PWM with STM8S

Jan 24, 2018 — by Junaid in Electronics

PWM is a great way to control brightness of LEDs, speed of motors, etc. Most microcontrollers including STM8S has built in ability to generate PWM signals. In this article I share simple code to generate PWM signal in STM8S microcontroller. However, I don't write anything about PWM theory other than basic calculation to get required frequency and duty cycle.

Calculation

In STM8S, there are total 7 CAPCOM channels. 4 of them are associated with Timer 1 and 3 with Timer 2. Here we will use Channel 1 of Timer 2 to generate PWM. This channel's output will be in PD4 pin.

We will not use any prescaler on timer 2, so it will be run as same clock frequency as system which is 2MHz by default. We want to generate PWM with 2KHz frequency. Then we should use 2,000,000/2000 = 1000 as upper limit of Timer 2. Counter starts from 0 so we should use 1000 – 1 = 999 to fill in Auto Reload Register.

We set our upper limit as 1000, so setting duty cycle is easy. For 25% duty cycle, we can set 249 (1000 x (50/100) – 1) in CCR1 registers.

Active High and Low

We can have PWM signal of active low or high. We are connecting LED to pin PD4 which is high sink capability pin as per datasheet, so it will be good to sink current to that pin than sourcing. So we will connect LED's cathode to PD4 and anode to VCC through a series resister. Thus we have to use PWM in active low mode.

Program

```
#include <stm8s.h>
#include <util.h>

const uint16_t timer2_arr = 999; // PWM with 2kHz
frequency
const uint16_t timer2_ccr1 = 249; // 25% duty cycle.
int main() {
  disable_interrupts();

  // As per datasheet of stm8s103f3 PD4 pin is timer 2
channel 1.
  set_bit(PD_DDR, 4); // 0b00010000 PD4 is now output
  set_bit(PD_CR1, 4); // 0b00010000 PD4 is now pushpull

  TIM2_PSCR = 0x00; // Prescaler = 1

  // Fill 16 bit timer2_arr to two 8 bit registers.
  // MSB register to be filled first.
  TIM2_ARRH = timer2_arr >> 8;
  TIM2_ARRL = timer2_arr & 0x00FF;

  // Fill 16 bit timer2_ccr1 to two 8 bit registers.
  // MSB register to be filled first.
  TIM2_CCR1H = timer2_ccr1 >> 8;
  TIM2_CCR1L = timer2_ccr1 & 0x00FF;

  set_bit(TIM2_CCER1, TIM2_CCER1_CC1P); // channel 1
active low
  set_bit(TIM2_CCER1, TIM2_CCER1_CC1E); // Enable channel
1 output

  // PWM mode 1.
  set_bit(TIM2_CCMR1, 6); // Set output compare mode as 6 (0b110)
  set_bit(TIM2_CCMR1, 5); // So channel 1 will be active while counter
  clear_bit(TIM2_CCMR1, 4);  // is lower than compare value.

  set_bit(TIM2_CR1, TIM2_CR1_CEN); // Enable counter

  enable_interrupts();
  while(1) {
    // Do nothing
  }
}
```

`stm8s.h` contains just register address definitions and `util.h` contains bit operation macros like `set_bit()` and `clear_bit()`. You can see more about source at repo https://github.com/junaidpv/stm8

8-bit electronics embedded led linux microcontroller
pwm sdcc stm8 stm8s

Comments

One response to “PWM with STM8S”

Anil
September 3, 2021

Sir Please make a post on timer1 pwm using timer1 and its equation like calculate switch freq.,Duty cycle change etc.

Reply

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment *

Name *

Email *

Website

Post Comment

← Previous: Bootloader-less Programming Arduino Pro Mini

Next: Switched to Hugo →

Junaid's Blog

Designed with WordPress