#### Normal mode

Normal mode timer counter is the simplest mode of operation, this time waveform generation mode control bit WGM0 [2: 0] = 0 Count maximum value TOP for MAX ( 0xFF ). In this mode, a counting mode for each clock count plus an increment, when the counter reaches TOP After the spill back BOTTOM Re-start accumulating. The count value TCNT0 The same count clock becomes zero set timer counter's overflow flag TOV0 . In this mode TOV0 The first sign is like 9 Count bit, but will only be set is not cleared. Overflow interrupt service routine will automatically clear TOV0 Logos, software can use it to improve the resolution of the timer counter. Normal mode is not to be considered a special case, a new count value can be written at any time. Set up OC0x Pin data direction register as an output a comparison signal to obtain an output OC0x Waveform. when COM0x = 1

When, flips compare match OC0x Signal, in this case the frequency waveform may be calculated using the following formula:

$$f_{oc0xnormal} = f_{sys} / ( 2 * N * 256)$$

among them, N It represents the prescale factor ( 1 , 8 , 64 , 256 or 1024 ).

Output Compare unit can be used to generate interrupts, but does not recommend the use of interrupts in the normal mode, it will take up too much CPU time.

#### CTC mode

Set up WGM0 [2: 0] = 2 When the timer counter 0 enter CTC Max mode, counting TOP for OCR0A . In this mode, a counting mode for each clock count plus an increment, when the value of the counter TCNT0 equal TOP When the counter is cleared. OCR0A It defines the maximum count, i.e., the resolution of the counter. This mode allows the user to easily control the frequency of the compare match output also simplifies the operation of the external event count. When the counter reaches a maximum count, an output compare match flag OCF0 Is set, an interrupt will be generated when the corresponding interrupt enable bit is set. Can be updated in the interrupt service routine OCR0A I.e., the maximum count register. In this mode

OCR0A Do not use double buffering, the counter prescaler to work under no or very low prescaler will be updated as close to the maximum value of the minimum time to be careful. If you write OCR0A The value is less than the time TCNT0 When the value of the counter will miss the compare match. Before a match occurs the next comparison, the first counter had counted to TOP And then from BOTTOM
To start counting OCR0A value. And normal mode, as the count value back BOTTOM The count clock in the set TOV0 Mark. Set up OC0x Pin data direction register as an output a comparison signal to obtain an output OC0x Waveform. when COM0x = 1
When, flips compare match OC0x Signal, in this case the frequency waveform may be calculated using the following formula:

$$f_{oc0xctc} = f_{sys} / ( 2 * N * (1 + OCR0x))$$

among them, N It represents the prescale factor ( 1 , 8 , 64 , 256 or 1024 ). As can be seen from the formula, when set OCR0A for 0x0 And when no prescaler, allowing for maximum frequency $f_{sys} / 2$ The output waveform.

### fast PWM mode

Set up WGM0 [2: 0] = 3 or 7 When the timer counter 0 Enter the fast PWM Mode, can be used to generate high frequency PWM Waveform, the counter maximum value TOP Respectively MAX ( 0xFF )or OCR0x . fast PWM Patterns and other PWM Except that it is a one-way mode operation. Counter from the minimum 0x00 To accumulate TOP Then came back BOTTOM Re-count. When the count value TCNT0 Arrivals OCR0x or BOTTOM , The output signal of the comparison OC0x It will be set or cleared, depending on the comparison output mode COM0x Setting, as detailed register description. Since the one-way operation, fast PWM Operating frequency of the phase correction mode is employed bi-directionally operable PWM Double mode. It makes the fast frequency PWM Mode is suitable for power regulation, rectification and DAC application. High-frequency signal can be reduced external components (capacitors, inductors) in size, thereby reducing system cost.

When the count value reaches the maximum value, the timer counter overflow flag TOV0 It will be set, and updates the value of the compare buffer