
How to set complementary pins as SPI2 function?

Questions:

How to configure PB13/PB14/PB15 pins correspondint to CH1C/CH2C/CH3C as SPI2_CS, SPI2_SCK, SPI2_MISO respectively when TMR1_CH1, CH2 and CH3 output PWM?

Answer:

1. At initialization, first enable TMR1 and then SPI2
2. When in use, after TMR1 output enable bit is enabled (OEN=1), it is not allowed to be disabled and enabled again
3. After OEN=1, the C1CP bit (channel complementary polarity) can only be set to 0. In other words, this bit and OEN bit cannot be set to 1 at the same time.
4. It is recommended to disable PWM by setting CxEN=0, and enable PWM by setting CxEN=1

Example:

```
#define PWM_OFF    TMR1->cctrl &= (~(u16)(0x0111));
#define PWM_ON     TMR1->cctrl |= (u16)(0x0111);
void tmr1_init(void)
{
    gpio_init_type gpio_init_struct;

    /* gpioa clock enable */
    crm_periph_clock_enable(CRM_GPIOA_PERIPH_CLOCK, TRUE);

    /* tmr1 clock enable */
    crm_periph_clock_enable(CRM_TMR1_PERIPH_CLOCK, TRUE);
    crm_periph_clock_enable(CRM_IOMUX_PERIPH_CLOCK, TRUE);
    gpio_pin_remap_config(TMR1_MUX_01, TRUE);

    gpio_default_para_init(&gpio_init_struct);

    gpio_init_struct.gpio_pins = GPIO_PINS_8 | GPIO_PINS_9 | GPIO_PINS_10;
    gpio_init_struct.gpio_out_type = GPIO_OUTPUT_PUSH_PULL;
    gpio_init_struct.gpio_pull = GPIO_PULL_NONE;
    gpio_init_struct.gpio_mode = GPIO_MODE_MUX;
    gpio_init_struct.gpio_drive_strength = GPIO_DRIVE_STRENGTH_STRONGER;
    gpio_init(GPIOA, &gpio_init_struct);

    /* compute the prescaler value */
    prescaler_value = (uint16_t)(system_core_clock / 24000000) - 1;
```

```
/* tmr1 time base configuration */
tmr_base_init(TMR1, 665, prescaler_value);
tmr_cnt_dir_set(TMR1, TMR_COUNT_TWO_WAY_3);
tmr_clock_source_div_set(TMR1, TMR_CLOCK_DIV1);

tmr_output_default_para_init(&tmr_oc_init_structure);
tmr_oc_init_structure.oc_mode = TMR_OUTPUT_CONTROL_PWM_MODE_A;
tmr_oc_init_structure.oc_idle_state = FALSE;
tmr_oc_init_structure.oc_polarity = TMR_OUTPUT_ACTIVE_HIGH;
tmr_oc_init_structure.oc_output_state = TRUE;
tmr_output_channel_config(TMR1, TMR_SELECT_CHANNEL_1, &tmr_oc_init_structure);
tmr_channel_value_set(TMR1, TMR_SELECT_CHANNEL_1, ccr1_val);
tmr_output_channel_buffer_enable(TMR1, TMR_SELECT_CHANNEL_1, TRUE);

tmr_output_channel_config(TMR1, TMR_SELECT_CHANNEL_2, &tmr_oc_init_structure);
tmr_channel_value_set(TMR1, TMR_SELECT_CHANNEL_2, ccr2_val);
tmr_output_channel_buffer_enable(TMR1, TMR_SELECT_CHANNEL_2, TRUE);

tmr_output_channel_config(TMR1, TMR_SELECT_CHANNEL_3, &tmr_oc_init_structure);
tmr_channel_value_set(TMR1, TMR_SELECT_CHANNEL_3, ccr3_val);
tmr_output_channel_buffer_enable(TMR1, TMR_SELECT_CHANNEL_3, TRUE);

tmr_period_buffer_enable(TMR1, TRUE);

tmr_output_enable(TMR1, TRUE);

/* tmr1 enable counter */
tmr_counter_enable(TMR1, TRUE);

PWM_OFF;
PWM_ON;
}

void spi2_init(void)
{
    gpio_init_type gpio_init_struct;
    spi_init_type spi_init_struct;
```

```
/* gpiob clock enable */
crm_periph_clock_enable(CRM_GPIOB_PERIPH_CLOCK, TRUE);

crm_periph_clock_enable(CRM_SPI2_PERIPH_CLOCK, TRUE);

gpio_default_para_init(&gpio_init_struct);

/* spi2 sck pin */
gpio_init_struct.gpio_pull      = GPIO_PULL_DOWN;
gpio_init_struct.gpio_mode      = GPIO_MODE_INPUT;
gpio_init_struct.gpio_pins = GPIO_PINS_13;
gpio_init(GPIOB, &gpio_init_struct);

/* slave miso pin */
gpio_init_struct.gpio_pull      = GPIO_PULL_UP;
gpio_init_struct.gpio_mode      = GPIO_MODE_MUX;
gpio_init_struct.gpio_pins = GPIO_PINS_14;
gpio_init(GPIOB, &gpio_init_struct);

/* spi2 mosi pin */
gpio_init_struct.gpio_pull      = GPIO_PULL_UP;
gpio_init_struct.gpio_mode      = GPIO_MODE_INPUT;
gpio_init_struct.gpio_pins = GPIO_PINS_15;
gpio_init(GPIOB, &gpio_init_struct);

spi_default_para_init(&spi_init_struct);
spi_init_struct.transmission_mode = SPI_TRANSMIT_SIMPLEX_RX;
spi_init_struct.master_slave_mode = SPI_MODE_SLAVE;
spi_init_struct.mclk_freq_division = SPI_MCLK_DIV_8;
spi_init_struct.first_bit_transmission = SPI_FIRST_BIT_LSB;
spi_init_struct.frame_bit_num = SPI_FRAME_8BIT;
spi_init_struct.clock_polarity = SPI_CLOCK_POLARITY_LOW;
spi_init_struct.clock_phase = SPI_CLOCK_PHASE_2EDGE;
spi_init_struct.cs_mode_selection = SPI_CS_SOFTWARE_MODE;
spi_init(SPI2, &spi_init_struct);

spi_enable(SPI2, TRUE);
}
```

Type: MCU applications

Applicable products: AT32F403, AT32F413, AT32F415, AT32F403A, AT32F407

Main function: TMR1, SPI2

Minor function: None

Document revision history

| Date | Revision | Changes |
|-----------|----------|-----------------|
| 2022.2.16 | 2.0.0 | Initial release |

IMPORTANT NOTICE – PLEASE READ CAREFULLY

Purchasers are solely responsible for the selection and use of ARTERY's products and services, and ARTERY assumes no liability whatsoever relating to the choice, selection or use of the ARTERY products and services described herein.

No license, express or implied, to any intellectual property rights is granted under this document. If any part of this document deals with any third party products or services, it shall not be deemed a license grant by ARTERY for the use of such third party products or services, or any intellectual property contained therein, or considered as a warranty regarding the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

Unless otherwise specified in ARTERY's terms and conditions of sale, ARTERY provides no warranties, express or implied, regarding the use and/or sale of ARTERY products, including but not limited to any implied warranties of merchantability, fitness for a particular purpose (and their equivalents under the laws of any jurisdiction), or infringement of any patent, copyright or other intellectual property right.

Purchasers hereby agrees that ARTERY's products are not designed or authorized for use in: (A) any application with special requirements of safety such as life support and active implantable device, or system with functional safety requirements; (B) any air craft application; (C) any automotive application or environment; (D) any space application or environment, and/or (E) any weapon application. Purchasers' unauthorized use of them in the aforementioned applications, even if with a written notice, is solely at purchasers' risk, and is solely responsible for meeting all legal and regulatory requirement in such use.

Resale of ARTERY products with provisions different from the statements and/or technical features stated in this document shall immediately void any warranty grant by ARTERY for ARTERY products or services described herein and shall not create or expand in any manner whatsoever, any liability of ARTERY.