Application Note

Document No.: AN1107

APM32F411 Series COMP Application Note

Version: V1.0

1 Introduction

This application note provides a guide on how to configure and use comparators on the APM32F411 series.

Two independent comparators (COMP1 and COMP2) are embedded in APM32F411. COMP1 supports ultra-low power operation, and COMP2 supports rail-to-rail input (the input voltage range is from minimum to maximum voltage), fast and slow modes. COMP1 and COMP2 can be combined to form a window comparator for use. The output signal of the comparators can be redirected to the timer, to be used as the braking signal of the timer, the input capture signal or the external clock source of the counter.

This application note will demonstrate the usage of the APM32F411 comparator and how it works together with other modules in the following three application scenarios:

- PWM control
- Frequency measurement
- Window comparator

Contents

1 Introduction 1

2 COMP introduction..3

2.1 COMP input..4

2.2 COMP output..4

2.3 COMP pin configuration 5

3 COMP application example 6

3.1 PWM control...6

3.2 Frequency measurement 8

3.3 Window comparator 10

4 Revision history 13

2 COMP introduction

COMP (comparator) is a module used to compare the input signals and generate corresponding outputs based on the comparison results, usually used to detect the size or status of input signals.

COMP has two input ends – non-inverting input (INP) and inverting input (INM). After COMP is enabled, it will compare the size of the non-inverting input signal and the inverting input signal. When the non-inverting input signal is greater than the inverting input signal, COMP will output a high level; when the non-inverting input signal is smaller than the inverting input signal, COMP will output a low level; when the non-inverting input signal is equal to the inverting input signal, the output state of COMP is uncertain. The output of COMP can be redirected to other modules for use as input signals.

Two COMP (COMP1 and COMP2) are embedded in APM32F411. In APM32F411, COMP does not have a separate clock enable control bit, and its clock is independent and synchronized with the PCLK clock. The comparator can only reset the modules through system reset. APM32F411 can configure the output polarity of COMP to achieve inverting output.

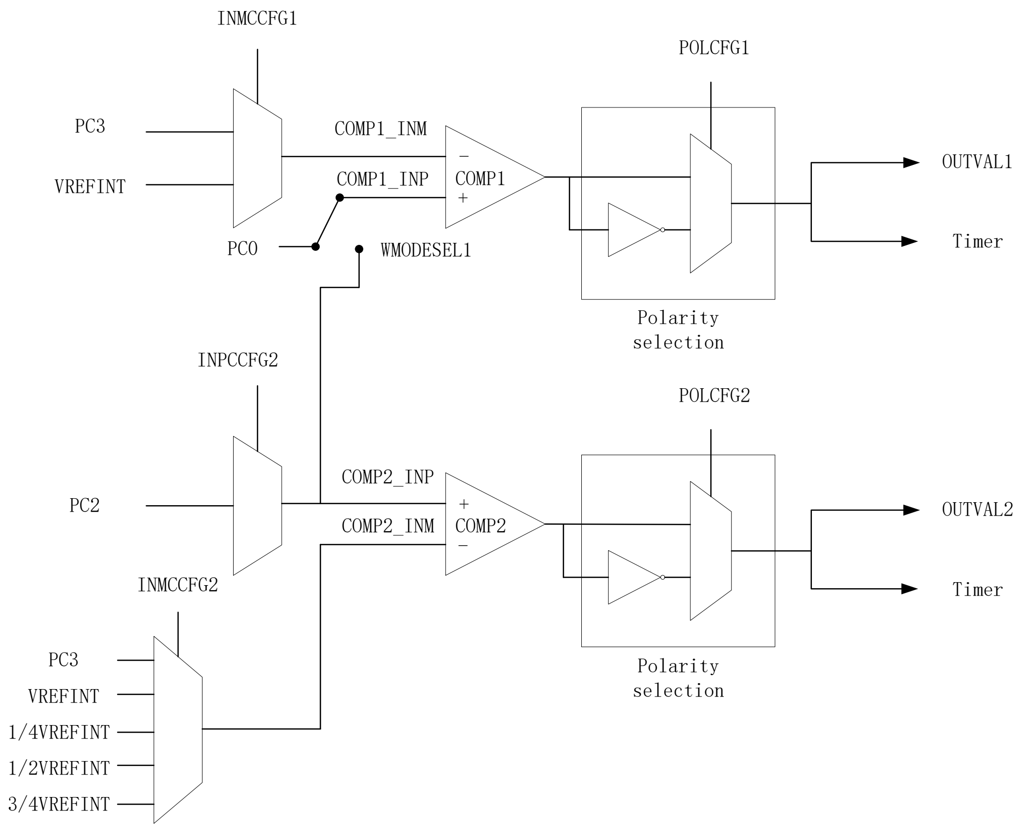

Figure 2 COMP Structure Block Diagram

2.1 COMP input

The non-inverting input end of COMP1 is connected to the external pin PC0, while the inverting input end can be connected to the external pin PC1 or reference voltage VREFINT; the non-inverting input end of COMP2 is connected to the external pin PC2, while the inverting input end can be connected to the external pin PC3, reference voltage VREFINT, or 1/4, 1/2, 3/4 of the reference voltage.

Table 1 COMP Non-inverting Input Pins

Module name	Input end	Pin
COMP1	Input Plus	PC0
COMP2	Input Plus	PC2

Table 2 COMP Inverting Input Pins

Module name	Input end	Pin
COMP1	Input Minus	PC1
COMP1	Input Minus	VREFINT (reference voltage)
COMP2	Input Minus	PC3
COMP2	Input Minus	VREFINT
COMP2	Input Minus	1/4 * VREFINT
COMP2	Input Minus	1/2 * VREFINT
COMP2	Input Minus	3/4 * VREFINT

2.2 COMP output

The COMP module output of APM32F411 can select not to be redirected to any device, or can be redirected to a timer for use as a timer braking signal, input capture signal, or ETRF (external reference clock) signal of a counter.

TMR1 (Timer 1), TMR8 (Timer 8), TMR3 (Timer 3), and TMR4 (Timer 4) are all optional objects for COMP output redirection. It is important to note that in APM32F411, only the advanced timers TMR1 and TMR8 have braking function. When the output of COMP is used as the timer braking signal, it can only be connected to TMR1 and TMR8.

There are 12 types of COMP output redirection states:

(1) No redirection object

(2) TMR1 braking signal

(3) TMR1 channel 1 input capture

(4) TMR1 external reference clock

(5) TMR8 braking signal

(6) TMR8 channel 4 input capture

(7) TMR8 external reference clock

(8) TMR2 channel 4 input capture

(9) TMR2 external reference clock

(10) TMR3 channel 1 input capture

(11) TMR3 external reference clock

(12) TMR4 channel 1 input capture

2.3 COMP pin configuration

The COMP module of APM32F411 uses pins PC0 and PC2 as the non-inverting input ends, and PC1 and PC3 as the inverting input ends.

When COMP uses these pins as input ends, the pins shall be configured to analog input mode. For example, when the external pin PC2 is used as the non-inverting input end of COMP2, the configuration example code is as follows:

```
void GPIO_Init(void)
{
     GPIO_Config_T gpioConfig;
     RCM_EnableAHB1PeriphClock(RCM_AHB1_PERIPH_GPIOC);

    GPIO_ConfigStructInit(&gpioConfig);
    gpioConfig.mode    = GPIO_MODE_AN;
    gpioConfig.pupd    = GPIO_PUPD_NOPULL;
    gpioConfig.pin     = GPIO_PIN_2;
    GPIO_Config(GPIOC, &gpioConfig);

}
```

3 COMP application example

This section describes how to apply the COMP module of APM32F411 in the example scenarios (PWM control, frequency measurement, window comparator, etc.).

3.1 PWM control

In APM32F411, the output of COMP can be used as the braking signal of advanced timers.

When the non-inverting input is greater than the inverting input, COMP will output a high-level braking signal, and the timer will stop outputting PWM waves; when the non-inverting input is smaller than the inverting input, the COMP output will return to a low level, and the timer will start to output PWM waves again in the next clock cycle. If the output polarity of COMP is inverting, the control effect of COMP on PWM will be exactly the opposite.

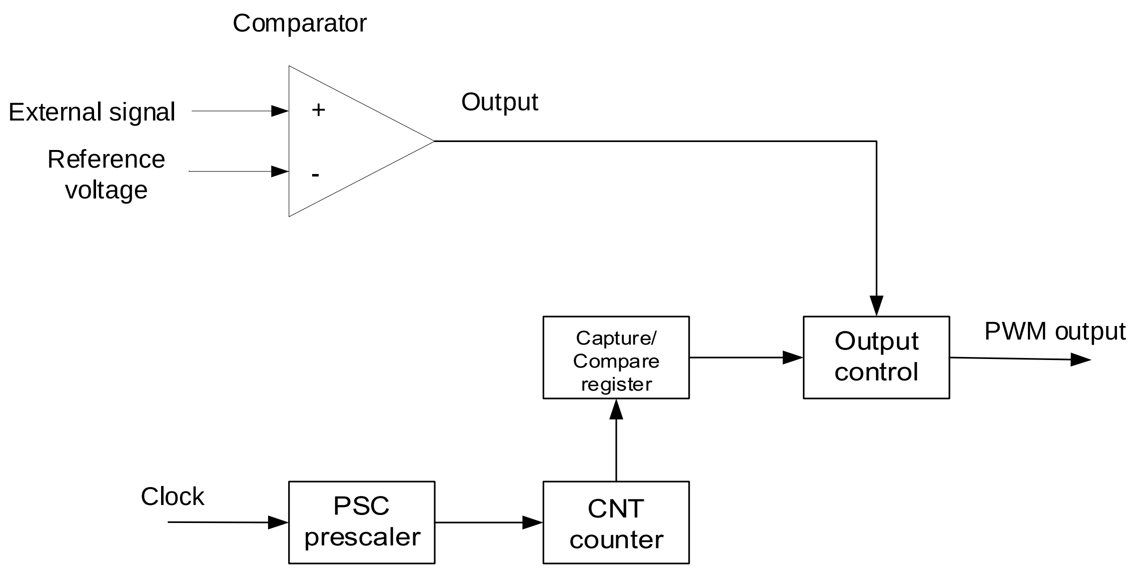

Figure 1 Structure Diagram

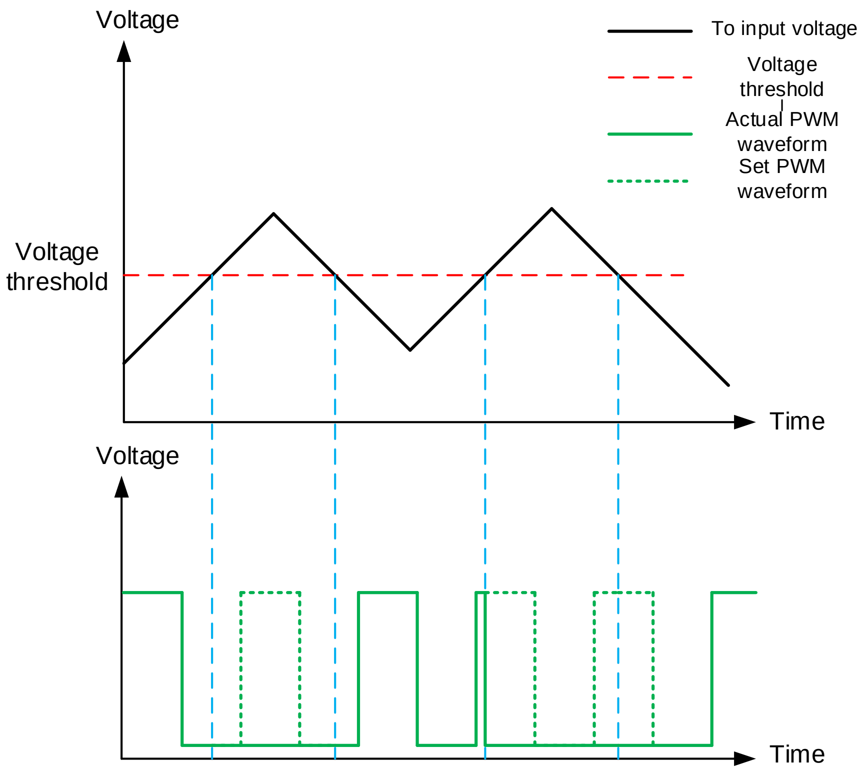

Figure 2 Input Voltage and PWM Output Effect

For example, when the output of COMP2 is redirected to TMR1 as the braking signal of TMR1, the configuration code of COMP is as follows:

```
void COMP_Init(void)
{
    COMP_Config_T compConfig;

    RCM_EnableAPB2PeriphClock(RCM_APB2_PERIPH_SYSCFG);

    COMP_ConfigStructInit(&compConfig);
    compConfig.invertingInput = COMP_INVERTING_INPUT_VREFINT;
    compConfig.mode = COMP_MODE_LOWSPEED;
    compConfig.output = COMP_OUTPUT_TIM1BKIN;
    compConfig.outputPol = COMP_OUTPUTPOL_NONINVERTED;
    COMP_Config(COMP_SELECT_COMP2, &compConfig);

    COMP_Enable(COMP_SELECT_COMP2);
}
```

The detailed source code can be seen in APM32F4xx_SDK/Example/COMP/COMP_PWMSignalControl.

3.2 Frequency measurement

The COMP module of APM32F411 can measure the frequency and pulse width of the part exceeding the voltage threshold. The voltage threshold is determined by the inverting input end. When measuring the frequency, the timer needs to be configured as the input capture mode, and when measuring the pulse width, the timer needs to be configured as the PWM input capture mode.

Capture when the timer is configured as input capture mode and the input channel is at the rising edge; configure the COMP input to redirect to the timer input capture channel, and the output polarity is not inverting.

When the non-inverting input is greater than the inverting input, COMP will output a high level to the timer input channel, the timer input channel will capture the rising edge signal, record the counter value to CC1 (channel 1 capture/compare register), and set the CC1IFLG bit of the timer STS (status register). If the timer interrupt is enabled, it will enter the timer interrupt service function; when the non-inverting input is greater than the inverting input for the second time, the timer input channel will capture the rising edge, record the counter value again to CC1 and set the CC1FLG bit of STS.

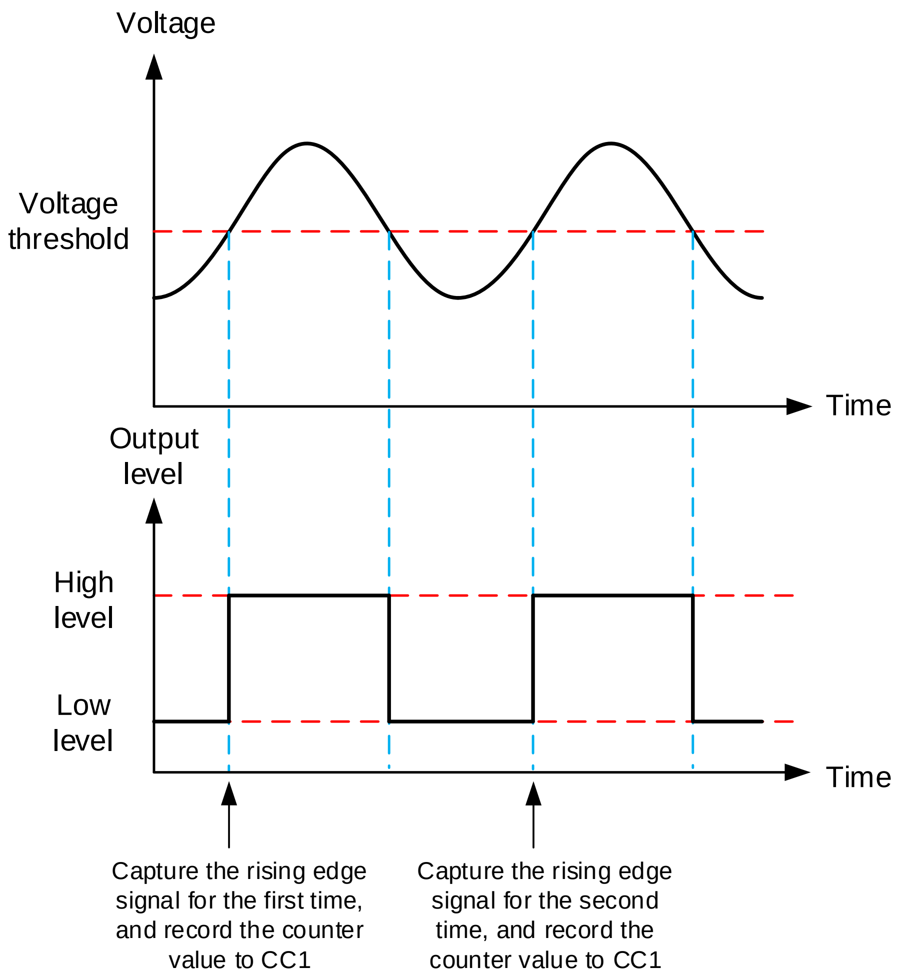

Figure 3 Schematic Diagram of COMP and Timer Frequency Measurement

Based on the recorded value [0] and value [1] of CC1, and the number of update events generated during the interval between two acquisitions, and in combination with the timer clock source frequency pclk1, the frequency of the COMP output waveform can be figured out:

frequency = pclk1 * 2 / (value[1] + update * 0xffff – value[0])

For example, when COMP2 and TMR3 (timer 3) are used for frequency measurement, the output of COMP2 needs to be redirected to the input capture channel 1 of TMR3. The configuration example code of COMP is as follows:

```
void COMP_Init(void)
{
    COMP_Config_T compConfig;

    /* Enable GPIOC clock */
    RCM_EnableAHB1PeriphClock(RCM_AHB1_PERIPH_GPIOC);
```

```
    /* Configure COMP2 */
    COMP_ConfigStructInit(&compConfig);
    compConfig.invertingInput = COMP_INVERTING_INPUT_VREFINT;
    compConfig.mode = COMP_MODE_HIGHSPEED;
    compConfig.output = COMP_OUTPUT_TIM3IC1;
    compConfig.outputPol = COMP_OUTPUTPOL_NONINVERTED;
    COMP_Config(COMP_SELECT_COMP2, &compConfig);

    /* Enable COMP2 */
    COMP_Enable(COMP_SELECT_COMP2);
}
```

TMR3 is configured as input capture mode, and the configuration example code is as follows:

```
void TMR_Init()
{
    TMR_ICConfig_T ICConfig;

    RCM_EnableAPB1PeriphClock(RCM_APB1_PERIPH_TMR3);

    /* TMR3 Input mode */
    ICConfig.channel = TMR_CHANNEL_1;
    ICConfig.polarity = TMR_IC_POLARITY_RISING;
    ICConfig.selection = TMR_IC_SELECTION_DIRECT_TI;
    ICConfig.prescaler = TMR_IC_PSC_1;
    ICConfig.filter = 0x00;
    TMR_ConfigPWM(TMR3, &ICConfig);

    /* Enable TMR3 Interrupt */
    TMR_Enable(TMR3);
     TMR_EnableInterrupt(TMR3, TMR_INT_UPDATE);
    TMR_EnableInterrupt(TMR3, TMR_INT_CC1);
    TMR_ClearIntFlag(TMR3, TMR_INT_CC1);
    NVIC_EnableIRQRequest(TMR3_IRQn, 0, 0);
}
```

3.3 Window comparator

The COMP1 and COMP2 of APM32F411 can be combined to form a window comparator. The enable switch of the window comparator is located in the status control register of COMP1. Enabling the window comparator function will short-circuit the non-inverting input ends of COMP1 and COMP2 and connect them to the external pin PC2.

When the input signal is transmitted from the non-inverting input end, it will be compared with the inverting inputs of COMP1 and COMP2. When the non-inverting input (INP) is greater than the inverting input of COMP1 (INM_{COMP1}), COMP1 will output a high level; when the non-inverting input is greater than the inverting input of COMP2, COMP2 will output a high level. From this, it can be seen that when the inverting inputs of COMP1 and COMP2 are not consistent, the window comparator will show three statuses.

Table 3 Window Comparator Output Status Table

COMP1 output status	COMP2 output status	Description
0	0	When INP＜INMCOMP1 and INP＜INMCOMP2
1	0	When INP＞INMCOMP1 and INP＜INMCOMP2
1	1	When INP＞INMCOMP1 and INP＞INMCOMP2

The stage of the input signal can be judged by the output status of COMP.

The configuration example code for enabling the window comparator function of COMP1 and COMP2 is as follows:

```
void COMP_Init(void)
{
    COMP_Config_T       compConfig;

    /* Enable COMP clock */
    RCM_EnableAPB2PeriphClock(RCM_APB2_PERIPH_SYSCFG);

    /* Configure COMP1 */
    COMP_ConfigStructInit(&compConfig);
    compConfig.invertingInput = COMP_INVERTING_INPUT_VREFINT;
    compConfig.mode = COMP_MODE_LOWSPEED;
    compConfig.output = COMP_OUTPUT_NONE;
    compConfig.outputPol = COMP_OUTPUTPOL_NONINVERTED;
    COMP_Config(COMP_SELECT_COMP1, &compConfig);

    compConfig.invertingInput = COMP_INVERTING_INPUT_1_2VREFINT;
    COMP_Config(COMP_SELECT_COMP2, &compConfig);

    COMP_EnableWindow();
```

```
    /* Enable COMP */
    COMP_Enable(COMP_SELECT_COMP1);
    COMP_Enable(COMP_SELECT_COMP2);
}
```

4 Revision history

Table 4 Document Revision History

Date	Version	Revision History
January 31, 2024	1.0	New edition

Statement

This manual is formulated and published by Zhuhai Geehy Semiconductor Co., Ltd. (hereinafter referred to as "Geehy"). The contents in this manual are protected by laws and regulations of trademark, copyright and software copyright. Geehy reserves the right to correct and modify this manual at any time. Please read this manual carefully before using the product. Once you use the product, it means that you (hereinafter referred to as the "users") have known and accepted all the contents of this manual. Users shall use the product in accordance with relevant laws and regulations and the requirements of this manual.

1. Ownership of rights

This manual can only be used in combination with chip products and software products of corresponding models provided by Geehy. Without the prior permission of Geehy, no unit or individual may copy, transcribe, modify, edit or disseminate all or part of the contents of this manual for any reason or in any form.

The "Geehy" or "Geehy" words or graphics with "®" or "TM" in this manual are trademarks of Geehy. Other product or service names displayed on Geehy products are the property of their respective owners.

2. No intellectual property license

Geehy owns all rights, ownership and intellectual property rights involved in this manual.

Geehy shall not be deemed to grant the license or right of any intellectual property to users explicitly or implicitly due to the sale and distribution of Geehy products and this manual.

If any third party’s products, services or intellectual property are involved in this manual, it shall not be deemed that Geehy authorizes users to use the aforesaid third party’s products, services or intellectual property, unless otherwise agreed in sales order or sales contract of Geehy.

3. Version update

Users can obtain the latest manual of the corresponding products when ordering Geehy products.

If the contents in this manual are inconsistent with Geehy products, the agreement in Geehy sales order or sales contract shall prevail.

4. Information reliability

The relevant data in this manual are obtained from batch test by Geehy Laboratory or cooperative third-party testing organization. However, clerical errors in correction or errors caused by differences in testing environment may occur inevitably. Therefore, users should understand that Geehy does not bear any responsibility for such errors that may occur in this manual. The relevant data in this manual are only used to guide users as performance parameter reference and do not constitute Geehy's guarantee for any product performance.

Users shall select appropriate Geehy products according to their own needs, and effectively verify and test the applicability of Geehy products to confirm that Geehy products meet their own needs, corresponding standards, safety or other reliability requirements. If loses are caused to users due to the user's failure to fully verify and test Geehy products, Geehy will not bear any responsibility.

5. Compliance requirements

Users shall abide by all applicable local laws and regulations when using this manual and the matching Geehy products. Users shall understand that the products may be restricted by the export, re-export or other laws of the countries of the product suppliers, Geehy, Geehy distributors and users. Users (on behalf of itself, subsidiaries and affiliated enterprises) shall agree and promise to abide by all applicable laws and regulations on the export and re-export of Geehy products and/or technologies and direct products.

6. Disclaimer

This manual is provided by Geehy "as is". To the extent permitted by applicable laws, Geehy does not provide any form of express or implied warranty, including without limitation the warranty of product merchantability and applicability of specific purposes.

Geehy will bear no responsibility for any disputes arising from the subsequent design and use of Geehy products by users.

7. Limitation of liability

In any case, unless required by applicable laws or agreed in writing, Geehy and/or any third party providing this manual "as is" shall not be liable for damages, including any general damages, special direct, indirect or collateral damages arising from the use or no use of the information in this manual (including without limitation data loss or inaccuracy, or losses suffered by users or third parties).

8. Scope of application

The information in this manual replaces the information provided in all previous versions of the manual.