Ameba-Z SDK User Configuration

This document introduces usage of rtl8710b_sleepcfg.c/rtl8710b_pinmapcfg.c/rtl8710b_intfcfg.c.

Table of Contents

1. SUMMARY 3

1.1. USER CONFIGURATION FILES 3

2. PINMAP CONFIGURATION 4

3. SLEEP CONFIGURATION 5

3.1. SLEEP_PWRMGT_CONFIG 5

3.2. SLEEP_WEVENT_CONFIG 5

3.3. SLEEP_WAKEPIN_CONFIG 6

4. DEEP STANDBY CONFIGURATION 7

4.1. DSTANDBY_WEVENT_CONFIG 7

4.2. DSTANDBY_WAKEPIN_CONFIG 7

5. DEEP SLEEP CONFIGURATION 8

5.1. DSLEEP_WEVENT_CONFIG 8

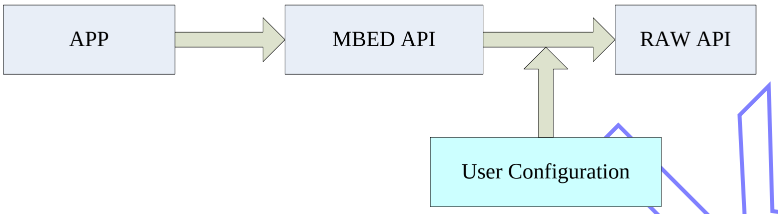
5.2. DSLEEP_WAKEPIN_CONFIG 8

6. INTERFACE CONFIGURATION 9

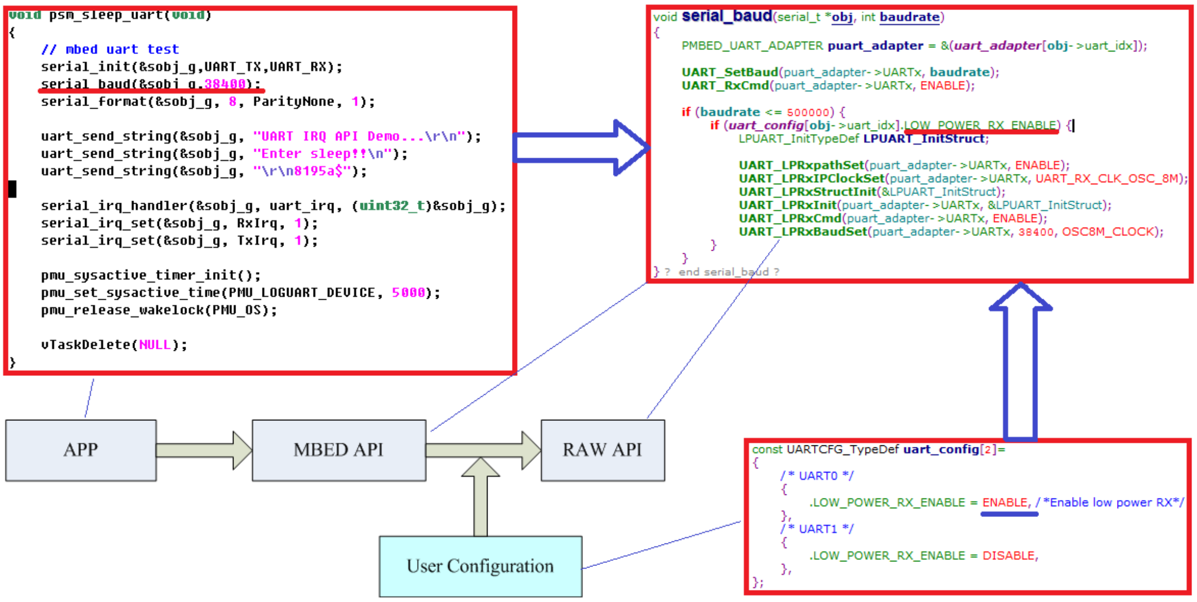
6.1. UART CONFIGURATION 9

1. Summary

RAW API is complex, but it provides rich parameters and functions to use hardware advanced features. MBED API is simple and standard, but lots of features cannot be used because API limitation. User Configuration is used between MBED and RAW to provide some advanced features and keep API simple.



For example, MBED API doesn’t support UART low power feature, but you can use UART low power feature without changing MBED API like following:



```
void psm_sleep_uart(void)
{
    // mbed uart test
    serial_init(&sobj_g,UART_TX,UART_RX);
    serial_baud(&sobj_g,38400);
    serial_format(&sobj_g, 8, ParityNone, 1);

    uart_send_string(&sobj_g, "UART IRQ API Demo...\r\n");
    uart_send_string(&sobj_g, "Enter sleep!!\n");
    uart_send_string(&sobj_g, "\r\n8195a$");

    serial_irq_handler(&sobj_g, uart_irq, (uint32_t)&sobj_g);
    serial_irq_set(&sobj_g, RxIrq, 1);
    serial_irq_set(&sobj_g, TxIrq, 1);

    pmu_sysactive_timer_init();
    pmu_set_sysactive_time(PMU_LOGUART_DEVICE, 5000);
    pmu_release_wakelock(PMU_OS);

    vTaskDelete(NULL);
}
```

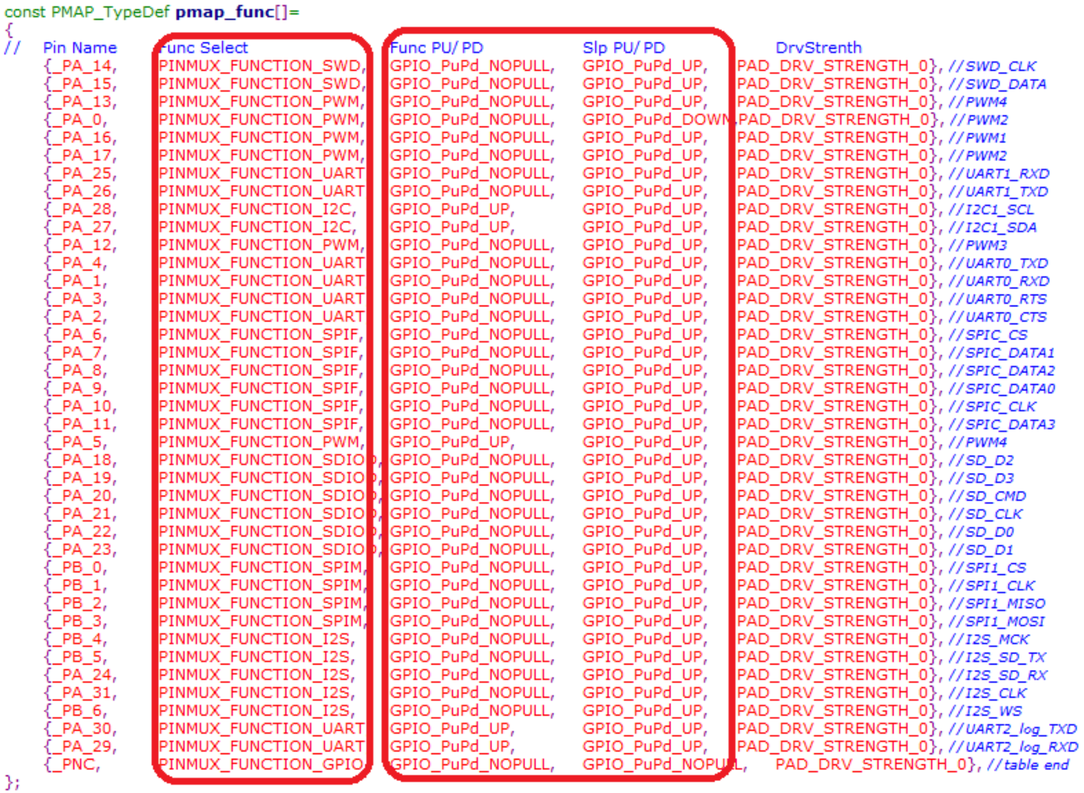
```
void serial_baud(serial_t *obj, int baudrate)
{
    PMBED_UART_ADAPTER puart_adapter = &(uart_adapter[obj->uart_idx]);

    UART_SetBaud(puart_adapter->UARTx, baudrate);
    UART_RxCmd(puart_adapter->UARTx, ENABLE);

    if (baudrate <= 500000) {
        if (uart_config[obj->uart_idx].LOW_POWER_RX_ENABLE) {
            LPUART_InitTypeDef LPUART_InitStruct;

            UART_LPRxpathSet(puart_adapter->UARTx, ENABLE);
            UART_LPRxIPClockSet(puart_adapter->UARTx, UART_RX_CLK_OSC_8M);
            UART_LPRxStructInit(&LPUART_InitStruct);
            UART_LPRxInit(puart_adapter->UARTx, &LPUART_InitStruct);
            UART_LPRxCmd(puart_adapter->UARTx, ENABLE);
            UART_LPRxBaudSet(puart_adapter->UARTx, 38400, OSC8M_CLOCK);
        }
    }
} ? end serial_baud ?
```

```
const UARTCFG_TypeDef uart_config[2]=
{
    /* UART0 */
    {
        .LOW_POWER_RX_ENABLE = ENABLE, /*Enable low power RX*/
    },
    /* UART1 */
    {
        .LOW_POWER_RX_ENABLE = DISABLE,
    },
};
```

1.1. User Configuration Files

User Configuration Files	*Configuration Items*
<Rtl8710b_intfcfg.c>	■ UART/I2C/SPI… interfaces
<Rtl8710b_pinmapcfg.c>	■ per pin function configuration ■ per pin function pull up & pull down ■ per pin sleep pull up & pull down, reduce power leakage when CM4 sleep
<Rtl8710b_sleepcfg.c>	■ Sleep power management ■ Sleep wakeup event ■ Sleep wake pin
<Rtl8710b_dstandbycfg.c>	■ Dstandby wakeup event ■ Dstandby wake pin
<Rtl8710b_dsleepcfg.c>	■ Dsleep wake pin

2. Pinmap Configuration

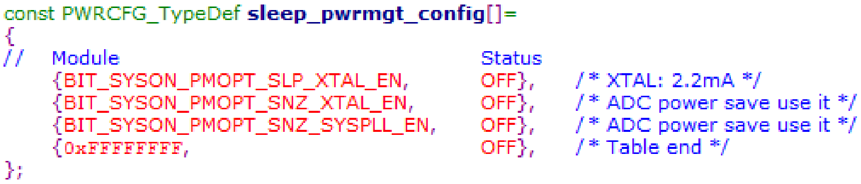


```
const PMAP_TypeDef pmap_func[]=
{
//    Pin Name      Func Select               Func PU/PD            Slp PU/PD            DrvStrenth
      {_PA_14,      PINMUX_FUNCTION_SWD,      GPIO_PuPd_NOPULL,     GPIO_PuPd_UP,        PAD_DRV_STRENGTH_0}, //SWD_CLK
      {_PA_15,      PINMUX_FUNCTION_SWD,      GPIO_PuPd_NOPULL,     GPIO_PuPd_UP,        PAD_DRV_STRENGTH_0}, //SWD_DATA
      {_PA_13,      PINMUX_FUNCTION_PWM,      GPIO_PuPd_NOPULL,     GPIO_PuPd_UP,        PAD_DRV_STRENGTH_0}, //PWM4
      {_PA_0,       PINMUX_FUNCTION_PWM,      GPIO_PuPd_NOPULL,     GPIO_PuPd_DOWN,PAD_DRV_STRENGTH_0}, //PWM2
      {_PA_16,      PINMUX_FUNCTION_PWM,      GPIO_PuPd_NOPULL,     GPIO_PuPd_UP,        PAD_DRV_STRENGTH_0}, //PWM1
      {_PA_17,      PINMUX_FUNCTION_PWM,      GPIO_PuPd_NOPULL,     GPIO_PuPd_UP,        PAD_DRV_STRENGTH_0}, //PWM2
      {_PA_25,      PINMUX_FUNCTION_UART      GPIO_PuPd_NOPULL,     GPIO_PuPd_UP,        PAD_DRV_STRENGTH_0}, //UART1_RXD
      {_PA_26,      PINMUX_FUNCTION_UART      GPIO_PuPd_NOPULL,     GPIO_PuPd_UP,        PAD_DRV_STRENGTH_0}, //UART1_TXD
      {_PA_28,      PINMUX_FUNCTION_I2C,      GPIO_PuPd_UP,         GPIO_PuPd_UP,        PAD_DRV_STRENGTH_0}, //I2C1_SCL
      {_PA_27,      PINMUX_FUNCTION_I2C,      GPIO_PuPd_UP,         GPIO_PuPd_UP,        PAD_DRV_STRENGTH_0}, //I2C1_SDA
      {_PA_12,      PINMUX_FUNCTION_PWM,      GPIO_PuPd_NOPULL,     GPIO_PuPd_UP,        PAD_DRV_STRENGTH_0}, //PWM3
      {_PA_4,       PINMUX_FUNCTION_UART      GPIO_PuPd_NOPULL,     GPIO_PuPd_UP,        PAD_DRV_STRENGTH_0}, //UART0_TXD
      {_PA_1,       PINMUX_FUNCTION_UART      GPIO_PuPd_NOPULL,     GPIO_PuPd_UP,        PAD_DRV_STRENGTH_0}, //UART0_RXD
      {_PA_3,       PINMUX_FUNCTION_UART      GPIO_PuPd_NOPULL,     GPIO_PuPd_UP,        PAD_DRV_STRENGTH_0}, //UART0_RTS
      {_PA_2,       PINMUX_FUNCTION_UART      GPIO_PuPd_NOPULL,     GPIO_PuPd_UP,        PAD_DRV_STRENGTH_0}, //UART0_CTS
      {_PA_6,       PINMUX_FUNCTION_SPIF,     GPIO_PuPd_NOPULL,     GPIO_PuPd_UP,        PAD_DRV_STRENGTH_0}, //SPIC_CS
      {_PA_7,       PINMUX_FUNCTION_SPIF,     GPIO_PuPd_NOPULL,     GPIO_PuPd_UP,        PAD_DRV_STRENGTH_0}, //SPIC_DATA1
      {_PA_8,       PINMUX_FUNCTION_SPIF,     GPIO_PuPd_NOPULL,     GPIO_PuPd_UP,        PAD_DRV_STRENGTH_0}, //SPIC_DATA2
      {_PA_9,       PINMUX_FUNCTION_SPIF,     GPIO_PuPd_NOPULL,     GPIO_PuPd_UP,        PAD_DRV_STRENGTH_0}, //SPIC_DATA0
      {_PA_10,      PINMUX_FUNCTION_SPIF,     GPIO_PuPd_NOPULL,     GPIO_PuPd_UP,        PAD_DRV_STRENGTH_0}, //SPIC_CLK
      {_PA_11,      PINMUX_FUNCTION_SPIF,     GPIO_PuPd_NOPULL,     GPIO_PuPd_UP,        PAD_DRV_STRENGTH_0}, //SPIC_DATA3
      {_PA_5,       PINMUX_FUNCTION_PWM,      GPIO_PuPd_UP,         GPIO_PuPd_UP,        PAD_DRV_STRENGTH_0}, //PWM4
      {_PA_18,      PINMUX_FUNCTION_SDIOD,    GPIO_PuPd_NOPULL,     GPIO_PuPd_UP,        PAD_DRV_STRENGTH_0}, //SD_D2
      {_PA_19,      PINMUX_FUNCTION_SDIOD,    GPIO_PuPd_NOPULL,     GPIO_PuPd_UP,        PAD_DRV_STRENGTH_0}, //SD_D3
      {_PA_20,      PINMUX_FUNCTION_SDIOD,    GPIO_PuPd_NOPULL,     GPIO_PuPd_UP,        PAD_DRV_STRENGTH_0}, //SD_CMD
      {_PA_21,      PINMUX_FUNCTION_SDIOD,    GPIO_PuPd_NOPULL,     GPIO_PuPd_UP,        PAD_DRV_STRENGTH_0}, //SD_CLK
      {_PA_22,      PINMUX_FUNCTION_SDIOD,    GPIO_PuPd_NOPULL,     GPIO_PuPd_UP,        PAD_DRV_STRENGTH_0}, //SD_D0
      {_PA_23,      PINMUX_FUNCTION_SDIOD,    GPIO_PuPd_NOPULL,     GPIO_PuPd_UP,        PAD_DRV_STRENGTH_0}, //SD_D1
      {_PB_0,       PINMUX_FUNCTION_SPIM,     GPIO_PuPd_NOPULL,     GPIO_PuPd_UP,        PAD_DRV_STRENGTH_0}, //SPI1_CS
      {_PB_1,       PINMUX_FUNCTION_SPIM,     GPIO_PuPd_NOPULL,     GPIO_PuPd_UP,        PAD_DRV_STRENGTH_0}, //SPI1_CLK
      {_PB_2,       PINMUX_FUNCTION_SPIM,     GPIO_PuPd_NOPULL,     GPIO_PuPd_UP,        PAD_DRV_STRENGTH_0}, //SPI1_MISO
      {_PB_3,       PINMUX_FUNCTION_SPIM,     GPIO_PuPd_NOPULL,     GPIO_PuPd_UP,        PAD_DRV_STRENGTH_0}, //SPI1_MOSI
      {_PB_4,       PINMUX_FUNCTION_I2S,      GPIO_PuPd_NOPULL,     GPIO_PuPd_UP,        PAD_DRV_STRENGTH_0}, //I2S_MCK
      {_PB_5,       PINMUX_FUNCTION_I2S,      GPIO_PuPd_NOPULL,     GPIO_PuPd_UP,        PAD_DRV_STRENGTH_0}, //I2S_SD_TX
      {_PA_24,      PINMUX_FUNCTION_I2S,      GPIO_PuPd_NOPULL,     GPIO_PuPd_UP,        PAD_DRV_STRENGTH_0}, //I2S_SD_RX
      {_PA_31,      PINMUX_FUNCTION_I2S,      GPIO_PuPd_NOPULL,     GPIO_PuPd_UP,        PAD_DRV_STRENGTH_0}, //I2S_CLK
      {_PB_6,       PINMUX_FUNCTION_I2S,      GPIO_PuPd_NOPULL,     GPIO_PuPd_UP,        PAD_DRV_STRENGTH_0}, //I2S_WS
      {_PA_30,      PINMUX_FUNCTION_UART      GPIO_PuPd_UP,         GPIO_PuPd_UP,        PAD_DRV_STRENGTH_0}, //UART2_log_TXD
      {_PA_29,      PINMUX_FUNCTION_UART      GPIO_PuPd_UP,         GPIO_PuPd_UP,        PAD_DRV_STRENGTH_0}, //UART2_log_RXD
      {_PNC,        PINMUX_FUNCTION_GPIO      GPIO_PuPd_NOPULL,     GPIO_PuPd_NOPULL,       PAD_DRV_STRENGTH_0}, //table end
};
```

Config Items	*Introduction*
<Func Select>	■ Set per pin pinmux based on PCB board
<Func PU/PD>	■ Set per pin PU/PD based on <Func Select>
<Slp PU/PD>	■ Set per pin PU/PD when CPU sleep, used to reduce power leakage
<DrvStrenth>	■ PD Driving Strength

3. Sleep Configuration

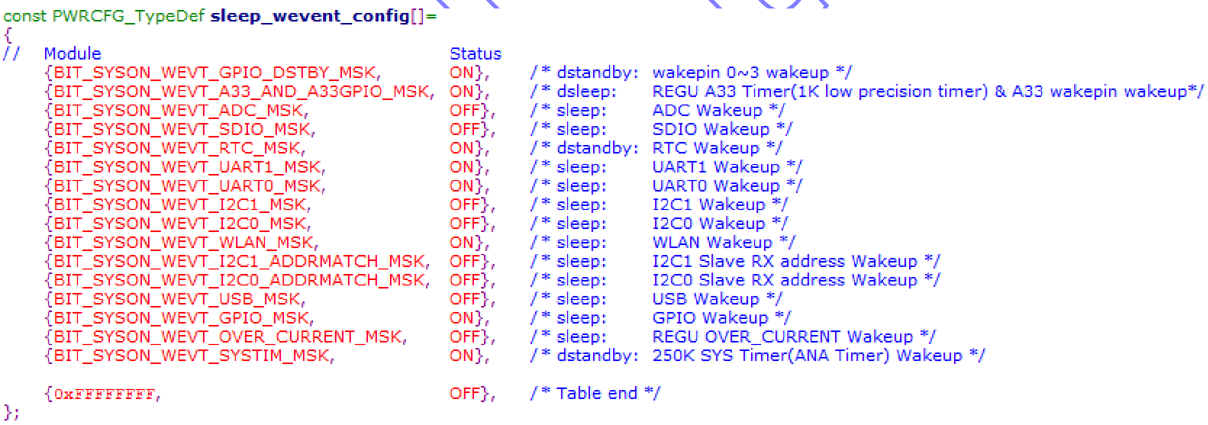
3.1. sleep_pwrmgt_config



```
const PWRCFG_TypeDef sleep_pwrmgt_config[]=
{
//    Module                              Status
      {BIT_SYSON_PMOPT_SLP_XTAL_EN,       OFF},    /* XTAL: 2.2mA */
      {BIT_SYSON_PMOPT_SNZ_XTAL_EN,       OFF},    /* ADC power save use it */
      {BIT_SYSON_PMOPT_SNZ_SYSPLL_EN,     OFF},    /* ADC power save use it */
      {0xFFFFFFFF,                        OFF},    /* Table end */
};
```

Config Items	*Introduction*
< BIT_SYSON_PMOPT_SLP_XTAL_EN >	■ Open when high speed UART wakeup used
< BIT_SYSON_PMOPT_SNZ_XTAL_EN >	■ Open when ADC power save mode used
< BIT_SYSON_PMOPT_SNZ_SYSPLL_EN >	■ Open when ADC power save mode used

3.2. sleep_wevent_config

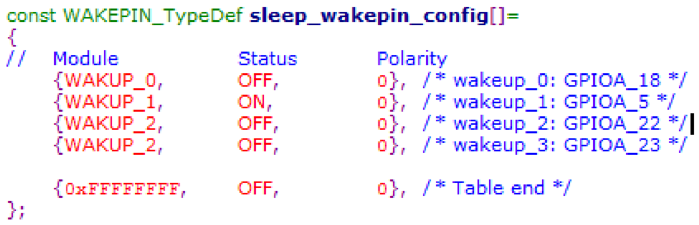


```
const PWRCFG_TypeDef sleep_wevent_config[]=
{
//    Module                              Status
      {BIT_SYSON_WEVT_GPIO_DSTBY_MSK,      ON},    /* dstandby:  wakepin 0~3 wakeup */
      {BIT_SYSON_WEVT_A33_AND_A33GPIO_MSK, ON},    /* dsleep:    REGU A33 Timer(1K low precision timer) & A33 wakepin wakeup*/
      {BIT_SYSON_WEVT_ADC_MSK,             OFF},   /* sleep:     ADC Wakeup */
      {BIT_SYSON_WEVT_SDIO_MSK,            OFF},   /* sleep:     SDIO Wakeup */
      {BIT_SYSON_WEVT_RTC_MSK,             ON},    /* dstandby:  RTC Wakeup */
      {BIT_SYSON_WEVT_UART1_MSK,           ON},    /* sleep:     UART1 Wakeup */
      {BIT_SYSON_WEVT_UART0_MSK,           ON},    /* sleep:     UART0 Wakeup */
      {BIT_SYSON_WEVT_I2C1_MSK,            OFF},   /* sleep:     I2C1 Wakeup */
      {BIT_SYSON_WEVT_I2C0_MSK,            OFF},   /* sleep:     I2C0 Wakeup */
      {BIT_SYSON_WEVT_WLAN_MSK,            ON},    /* sleep:     WLAN Wakeup */
      {BIT_SYSON_WEVT_I2C1_ADDRMATCH_MSK,  OFF},   /* sleep:     I2C1 Slave RX address Wakeup */
      {BIT_SYSON_WEVT_I2C0_ADDRMATCH_MSK,  OFF},   /* sleep:     I2C0 Slave RX address Wakeup */
      {BIT_SYSON_WEVT_USB_MSK,             OFF},   /* sleep:     USB Wakeup */
      {BIT_SYSON_WEVT_GPIO_MSK,            ON},    /* sleep:     GPIO Wakeup */
      {BIT_SYSON_WEVT_OVER_CURRENT_MSK,    OFF},   /* sleep:     REGU OVER_CURRENT Wakeup */
      {BIT_SYSON_WEVT_SYSTIM_MSK,          ON},    /* dstandby:  250K SYS Timer(ANA Timer) Wakeup */

      {0xFFFFFFFF,                         OFF},   /* Table end */
};
```

Config Items	*Wake Source*
< BIT_SYSON_WEVT_GPIO_DSTBY_MSK>	■ WAKEUP_0 ■ WAKEUP_1 ■ WAKEUP_2 ■ WAKEUP_3
<BIT_SYSON_WEVT_A33_AND_A33GPIO_MSK>	■ 1K low precision timer Timeout(140min)
<BIT_SYSON_WEVT_ADC_MSK>	■ ADC interrupt
<BIT_SYSON_WEVT_SDIO_MSK>	■ SDIO interrupt
<BIT_SYSON_WEVT_RTC_MSK>	■ RTC Alarm Interrupt
<BIT_SYSON_WEVT_UART1_MSK>	■ UART1 Interrupt
<BIT_SYSON_WEVT_UART0_MSK>	■ UART1 Interrupt
<BIT_SYSON_WEVT_I2C1_MSK>	■ I2C1 Interrupt
<BIT_SYSON_WEVT_I2C0_MSK>	■ I2C0 Interrupt
<BIT_SYSON_WEVT_WLAN_MSK>	■ WLAN WOWLAN

<BIT_SYSON_WEVT_I2C1_ADDRMATCH_MSK>	■ I2C1 Master address match
<BIT_SYSON_WEVT_I2C0_ADDRMATCH_MSK>	■ I2C0 Master address match
<BIT_SYSON_WEVT_USB_MSK>	■ USB Interrupt
<BIT_SYSON_WEVT_GPIO_MSK>	■ GPIOA_X Interrupt
<BIT_SYSON_WEVT_OVER_CURRENT_MSK>	■ Regulator OVER_CURRENT
<BIT_SYSON_WEVT_SYSTIM_MSK>	■ 250K System Timer Timeout (<8s)

3.3. sleep_wakepin_config

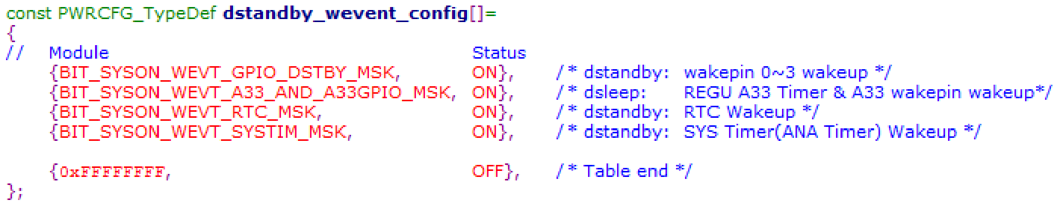


```
const WAKEPIN_TypeDef sleep_wakepin_config[]=
{
//   Module          Status       Polarity
     {WAKUP_0,       OFF,         0},  /* wakeup_0: GPIOA_18 */
     {WAKUP_1,       ON,          0},  /* wakeup_1: GPIOA_5 */
     {WAKUP_2,       OFF,         0},  /* wakeup_2: GPIOA_22 */
     {WAKUP_2,       OFF,         0},  /* wakeup_3: GPIOA_23 */

     {0xFFFFFFFF,    OFF,         0},  /* Table end */
};
```

Config Items	*pinname*	*Polarity*
< ***WAKEUP_0***>	GPIOA_18	■ 0: Low level wakeup ■ 1: High level wakeup
< ***WAKEUP_1***>	GPIOA_5	■ 0: Low level wakeup ■ 1: High level wakeup
< ***WAKEUP_2***>	GPIOA_22	■ 0: Low level wakeup ■ 1: High level wakeup
< ***WAKEUP_3***>	GPIOA_23	■ 0: Low level wakeup ■ 1: High level wakeup

4. Deep Standby Configuration

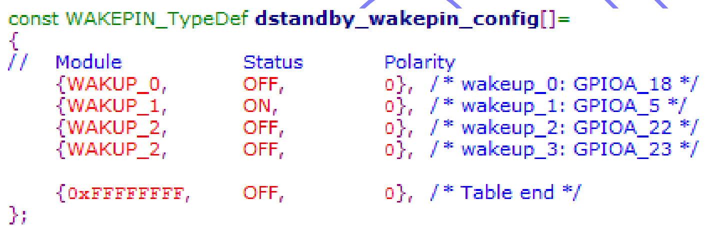
4.1. dstandby_wevent_config



```
const PWRCFG_TypeDef dstandby_wevent_config[]=
{
//   Module                              Status
     {BIT_SYSON_WEVT_GPIO_DSTBY_MSK,      ON},     /* dstandby:  wakepin 0~3 wakeup */
     {BIT_SYSON_WEVT_A33_AND_A33GPIO_MSK, ON},     /* dsleep:    REGU A33 Timer & A33 wakepin wakeup*/
     {BIT_SYSON_WEVT_RTC_MSK,             ON},     /* dstandby:  RTC Wakeup */
     {BIT_SYSON_WEVT_SYSTIM_MSK,          ON},     /* dstandby:  SYS Timer(ANA Timer) Wakeup */

     {0xFFFFFFFF,                         OFF},    /* Table end */
};
```

Config Items	*Wake Source*
< ***BIT_SYSON_WEVT_GPIO_DSTBY_MSK***>	■ WAKEUP_0 ■ WAKEUP_1 ■ WAKEUP_2 ■ WAKEUP_3
<***BIT_SYSON_WEVT_A33_AND_A33GPIO_MSK***>	■ 1K low precision timer Timeout(140min)
<***BIT_SYSON_WEVT_RTC_MSK***>	■ RTC Alarm Interrupt
<***BIT_SYSON_WEVT_SYSTIM_MSK***>	■ 250K System Timer Timeout (<8s)

4.2. dstandby_wakepin_config

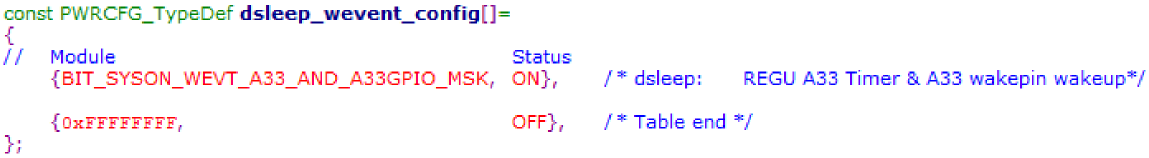


```
const WAKEPIN_TypeDef dstandby_wakepin_config[]=
{
//    Module          Status        Polarity
      {WAKUP_0,       OFF,          0},  /* wakeup_0: GPIOA_18 */
      {WAKUP_1,       ON,           0},  /* wakeup_1: GPIOA_5 */
      {WAKUP_2,       OFF,          0},  /* wakeup_2: GPIOA_22 */
      {WAKUP_2,       OFF,          0},  /* wakeup_3: GPIOA_23 */

      {0xFFFFFFFF,    OFF,          0},  /* Table end */
};
```

Config Items	*pinname*	*Polarity*
< ***WAKEUP_0***>	GPIOA_18	■ 0: Low level wakeup ■ 1: High level wakeup
< ***WAKEUP_1***>	GPIOA_5	■ 0: Low level wakeup ■ 1: High level wakeup
< ***WAKEUP_2***>	GPIOA_22	■ 0: Low level wakeup ■ 1: High level wakeup
< ***WAKEUP_3***>	GPIOA_23	■ 0: Low level wakeup ■ 1: High level wakeup

5. Deep Sleep Configuration

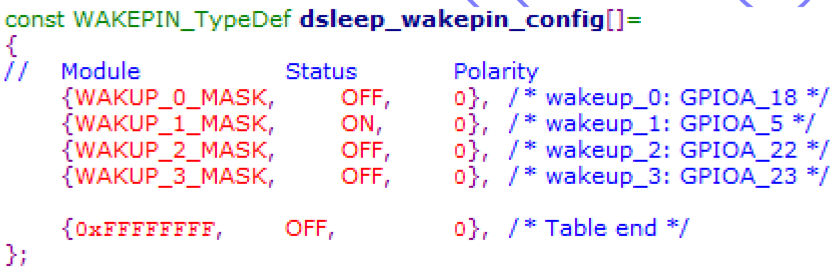
5.1. dsleep_wevent_config



```
const PWRCFG_TypeDef dsleep_wevent_config[]=
{
//    Module                                    Status
      {BIT_SYSON_WEVT_A33_AND_A33GPIO_MSK,      ON},      /* dsleep:     REGU A33 Timer & A33 wakepin wakeup*/

      {0xFFFFFFFF,                              OFF},     /* Table end */
};
```

Config Items	*Wake Source*
<BIT_SYSON_WEVT_A33_AND_A33GPIO_MSK>	■ WAKEUP_0 ■ WAKEUP_1 ■ WAKEUP_2 ■ WAKEUP_3 ■ 1K low precision timer Timeout(140min)

5.2. dsleep_wakepin_config

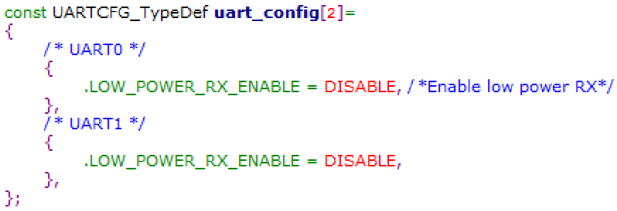


```
const WAKEPIN_TypeDef dsleep_wakepin_config[]=
{
//    Module              Status        Polarity
      {WAKUP_0_MASK,         OFF,       0},   /* wakeup_0: GPIOA_18 */
      {WAKUP_1_MASK,         ON,        0},   /* wakeup_1: GPIOA_5 */
      {WAKUP_2_MASK,         OFF,       0},   /* wakeup_2: GPIOA_22 */
      {WAKUP_3_MASK,         OFF,       0},   /* wakeup_3: GPIOA_23 */

      {0xFFFFFFFF,        OFF,          0},   /* Table end */
};
```

Config Items	*pinname*	*Polarity*
< *WAKEUP_0*>	GPIOA_18	■ 0: Low level wakeup ■ 1: Not support
< *WAKEUP_1*>	GPIOA_5	■ 0: Low level wakeup ■ 1: Not support
< *WAKEUP_2*>	GPIOA_22	■ 0: Low level wakeup ■ 1: Not support
< *WAKEUP_3*>	GPIOA_23	■ 0: Low level wakeup ■ 1: Not support

6. Interface Configuration

6.1. UART Configuration



```
const UARTCFG_TypeDef uart_config[2]=
{
    /* UART0 */
    {
        .LOW_POWER_RX_ENABLE = DISABLE, /*Enable low power RX*/
    },
    /* UART1 */
    {
        .LOW_POWER_RX_ENABLE = DISABLE,
    },
};
```

Config Items	*Polarity*
< ***LOW_POWER_RX_ENABLE***>	■ ENABLE: Low power UART ENABLE ■ DISABLE: Low power UART DISABLE