



# Piscine embarquée

Module09 : expandeur et 7 segments

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

*Résumé: Quand on a pas assez de GPIO*

*Version: 1*

# Chapitre I

## Préambule

Les expandeurs sont utilisés en électronique pour étendre les capacités d'un microcontrôleur en termes de nombre d'entrées/sorties (I/O).

Ils permettent de connecter plusieurs périphériques à un seul bus I2C sans occuper trop de broches sur le microcontrôleur.

Ce qui est utile pour les projets qui nécessitent de nombreuses entrées/sorties mais n'ont pas suffisamment de broches disponibles.

Les expandeurs I2C sont également très pratiques car ils peuvent être facilement configurés et contrôlés en utilisant simplement des commandes I2C, ce qui rend le développement de projets plus rapide et plus facile.

# Chapitre II


## Consignes générales

Sauf contradiction explicite, les consignes suivantes seront valables pour tous les exercices.

- 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.
- Toutes les réponses à vos questions techniques se trouvent dans les **datasheets** ou sur Internet. À 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.
- Vous avez une question ? Demandez à votre voisin de droite ou de gauche. Vous pouvez demander sur le salon dédié dans le Discord de la piscine ou en dernier recours à un staff.

# Chapitre III

## Finger display

	Exercice : 00
Une lueur d'espoir (encore...)	
Dossier de rendu : <i>ex00/</i>	
Fichiers à rendre : <b>Makefile</b> , <b>*.c</b> , <b>*.h</b>	
Fonctions Autorisées : <b>avr/io.h</b> , <b>util/delay.h</b>	


Je vais vous demander de juste allumer une LED pour cet exercice.  
Et comme c'est le D(ay)9 vous allez devoir faire cligoner la LED D9 à 1 Hz.  
Malheureusement la LED D9 n'est pas reliée directement au MCU.  
Mais vous êtes grand maintenant, à vous de trouver une solution !



Une petite relecture du schéma s'impose.

# Chapitre IV


## Easy

	Exercice : 01
Compteur binaire (encore?)	
Dossier de rendu : <i>ex01/</i>	
Fichiers à rendre : <b>Makefile</b> , <b>*.c</b> , <b>*.h</b>	
Fonctions Autorisées : <b>avr/io.h</b> , <b>util/delay.h</b>	

Super ! Vous avez réussi à utiliser U4 en mode output. Maintenant on va aussi tester les inputs.

Vous devez écrire un programme qui :

- chaque fois que vous pressez le bouton SW3, incrémente un compteur ;
- et affiche en permanence sa valeur sur les LEDs D9 D10 D11 en binaire.


	Exercice : 02
Afficher un nombre	
Dossier de rendu : <i>ex02/</i>	
Fichiers à rendre : <b>Makefile</b> , <b>*.c</b> , <b>*.h</b>	
Fonctions Autorisées : <b>avr/io.h</b> , <b>util/delay.h</b>	

Maintenant on va s'intéresser aux 7 segments.

- Vous devez écrire un programme qui affiche sur le digit le plus à droite "2".
- Les 3 autres digits ne doivent pas être allumés.


# Chapitre V

## Medium

	Exercice : 03
Père castor	
Dossier de rendu : <i>ex03/</i>	
Fichiers à rendre : <b>Makefile</b> , <b>*.c</b> , <b>*.h</b>	
Fonctions Autorisées : <b>avr/io.h</b> , <b>util/delay.h</b>	

Vous devez écrire un programme qui affiche un chiffre sur le digit le plus à droite.

- Ce chiffre doit s'incrémenter chaque seconde.
- Quand le chiffre dépasse 9 il retourne à 0.
- Les 3 autres digits ne doivent pas être allumés.

	Exercice : 04
Agent double	
Dossier de rendu : <i>ex04/</i>	
Fichiers à rendre : <code>Makefile</code> , <code>*.c</code> , <code>*.h</code>	
Fonctions Autorisées : <code>avr/io.h</code> , <code>util/delay.h</code>	

Maintenant vous allez devoir afficher "42" sur les digits les plus à droite.

- Ça paraît plutôt simple comme ça et pourtant !
- Les 2 autres digits ne doivent pas être allumés.




Les 2 nombres doivent être visibles en même temps par un être humain sans clignotement "**visible**".




# Chapitre VI

## Very hard ( :

	Exercice : 05
Le bus magique	
Dossier de rendu : <i>ex05/</i>	
Fichiers à rendre : <b>Makefile</b> , <b>*.c</b> , <b>*.h</b>	
Fonctions Autorisées : <b>avr/io.h</b> , <b>util/delay.h</b>	

- Vous devez écrire un programme qui affiche un nombre sur les 4 digits.
- Ce nombre doit s'incrémenter chaque seconde.
- Quand le nombre dépasse 9999 il retourne à 0.

	Exercice : 06
Twitter	
Dossier de rendu : <i>ex06/</i>	
Fichiers à rendre : <i>Makefile, *.c, *.h</i>	
Fonctions Autorisées : <i>avr/io.h, util/delay.h</i>	

- Vous allez devoir récupérer la valeur de RV1 sur 10 bits.
- Et l'afficher sur le 7 segments.



Quand l'ADC renvoie 0 vous devez afficher "0000"  
Quand l'ADC renvoie 42 vous devez afficher "0042"  
Quand l'ADC est au max vous devez afficher "1023"