

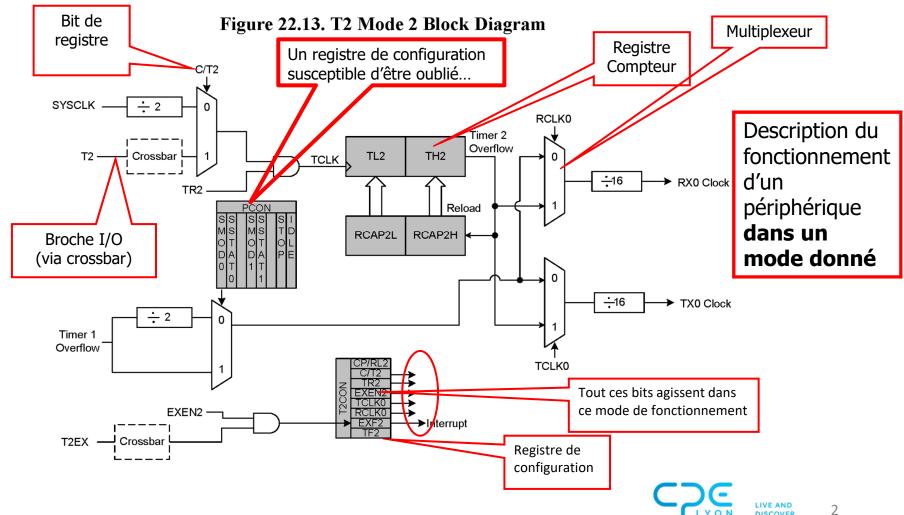
Bases des systèmes embarqués

A03 – Timers – 8051F020

Version 2022 - 21/11/2022 18:56







Descriptif d'un registre dans la doc 8051F020 (bis)

Accessibilité des bits en lecture ou en écriture

Figure 22.5. TCON: Timer Control Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Andress:
						(t	oit addressal	ble) 0 x 88
Bit7:	TF1: Timer 1 Set by hardwa cleared when 0: No Timer 1	re when Tir the CPU vec overflow d	ner 1 overflo ctors to the T etected.	-		•	are but is	automatically
Bit6:	1: Timer 1 has TR1: Timer 1 0: Timer 1 dis 1: Timer 1 ena	Run Contro abled.				Indique s registre adressable bit	est	Contenu du registre après Reset

Bit5: TF0: Timer 0 Overflow Flag.

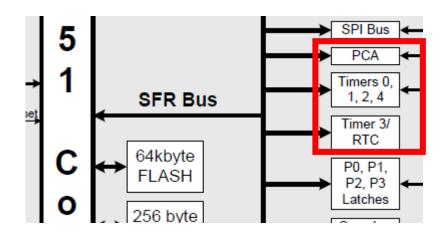
Set by hardware when Timer 0 overflows. This flag can be cleared by software but is automatically cleared when the CPU vectors to the Timer 0 interrupt service routine.

0: No Timer 0 overflow detected.

1: Timer 0 has overflowed.

Les Timers dans le 8051F020

- Timer0 et 1: présents sur tous les 8051. Le mode 13 bits n'est utilisé que pour assurer la compatibilité avec de vieux codes.
- Timer2 et Timer4: timers 16 bits polyvalents et simples d'emploi
- Timer 3 monomode
- PCA (Program counter Array) multimodes utilisations plus complexes



Fonction essentielle des Timers: « Compter »

Dans les microcontrôleurs 8051, les timers ne fonctionnent qu'en comptage

La résolution de ce compteur est exprimée en nombre de bits N. Un compteur N bits peut ainsi compter de 0 à $2^{N}-1$.

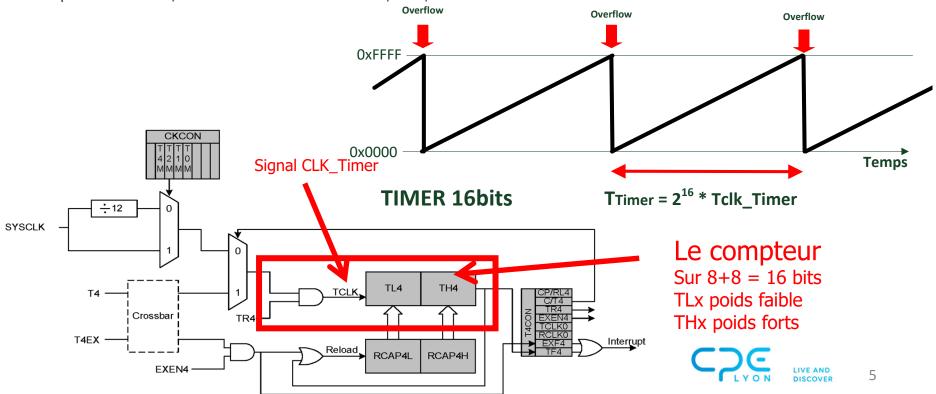
Dans les 8051, les résolutions des timers est de 16 bits (sauf dans le cas particulier des Timer 0 et 1, qui fonctionnent en 8 bits dans certains modes)

Dans un mode sans modulo, arrivé à $2^N - 1$, le compteur repart à 0.

Un signal d'horloge appelé CLK_Timer provoque l'incrémentation du compteur

La période TTimer est égale à 2^N * Période CLK_Timer avec N, résolution du Compteur

Cette période Ttimer pourra aussi être nommée « Temps de parcours du timer »





Bases des systèmes embarqués

Acte I - Première approche des Timers par un cas d'école:

le Timer utilisé en base de temps



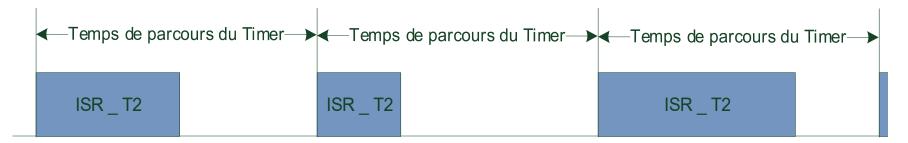
Les timers par la pratique – un cas d'école: Utilisation du Timer 2 en base de temps

Objectifs de la génération d'une base de temps

> Déclencher une interruption périodique

Avec une interruption périodique, il devient alors possible de gérer, de piloter, de nombreux évènements (des temporisations par exemple...).

On cherchera aussi à pouvoir facilement modifier la durée « Temps de parcours du Timer » en agissant sur la configuration du timer



Pourquoi le Timer2?

Il est à la fois relativement simple à mettre en œuvre, tout en offrant de nombreuses possibilités de mise en œuvre.

Dans cet exemple, nous chercherons à produire une interruption toutes les 25ms

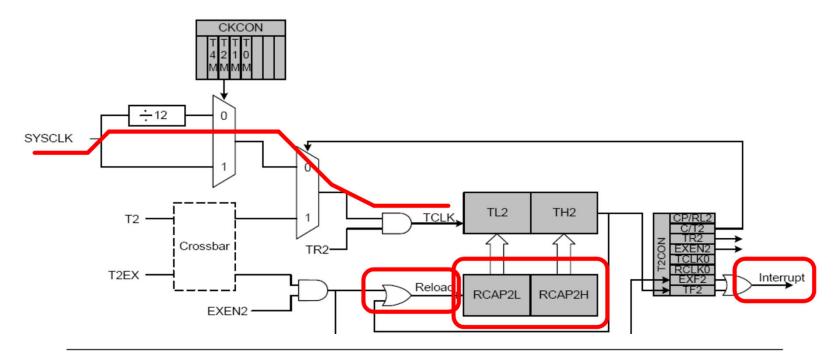


Timer 2 en base de temps - Configuration

Choix du mode de fonctionnement du timer

➤ Mode 1 - Timer en Auto-rechargement

Ce mode permet d'obtenir un temps de parcours du timer configurable en agissant sur le contenu des registres de rechargement RCAP2H-RC1AP2L



22.1.2. Mode 1: 16-bit Counter/Timer with Auto-Reload

The Counter/Timer with Auto-Reload mode sets the TF2 timer overflow flag when the counter/timer register overflows from 0xFFFF to 0x0000. An interrupt is generated if enabled. On overflow, the 16-bit value held in the two capture registers (RCAP2H, RCAP2L) is automatically loaded into the counter/timer register and the timer is restarted.

Registres impliqués pour utiliser le Timer 2

Le registres évidents qui apparaissent clairement dans la doc:

- T2CON Registre de contrôle: configuration et état
- TH2, TL2 registres de comptage
- RCAP2H, RCAP2L registres de rechargement (mode 1) ou de capture (mode 0)

Les registres moins évidents, mais non moins essentiels:

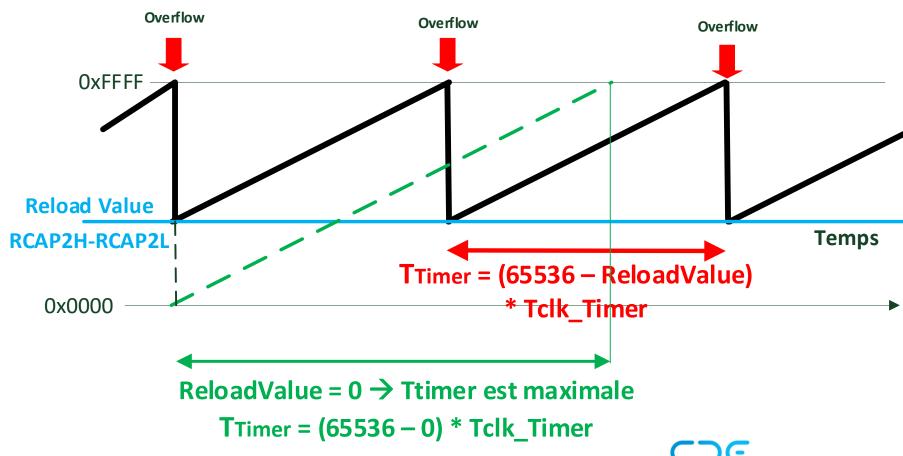
- CKCON: 1 bit CKCON.5 (T2M) Diviseur d'horloge pour CLK Timer
- PCON: 1 bit PCON,7 (SMOD0) Mode 2 uniquement.

RCLK0	TCLK0	CP/RL2	TR2	Mode
0	0	1	1	16-bit Counter/Timer with Capture
0	0	0	1	16-bit Counter/Timer with Auto-Reload
0	1	X	1	Baud Rate Generator for UART0
1	0	X	1	Baud Rate Generator for UART0
1	1	X	1	Baud Rate Generator for UART0
X	X	X	0	Off

Timer 2 en base de temps – Mode Auto-reload

Le timer compte en continu de la valeur de rechargement contenue dans RCAP2H-RCAP2L (ReloadValue) jusqu'à 0xFFFF (valeur max sur 16 bits)

TIMER2 - 16bits - Mode Auto-reload



Configuration Registre T2CON_ Bits 7-6-5

Figure 22.14. T2CON: Timer 2 Control Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Valu
	1	 	 		1	1	1	¬
TF2	EXF2	RCLK0	TCLK0	EXEN2	TR2	C/T2	CP/RL2	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address
Bit7:	TF2: Timer 2	Overflow Fl		Les noms des l déclarés dans C8051F020.h	oits sont		bit addressable	0xC8

Pending

Flag

Flag

Bit5:

Set by hardware when Timer 2 overflows. When the Timer 2 interrupt is enabled, setting this bit causes the CPU to vector to the Timer 2 interrupt service routine. This bit is not automatically cleared

by hardware and must be cleared by software. TF2 will not be set when RCLK0 and/or TCLK0 are logic 1.

Bit6: EXF2: Timer 2 External Flag.

Set by hardware when either a capture or reload is caused by a high-to-low transition on the T2EX **Pending**

input pin and EXEN2 is logic 1. When the Timer 2 interrupt is enabled, setting this bit causes the

CPU to vector to the Timer 2 Interrupt service routine. This bit is not automatically cleared by hard-

ware and must be cleared by software. RCLK0: Receive Clock Flag for UART0.

Selects which timer is used for the UART0 receive clock in modes 1 or 3. 0: Timer 1 overflows used for receive clock.

1: Timer 2 overflows used for receive clock.

RCLK0	TCLK0	CP/RL2	TR2	Mode
0	0	1	1	16-bit Counter/Timer with Capture
0	0	0	1	16-bit Counter/Timer with Auto-Reload

Configuration Registre T2CON_ Bits 4-3-2-1-0

Bit4: TCLK0: Transmit Clock Flag for UART0.

Selects which timer is used for the UART0 transmit clock in modes 1 or 3.

0: Timer 1 overflows used for transmit clock.

1: Timer 2 overflows used for transmit clock.

Bit3: EXEN2: Timer 2 External Enable.

Enables high-to-low transitions on T2EX to trigger captures or reloads when Timer 2 is not operating

in Baud Rate Generator mode.

0: High-to-low transitions on T2EX ignored.

1: High-to-low transitions on T2EX cause a capture or reload.

Bit2: TR2: Timer 2 Run Control.

This bit enables/disables Timer 2.

0: Timer 2 disabled.

1: Timer 2 enabled.

Bit1: C/T2: Counter/Timer Select.

0: Timer Function: Timer 2 incremented by clock defined by T2M (CKCON.5).

1: Counter Function: Timer 2 incremented by high-to-low transitions on external input pin (T2).

Bit0: CP/RL2: Capture/Reload Select.

This bit selects whether Timer 2 functions in capture or auto-reload mode. EXEN2 must be logic 1 for high-to-low transitions on T2EX to be recognized and used to trigger captures or reloads. If RCLK0

or TCLK0 is set, this bit is ignored and Timer 2 will function in auto-reload mode.

0: Auto-reload on Timer 2 overflow or high-to-low transition at T2EX (EXEN2 = 1).

1: Capture on high-to-low transition at T2EX (EXEN2 = 1).

Configuration Registre CKCON

Figure 22.1. CKCON: Clock Control Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	T4M	T2M	T1M	T0M	Reserved	Reserved	Reserved	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0x8E

Bit7: UNUSED. Read = 0b, Write = don't care.

Bit6: T4M: Timer 4 Clock Select.

> This bit controls the division of the system clock supplied to Timer 4. This bit is ignored when the timer is in baud rate generator mode or counter mode (i.e. C/T4 = 1).

Q: Timer 4 uses the system clock divided by 12.

Ne pas modifier T: Timer 4 uses the system clock.

Bit5: T2M: Timer 2 Clock Select.

This bit controls the division of the system clock supplied to Timer 2. This bit is ignored when the

timer is in baud rate generator mode or counter mode (i.e. C/T2 = 1).

0: Timer 2 uses the system clock divided by 12. Ne pas modifier

1: Timer 2 uses the system clock.

Bit4: T1M: Timer 1 Clock Select.

> ou seront, This bit controls the division of the system clock supplied to Timer 1.

0: Timer 1 uses the system clock divided by 12.

potentiellement Ne pas modifier Timer 1 uses the system clock. utilisés par

Bit3: T0M: Timer 0 Clock Select.

This bit controls the division of the system clock supplied to Counter/Timer 0. d'autres 0: Counter/Timer uses the system clock divided by 12. périphériques

Ne pas modifier dounter/Timer uses the system clock.

Bits2-0: Reserved. Read = 000b, Must Write = 000. les bits qui sont,

contr

Modifier proprement les registres de contrôle



Cas typique: modifications de quelques bits du registre

- Ne pas écraser les bits non concernés
- Exemple Config de CKCON pour le Timer2
- On veut mettre à zéro le bit 5 de CKCON sans altérer la valeur des autres bits de ce registre

```
CKCON = 0X00; Conséquence?
```

- \checkmark CKCON &= \sim 0x20; 1101 1111
- ✓ Équivalent à CKCON = CKCON & 0xDF;
- ✓ Équivalent aussi à CKCON &= ~ (1<<5);





- Même si, après un Reset, aucun registre n'est initialisé de manière aléatoire.
- Même si, la documentation contient la valeur post-reset de chaque registre.

On prendra soin de faire abstraction de cette initialisation postreset.

Pourquoi?

- Exemple d'une fonction d'initialisation ré-appelée en cours de programme....
- Débogage facilité



Tentative de calcul - Registres de rechargement RCAP2H et

RCAP2L

Cette valeur n'est pas choisie par hasard, elle correspond à la fréquence du quartz présent sur la carte d'évaluation

Calcul du contenu des registres de re-chargement.

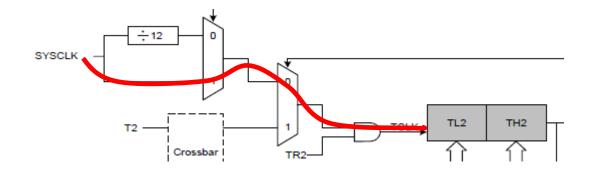
Hypothèse: SYSCLK de 22,1184 Mhz et période de 25mS souhaitée (Tick Time ou aussi appelée TClkTimer).

On veut faire en sorte que CLK_Timer: SYSCLK= 22,1184MHz, soit une période de 45,2nS Nombre d'incréments Timer nécessaires:

Tick Time / CLK_Timer Period = 553097 incréments > 65536

Il n'est pas possible avec un timer 16 bits piloté par une horloge de 22,1184MHz de produire une base de temps à 25ms.

Il est donc indispensable d'utiliser comme entrée d'horloge pour le Timer, le signal SYSCLK divisé par 12.



Registres de rechargement RCAP2 - Calcul

Calcul du contenu des registres de re-chargement.

Hypothèse: SYSCLK de 22,1184 Mhz et période de 25mS souhaitée (Tick Time ou aussi appelée TTimer).

CLK_Timer: SYSCLK/12= 1,8432MHz, soit une période de 0,542 μS

Nombre d'incréments Timer nécessaires:

Tick Time / CLK_Timer Period = 46080 incréments soit B400 en hexadécimal Le timer compte de sa valeur de pré-chargement jusqu'à 0xFFFF, puis repasse à la valeur de pré-chargement.

La valeur de rechargement est donc de (0xFFFF+1) - 0xB400 = 0x4C00

RCAP2L = 0x00

RCAP2H = 0x4C

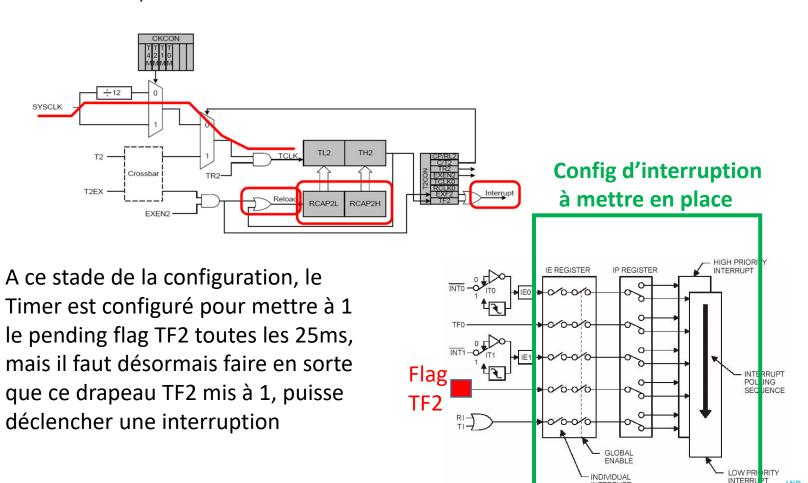
Le timer va donc compter en continu de 0x4C00 (valeur de rechargement) à 0xFFFF (Overflow)

Et les registres TL2 et TH2? Quel contenu?



Autres registres à configurer?

A-t-on configuré tous les registres nécessaires pour la mise en place d'un Timer fonctionnant en auto rechargement et produisant une interruption toutes les N ms?



Mise en place des interruptions





Ne pas oublier de mettre en place les interruptions.....

Interrupt Source	Interrupt Vector	Priority Order	Pending Flag	Bit addressable?	Cleared by HW?	Enable Flag	Priority Control
Reset	0x0000	Тор	None	N/A	N/A	Always Enabled	Always Highest
External Interrupt 0 (/INT0)	0x0003	0	IE0 (TCON.1)	Y	Y	EX0 (IE.0)	PX0 (IP.0)
Timer 0 Overflow	0x000B	1	TF0 (TCON.5)	Y	Y	ET0 (IE.1)	PT0 (IP.1)
External Interrupt 1 (/INT1)	0x0013	2	IE1 (TCON.3)	Y	Y	EX1 (IE.2)	PX1 (IP.2)
Timer 1 Overflow	0x001B	3	TF1 (TCON.7)	Y	Y	ET1 (IE.3)	PT1 (IP.3)
UART0	0x0023	4	RI0 (SCON0.0)	Y		ES0 (IE.4)	PS0 (IP.4)
Timer 2 Overflow (or EXF2)	0x002B	5	TF2 (T2CON.7)	Y		ET2 (IE.5)	PT2 (IP.5)

Mise en place des interruptions

Figure 12.9. IE: Interrupt Enable

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
EA	IEGF0	ET2	ES0	ET1	EX1	ET0	EX0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	FR Address:
							(bit addressable)	0xA8

Bit7: EA: Enable All Interrupts.

This bit globally enables/disables all interrupts. When set to '0', individual interrupt mask settings are overridden.

0: Disable all interrupt sources.

1: Enable each interrupt according to its individual mask setting.

Bit6: IEGF0: General Purpose Flag 0.

This is a general purpose flag for use under software control.

Bit5: ET2: Enabler Timer 2 Interrupt.

This bit sets the masking of the Timer 2 interrupt.

0: Disable Timer 2 interrupt.

1: Enable interrupt requests generated by the TF2 flag (T2CON.7).

Bit4: ES0: Enable UART0 Interrupt.

This bit sets the masking of the UART0 interrupt.

0: Disable UART0 interrupt.

1: Enable UART0 interrupt.

Mise en place des interruptions

Figure 12.10. IP: Interrupt Priority

R/W	R/W -	R/W PT2	R/W PS0	R/W PT1	R/W PX1	R/W PT0	R/W PX0	Reset Value				
Bit7	Bit7 Bit6 Bit5 Bit4 Bit3 Bit2 Bit1 Bit0 (bit addressable)											
Bits7-6:	Bits7-6: UNUSED. Read = 11b, Write = don't care.											
Bit5:	PT2: Timer 2 This bit sets th 0: Timer 2 int 1: Timer 2 int	ne priority o errupt priori	f the Timer 2 ty determine	interrupt. d by default	priority order	ī.						
Bit4: PS0: UART0 Interrupt Priority Control. This bit sets the priority of the UART0 interrupt. 0: UART0 interrupt priority determined by default priority order. 1: UART0 interrupts set to high priority level.												





Fonction: CFG Timer2 BT

De préférence dans l'ordre:

- Stopper le Timer2 (au cas où) Action sur TR2 de T2CON
- Configurer CKCON pour positionner T2M: Timer 2 Clock Select Choix effectué: CLK Timer2 = sysclk/12
- Configuration T2CON
 - Effacer le flag d'overflow TF2
 - Effacer le flag d'overflow EXF2 (pour éviter une interruption intempestive)
 - Configurer le mode 1 (action sur CPRL2, RCLK0 et TCLK0)
 - Fonctionnement en mode Timer Action sur C/T2
 - Dévalider l'action possible de l'entrée T2EX Action sur EXEN2
- Programmation des registres de rechargement RCAP2L et RCAP2H
- Initialisation (facultative) des registres de comptage du Timer T2L et T2H
- Configuration de la priorité pour interruption Timer2 action sur le registre IP
- Validation de l'interruption Timer2 action sur le registre IE
- Timer 2 démarré Action sur T2CON



La validation globale des interruptions (bit EA) sera insérée dans le code du Main

Mise en place de la fonction d'interruption ISR_Timer2

NB: Visu_INT_Timer2 est un drapeau matériel qui aurait pu être déclaré de la manière suivante:

sbit Visu_INT_Timer2 = Px^y; (Ne pas oublier de configurer la broche...) (uniquement valable sur les ports PO à P3

L'intérêt de ce drapeau est de pouvoir observer sur un oscilloscope l'exécution de l'interruption, et donc vérifier la périodicité et la fréquence de cette interruption.



Attention: l'évènement « Overflow » fait passer le « pending flag » TF2 du Timer2 à 1, mais il n'y a pas de mécanisme qui le remet automatiquement à zéro. Il faut donc le remettre à zéro par programme (d'où le TF2 = 0).

Si vous oubliez cette remise à zéro, le microcontrôleur après avoir exécuté l'interruption Timer 2, va aussitôt de nouveau exécuter l'interruption Timer 2 et ceci jusqu'à la fin des temps (ou presque...)





Bases des systèmes embarqués

Acte II - Généralisation de l'étude des Timers dans le 8051F020



Rappel: constitution d'un timer

Un timer est constitué au minimum des éléments suivants:

- Un compteur accessible au travers d'un ou plusieurs registres (TL2, TH2)
- Une logique de gestion et de contrôle accessible au travers d'un ou plusieurs registres (CKCON,T2CON)
- Une logique de gestion d'horloge pour gérer le comptage configurable via les registres de contrôle
- Un mécanisme de production d'interruptions selon les évènement liés au comptage

Très fréquemment, on rencontre aussi des registres destinés à capturer l'état du compteur à un instant donné ou à fournir au compteur des valeurs de rechargement (RCAP2L et RCAP2H)

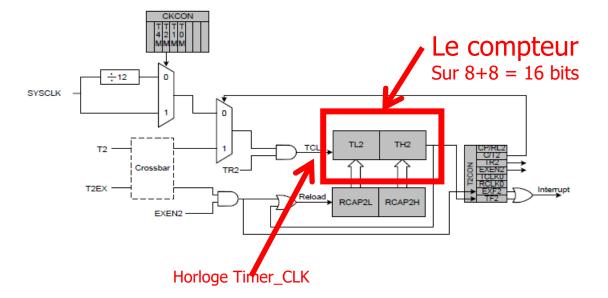


Figure 22.12. T2 Mode 1 Block Diagram

Sources du signal d'horloge d'un timer – Mode Timer/Counter

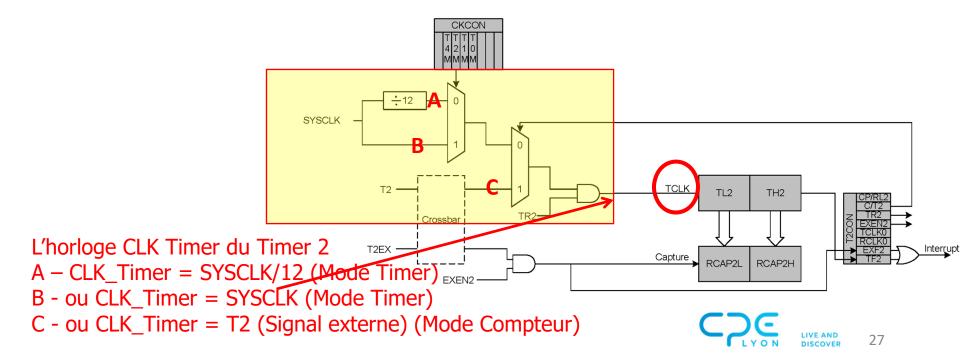
Cas général: la source d'horloge d'un Timer (Signal CLK_Timer) peut être issue de:

- L' horloge du processeur divisée ou pas
- D'un périphérique
 - Un autre Timer
 - Un périphérique spécifique
- Un signal extérieur

Le choix de la source d'horloge permet de distinguer 2 usages différents du Timer:

- Le mode « Temporisateur» (Timer) (comptage de temps). La source d'horloge est un signal périodique interne, souvent dérivée de l'horloge système du processeur
- Le mode « compteur » (Counter) (comptage d'évènements) La source d'horloge est dans ce cas un signal externe qui sera périodique ou pas...

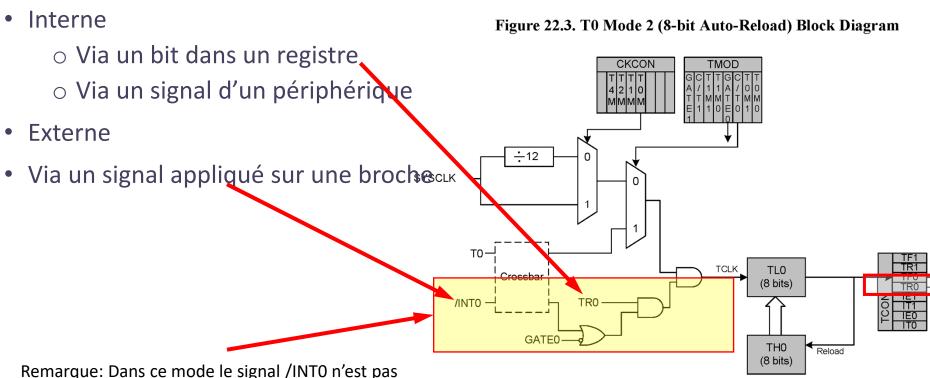
Figure 22.11. T2 Mode 0 Block Diagram



Validation du timer - Commande de Marche/Arrêt

La commande de validation du Timer agit sur le signal d'horloge CLK_Timer. Cette commande permet la mise en route ou l'arrêt du Timer, voire un arrêt temporaire (comme par exemple durant la modification de configuration).

Cette commande peut être



utilisé comme entrée d'interruption, mais comme entrée de validation du Timer0

Evènement essentiel dans un Timer: l'overflow

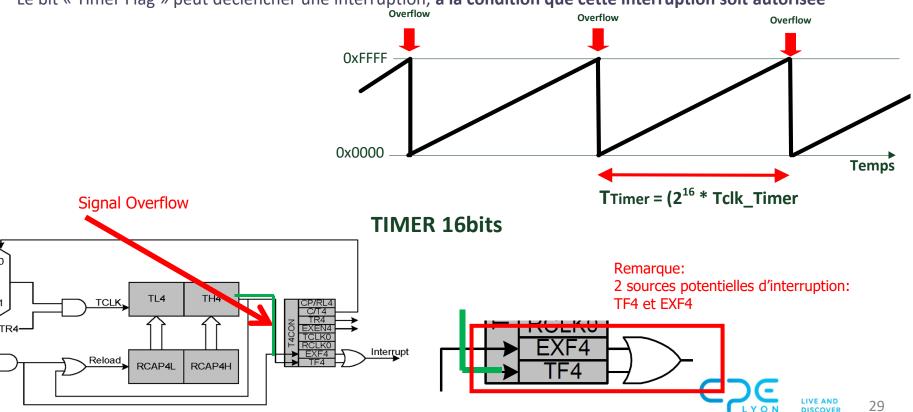
L'overflow (dépassement) correspond à l'instant ou le compteur atteint sa valeur maximale (0xFFFF pour un Timer 16 bits). Selon le mode de fonctionnement adopté, il revient alors à la valeur 0 (dans le cas du timer sans modulo) ou se recharge avec la valeur (Reload_Value) contenue dans les registres de rechargement.

L'overflow est donc un évènement périodique de période égale à la période Ttimer (Ttimer sera périodique à la condition que le signal CLK_Timer soit périodique)

Cet évènement Overflow met à un drapeau « Timer Overflow Flag » dans le registre de contrôle du Timer

Dans l'exemple ci-dessous, l'overflow du Timer4, met à 1, le bit TF4 du registre T4CON

Le bit « Timer Flag » peut déclencher une interruption, à la condition que cette interruption soit autorisée



Le comptage modulo (Mode Auto-reload)

Objectif général : obtenir une période TTimer différente de 2^N * Période CLK_Timer

On peut agir sur 2 paramètres: la fréquence de pilotage du Timer, ou jouer sur la résolution du timer (nombre de bits du compteur) .

- La fréquence de l'horloge CLK_Timer: dans le 8051, le choix se limite à 2 fréquences d'horloge interne, ou à une entrée de signal d'horloge externe (si on peut éviter de rajouter du matériel...)
- La résolution: changer la résolution est une opération matérielle lourde et peu souple (le changement de période évolue sur des puissances de 2)

Au lieu de changer la résolution du timer, et en s'appuyant sur l'évènement « Overflow », on peut alors agir sur la valeur de rechargement du timer juste après l'overflow.

Ainsi on remplace le parcours Timer $0 \rightarrow 2^{N}-1$ en continu, par un parcours de Reload_value $\rightarrow 2^{N}-1$ avec

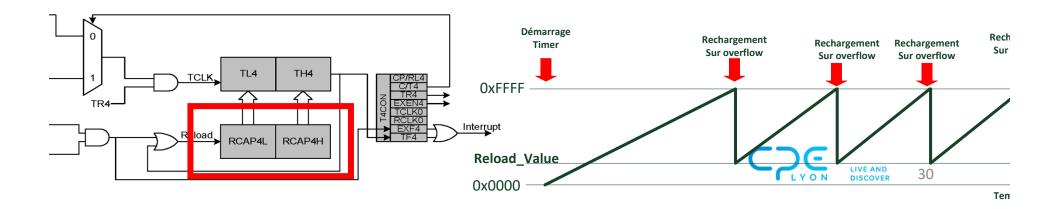
0 <= Reload_value <= 2^N-1

Cette valeur de rechargement est dans l'exemple ci-dessous du Timer4, stockée dans les registres RCAP4L et RCAP4H



La période Ttimer devient:

Ttimer = $Tclk_Timer * (2^N - Reload_value)$



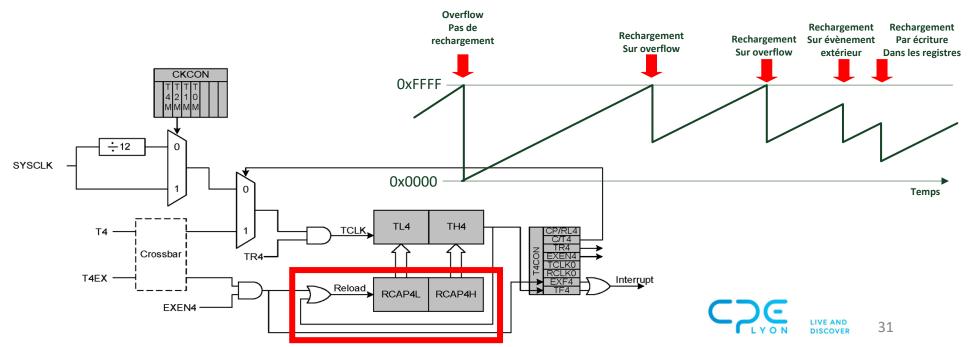
Rechargement du Timer – différentes manières de le provoquer

Définition élargie de « Rechargement du Timer »: on appelle ainsi toute opération qui modifie à un instant donné le contenu des registres de comptage

Cas du rechargement dans les modes auto-rechargement des timers: Arrivé à sa valeur maximale (0xFFFF), le timer au lieu de repasser à la valeur 0, les registres de comptage sont rechargés avec la valeur contenue dans les registres de rechargement (RCAP4L et RCAP4H recopiés dans TL4 et TH4).

Mais le rechargement des registres de comptage peut aussi être provoqué par:

- Ecriture directe dans les registres de comptage du timer s'imposent(TL4, TH4 ci-dessous) en prenant les précautions qui s'imposent (arrêter le timer)
- Rechargement commandé par l'extérieur (signal T4EX) via les registres de rechargement ((RCAP4L et RCAP4H recopiés dans TL4 et TH4) (A noter que cet évènement extérieur, outre le rechargement est aussi susceptible de provoquer une interruption grâce au drapeau EXF4 Timer External Flag)



Accéder au registres de comptage – Risques et limites

A tout instant, Il est possible de lire le contenu du compteur voire de le modifier.

Dans l'exemple ci-dessous concernant le Timer4, on accède au compteur 16 bits au travers de 2 registres:

- TL4 Poids faibles du compteur 16 bits
- TH4 _ Poids forts du compteur 16 bits
- De par la position de ces registres dans l'espace SFR, une déclaration SFR16 du type « sfr16 T4 = 0xf4; » est possible

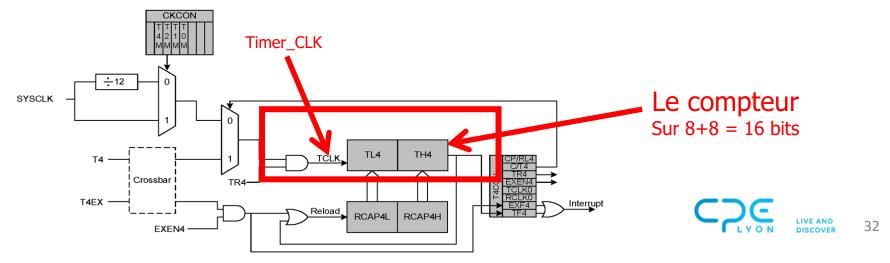
Ainsi on pourrait écrire (par exemple) if $(T4 \ge 0x1000)$ T4 = 0; //comptage de 0 à 0x1000

MAIS cette méthode est imprécise, peu fiable et consomme beaucoup de ressource processeur.

Consommation de ressource processeur: car on travaille en scrutation

Imprécise: de par le mode de scrutation. Si on veut gagner en précision, on augmente la vitesse de scrutation et on consomme encore plus de ressource...

Peu fiable: au niveau du processeur, l'accès aux registres de comptage, se fait séquentiellement, registre poids fort, puis registre poids faible (ou l'inverse), si dans le laps de temps qui sépare ces 2 accès, le timer s'incrémente, alors la comparaison sera faussée... (On peut palier le problème en arrêtant le compteur durant l'opération de lecture du timer, mais on perdra en précision....)



Autre évènement dans un Timer: la capture

La capture dans un Timer est la possibilité de « capturer » à un instant donné la valeur des registres de compta<mark>ge, et de stocker cette valeur capturée dans des registres dédiés: les registre de capture.</mark>

C'est le principe de lecture du temps intermédiaire dans un chronomètre. On fige l'affichage pour lire un temps intermédiaire, tandis que le chronomètre continue à compter.

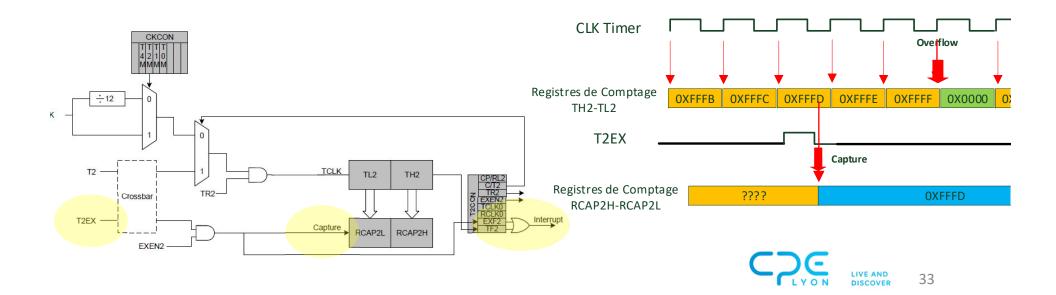
Cette « Capture », dans le cas du 8051 ne peut être provoquée que par un évènement extérieur (signal T2EX)

Dans le cas du Timer2 (schéma bloc ci-dessous), sur un front descendant du signal T2EX, une capture est exécutée. Le contenu des registres de comptage TL2 et TH2 est instantanément recopié dans les registres RCAP2L et RCAP2H sans stopper, ni altérer le comptage en cours.

Cet évènement « Capture » met en outre à un, le drapeau « Timer External Flag » dans le registre de contrôle du timer

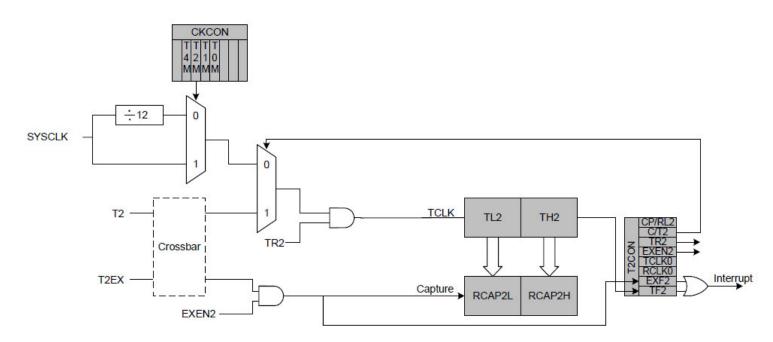
Dans l'exemple ci-dessous, la capture du Timer2, met à 1, le bit EXF2 du registre T2CON

Le bit « Timer External Flag » peut déclencher une interruption, à la condition que cette interruption soit autorisée



Précisions sur le mode « capture » dans les Timers 2 et 4

- Ce mode ne désactive pas l'évènement « overflow ». Ainsi même en mode capture, on pourra avoir le flag « Overflow flag » (TF2, TF4) mis à 1 lors de chaque Overflow. Par contre, la fonctionnalité « auto-rechargement » n'est pas possible dans ce mode car les registres « Reload-Capture » (RCAP2L/RCAP2H et RCAP4L/RCAP4H) sont réservés pour le stockage des valeurs capturées.
- Dans ce mode, le signal de capture (T2EX, T4EX) peut être invalidé par le bit EXEN (EXEN2, EXEN4) du registre de contrôle.
- Il est possible de fonctionner aussi bien en mode Timer (source d'horloge interne) qu'en mode compteur (source d'horloge externe).



Une interruption avec 2 sources possibles d'interruption

Dans le cas, par exemple du Timer2, 2 drapeaux (flags) sont susceptibles de provoquer une interruption Timer2:

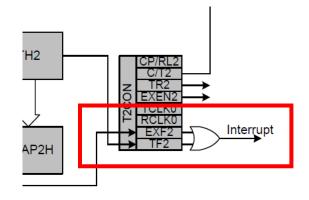
- Le drapeau TF2, activé par un évènement « overflow »
- Le drapeau EXF2 activé par un évènement extérieur sur le signal EXF2 (actif front descendant). Cet évènement EXF2 est susceptible de plus, en mode capture, de provoquer une capture des registres de comptage.

Aussi dans la fonction d'interruption liée au timer 2, il conviendra des déterminer quelle est la véritable source d'interruption (TF2 ou EXF2) afin de faire le bon traitement.

Exemple de squelette de code d'interruption:

```
void ISR_INT_TIMER2() interrupt 5
if (EXF2 == 1) { // Traitement évènement EXF2
}
if (TF2 == 1) { // Traitement évènement overflow
}
```

Remarque: éviter le « if » suive de « else », car pour une interruption donnée, on peut très bien avoir les 2 flags à 1 simultanément.





Signaux produits par les timers

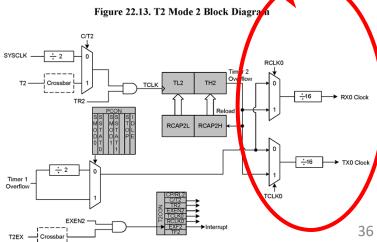
Les timers, outre les drapeaux qu'ils positionnent dans leurs registres de contrôle, peuvent produire des signaux.

Ces signaux pourraient être routés sur des broches d'entrée sortie ou être utilisé par d'autres périphériques.

Dans le cas du 8051F020, pour les timers 0 à 4, il n'est pas possible de router sur des broches I/O des signaux tels que l'overflow (par exemple).

Par contre, ce signal d'overflow pourra être utilisé par un autre périphérique (L'UART – un périphérique de communication série, l'ADC – Convertisseur Analogique- Numérique)

Dans le cas du Timer PCA du 8051F020, il sera possible de produire des signaux accessible sur les broches I/O dans différents modes de fonctionnement de ce Timer.



Timers et Crossbar

Un certain nombre de broches PIO peuvent être réservées pour les timers.

Par exemple, pour le Timer 2, 2 broches PIO peuvent être requises pour les signaux T2 et T2EX

Mais, selon le mode d'utilisation de ces timers, ces signaux ne seront pas forcément mis en œuvre.

Aussi, il ne sera pas forcément nécessaire de router ces signaux via le crossbar.

Exemple: pour produire une base de temps en mode auto-rechargement sur le Timer2, les signaux T2 et T2EX ne sont pas requis. Il ne sera donc pas utile de router via le crossbar ces mêmes signaux.

				-	P0				ı			F	21				П				P2								P3					Crossbar Register Bit
PIN I/O	0	1	2	3	4	5	6	7	0	1	2	3	4	- 5	6	7		0 1	2	3	4	- 5	- 6	3 7	7 (0	1 2		3	4	5	6	7	orecond regions an
TX0	•																																	UARTOEN: XBR0.2
RX0		•																																OAKTOLK, ALKO,E
sck	•		•			П											Т																	
MISO		•		•																														SPI0EN: XBR0.1
MOSI			•		•																													SPIOLIN. ABRO. I
NSS				•		•																												
SDA	•		•		•		•										Т																	SMB0EN: XBR0.0
SCL		•		•		•		•																										SINDUEN. ABRU.U
TX1	•		•		•		•		•								Т																	UART1EN: XBR2.2
RX1		•		•		•		•	•	•																								OARTIEN, ABRZ.Z
CEX0	•		•		•		•		•	•	•						Т																	
CEX1		•		•		•		•	•	•	•	•																						
CEX2			•		•		•		•	•	•	•	•																					PCA0ME: XBR0.[5
CEX3				•		•		•	•	•	•	•	•	•																				
CEX4					•		•		•	•	•	•	•	•	•																			
ECI	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		,																	ECI0E: XBR0.6
CP0	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•																CP0E: XBR0.7
CP1	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	,															CP1E: XBR1.0
то	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•)														T0E: XBR1.1
/INTO	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•														INT0E: XBR1.2
T1	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•	•													T1E: XBR1.3
/INT1	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•												INT1E: XBR1.4
T2	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•										T2E: XBR1.5
T2EX	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	1	•	•		•	•	•		•									T2EXE: XBR1.6
T4	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•			•								T4E: XBR2.3
T4EX	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•			•	•							T4EXE: XBR2.4
SYSCLK	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•		•	•	•			•	•	•						SYSCKE: XBR1.7
CNVSTR	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		•		•	•	•					CNVSTE: XBR2.0
									90	6	9	F	12	13	14	ı,	2		-															
									IN1.0/A8	IN1.1/A9	IN1.2/A10	IN1.3/A11	IN1.4/A12	IN1.5/A13	N1.6/A14	N1 7/415	9	Sm/A1	10m/A2	11m/A3	12m/A4	13m/A5	14m/AR	15m/47	DONO	3 2	מונים	1 0	3	D4/D4	DS/DS	D6/D6	D7/D7	
						Щ	2	¥	Ξ	E	Ξ	E	E	Ξ	Z	- 5		E 5	å	-	22	8	4	f ú	5 5	3	5 6	3 6	3	4	25	90	1/2	



Bases des systèmes embarqués

Acte III – Exemple d'utilisation des Timers dans le 8051F020





Exemple: Timer 2 utilisé dans une temporisation logicielle paramétrable (Delay();)

ole

Exemple de fonction de temporisation paramétrable L'argument passé est un entier non signé – L'unité de l'argument est en millisecondes

Fonction Timer_Delay (unsigned int Val_millisecond)

Configuration Timer:

- Choisir l'horloge Timer (CLK Timer = SYSCLK ou SYSCLK/12)
- Configurer le Timer en mode Autorechargement
- Effacer le flag overflow
- Régler les valeurs de rechargement pour un temps de parcours timer de 1ms
- Initialiser les registre timer avec la valeur de rechargement
- Timer démarré

Boucle

- Faire N fois pour une temporisation de N millisecondes:
 - Attendre l'overflow Test du bit Overflow à 1
 - Remettre le bit d'overflow à zéro
- Timer stoppé
- FIN

Par rapport à une fonction de délai purement logicielle, Cette temporisation est précise et peut être facilement rendue Indépendante de la fréquence d'horloge système Par contre, on fonctionne toujours en scrutation... Configuration du timer Tconfig<< TTimer

Scrutation Overflow N fois



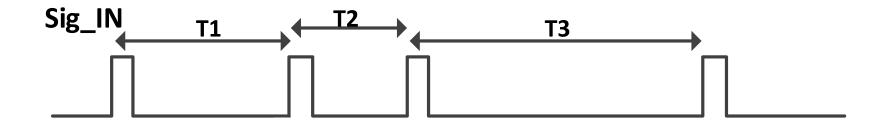


Mesure de temps



Soit un signal périodique dont on souhaite connaître la période T instantanée (mesure de T1, T2, T3, etc...).

Quelles sont les solutions pour traiter ce problème?

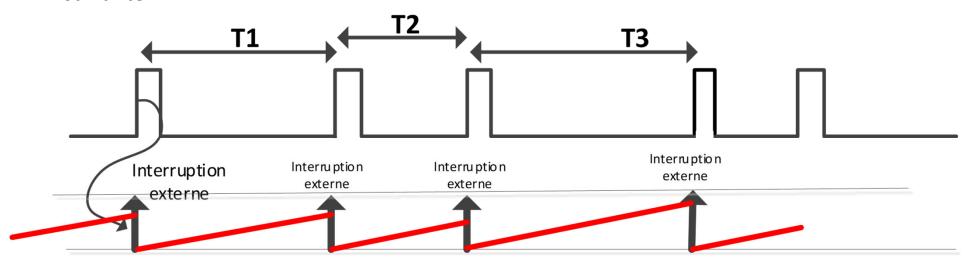


Mesure de temps – Solution 1

- Une Interruption Externe déclenchée par un front (montant ou descendant du signal SIG_IN)
- Un Timer configuré en comptage « Timer » (Horloge interne) sans modulo (mode capture sans capture ou mode auto-rechargement avec rechargement à 0)

A chaque interruption:

- Lecture du contenu des registres de comptage du Timer pour calculer la durée écoulée depuis l'Interruption précédente Tn = CPTimer*TCLKTimer
- Puis, Remise à zéro des registres de comptage du Timer pour préparer la mesure suivante



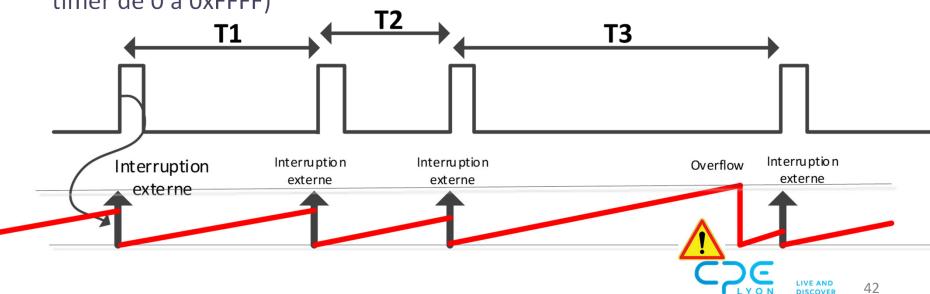
Mesure de temps – Limites de la solution 1

Pour:

- Simplicité de mise en œuvre
- Fonctionnement évident

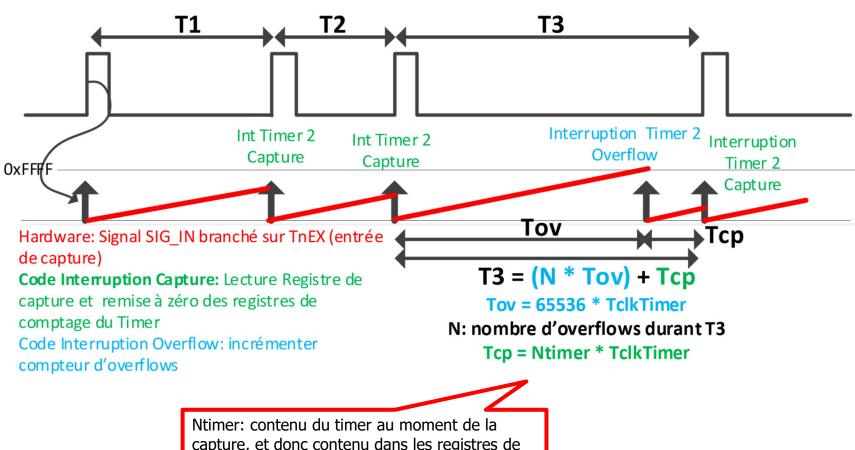
Contre:

- Manque de précision
 - Temps de latence de l'interruption et temps d'exécution de l'interruption
 - Nécessité d'arrêter le timer pour lire le contenu des registres Timer (sinon risque de lecture erronée...)
- Consommation d'un timer et d'une interruption externe
- Gestion du dépassement timer (ci-dessous: T3 > Temps de comptage du timer de 0 à 0xFFFF)



Mesure de temps – Solution 2

- Timer configuré en mode Capture et comptage « Timer » (CLK interne)
- Branchement du signal SIG_IN sur l'entrée TnEX du timer (entrée de capture)
- Une seule interruption Timer, mais 2 sources possibles (Capture et Overflow)



capture, et donc contenu dans les registres de capture...





Rappel:

- Timer configuré en mode Capture et comptage « Timer » (CLK interne)
- Branchement du signal SIG_IN sur l'entrée TnEX du timer (entrée de capture)
- Une seule interruption Timer, mais 2 sources possibles (Capture et Overflow)

Pour:

- Meilleure précision apportée par le mécanisme de capture matériel du contenu timer
- La mesure du temps sur plusieurs cycles Timer est possible

Contre:

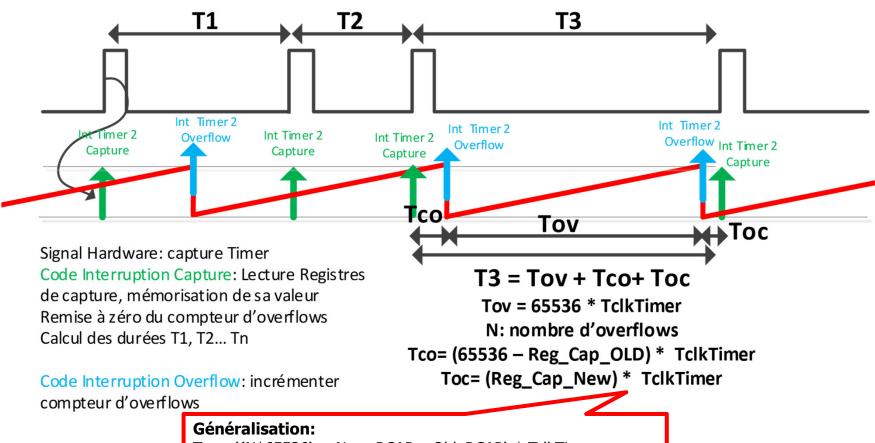
- La remise à zéro du Timer est faite par logiciel (précision!!)
- Attention, sur le comptage multi-cycles Timer, envisager à gérer un déclenchement quasi simultané de l'overflow et de la capture

La résolution de ces 2 points précédents est permis par la solution 3....



Mesure de temps – Solution 3

- Timer configuré en mode Capture et comptage « Timer » (CLK interne)
- Branchement du signal SIG_IN sur l'entrée TnEX du timer (entrée de capture)
- Une seule interruption Timer, mais 2 sources possibles (Capture et Overflow)
- Identique à solution 2, mais pas de remise à zéro du compteur Timer



Tn = ((N*65536) + New_RCAP – Old_RCAP) * TclkTimer Avec N: nombre d'overflows depuis la Capture précédente New_RCAP : valeur des registres de capture pour la capture N Old_RCAP: valeur des registres de capture pour la capture N-1



Application: Comptage d'évènements



On souhaite compter des évènements extérieurs

Condition préalable: le signal doit être conditionné pour produire des signaux digitaux compatibles avec le microcontrôleur (CMOS 3V3)

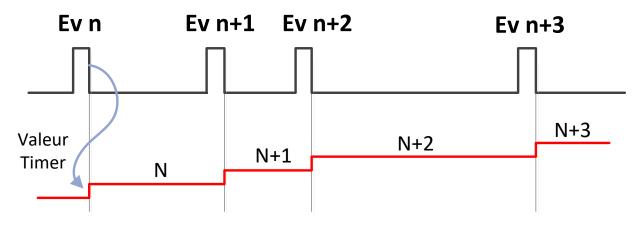
Ev n	Ev n+1	Ev n+2	Ev n+3
П	П		

Comptage d'évènements – Généralités

- ELEMENT ESSENTIEL: bien que l'on puisse faire sans (interruptions externe à chaque évènement et comptage logiciel), un timer permet de fortement réduire la charge de travail du processeur.
- **ELEMENT ESSENTIEL**: Utilisation du Timer configuré en mode « **Counter** » (CLK externe) L'horloge qui permet l'incrémentation des registres de comptage est en fait le signal dont on souhaite compter les évènements. Autrement dit, le signal dont on souhaite compter les évènements doit être connecté sur l'entrée d'horloge externe du timer (Par exemple T2 sur le Timer2)

Pour le reste, à chaque situation, une configuration particulière au niveau du timer

- Possibilité de travailler en interruption ou en scrutation.
- Attention, certains timers peuvent déclencher des interruptions sur 2 évènements (Overflow et Capture), et il n'est pas possible d'autoriser une demande d'interruption de l'un sans l'autre... Il faudra donc dans le code d'interruption s'assurer quelle est la source de déclenchement de l'interruption.
- En scrutation, attention à la lecture de la valeur du timer: risque de lecture erronée car le timer peut être incrémenté à tout instant (Solution: arrêt du Timer, mais risque de perte d'impulsions de comptage, ou capturer....)
- L'utilisation de la variable de type SFR16 pour les registres timer ne supprime pas le risque de lecture erronée. L'accès se fait toujours octet par octet



Comptage d'évènements – Exemples de solutions

Exemple 1: On souhaite être averti à chaque fois que l'on a compté N événements (N <65535): utilisation du mode Auto-Reload de manière à pouvoir produire périodiquement une interruption tous les N évènements comptés (configurer au préalable les registres de rechargement avec la valeur 65536-N)

• Exemple 2: On souhaite pouvoir à tout instant connaître le nombre d'évènements comptés (pas de limitation sur la valeur de N): utilisation du mode Capture pour que le timer compte de 0 à 65535 en continu. L'interruption sur Overflow permet de mettre à jour une variable de comptage global (+65536 évènements). De plus une broche GPIO pourra déclencher une capture (par exemple, action sur T2EX pour le timer2) et donc une interruption « Capture - Flag EXF2 sur Timer2». A cet instant le nombre d'évènements compté sera égal à la somme de la variable de comptage global et du contenu des registres de capture.

Bilan sur l'utilisation des Timers

Face à tout problème d'utilisation des Timers, il y a toujours de nombreuses solutions possibles...

- Privilégiez les solutions qui économisent le temps processeur en supprimant si possible toutes les scrutations et en minimisant les durées des interruptions.
- Avant de choisir il faut prendre le temps d'analyser le problème en évaluant les valeurs limites des divers paramètres (par exemple un timer sur 16 bits ne peut théoriquement pas compter plus de 65536 évènements).
- Prévoir la gestion de ces mêmes paramètres quand ils sont hors limites.
- Ne pas oublier qu'un Timer fonctionne soit en timer (Clk interne) pour mesurer le temps soit en compteur (clk externe) pour compter des événements.
- L'utilisation en compteur (CLK externe) pour mesurer le temps est à proscrire....

Les questions à se poser quand on utilise un Timer

- Quelle horloge? Interne pour un mode « Timer », ou Externe pour un mode « Counter »
- Quel mode? Auto-reload, Capture, voire Baud Rate Generator?
- Quelle interruption? Où plutôt quel type d'évènement d'interruption devrez vous traiter?



Le Timer PCA dans le 8051F020

Le PCA « programmable Counter Array »

- Timer évolué
- Allégement des tâches « timer » pour le processeur

