



Haute école d'ingénierie et d'architecture Fribourg
Hochschule für Technik und Architektur Freiburg

Systèmes Embarqués 1 & 2

tp.05 - Pilote d'un timer du AM335x

Classes T-2/I-2 // 2016-2017

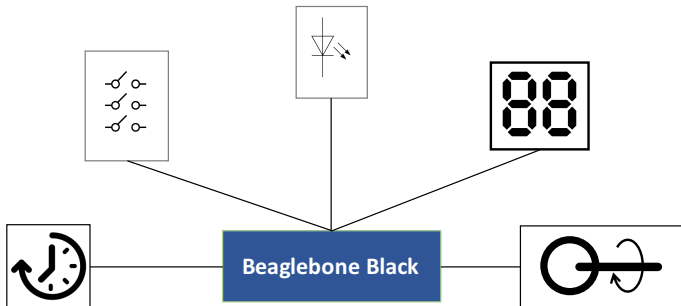
Daniel Gachet | HEIA-FR/TIC
tp.05 | 22.11.2016



- A la fin du laboratoire, les étudiant-e-s seront capables de
 - ▶ Etudier le datasheet d'un contrôleur et en comprendre le contenu
 - ▶ Concevoir un pilote de périphérique en C
 - ▶ Concevoir le pilote pour le contrôleur DMTimer du μ P AM3358
 - ▶ Faire ses premiers pas avec les pointeurs en C
 - ▶ Intégrer des composants développés lors de travaux précédents
 - ▶ Concevoir une application originale, modulaire et générique
 - ▶ Debugger une application en C sous Eclipse
- Durée
 - ▶ 2 séances de laboratoire (8 heures)
- Rapport
 - ▶ Rapport de laboratoire avec le code source à la fin du 2^e laboratoire



- Développer une application offrant deux fonctionnalités, un chronomètre et un compte à rebours (countdown).





- Choix de la fonction
 - ▶ chronomètre : bouton S1 pour choisir le chronomètre
 - ▶ countdown : bouton S2 pour choisir le countdown
 - ▶ reset : bouton S3 pour stopper l'activité et revenir à l'état initial
- Indication de la fonction en cours
 - ▶ chronomètre : LED1 allumée
 - ▶ countdown : LED2 allumée
- Affichage du temps sur le display 7-segments
 - ▶ temps inférieur à 10 secondes : affiché en 1/10 secondes
 - ▶ temps supérieur à 10 secondes : affiché en secondes
- L'encodeur rotatif permettra de choisir la valeur du countdown
- Activation de la fonction avec le bouton poussoir de l'encodeur
 - ▶ 1 pression pour démarrer le chronomètre ou le countdown
 - ▶ 1 pression pour stopper le chronomètre ou le countdown



Travail à réaliser (III)

■ Horloge / timer

- ▶ Le timer DMTimer 1ms du μ P AM3358 servira d'horloge
- ▶ Le timer sera contrôlé par un pilote de périphérique, lequel offrira les services suivants
 - ◇ Méthode pour initialiser le timer
 - ◇ Méthode pour lire la valeur actuelle du timer
 - ◇ Méthode pour lire la fréquence à laquelle le timer est cadencé
- ▶ Le fonctionnement du pilote de périphérique devra être validé et vérifié



- Les conditions d'exécution sont les suivantes
 - ▶ Le squelette du projet se trouve sur le dépôt centralisé
 - ▶ Pour le télécharger, tapez les commandes suivantes

```
$ cd ~/workspace/se12/tp
$ git pull upstream master
```
- Le code et le rapport seront rendus au travers du dépôt Git centralisé
 - ▶ *sources : .../tp/tp.05*
 - ▶ *rapport : .../tp/tp.05/doc/report.pdf*
- Délai
 - ▶ Le journal et le code doivent être rendus au plus tard 6 jours après la 2^e séance de laboratoire à minuit



■ Pour mettre à jour la bibliothèque spécialisée du Beaglebone

```
$ cd ~/workspace/se12
```

```
$ git pull
```

```
$ make -C ~/workspace/se12/bbb/source
```

■ Pour mettre à jour les paths des includes dans eclipse

- ▶ ouvrir **Properties** pour votre projet
- ▶ aller **C/C++ General** → **Paths and Symbols**
- ▶ ouvrir **Includes** → **GNU C**
- ▶ ajouter **/home/lmi/workspace/se12/bbb/source**



- Quelle est la signification du qualificatif **volatile** ?
- Quelle est l'utilité du qualificatif **volatile** , quand il est associé à un pointeur ?
- Comment peut-on efficacement définir des registres et leur contenu en c ?
- Comment peut-on accéder aux registres d'un contrôleur de périphérique situés dans l'espace d'adressage du μP ?
- Comment sont placés les attributs d'une structure dans la mémoire ?
- Quelle est l'utilité des conversions de type (type casting) en C ?
- Comment réalise-t-on un "type cast" en C ?



- Le μ P dispose de 8 timers distincts (DMTimer 0 à 7)
- Pour notre application, nous allons utiliser le DMTimer 1ms

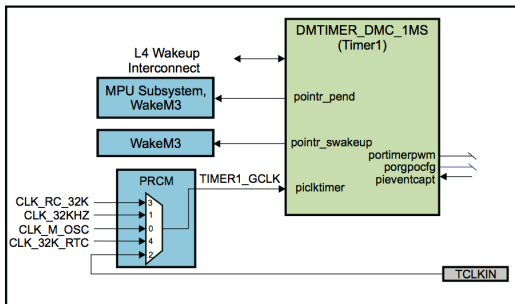


Figure 20-24. DMTimer 1 ms Integration

- Pour enclencher son horloge, appelez la fonction
`"am335x_clock_enable_timer_module(AM335X_CLOCK_TIMER1)"`

⁰Réf. 06_am335x_technical_reference_manual.pdf, chapter 20



⁰Réf. 06_am335x technical reference manual.pdf, chapter 20



Figure 20-25. TCRR Timing Value

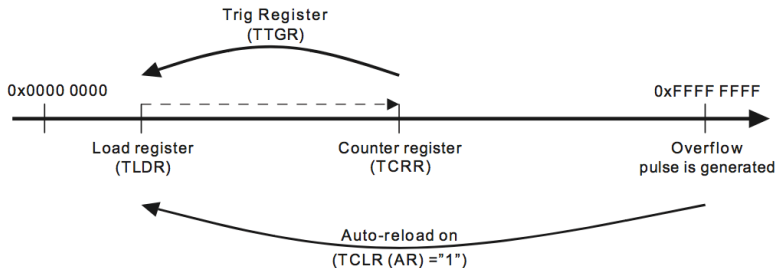




Table 20-35. DMTIMER_1MS REGISTERS

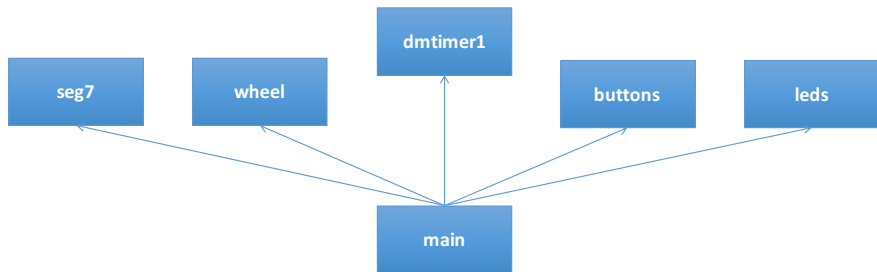
| Offset | Acronym | Register Name | Section |
|--------|-----------|---|-----------------------------------|
| 0h | TIDR | This register contains the IP revision code | Section 20.2.4.1 |
| 10h | TIOCP_CFG | This register controls the various parameters of the OCP interface | Section 20.2.4.2 |
| 14h | TISTAT | This register provides status information about the module, excluding the interrupt status information | Section 20.2.4.3 |
| 18h | TISR | The Timer Status Register is used to determine which of the timer events requested an interrupt. | Section 20.2.4.4 |
| 1Ch | TIER | This register controls (enable/disable) the interrupt events | Section 20.2.4.5 |
| 20h | TWER | This register controls (enable/disable) the wakeup feature on specific interrupt events | Section 20.2.4.6 |
| 24h | TCLR | This register controls optional features specific to the timer functionality | Section 20.2.4.7 |
| 28h | TCRR | This register holds the value of the internal counter | Section 20.2.4.8 |
| 2Ch | TLDR | This register holds the timer's load value | Section 20.2.4.9 |
| 30h | TTGR | This register triggers a counter reload of timer by writing any value in it. | Section 20.2.4.10 |
| 34h | TWPS | This register contains the write posting bits for all writable functional registers | Section 20.2.4.11 |
| 38h | TMAR | This register holds the match value to be compared with the counter's value | Section 20.2.4.12 |
| 3Ch | TCAR1 | This register holds the value of the first counter register capture | Section 20.2.4.13 |
| 40h | TSICR | Timer Synchronous Interface Control Register | Section 20.2.4.14 |
| 44h | TCAR2 | This register holds the value of the second counter register capture | Section 20.2.4.15 |
| 48h | TPIR | This register is used for 1ms tick generation. The TPIR register holds the value of the positive increment. The value of this register is added with the value of the TCVR to define whether next value loaded in TCRR will be the sub-period value or the over-period value. | Section 20.2.4.16 |
| 4Ch | TNIR | This register is used for 1ms tick generation. The TNIR register holds the value of the negative increment. The value of this register is added with the value of the TCVR to define whether next value loaded in TCRR will be the sub-period value or the over-period value. | Section 20.2.4.17 |
| 50h | TCVR | This register is used for 1ms tick generation. The TCVR register defines whether next value loaded in TCRR will be the sub-period value or the over-period value. | Section 20.2.4.18 |
| 54h | TOCR | This register is used to mask the tick interrupt for a selected number of ticks. | Section 20.2.4.19 |
| 58h | TOWR | This register holds the number of masked overflow interrupts. | Section 20.2.4.20 |



- L'étude d'un datasheet n'est jamais très aisée
- La liste ci-dessous indique les chapitres et les registres les plus intéressants pour la réalisation du projet
 - ▶ Document : *06_am335x_technical_reference_manual.pdf*
 - ▶ Contrôleur : *20 Timers*
 - ▶ Les chapitres
 - ◇ 20.2.1 et 21.2.2
 - ◇ 21.2.3.1
 - ▶ Les registres
 - ◇ TIOCP_CFG et TISTAT
 - ◇ TCLR, TCRR, TLDR et TTGR



Structure générale de l'application



- "main" module principal de l'application
- "seg7" module pour le pilotage de l'afficheur 7-segments
- "wheel" module pour le pilotage de l'encodeur rotatif
- "dmtimer1" module pour le pilotage du timer 1 du μP
- "buttons" module pour le pilotage des boutons poussoir
- "leds" module pour le pilotage des leds