ECE 271 – Microcomputer Architecture and Applications Lecture 18

Vince Weaver

http://web.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

2 April 2019

Announcements

- Read Chapter 15
- Chuck Peddle on Friday

Concurrency Example from Last Time

```
volatile int m,s;

void seconds_reset_button_interrupt(void) {
    s=0;
}

int main() {
    while(1) {
        s++;
        if (s==60) {
            m++;
            s=0;
        }
        sleep(1);
    }

}
```

- Can we interrupt at in-opportune time? What if the reset of s=0 happens in the *middle* of s++?
- Race condition. Won't happen often, but in some cases the reset could be missed.

```
// s++
    ldr r0,=s
        IRQ HAPPENS, mov r0,#0 ; str r0,=s
    add r0,r0,#1
    str r0,=s    ; this store un-does the reset in the interrupt
```

Advanced Control-Timers

- Note these are *not* the same as the SysTick timer
- Chapter 30 (p1007)
- TIM1/TIM8
- 16-bit auto-reload timers
- 16-bit prescalar
- Unrelated to the other TIM timers
- Each timer has 6 channels
- Key registers
  - Counter register `TIM1_CNT`

- Prescaler register `TIM1_PSC`
- Auto-reload register `TIM1_ARR`
- Repetition count register `TIM1_RCR`

- Upcounting mode
  - Counts from 0 to ARR then restarts at 0
  - The repetition counter says how many times to count before triggering an interrupt?
- Downcounting mode
  - Counts starting at ARR down to 0, reloads ARR
- Center-count mode
  - Counts up then down then up then down

Various Modes

- Can put counter in various modes. We'll use some of them next lab.
- Like 100 pages on this.
- We are setting PWM mode.
- Period set by ARR register, duty cycle by CCMR register
- Can do complicated things like assymmetric mode (with phase shift) or have different channels with different phases.
- Complicated way of setting phase

- Can generate both signal and inverse, and can add “dead time” to the transition
- Can xor together to generate more complex waveforms.

Where does the output go?

- This is configurable too.
- Can put to some but not all of the GPIO pins. Conveniently PE8 (green LED) gets the TIM1 output, but CH1N (so the inverted output)
- Have to set the AF (alternate function) register to pick. Somehow wasn't able to find where these are listed but it is in Appendix I of the textbook.

Servo Motors

- A motor with gearing
- Feedback. When you instruct it to g to 90 degrees, it will go there and using feedback (usually a potentiometer hooked to the last gear) will adjust to keep it held there even under load
- There is a processor on board that does this, plus maybe an H-bridge
- The typical interface for this is a 50Hz signal (20ms) and 1ms means 90 degrees to left, 1.5ms means center,

and 2ms means 90 degrees to right.

- While some watch duty cycle, others just look for the length of the high value.
- Why 50 Hz? Often used in radio-control apps and it was an easy frequency to convert to from the wireless signal.
- You might actually find 1ms/1.5ms/2ms while 1.5ms is exact, the other two might vary a bit and you might have to trial and error a bit to get it the full 90/-90 swing.

Looking ahead to Lab #9

Input Capture

- Input capture is measuring the time between two transitions on a signal
  - Rising/Rising or Falling/Falling or Rising/Falling or Falling/Rising
  - When transition happens, the current timer count (CNT) is saved to the CCR (Compare and capture register)
  - Also an interrupt (or other event such as DMA) can be triggered

- Time elapsed can be calculated by taking the previous timestamp and subtracting from the current one.