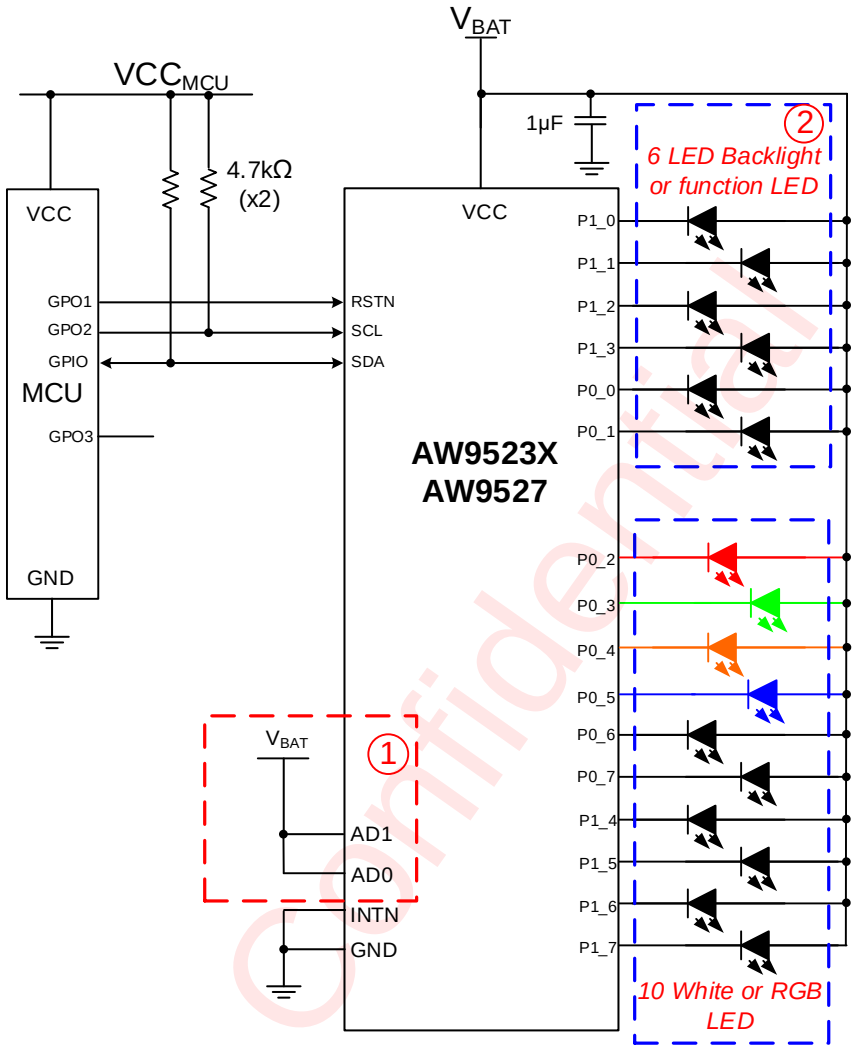
AW9527 support **linear current regulation by GCC**. AW9527 supports **BLINK mode** and has flexible and efficient lighting effect programming function, which can reduce controller resource occupation.

When ports are used as GPIO, each port can be configured to input or output mode independently. When the output mode is set to output mode, the output type of port P1 is Push-Pull output. The output type of the P0 port defaults to Open-Drain output and can be configured to Open-Drain or Push-Pull output via a register. When the port is configured in input mode, the GPIO status can be obtained by reading the register. If the port interrupt function is enabled in input mode, the chip will generate an interrupt signal in the INTN pin when the level of the GPIO port changes.



1. When LED anode is connected to VBAT, the AD1/AD0 PIN must be connected to VBAT to assure that the default value of GPIO after POWER ON is High or Hi-Z so that LED cannot be lighted falsely. The default value of GPIO after POWER ON is decided by AD1/AD0 PIN (refer to table 1).
2. The resistor R_R /R_G /R_B are only thermal reduction. And they are determined by V_{LED}, VF of LED, V_{drop} of LEDx and I_{LED}. $R_X=(V_{LED}-VF_X-V_{drop})/I_{LED}$.
3. The Dropout performance of the low 6 LED Ports（P1_0~P1_3, P0_0~P0_1）is optimized, so these ports are recommended if you want to use AW9523X_AW9527 to drive LED backlight.

Figure 1 AW9523X_AW9527 GPIO Mode Typical Application Circuit



1. When LED anode is connected to VBAT, the AD1/AD0 PIN must be connected to VBAT to assure that the default value of GPIO after POWER ON is High or Hi-Z so that LED cannot be lighted falsely.
The default value of GPIO after POWER ON is decided by AD1/AD0 PIN (refer to table 1).

2. The Dropout performance of the low 6 LED Ports (P1_0~P1_3, P0_0~P0_1) is optimized, so these ports are recommended if you want to use AW9523X_AW9527 to drive LED backlight.

Figure 2 AW9523X_AW9527 LED Mode Typical Application Circuit

2 Software Design

2.1 I²C Address

The I2C address of the AW9523X_AW9527 series chip is configured by pins AD0 and AD1, supporting 4 different device addresses. Bit7 to Bit3 of I2C address are fixed values of "10110". The values of Bit2 and Bit1 are determined by the level of AD1 and AD0 respectively. Bit0 represents the read/write control bit of I2C protocol.

Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	0	1	1	0	AD1	AD0	R/W

Table 1 Relationship between AD0/AD1 and I2C address

When AD1=GND, AD0=GND, Bit2-Bit1 is "00", I2C 7-bit address is 0x58,

When AD1=GND, AD0=VCC, Bit2-Bit1 is "01", I2C 7-bit address is 0x59,

When AD1=VCC, AD0=GND, Bit2-Bit1 is "10", I2C 7-bit address is 0x5A,

When AD1=VCC, AD0=VCC, Bit2-Bit1 is "11", I2C 7-bit address is 0x5B.

2.2 Power On And Off

2.2.1 Power On

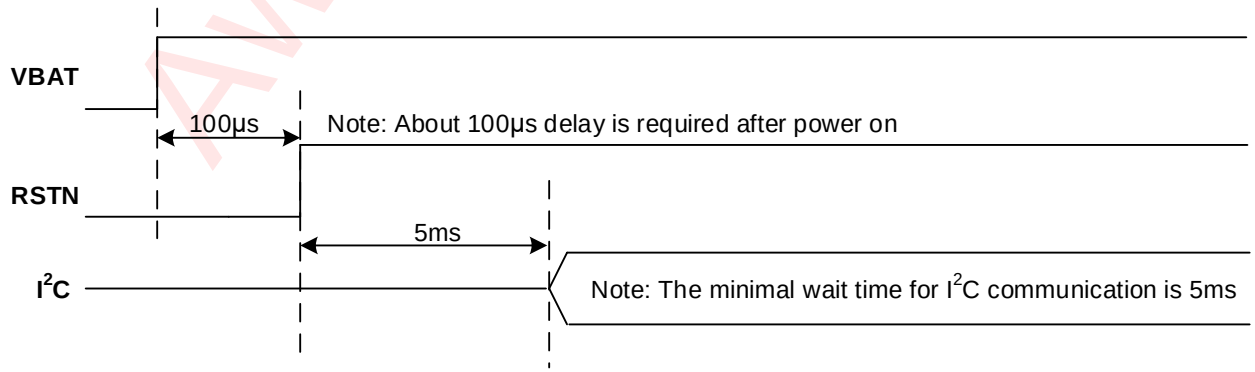


Figure 3 AW9523X_AW9527 Power On Timing

For AW9523X_AW9527 series chips to work properly, a certain power-on sequence must be met. After AW9523X_AW9527 series chips are powered on, at least 100μs delay is required to pull up the RSTN pin. After pulling up the RSTN pin, at least 5ms delay is required before the main control can communicate with AW9523X_AW9527 series chips through the I2C interface.

After the AW9523X_AW9527 series chip is powered on, all ports work in GPIO output mode by default. The state of the port is determined by the level state of the AD1 and AD0 pins. For details about the relationship between port status and address selection pins of P0 and P1 after power-on, see Table 2.

AD1	AD0	P1_7	P1_6	P1_5	P1_4	P1_3	P1_2	P1_1	P1_0	P0_7	P0_6	P0_5	P0_4	P0_3	P0_2	P0_1	P0_0
GND	GND	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GND	VBAT	0	0	0	0	1	1	1	1	0	0	0	0	Hi-Z	Hi-Z	Hi-Z	Hi-Z
VBAT	GND	1	1	1	1	0	0	0	0	Hi-Z	Hi-Z	Hi-Z	Hi-Z	0	0	0	0
VBAT	VBAT	1	1	1	1	1	1	1	1	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z

Table 2 Default port status after the chip is powered on

The chip power-on reference code is as follows:

```
/* AW9523X_AW9527 chip enable */
static void aw9523x_aw9527_chip_enable(AW_BOOL enable)
{
    if (enable) {
        GPIO_EN_HIGH();
        /* About 5ms delay is required after power on */
        delay_ms(5);
    } else {
        GPIO_EN_LOW();
    }
}
```

2.2.2 Power Off

There is no special timing requirement for powering off the chip.

2.3 Chip Reset

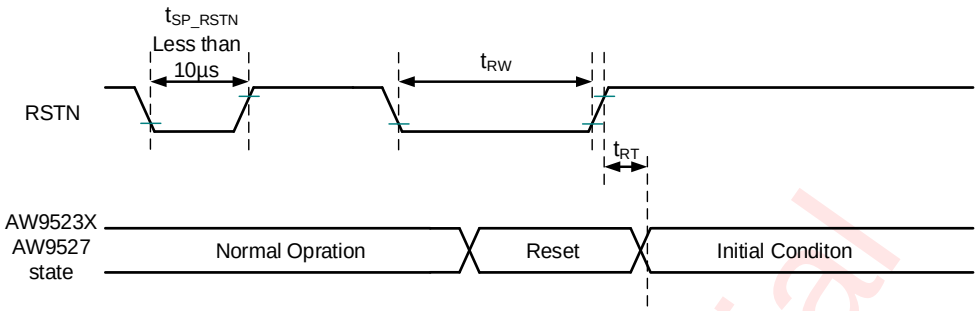


Figure 4 AW9523X_AW9527 Hardware Reset Timing

AW9523X_AW9527 chip reset function is as follows:

- Power On Reset: After the chip is powered on for 5ms, the chip is initialized to the default state.
- Hardware Reset: The RSTN pin remains in a low state for more than 20μs, and all circuits inside the chip are reset.
- Software Reset: Write 00H to the SW_RSTN (7FH) register, and all the internal circuits of the chip are reset.

The reference code for chip software reset is as follows:

```
1.  /* AW9523X_AW9527 software reset */
2.  static void aw9523x_aw9527_soft_rst(void)
3.  {
4.      aw_i2c_write_reg(SW_RSTN, 0x00);
5.      /* delay 2ms at least */
6.      delay_ms(2);
7.  }
```

2.4 Interrupt Function

The AW9523X_AW9527 series chips detects the level status of the input port when the port is configured to GPIO input mode and interrupt is enabled. When the input port level status

changes, the chip will pull the INTN pin down. The host can clear the interrupt state of the chip by reading the INPUT_PORT0 (00H) or INPUT_PORT1 (01H) register. After the interrupt clears, the chip will stop pulling down the INTN pin. In application, the INTN pin needs an external pull-up resistor. Reference Figure 5 for interrupt trigger timing.

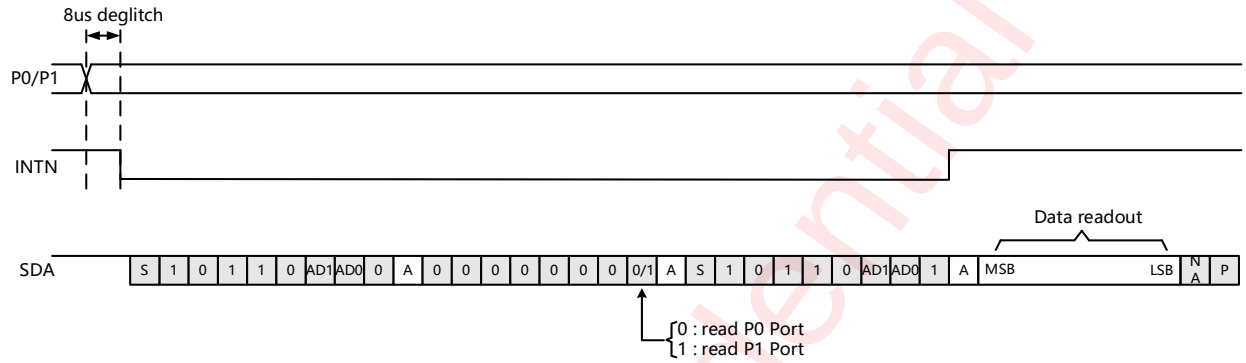


Figure 5 Interrupt generation and clear

1. When the interrupt-enabled input port detects a level change, from high level to low level or from low level to high level. 8μs anti-peak interference processing is carried out automatically inside the chip to ensure the effectiveness of state change.
2. When the electrical port change is effective, the chip will pull down the INTN pin.
3. When an interrupt is triggered and the INTN pin is pulled down, the master controller can clear the interrupt event by reading the INPUT_PORT0 (00H) and INPUT_PORT1 (01H) registers. After clearing the interrupt, the chip will stop pulling down the INTN pin. INTN pin state will be determined by the external level.

2.5 Work Mode

2.5.1 Linear Dimming

AW9523X_AW9527 series chips as an LED driver, provides four maximum current level configuration, all ports can be configured to LED drive mode, each port supports independent control of 256 levels of dimming, the maximum current of 37mA.

The steps to configure the chip to the LED dimming mode are as follows:

1. Power on the chip and pull up the RSTN pin with a delay of at least 5ms.
2. Set the port to the LED working mode based on the application. Port P0_7-P0_0 of P0 is controlled by [Bit7-Bit0] of register 12H, and port P1_7-P1_0 of P1 is controlled by [Bit7-Bit0] of register 13H. When the Bit of the corresponding register of port is set to 0, the port works in LED mode.
3. Set the global maximum output current and set ISEL[Bit1-Bit0] of the CTL (11H) register. The control logic of the register is: when ISEL[Bit1-Bit0]=00, the maximum dimming current is Imax; When ISEL[Bit1-Bit0]=01, the maximum dimming current is Imax*3/4. When ISEL[Bit1-Bit0]=10, the maximum dimming current is Imax*2/4. When ISEL[Bit1-Bit0]=11, the maximum dimming current is Imax*1/4. Where, Imax=37mA.
4. Set the port brightness value. Adjust the current value of different ports by configuring the DIMx (x=0~15) (20H-2FH) register, that is, change the brightness of the LED. See Table 3 for the mapping between registers and ports.
5. If you only need to turn off some leds, you can set the current value of the channel corresponding to the LED to 0.

6. If the entire chip does not need to work, pull down the RSTN pin to reduce power consumption. To turn it on again, perform Steps 1-4.

Address	Name	Description	Default
20H	DIM0	P1_0 port LED current control	00H
21H	DIM1	P1_1 port LED current control	00H
22H	DIM2	P1_2 port LED current control	00H
23H	DIM3	P1_3 port LED current control	00H
24H	DIM4	P0_0 port LED current control	00H
25H	DIM5	P0_1 port LED current control	00H
26H	DIM6	P0_2 port LED current control	00H
27H	DIM7	P0_3 port LED current control	00H
28H	DIM8	P0_4 port LED current control	00H
29H	DIM9	P0_5 port LED current control	00H
2AH	DIM10	P0_6 port LED current control	00H
2BH	DIM11	P0_7 port LED current control	00H
2CH	DIM12	P1_4 port LED current control	00H
2DH	DIM13	P1_5 port LED current control	00H
2EH	DIM14	P1_6 port LED current control	00H
2FH	DIM15	P1_7 port LED current control	00H

Table 3 Mapping between the port brightness value and the port

The reference code for the software configuration of LED dimming mode is as follows:

```
//Function declaration
void aw_i2c_write_byte(unsigned char reg_addr, unsigned char reg_val);
unsigned char aw_i2c_read_byte(unsigned char reg_addr);
void set_rstn_pin_high(); //pull RSTN pin

/* init chip */
set_rstn_pin_high(); //pull up RSTN pin
delay_ms(10);

aw_i2c_write_byte(0x12, 0x00); //set P0 port is LED mode
aw_i2c_write_byte(0x13, 0x00); //set P1 port is LED mode
val = aw_i2c_read_byte(0x11);
aw_i2c_write_byte(0x11, val&(~0x03)|0x00); //set globe LED Imax is 37mA

/* LED brightness control */
aw_i2c_write_byte(0x20, 0xFF); // Set the LED brightness to port P1_0 to max is 0xFF
aw_i2c_write_byte(0x21, 0xFF); // Set the LED brightness to port P1_1 to max is 0xFF
aw_i2c_write_byte(0x22, 0xFF); // Set the LED brightness to port P1_2 to max is 0xFF
aw_i2c_write_byte(0x23, 0xFF); // Set the LED brightness to port P1_3 to max is 0xFF
aw_i2c_write_byte(0x24, 0xFF); // Set the LED brightness to port P0_0 to max is 0xFF
```

```
aw_i2c_write_byte(0x25, 0xFF); // Set the LED brightness to port P0_1 to max is 0xFF
aw_i2c_write_byte(0x26, 0xFF); // Set the LED brightness to port P0_2 to max is 0xFF
aw_i2c_write_byte(0x27, 0xFF); // Set the LED brightness to port P0_3 to max is 0xFF
aw_i2c_write_byte(0x28, 0xFF); // Set the LED brightness to port P0_4 to max is 0xFF
aw_i2c_write_byte(0x29, 0xFF); // Set the LED brightness to port P0_5 to max is 0xFF
aw_i2c_write_byte(0x2A, 0xFF); // Set the LED brightness to port P0_6 to max is 0xFF
aw_i2c_write_byte(0x2B, 0xFF); // Set the LED brightness to port P0_7 to max is 0xFF
aw_i2c_write_byte(0x2C, 0xFF); // Set the LED brightness to port P1_4 to max is 0xFF
aw_i2c_write_byte(0x2D, 0xFF); // Set the LED brightness to port P1_5 to max is 0xFF
aw_i2c_write_byte(0x2E, 0xFF); // Set the LED brightness to port P1_6 to max is 0xFF
aw_i2c_write_byte(0x2F, 0xFF); // Set the LED brightness to port P1_7 to max is 0xFF
/* if you need to close led, sush as close P1_0 */
aw_i2c_write_byte(0x20, 0x00);
```

Note: The above code is pseudocode and is only used for reference in the configuration process.

2.5.2 GPIO Function

All 16 ports of the AW9523X_AW9527 series chips can be configured in GPIO mode. In GPIO mode, both input and output directions can be configured. To configure a port to GPIO mode, perform the following steps:

1. Power on the chip and pull up the RSTN pin to delay at least 5ms.
2. Set all GPIO outputs to low level. Port P0_7-P0_0 of P0 can be set to 0 for [Bit7-Bit0] of the OUTPUT_PORT0 (02H) register, and port P1_7-P1_0 of P1 can be set to 0 for [Bit7-Bit0] of the OUTPUT_PORT1 (03H) register.
3. Set the port mode to GPIO. Port P0_7-P0_0 of P0 can be configured with [Bit7-Bit0] of the 12H register as 1, and port P1_7-P1_0 of P1 can be configured with [Bit7-Bit0] of the 13H register as 1.
4. Set the port mode to Input or output. P0_7-P0_0 port mode is controlled by [Bit7-Bit0] in the CONFIG_PORT0 (04H) register and P1_7-P1_0 port mode is controlled by [Bit7-Bit0] in the CONFIG_PORT1 (05H) register. When the Bit corresponding to a port is set

to 0, the port works in output mode. When the Bit corresponding to a port is set to 1, the port works in input mode.

5. In output mode, the output status of port P0_7-P0_0 is controlled by [Bit7-Bit0] of OUTPUT_PORT0 (02H) register, and port P1_7-P1_0 is controlled by [Bit7-Bit0] of OUTPUT_PORT1 (03H) register. When the Bit corresponding to a port is set to 0, the port output is high. When the Bit corresponding to a port is set to 1, the port output is low.

   a). P1 port output is Push-Pull output by default and cannot be reconfigured.

   b). When P0 port is set to output, the default mode is Open-Drain. The CTL (11H) register [Bit4] can be set to Push-Pull output or Open-Drain output.

6. If set to input mode, the status of ports P0_7-P0_0 can be represented by [Bit7-Bit0] of register INPUT_PORT0 (00H), and port P1_7-P1_0 can be represented by [Bit7-Bit0] of register INPUT_PORT1 (01H), respectively. INPUT_PORT0 (00H) and INPUT_PORT1 (01H) are read-only registers and cannot be written. Based on the input mode, INT_PORT0 (06H) and INT_PORT1 (07H) can be configured to control input interrupts on ports P0_7-P0_0 and P1_7-P1_0, disabling the interrupt when the corresponding Bit is 1 and enabling the interrupt when the corresponding Bit is 0.

7. If the whole chip does not need to work, the RSTN pin can be pulled down to reduce power consumption.

Take port P0 as input and port P1 as output. The software configuration reference code is as follows:

```
1. //Function declaration
2. void aw_i2c_write_byte(unsigned char reg_addr, unsigned char reg_val);
3. unsigned char aw_i2c_read_byte(unsigned char reg_addr);
4. void set_shdn_port_high();
5.
```



```
/* Suppose P0 is set as input and P1 set outputt */
/* chip init */
set_rstn_pin_high(); // pull up RSTN pin
delay_ms(10);

aw_i2c_write_byte(0x02, 0x00); //set P0 port output low level
aw_i2c_write_byte(0x03, 0x00); //set P1 port output low level
aw_i2c_write_byte(0x12, 0xff); //set P0 port is GPIO mode
aw_i2c_write_byte(0x13, 0xff); //set P1 port is GPIO mode
aw_i2c_write_byte(0x04, 0xff); //set P0 input mode
aw_i2c_write_byte(0x06, 0x00); //enable P0 interrupt
aw_i2c_write_byte(0x05, 0x00); //set P1 output mode
val = aw_i2c_read_byte(0x11);
aw_i2c_write_byte(0x11, val|0x01<<4); //set P0 Push-Pull output mode

/* gpio control */
/* set P1 all high */
aw_i2c_write_byte(0x03, 0xff); //set P1 port all high

/* get P0 input state */
val = aw_i2c_read_byte(0x00);

/* set P1_x state */
val = aw_i2c_read_byte(0x03);
aw_i2c_write_byte(0x03, val&(~0x01<<0)); //set P1_0 low
aw_i2c_write_byte(0x03, val&(~0x01<<1)); //set P1_1 low

/* When using input, pay attention to clear the interrupt after triggering the interrupt,
   and the clear interrupt must be separated and read 0x00 and 0x01 registers*/
val = aw_i2c_read_byte(0x00);
val = aw_i2c_read_byte(0x01);
```

Note: The above code is pseudocode and is only used to illustrate the configuration process.

2.5.3 BLINK mode

Ports P0_0~P0_1 and P1_0~P1_3 of the AW9527 support BLINK mode. In BLINK mode, the periodic breathing effect can be automatically completed until you exit BLINK mode or turn off the breathing function.

The process for configuring a chip port to work in BLINK mode is as follows:

1. Power on the chip and pull up the RSTN pin to delay at least 5ms.
2. Set the port to the LED working mode based on the application. P0_1~P0_0 is controlled by [Bit1-Bit0] of the P0WKMD (12H) register, P1_3~P1_0 is controlled by the [Bit3-Bit0] of the P1WKMD (13H) register. If it is set to 0, it indicates that the corresponding port works in LED mode.
3. Set the current value of the breathing port to 0. The current values of P1_0~P1_3 are controlled by the DIM0 to DIM3 (20H to 23H) registers, the current values of P0_0~P0_1 are controlled by the DIM04to DIM5 (24H to 25H) registers.
4. Set [Bit5-Bit0] of the PATEN (14H) register to 1 according to the application to enable the breathing function of P1_0~P1_3、P0_0~P0_1.
5. Set the working mode of P1_0~P1_3、P0_0~P0_1 to BLINK based on the application, and set [Bit1-Bit0] of register CONFIG_PORT0 (04H) and [Bit3-Bit0] of register CONFIG_PORT1 (05H) to 1.
6. Set the fade in and fade out time of LED and configure the [Bit5-Bit0] of FDTMR (15H) register.
7. Set the full light and dark time of LED, and configure the FLTMR (16H) register [Bit5-Bit0].
8. Set the maximum working current of LED and configure the [Bit1-Bit0] of the GCR (11H) register.
9. Trigger BLINK breathing function and set [Bit7] of GCR (11H) register to 1.
10. To disable BLINK breathing, perform the following steps in sequence to ensure that DIMx (x = 0~5) is set when breathing is disabled.

a) Turn off the respiratory function first, and set [Bit5-Bit0] of the PATEN (14H) register to 0.

b) Then clear the corresponding DIM value of LED to 0 and set the value of DIM0~DIM5 (20H~25H) register to 0, so that after the chip exits the BLINK mode, the LED will be in the state of off.

11. If the whole chip does not need to work, pull down the SHDN pin. To re-enable the function, perform 1 to 9 again.

Note: AW9527 only allow the GO control bit (the [Bit7] bit of the GCR (11H) register) to be enabled once. Before you can re-enable the GO control bit, you need to reset the chip and reconfigure the BLINK mode related registers.

The BLINK mode configuration reference code is as follows:

```
1.  //Function declaration
2.  void aw_i2c_write_byte(unsigned char reg_addr, unsigned char reg_val);
3.  unsigned char aw_i2c_read_byte(unsigned char reg_addr);
4.  void set_rstn_port_high();
5.
6.  /* init chip */
7.  set_rstn_port_high();
8.  delay_ms(10);
9.  aw_i2c_write_byte(0x12, 0x00); //set mode of P0_0-P0_1 is led
10. aw_i2c_write_byte(0x13, 0x00); //set mode of P1_0-P1_3 is led
11. aw_i2c_write_byte(0x11, 0x00); //set dim range 0-37mA
12.
13. aw_i2c_write_byte(0x20, 0x00); //set brightness of P1_0 is 0x00
14. aw_i2c_write_byte(0x21, 0x00); //set brightness of P1_1 is 0x00
15. aw_i2c_write_byte(0x22, 0x00); //set brightness of P1_2 is 0x00
16. aw_i2c_write_byte(0x23, 0x00); //set brightness of P1_3 is 0x00
17. aw_i2c_write_byte(0x24, 0x00); //set brightness of P0_0 is 0x00
18. aw_i2c_write_byte(0x25, 0x00); //set brightness of P0_1 is 0x00
19.
20. aw_i2c_write_byte(0x14, 0x1F); //enable breath mode
21. aw_i2c_write_byte(0x04, 0x03); //select P0_0-P0_1 is blink mode
22. aw_i2c_write_byte(0x05, 0x0F); //select P1_0-P1_3 is blink mode
23.
```



```
aw_i2c_write_byte(0x15, 0x09);
//set fade off/on param of blink mode <00[001][001]> fade on = fade off = 315ms
aw_i2c_write_byte(0x16, 0x12);
//set stay off/on param of blink mode <00[010][010]> stay on = stay off = 630ms

val = aw_i2c_read_byte(0x11);
aw_i2c_write_byte(0x11, val|0x01<<7); //start blink
```

2.5.4 SMART-FADE mode

The SMART-FADE mode is a semi-autonomous breathing mode. The P0_0~P0_1 and P1_0~P1_3 port on the AW9527 chip supports the SMART-FADE mode.

The process for configuring the SMART-FADE mode is as follows:

1. Power on the chip and pull up the RSTN pin to delay at least 5ms.
2. Set the port to the LED working mode based on the application. P0_1~P0_0 is controlled by [Bit1-Bit0] of the P0WKMD (12H) register, P1_3~P1_0 is controlled by the [Bit3-Bit0] of the P1WKMD (13H) register. If it is set to 0, it indicates that the corresponding port works in LED mode.
3. Enable the breathing function of ports according to the application, and set [Bit5:Bit0] of the PATEN (14H) register to 1.
4. Select the LEDx (x = 0~5) working mode as SMART-FADE, and set [Bit1-Bit0] of the CONFIG_PORT0 (04H) register and [Bit3-Bit0] of the CONFIG_PORT1 (05H) register to 0.
5. Set the fade in and fade out time of LED and configure the [Bit5-Bit0] of FDTMR (15H) register.

6. Enable LEDx (x = 0~5) to fade in, set [Bit1-Bit0] of the OUTPUT_PORT0 (02H) register and [Bit3-Bit0] of the OUTPUT_PORT1 (03H) register to 1, and LEDx (x = 0~5) starts to lighten.
7. Wait for the set LEDx (x = 0~5) brightening time.
8. Enable LEDx (x = 0~5) to fade out, set [Bit1-Bit0] of the OUTPUT_PORT0 (02H) register and [Bit3-Bit0] of the OUTPUT_PORT1 (03H) register to 0, and LEDx (x = 0~5) starts to dim.
9. Wait for the set LEDx (x = 0~5) to dim.
10. Repeat 6-9 if you need to turn on and off.
11. If the breathing parameters need to be modified, perform Step 5-9 again.
12. If the chip is not required to work, pull down the RSTN pin to reduce power consumption. Run 1-9 to run the SMART-FADE command again.

The reference code for configuring the SMART-FADE mode is as follows:

```
1.  //Function declaration
2.  void aw_i2c_write_byte(unsigned char reg_addr, unsigned char reg_val);
3.  unsigned char aw_i2c_read_byte(unsigned char reg_addr, unsigned char reg_val);
4.  void set_shdn_port_high();
5.
6.  /* init chip */
7.  set_shdn_port_high();
8.  delay_ms(10);
9.  aw_i2c_write_byte(0x12, 0x00); //set mode of P0_0-P0_1 is led
10. aw_i2c_write_byte(0x13, 0x00); //set mode of P1_0-P1_3 is led
11. aw_i2c_write_byte(0x11, 0x00); //set dim range 0-37mA
12.
13. aw_i2c_write_byte(0x20, 0x00); //set brightness of P1_0 is 0x00
14. aw_i2c_write_byte(0x21, 0x00); //set brightness of P1_1 is 0x00
15. aw_i2c_write_byte(0x22, 0x00); //set brightness of P1_2 is 0x00
16. aw_i2c_write_byte(0x23, 0x00); //set brightness of P1_3 is 0x00
17. aw_i2c_write_byte(0x24, 0x00); //set brightness of P0_0 is 0x00
18. aw_i2c_write_byte(0x25, 0x00); //set brightness of P0_1 is 0x00
19.
```



```
aw_i2c_write_byte(0x14, 0x1F); //enable breath mode

aw_i2c_write_byte(0x04, 0x00); //select P0_0-P0_1 is smart-fade mode
aw_i2c_write_byte(0x05, 0x00); //select P1_0-P1_3 is smart-fade mode
aw_i2c_write_byte(0x15, 0x09);
//set fade off/on param of blink mode <00[001][001]> fade on = fade off = 315ms

aw_i2c_write_byte(0x02, 0x03); //P0_0-P0_1 start fade on
aw_i2c_write_byte(0x03, 0x0F); //P1_0-P1_3 start fade on

aw_i2c_write_byte(0x02, 0x00); //P0_0-P0_1 start fade off
aw_i2c_write_byte(0x03, 0x00); //P1_0-P1_3 start fade off
```


3 Matters Needing Attention

3.1 Application Description of GPIO Mode

1. The default HiZ port on the chip cannot be used for GPIO directly. You need to add weak pull-up or weak pull-down to the port to ensure that the port remains stable after initialization.
2. When the interrupt function is used, the INTN pin has a Open-Drain architecture inside, so the external pin needs to be pulled up through a KΩ pull-up resistor. Once an interrupt is generated, register clear interrupt is performed on each port. That is, register INPUT_PORT0 (00H) and register INPUT_PORT1 (01H) are read separately.

3.2 LED Mode application description

When using AW9527 to configure PORT as LED mode, when configuring breath parameters (Corresponding to FDTMR (15H), FLTMR (16H) registers), Note that the 3-bit parameters of fade-on, fade-off, all-on, and all-off can only be set to 001, 010, 011, 100, and 101. The corresponding relationships are as follows:

3bit breath parameters	Breathing time
001	315ms
010	630ms
011	1260ms
100	2520ms
101	5040ms

Revision History

Version	Date	Description
V1.0	2019-08-04	Initial release of the software design guide
V1.1	2021-11-24	Modify clear interrupt
V1.2	2023-01-31	Modified process for configuring GPIO functions
V1.3	2023-04-07	Updating a document template
V1.4	2024-12-04	1.Added AW9527QNR 2. Update declaration
V1.5	2025-01-22	Added description of BLINK mode

Declaration

All trademarks are the property of their respective owners. Information in this document is believed to be accurate and reliable. However, Shanghai AWINIC Technology Co., Ltd (AWINIC Technology) does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information.

AWINIC Technology reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. Customers shall obtain the latest relevant information before placing orders and shall verify that such information is current and complete. This document supersedes and replaces all information supplied prior to the publication hereof.

AWINIC provides the reference design and document for typical application. Please modify and verify according to the actual application and AWINIC disclaims all warranties.

Reproduction of AWINIC information in AWINIC technical document is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. AWINIC is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

