

Incremental Interface Mode

des Mikrocontrollers C167 von Infineon

Bernhard Putz

1. Allgemeines

Ein Timer ist ein mit Befehlen initialisierter Hardware-Zähler, der programmunabhängig läuft. Er wird vor allem für Aufgaben eingesetzt, die sich durch Software nur sehr aufwendig programmieren lassen, z.B.:

- PWM-Signal zur Ansteuerung von Motoren
- Ereigniszähler für Signalfanken
- periodische Interrupts als Zeitgeber
- Frequenzgenerator und -messer

Als Taktquelle dient der interne Prozessortakt oder ein externes Signal. Der Gate-Eingang steuert den Takt des Timers. Bei einem Nulldurchgang des Zählers wird ein Interrupt ausgelöst und kann ein Signal nach außen abgegeben werden. Hilfsregister dienen zum Nachladen, Vergleichen und Auslesen des Zählerstandes und mit Hilfe der Steuerregister können verschiedene Betriebsarten eingestellt werden, die wichtigsten sind:

- **Timer-Mode:** programmierbarer Frequenzteiler mit internem Takt,
- **Gated Timer-Mode:** Frequenzteiler mit Gate-Steuerung,
- **Count-Mode:** Ereigniszähler für externe Signalfanken,
- **Reload-Mode:** Nachladbare Frequenzteiler,
- **Capture-Mode:** Auslesen des Zählerstandes bei einem bestimmten Ereignis,
- **Compare-Mode:** Ausgabe eines Signals bei einem bestimmten Zählerstand,

- **Incremental Interface-Mode:** Erfassen phasenverschobener Positionssignale.

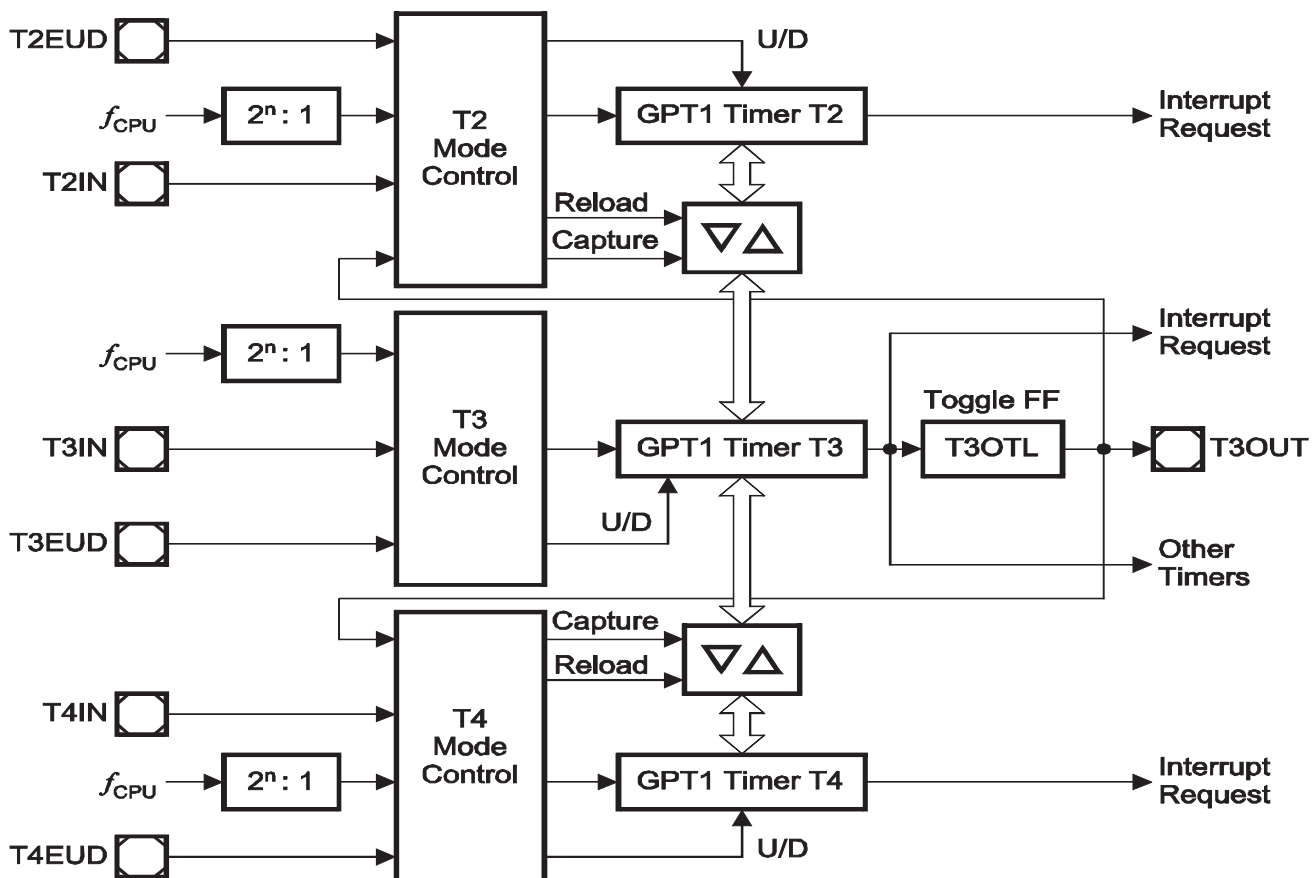
Es ist allerdings zu beachten, dass nicht jede Betriebsart von jedem Timer unterstützt wird.

2. Mehrzweck-Timereinheiten

Die Mehrzweck-Timereinheiten (*General Purpose Timer*) GPT1 (**Abbildung 1**) und GPT2 des Mikrocontrollers C167 arbeiten völlig unabhängig voneinander und setzen sich aus 3 (GPT1) bzw. 2 (GPT2) 16-Bit Einzelzählern zusammen, wobei jeweils einer der Hauptzähler ist (*Core Timer*). Jeder dieser 5 Timer kann entweder für sich alleine oder in Kombination (z.B. kaskadiert) betrieben werden. Die Auflösung reicht je nach Einstellung und Timer von 160 ns bis ca. 41 μ s, die maximale Periode liegt damit zwischen 10 ms und 2,68 s (alle Werte bei 25 MHz Taktfrequenz).

Über die externen Steuereingänge TxEUD kann die Zählrichtung der Timer verändert werden. Welcher Pegel hier welche Zählrichtung einstellt, ist zudem wählbar, so dass eine Anpassung an die verschiedensten Einsatzbedingungen möglich ist. Die interne Hardware der Timer ist so aufgebaut, dass bei jedem Zählimpuls, sei er vom internen Takt abgeleitet oder durch eine Flanke am externen Zähleringang TxIN erzeugt, gleichzeitig der Zustand der Zählrichtungseinstellung überprüft und der Timer dann entsprechend inkrementiert oder dekrementiert wird. Insbesondere im Counter-Betrieb mit externer Richtungseinstellung werden die beiden Eingänge TxIN und TxEUD zum selben Zeitpunkt abgefragt. Liegt kein Zählimpuls vor, hat der Zustand der Richtungseinstellung keine Bedeutung.

Abbildung 1: General Purpose Timer Unit GPT1



3. Erfassen phasenverschobener Positionssignale

Diese gleichzeitige Abfrage der Richtungseingänge lässt sich gut ausnutzen, um phasenverschobene Signale, wie sie durch Wegstrecken-Sensoren oder ähnliches erzeugt werden, zu verarbeiten.

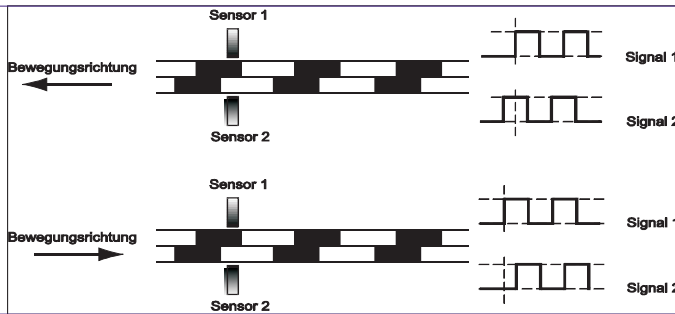


Abbildung 2: Generierung phasenverschobener Positionssignale

Abbildung 2 zeigt die Generierung solcher Signale über z.B. eine spezielle Skala mit gegeneinander verschobenen Schwarz-Weiß-Balken, die mit optischen Sensoren abgetastet wird. Durch die Phasenverschiebung steckt in den beiden Signalen der Sensoren nicht nur die Zählinformation (also z.B. das Wegstrecken-Inkrement), sondern auch eine Richtungsinformation. Kommt die steigende Flanke von Signal 1 vor der steigenden Flanke von Signal 2, bedeutet dies z.B. die Richtung Rechts. Dreht sich die Richtung der Bewegung um, so kommt diesmal die steigende Flanke von 1 nach der steigenden Flanke von Signal 2. Wie kann nun diese Information mit einem Timer verarbeitet werden?

Variante 1 (Count-Mode)

Die beiden Signale sind ja auf den ersten Blick nicht trennbar in ein Zählsignal, das an den Count-Eingang TxIN anzuschließen wäre, und ein Richtungssignal, das den Eingang TxEUD steuert. Es ist jedoch einfach und ohne zusätzliche externe Gatter möglich, diese beiden Signale korrekt zu verarbeiten, allerdings mit der Einschränkung der geringeren Frequenz. Die Lösung des Problems liegt in der oben beschriebenen Tatsache, dass die Richtungsinformation am Pin TxEUD nur dann benötigt wird, wenn ein Zählimpuls anliegt. Schließt man nun Signal 1 an den Count-Eingang TxIN an und programmiert den Timer im Zählerbetrieb so, dass er z.B. die positive Flanke des Signals zählt, so kann man aus **Abbildung 3** sehen, dass der momentane Zustand des Signals 2 zum Zeitpunkt der positiven Flanke von Signal 1 die Richtungsinformation wiedergibt.

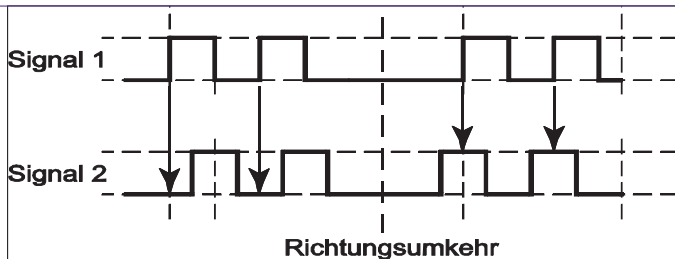


Abbildung 3: Bewertung des Richtungseingangs nur bei positiver Zählflanke

Man kann also Signal 2 direkt an den Steuereingang TxEUD legen und erhält so die gewünschte Funktion ohne zusätzlichen Aufwand. **Abbildung 4** zeigt den Verlauf des Zählerstandes für verschiedene Bewegungsrichtungen.

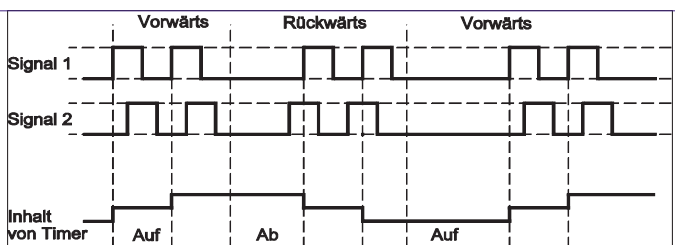


Abbildung 4: Zählerstand bei verschiedenen Bewegungsrichtungen

Allerdings arbeitet diese Lösung nur mit einer geringeren Frequenz im Vergleich zu speziellen ASIC-Bausteinen für diesen Zweck (Quadratur-Zähler). Bei dieser Methode wird ja nur eine Flanke von einem der beiden Signale als Zähl-Inkrement benutzt, während es mit zusätzlichem Logik-Aufwand möglich ist, jede Flanke beider Signale auszuwerten und so eine zwei- oder vierfach höhere Zählfrequenz zu erreichen (= höhere Genauigkeit).

Variante 2 (Incremental Interface-Mode)

Der Mehrzweck-Timer GPT1 stellt diesen zusätzlichen Logik-Aufwand in Form des "Incremental Interface-Mode" zur Verfügung. Dabei kann der Eingang TxEUD nicht nur für die Richtungsinformation, sondern auch zur Flanken-zählung verwendet werden.

Weiters besteht nun die Möglichkeit, sowohl steigende als auch fallende Flanken mit entsprechender Richtung zu zählen. Somit erreicht man eine bis zu vierfach höhere Zählfrequenz gegenüber Variante 1. Die anschließenden Erklärungen beziehen sich immer auf Timer T2, sie gelten aber genauso für die beiden anderen Timer von GPT1 (Timer 1 und 3).

Folgende Register gehören zu T2:

T2: Timer Datenregister: Dieses Register enthält den aktuellen Zählerwert. Je nach Drehrichtung des Signalgebers wird der Wert erhöht oder verringert.

T2IC: Interrupt Control Register: Beim Überlauf des Timerregisters T2 von FFFFH auf 0000H (Aufwärtszählen) bzw. 0000H auf FFFFH (Abwärtszählen) wird ein Interrupt ausgelöst und T2IR gesetzt. Sind gleichzeitig das Einzel-Freigabebit T2IE = 1 und das Gesamt-Freigabebit IEN=1, so liegt die Anforderung an der Interrupt-Steuerung an und es wird die passende Interrupt Service Routine aufgerufen. Die Bitfelder ILVL und GLVL legen die Interrupt-Prioritätsebene fest.

T2IC															
Timer 2 Intr. Ctrl. Reg. SFR (FF60 _H /B0 _H)															
Reset value: -- 00 _H															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								T2IR	T2IE	ILVL				GLVL	
								rwh	rw	rw				rw	

Die neuen 16-Bit-Mikrocontroller der Serie XC16x von Infineon [2] bieten eine Erweiterung, die bei jeder Flanke aber auch bei einem Richtungswechsel einen Interrupt auslöst. Das Control Register enthält drei zusätzliche Flags für die aktuelle Richtung, Richtungswechsel- und Flankenerkennung.

T2CON: Control Register: Zum Setzen des Modus und Konfiguration des Timers.

T2CON															
Timer 2 Control Register SFR (FF40 _H /A0 _H)															
Reset value: 0000 _H															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						T2UDE	T2UD	T2R	T2M				T2I		
						rw	rw	rw	rw				rw		

Tabelle 1 und 2 erklären die Bedeutung der einzelnen Bitfelder und Bits.

Bit	Funktion
T2I	Timer 2 Input Selection: abhängig von Modus T2M (siehe Tabelle 4)
T2M	Timer 2 Mode Control: 000: Timer Mode 001: Counter Mode 010: Gated Timer mit Gate active low 011: Gated Timer mit Gate active high 100: Reload Mode 101: Capture Mode 110: Incremental Interface Mode 111: Reserviert. Nicht verwenden.
T2R	Timer 2 Run Bit: 0: Timer 2 deaktiviert 1: Timer 2 aktiviert
T2UD	Timer 2 Up/Down Control: Siehe Tabelle 2
T2UDE	Timer 2 External Up/Down Enable: Siehe Tabelle 2

Tabelle 1: Timer 2 Control Register Bits

Pin T2EUD	Bit T2UDE	Bit T2UD	Zählrichtung
X	0	0	Aufwärts
X	0	1	Abwärts
0	1	0	Aufwärts
1	1	0	Abwärts
0	1	1	Abwärts
1	1	1	Aufwärts

Tabelle 2: Timer 2 Zählrichtungskontrolle

Um den *Incremental Interface-Mode* ausführen zu können, müssen folgende Bedingungen erfüllt sein:

- Bitfeld **T2M** muss auf **#110b** gesetzt sein.
- Beide Pins (**T2IN** und **T2EUD**) müssen auf Input geschaltet sein, d.h. die entsprechenden *Direction Control Bits* sind **#0b**.
- Bit **T3UDE** muss **#1b** sein, damit die automatische Richtungskontrolle aktiviert wird. **Tabelle 3** zeigt die daraus resultierenden Richtungskombinationen.

Level des entsprechend anderen Eingangs	T2IN Eingangsflanke		T2EUD Eingangsflanke	
	Steigend	Fallend	Steigend	Fallend
High	Abwärts	Aufwärts	Aufwärts	Abwärts
Low	Aufwärts	Abwärts	Abwärts	Aufwärts

Tabelle 3: Automatische Zählrichtungen bei Incremental Interface-Mode

Wie oben bereits erwähnt, kann in diesem Modus die Zählfrequenz variiert werden. **Tabelle 4** zeigt die möglichen Einstellungen und **Abbildung 5 bzw. 6** passende Beispiele.

T2I	Funktion
000	Keine. Zähler T2 stoppt.
001	Bei jeder Flanke an T2IN zählen.
010	Bei jeder Flanke an T2EUD zählen.
011	Bei jeder Flanke an T2IN und T2EUD zählen.
1xx	Reserviert. Nicht verwenden.

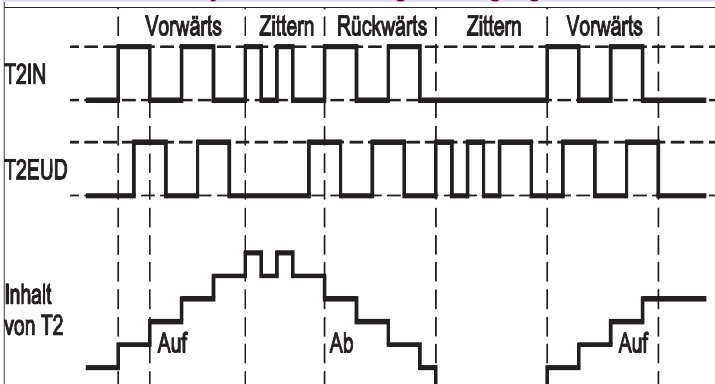
Tabelle 4: Timer 2 Input Selection T2I bei Incremental Interface-Mode

4. Beispiel

Der Timer **T2** soll benutzt werden, um die zwei Ausgangssignale eines magnetischen Impulsgebers zu dekodieren. Diese sind an den I/O-Pins **T2IN** und **T2EUD** des C167 angeschlossen. Damit ergibt sich folgende Einstellung für **T2CON** (#0173h):

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
.	1	0	1	1	1	0	0	1	1
-	-	-	-	-	-	-	rw	rw	rw	rw	rw	rw	rw	rw	rw

Abbildung 5: Timer zählt bei jeder Veränderung von Eingang T3IN



Programm Code

Im folgenden Programmteil werden die Drehzahlen zweier Antriebsmotoren eines Roboters über die Ausgänge magnetischer Signalgeber gemessen. Dazu wird Timer 2 und 4 der Einheit **GPT1** verwendet. Die Einstellungen zu Timer 4 erfolgen analog zu Timer 2 (siehe oben). Die Funktion *"increm_initialize"* bereitet die notwendigen Register vor, *"increm_update"* aktualisiert die Geschwindigkeiten **"v_outL"** und **"v_outR"** des linken bzw. rechten Rades. Die Funktion muss regelmäßig aufgerufen werden (in diesem Fall jede Millisekunde) um einen Timerüberlauf zu verhindern. Aus Performance-Gründen wurde der Programm-Kode in Assembler geschrieben: **Datei "increm_f.A66" siehe Webversion dieses Artikels.**

5. Anwendung und Tests

Die Funktionalität der Enkoder-Einheit wurde anhand einer Motorsteuerung getestet. Im vorliegenden Testfall wurde die Drehzahl zweier über ein PWM-Signal angesteuerte DC-Motoren geregelt. Als Referenzwert dienen die Impulse der direkt an den DC-Motoren angebrachten digitalen 2-Kanal Enkoder. Dieser Enkoder lieferte pro Umdrehung 512 Impulse auf 2 Kanälen. Die im C167er implementierte Enkoder-Einheit ist in der Lage diesen Wert intern zu vervierfachen, wodurch sich 2048 Impulsen pro Umdrehung ergaben. Als Testboard wurde das von **Gregor Novak** und **Christian Perschl** entwickelte ub167 verwendet. Das ub167 ist eine komplette, kostengünstige Plattform für einfache und komplexe Aufgaben der Steuerungs-, Regelungs- und Messtechnik. Zur Plattform gehört neben der universellen Hardware auch das Softwarekonzept: vorkonfigurierte Musterprojekte, Softwaremodule in C-Source und das Tool Minimon für Download und Test. Das Gehirn von ub167 ist ein 16-Bit Mikrocontroller der Firma Infineon, der C167CR, welcher mit 20 MHz getaktet wird (10 MIPS Rechenleistung). Er enthält einen 10 Bit A/D Wandler mit 16 Kanälen und 9,7µs Wandlungszeit, eine leistungsfähige CAN Bus Schnittstelle, eine asynchrone und eine schnelle synchrone serielle Schnittstelle, eine 4-Kanal PWM Einheit, mehr als 70 freikonfigurierbare digitale Ein-/Ausgänge, 9 Timer und vieles mehr. All diese Peripherie ist in Silicon und kann ohne zusätzliche Hardware verwendet werden.

Nähere Information und die Bezugsquelle des ub167 findet sich unter www.bluetechneix.at/ub167.

6. Literatur

- [1] C167CR Derivatives User's Manual v3.1, Infineon Technologies AG, 2000
- [2] MC-Tools 17 - Arbeiten mit C166-Controllern, Karl-Heinz Mattheis / Steffen Storandt, 1995
- [3] Mikrocomputertechnik mit dem Controller C167, Günter Schmitt, 2000
- [4] Einsatz von Mikrocontrollern in der Robotertechnik, Christian Perschl, 2000
- [5] XC161 User's Manual v2.0 – Peripheral Units, Infineon Technologies AG, 2003

Abbildung 6: Timer zählt bei jeder Veränderung von Eingang T3IN und T3EUD

