



MSP430 Timers

References:

MSP430 Data Sheet (MSP430f2013_annotated.pdf)

MSP430 Users Guide (msp430x2xx_Users_Guide_slau144b.pdf)

Timer_A

- Generates interrupts – has a dedicated interrupt vector
- 2 Timers available: Timer_A0 and Timer_A1
- Each are 16 bit...can therefore count from 0 to 0xFFFF (0 to 65535) - 2^{16}
- Many modes – very configurable. Some involve external hardware, others just used internally for timing.
- Most common usage – to provide an internal “heartbeat” – the base unit of timing for the USER programming (not the same as the timing provided by the crystal, which is for microprocessor timing)

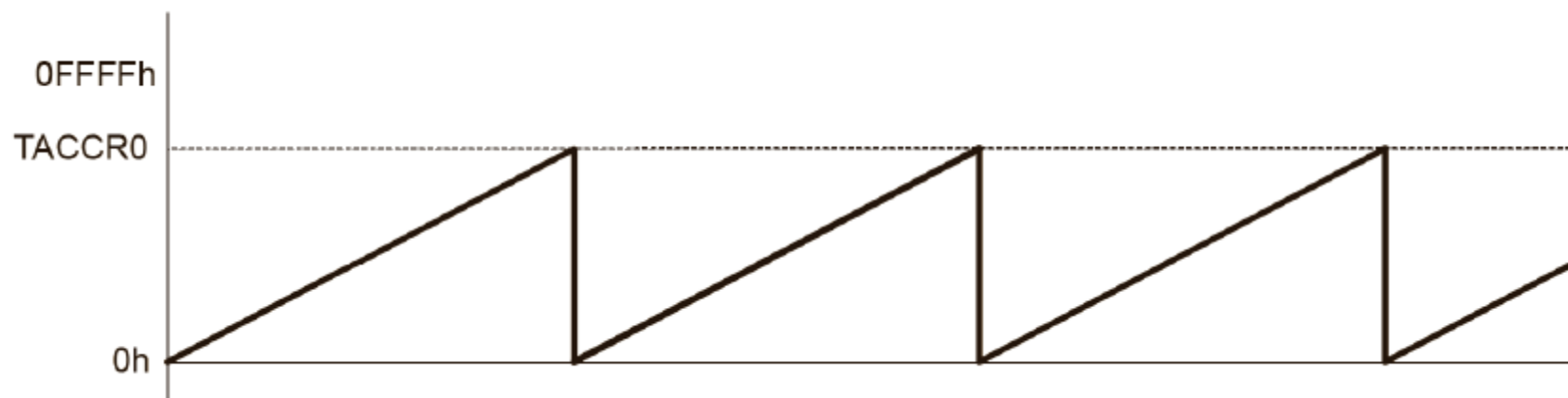
Timer Modes (4)

Four modes are available (configurable):

- Stop
- Up
- Continuous
- Up/Down

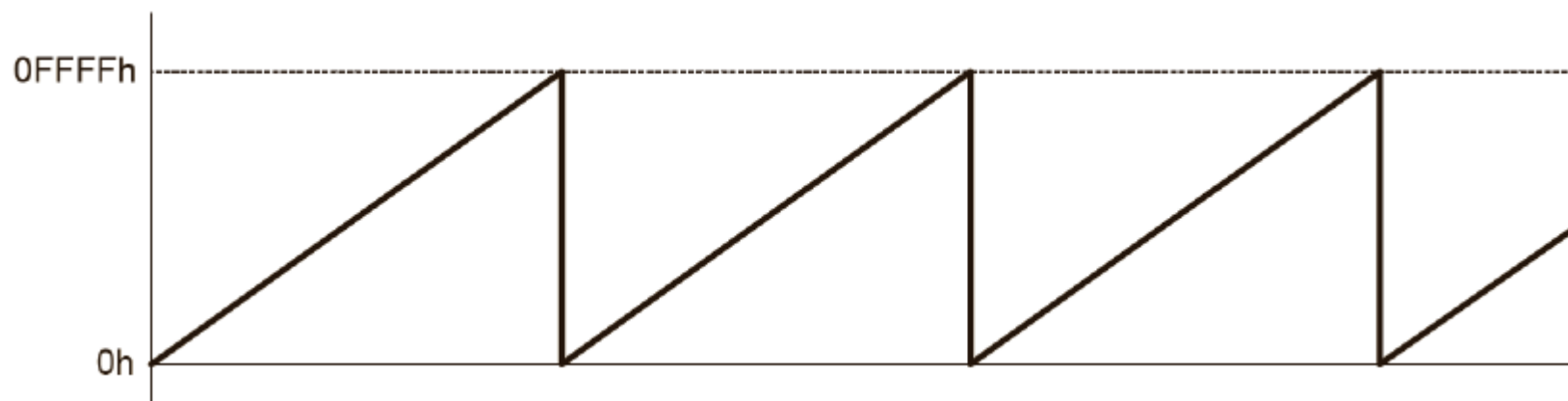
Up Mode

- Timer counts UP from zero to TACCRO
- **Interrupt occurs** when timer goes back to zero
- Useful for periods other than 0xFFFF
- Can have timing errors (will discuss later)



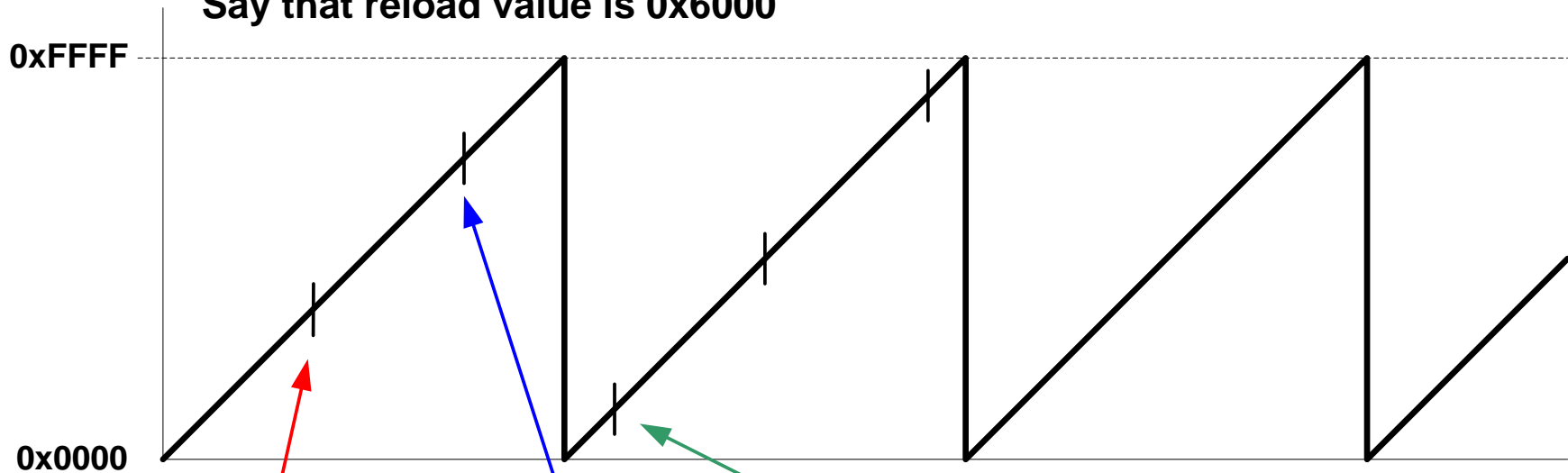
Continuous Mode

- Timer counts from 0 to 0xFFFF
- Fewer timing errors because timer never stops – keeps counting up until it reaches 0xFFFF and rolls over to 0 and keeps going.
- The ACTUAL VALUE of the timer does not matter – only the RELOAD VALUE matters – this controls the period of the interrupt.
- Interrupt DOES NOT OCCUR AT 0 OR 0xFFFF!
Occurs when timer reaches current TACCR0 value!



More on Continuous Mode

Continuous Timer Mode 2:
Say that reload value is 0x6000



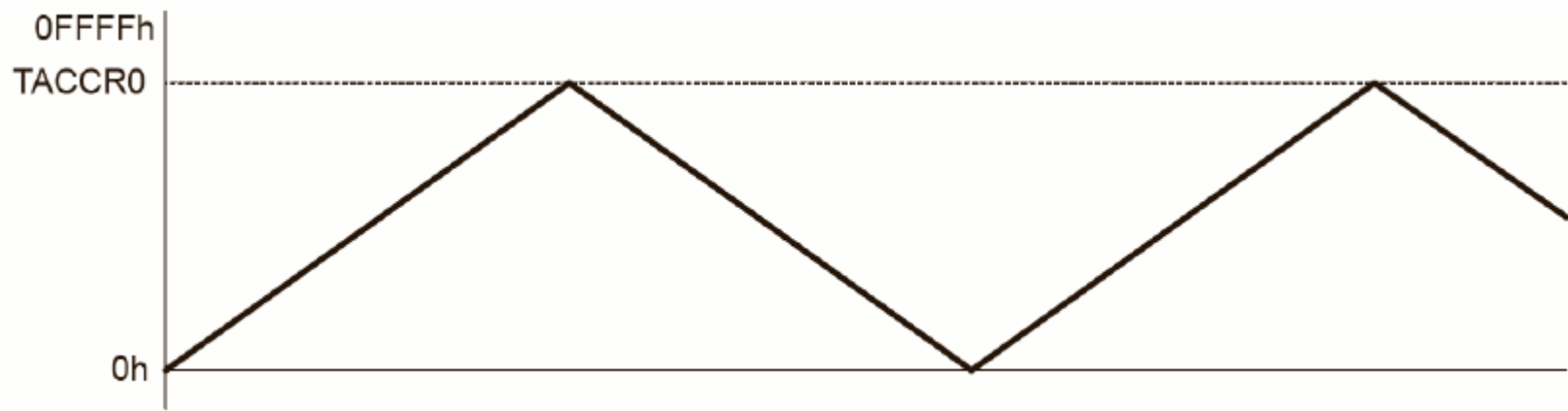
1) Timer reaches 0x6000, fires interrupt, inside ISR we ADD the period to the CURRENT VALUE of the timer:
 $TACCR0 += 0x6000$
 Therefore $TACCR0 = 0xC000$, and next interrupt will fire when timer reaches this value.

2) Timer now reaches 0xC000, fires interrupt, inside interrupt
 $TACCR0 += 0x6000$
 New $TACCR0 = 0x2000$ (rolls over 0xFFFF)

3) Timer rolls over 0xFFFF and now reaches 0x2000. Cycle repeats.

Up/Down Mode

- Timer counts from 0 to TACCRO, then back down to 0
- Used when timer period must be different from 0xFFFF and when pulse needs to be symmetric
- Good for driving motors (ON pulse to control speed)



???????

- Which mode should you use for Lab 2?

Configuring the Timer

- From the 430 peripheral map on Page 19 (MSP430f2013_annotated.pdf)
- TACTL – Timer A Control Register
 - This register used to configure how the timer runs
- TACCTL0 – Capture/Compare Control Register
 - For enabling and disabling TimerA0 interrupt
- TACCR0 – Capture/Compare Register
 - This registers holds the value YOU define to configure the timing
 - Sometimes called the “period” register (Hint - remember “period” as the denominator for the first LED flash lab? It is the ON + OFF time.)
 - Writing to this register actually STARTS the timer counting!
- You can confirm these #defines in the msp header file starting line 428. Some provided code uses alternate #defines starting line 447. Same function, just shorter #define name. Don't get confused.

To Make Timer INT Work

1. Must configure Timer MODE
 - TACTL – Timer A Control Register
 - Users Guide 8-7, 8-20
2. Set the interrupt period
 - TACCR0 – Capture/Compare Register
 - Users Guide 8-5
3. Enable Timer A0
 - TACCTL0 – Capture/Compare Control Register
 - Users Guide 8-22
4. Enable ALL interrupts (GIE in status register)
 - `__enable_interrupt()`

Lab 2 Flashing LED

- There is code on the internet to complete this lab. Be warned...
- 1) It is not well documented
- 2) It uses shorthand for registers
- 3) It assumes there is no other software that needs to run, and immediately puts the processor in low power mode.
-You can use it as an educational resource, but don't turn that in and expect to get credit.