**Timer Module Description Document**

**1. Introduction**  
Timer0/1 are used for countdown operations based on the system clock. When the counter value decreases to 0, it will automatically reload the initial value. If the Timer interrupt is enabled, a Timer interrupt will be generated.

**2. Main Features**

 24-bit timer, using the system clock as the clock source.

 Automatically reloads the initial value.

**3. Function Descriptions**

void Timer0Set(uint32\_t Time); Configure Timer0 and start counting. Time is the Timer0 interrupt period in microseconds (us), range: 1~174000.

void Timer1Set(uint32\_t Time); Configure Timer1 and start counting. Time is the Timer1 interrupt period in microseconds (us), range: 1~174000.

void Timer0HaltModeSet(bool Enable); When Enable=1, pause Timer0 counting; When Enable=0, resume Timer0 counting.

void Timer1HaltModeSet(bool Enable); When Enable=1, pause Timer1 counting; When Enable=0, resume Timer1 counting.

void Timer0IntClr(void); Clear Timer0 interrupt flag.

void Timer1IntClr(void); Clear Timer1 interrupt flag.

void Timer0Close(void); Disable Timer0.

void Timer1Close(void); Disable Timer1.

**4. Configuration Example**

Initialization process:

Timer1Set(1000); // Initialize Timer1

NVIC\_EnableIRQ(TMR1\_IRQn); // Enable Timer1 interrupt

In the Timer interrupt function, you need to clear the interrupt flag:

Timer1IntClr();