



# Electronics Project

## TP Prog 1 : Timers

[contact@42chips.fr](mailto:contact@42chips.fr)

*Résumé: Initiation à l'utilisation des timers hardware.*

# Table des matières

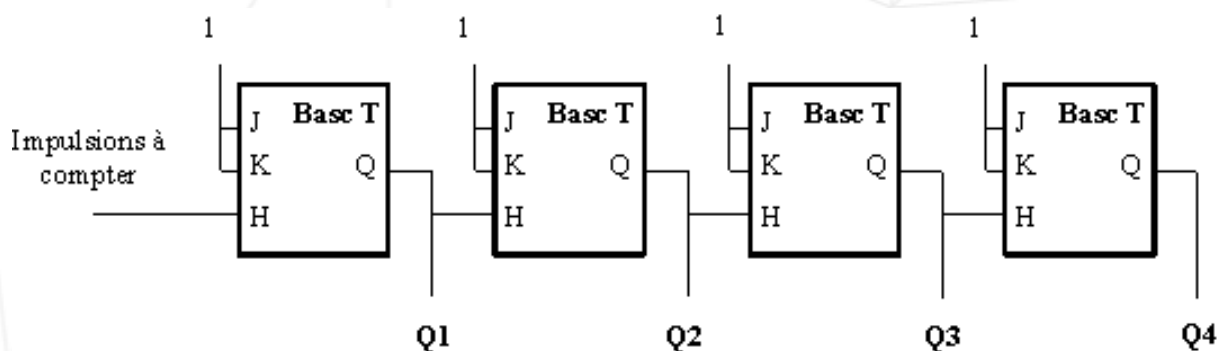
# Chapitre I

## Préambule

En électronique, un compteur est un circuit intégré numérique destiné à compter le nombre d'impulsions appliquées à son entrée. Il est composé d'un certain nombre de [bascules](#) D, T ou JK.

Le compteur le plus simple est obtenu en mettant en cascade une série de bascules T, le signal à compter étant appliqué à l'entrée de la première bascule ; la sortie de cette bascule pilote l'entrée de la deuxième bascule et ainsi de suite.

Le résultat du comptage apparaît sous forme de nombre binaire, la première bascule indiquant le chiffre le moins significatif.



La capacité du compteur, c'est le nombre maximum d'impulsions qu'il peut totaliser. Elle vaut  $2^N$  pour un compteur binaire.

Si l'on dépasse la capacité, le compteur revient à 0 et se remet à compter.

Mais bon, trêve de plaisanteries, voyons maintenant ce qui nous intéresse le plus. Comment le faire sur [minecraft](#)

# Chapitre II


## Consignes générales

Sauf contradiction explicite, les consignes suivantes seront valables pour tous les TPs

- Le langage utilisé pour ce projet est le C.
- Il n'est pas nécessaire de coder à la norme de 42.
- Les exercices sont très précisément ordonnés du plus simple au plus complexe. En aucun cas nous ne prendrons en compte ni n'évaluerons un exercice complexe si un exercice plus simple n'est pas parfaitement réussi.
- Vos exercices seront évalués par des responsables de l'association 42Chips.
- Vous ne devez laisser aucun autre fichier que ceux explicitement spécifiés par les énoncés des exercices dans votre répertoire lors de la peer-évaluation.
- Vous avez une question ? Demandez à votre voisin de droite ou de gauche. Vous pouvez demander sur le salon dédié dans le discord [42Chips](#) ou en dernier recours à un responsable 42Chips.
- Toutes les réponses à vos questions techniques se trouvent dans les **datasheets** ou sur Internet. A vous d'utiliser et d'abuser de ces sujets pour comprendre comment réaliser votre exercice.
- Vous devez utiliser la datasheet du microcontrôleur qui vous est fourni et commenter les parties importantes de votre programme en renseignant où vous avez trouvé les indices dans le document, et, si nécessaire, expliquer votre démarche. Ne faites pas des pavés non plus. Il faut que cela reste clair.
- Écoutez attentivement les encadrants lors des séances de TP, ils vous donneront des éléments essentiels sur le fonctionnement du microcontrôleur.

# Chapitre III

## Exercice 00 : Timer1

	Exercice : 00
Timer1	
Dossier de rendu : <i>ex00/</i>	
Fichiers à rendre : <code>Makefile</code> , <code>main.c</code>	
Fonctions interdites : <code>PORTX</code> , <code>avr/interrupt.h</code>	


- Vous devez écrire un programme qui permet d'allumer et éteindre la LED D3 (PB1) à une fréquence de 1Hz.
- Vous devez configurer les registres du `Timer1` pour commander la LED.
- La boucle infinie du programme doit rester vide.



Vous devez à chaque fois expliquer la fonction et les valeurs assignées aux registres en commentaire !

# Chapitre IV

## Exercice 01 : Rapport cyclique

	Exercice : 01
Rapport cyclique	
Dossier de rendu : <i>ex01/</i>	
Fichiers à rendre : <b>Makefile</b> , <b>main.c</b>	
Fonctions interdites : <b>PORTX</b> , <b>avr/interrupt.h</b>	

- Vous devez écrire un programme qui permet d'allumer et éteindre la LED D1 (PB3) à une fréquence de 1Hz et avec un [rapport cyclique](#) de 10%.
- Vous devez configurer les registres du **Timer1** pour commander la LED.
- La boucle infinie de votre programme doit rester vide.