

CT Übungsaufgaben

Timer/Counter, PWM

Aufgabe 1

- a) Erklären Sie in Stichworten die einzelnen Funktionseinheiten eines Timers anhand der Tabelle.

Register	Inhalt	Funktion(en)
Prescaler	Divisor für Eingangssignal	Es wird nur jeder n-te Wert gezählt.
Counter	Aktueller Timerwert	Der Wert wird mit jedem n-ten Tick um eins erhöht oder erniedrigt
Reload	Wert für Timer-überlauf	<u>Upcounter</u> : Timer zählt bis zu diesem Wert, dann Überlauf <u>Downcounter</u> : Startwert für Timer
Capture/Compare	Vergleichswert	<u>Capture</u> : Bei einem Event wird der Wert des Counters hier gespeichert <u>Compare</u> : Sobald der Wert vom Counter erreicht wird, wird ein Event ausgelöst.

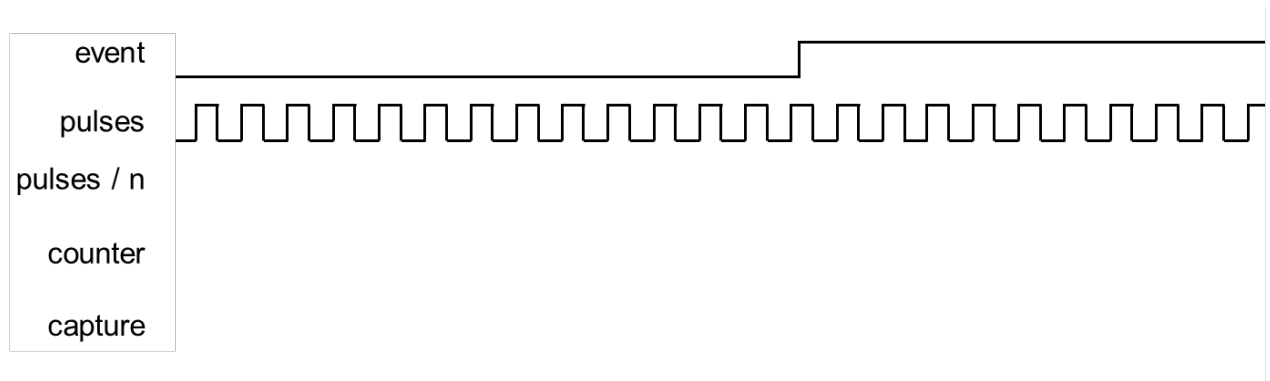
- b) Erläutern Sie die Funktion Capture.

Bei einem Event wird der Inhalt des Counter Registers in das Capture / Compare Register kopiert. Der Counter läuft weiter.

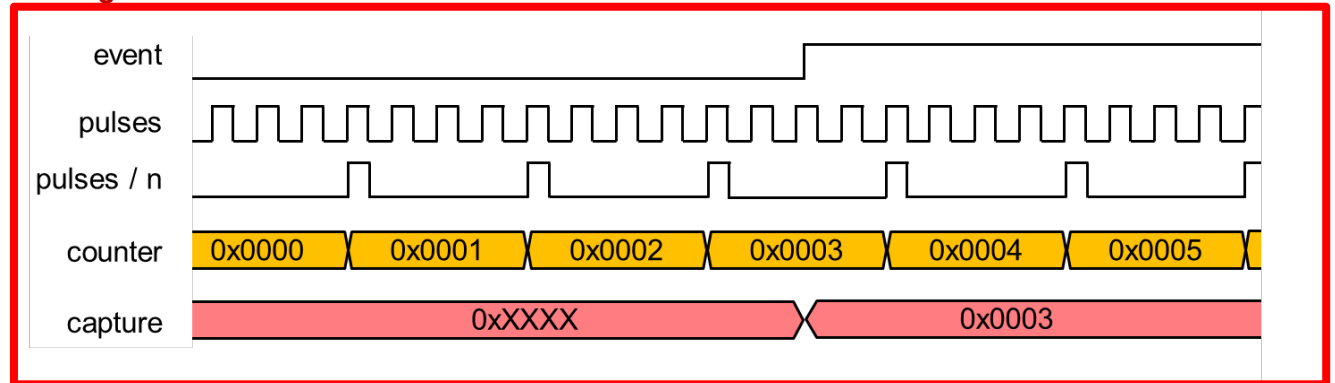
- c) Erläutern Sie die Funktion Compare.

Sobald der Counter den Wert des Capture / Compare Register erreicht hat, wird ein Event oder ein Interrupt ausgelöst. Der Counter läuft weiter.

- d) Ergänzen Sie das gegebene Timing Diagram. Die Funktion Capture wird bei steigender Flanke des Signals „event“ ausgelöst. Der Timer ist als Upcounter konfiguriert, und der Prescaler ist auf 4 (im Register steht 0x03) eingestellt. Die Startwerte entsprechen sonst dem Zustand nach einem Reset.



Lösung:



Aufgabe 2

Es soll Timer 3 des STM32F429 konfiguriert werden. Verwenden Sie das Reference Manual (zu finden in OLAT) zur Lösung der Aufgabe. Geben Sie die entsprechende Codezeilen in C an.

- a) Als Source soll die interne Clock CK_INT mit 84 MHz verwendet werden. Setzen Sie die Bits im entsprechenden Register auf die notwendigen Werte. Hinweis: andere Bits des Registers sollen nicht verändert werden.

```
TIM3_SMCR &= 0xFFF8;    // TIM3_SMCR [2:0]  0b000
```

- b) Der Timer soll als Upcounter konfiguriert werden. Hinweis: andere Bits des Registers sollen nicht verändert werden.

```
TIM3_CR1 &= 0xFF8F
```

- c) Die Zeit für den Timerüberlauf soll 200 ms betragen. Welche Werte müssen Sie in die Register PSC und ARR schreiben (Angabe hexadezimaler Werte)? Hinweis: Es sind verschiedene richtige Lösungen möglich.

```
TIM3_PSC = 0x20CF    // (8400 - 1) entspricht 10 kHz  
TIM3_ARR = 0x07CF    // (2000 - 1) entspricht 200 ms
```

```
TIM3_PSC = 0x0347    // (840 - 1) entspricht 100 kHz  
TIM3_ARR = 0x4E1F    // (20000 - 1) entspricht 200 ms
```

```
TIM3_PSC = 0x01A3    // (420 - 1) entspricht 200 kHz  
TIM3_ARR = 0x9C3F    // (40000 - 1) entspricht 200 ms
```

Aufgabe 3

Timer 4 des STM32F429 ist bereits als Upcounter konfiguriert und läuft. Das Reload Register enthält folgenden Wert:

TIM4_ARR = 0x9C3F

- a) Geben Sie den Wert für das CCR-Register an, damit ein Duty Cycle von 25% mittels PWM Mode 1 erzeugt wird.

TIM4_CCR1 = 0x2710 // 0x9C3F=(40000-1) -> 0x2710=(10000)

TIMx_CNT zählt von 0 ...39999 = 40000 Ticks
-> Duty Cycle 25% entspricht 10000 Ticks

PWM Mode 1 (Upcounting): OC_REFx='1' as long as TIMx_CNT<TIMx_CCR
otherwise OC_REFx= '0'

-> 0...9999 = 10000 Ticks -> TIMx_CNT<TIMx_CCR -> OC_REFx= '1'
-> 10000...39999 = 30000 Ticks -> TIMx_CNT>=TIMx_CCR -> OC_REFx= '0'

- b) Nun ist Timer 4 als Downcounter im PWM Mode 2 statt als Upcounter im PWM Mode 1 konfiguriert. Was müssen Sie ändern, um ein identisches elektrisches Signal zu erhalten (Werte)?

TIM4_CCR1 anpassen

TIM4_CCR1 = 0x752F // entspricht (30000-1)

PWM Mode 2 (Downcounting): OC_REFx='1' as long as TIMx_CNT>TIMx_CCR
otherwise OC_REFx= '0'

-> 39999...30000 = 10000 Ticks -> TIMx_CNT>TIMx_CCR -> OC_REFx= '1'
-> 29999...0 = 30000 Ticks -> TIMx_CNT<=TIMx_CCR -> OC_REFx= '0'

Aufgabe 4

Der Timer ist als Upcounter konfiguriert. Bestimmen Sie das generierte PWM-Signal am Ausgang (Zahlen + Skizze). Die Source liefert ein Signal der Frequenz 0,5 MHz.

Die relevanten Konfigurationsregister sind wie folgt initialisiert:

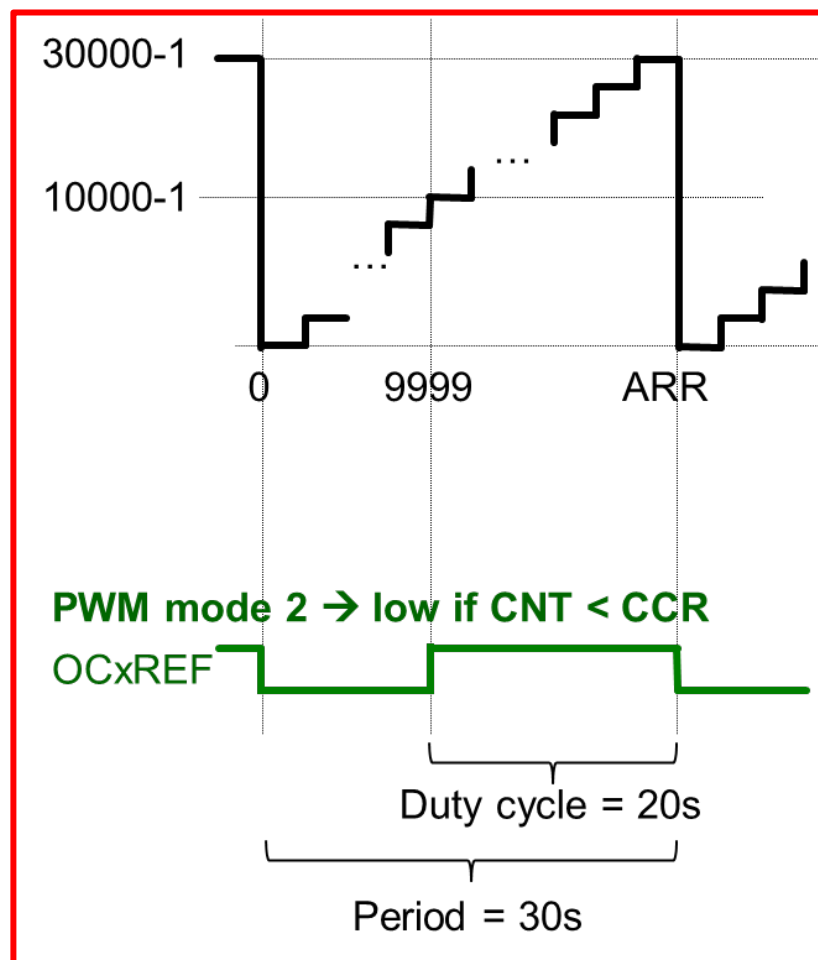
```
TIM3_PSC      = 0x01F3  
TIM3_ARR      = 0x752F  
TIM3_CCR1     = 0x2710  
TIM3_CCMR1    = 0x0070
```

Welches Signal wird erzeugt? Zeichnen Sie das Signal und geben Sie die Werte für Periode und Duty Cycle an (Zeitangaben).

Lösung:

Gegebene Register entsprechen:

```
TIM3_PSC      = 0x01F3    // (500-1) -> 1 kHz  
TIM3_ARR      = 0x752F    // (30000-1) -> Periode 30 Sekunden  
TIM3_CCR1     = 0x2710    // (10000) -> 10 Sekunden  
TIM3_CCMR1    = 0x0070    // PWM Mode 2
```



Periode: 30 Sekunden, Duty Cycle: (30 Sekunden-10Sekunden) = 20 Sekunden