



Piscine embarquée

Module07 : EEPROM

contact@42chips.fr

Résumé: Because writing too many times at the same spot is what made cars crash

Version: 1

Chapitre I

Préambule

Une EEPROM (Electrically Erasable Programmable Read-Only Memory) est un type de mémoire non-volatile qui peut être utilisée pour stocker des données sur un microcontrôleur.

Elle peut être écrite et effacée plusieurs fois et conserve ses données lorsque l'alimentation est coupée.

Les EEPROM sont utiles pour stocker des données qui doivent être conservées même lorsque le microcontrôleur n'est pas alimenté, telles que des paramètres de configuration ou des données d'étalonnage.

Elles sont plus lentes et ont une capacité moindre que d'autres types de mémoire, tels que la SRAM ou la mémoire flash, mais sont toujours utiles dans une variété d'applications.

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

Tutoriel


	Exercice : 00
Mais quel bordel !	
Dossier de rendu : <i>ex00/</i>	
Fichiers à rendre : <i>Makefile</i> , <i>*.c</i> , <i>*.h</i>	
Fonctions Autorisées : <i>avr/io.h</i> , <i>avr/eeprom.h</i>	

Écrivez un programme qui affiche l'état de toute l'EEPROM sur l'UART en format hexdump.

```
0000000 6f43 7262 3a61 5020 4948 4c4c 5049 2045
0000010 2121 2021 654a 7320 6961 2073 756f 7420
0000020 2075 6574 6320 6361 6568 2073 0a21 6956
0000030 6e65 2073 6369 2069 7571 2065 656a 7420
0000040 2065 7562 6574 6520 636e 6c75 656c 2120
0000050 2121 500a 6968 6c6c 7069 3a65 5420 2061
0000060 7567 7565 656c 2120 2121 000a 0000 0000
...
```



Ne supposez pas que l'EEPROM est initialisée à 0 lorsque vous n'avez pas encore écrit dedans.

	Exercice : 01
Sniper!	
Dossier de rendu : <i>ex01/</i>	
Fichiers à rendre : <i>Makefile, *.c, *.h</i>	
Fonctions Autorisées : <i>avr/io.h, avr/eeprom.h</i>	



Le nombre de cycles d'écritures des EEPROM sont limités, mais pas la lecture.

Écrivez un programme en ligne de commande sur l'UART.


Quand on lui donne une adresse valide sur l'EEPROM en hexadécimal il y écrit un octet en hexadécimal puis affiche le contenu de l'EEPROM avec l'octet en rouge.

Si jamais l'octet avait déjà cette valeur il ne doit rien se passer.

```
00000010 7f
00000000 4a6f 686e 2046 6974 7a67 6572 616c 6420
00000010 7f65 6e6e 6564 792c 2064 6974 204a 6163
00000020 6b20 4b65 6e6e 6564 792c 0a63 6f6d 6d75
00000030 6ec3 a96d 656e 7420 6170 7065 6cc3 a920
00000040 4a6f 686e 2046 2e20 4b65 6e6e 6564 7920
00000050 6574 2070 6172 2073 6573 2069 6e69 7469
00000060 616c 6573 204a 464b 2c0a 6ec3 a920 6c65
00000070 2032 3920 6d61 6920 3139 3137 20c3 a020
00000080 4272 6f6f 6b6c 696e 6520 284d 6173 7361
00000090 6368 7573 6574 7473 290a 6574 206d 6f72
000000a0 7420 6173 7361 7373 696e c3a9 206c 6520
000000b0 3232 206e 6f76 656d 6272 6520 3139 3633
000000c0 20c3 a020 4461 6c6c 6173 2028 5465 7861
000000d0 7329 0a00 0000 0000 0000 0000 0000 0000
...
```

Chapitre IV

This is not EEPROMalloc

	Exercice : 02
Um Ackchyually	
Dossier de rendu : <i>ex02/</i>	
Fichiers à rendre : <i>Makefile, *.c, *.h</i>	
Fonctions Autorisées : <i>avr/io.h, avr/eeprom.h</i>	



Il est fortement conseillé d'utiliser des nombres magiques pour signifier la présence de vos données dans l'EEPROM. Préférez aussi utiliser un nombre magique qui n'est pas dans la table ASCII standard. Par exemple 7f.

Écrire une interface en ligne de commande sur le port UART de votre microcontrôleur.

Elle peut stocker une paire de clé/valeur de chaînes de caractères d'une longueur maximum de 32 caractères ASCII standard qui ne doivent pas être perdues lors du redémarrage.

Elle peut prendre 4 commandes :

- **READ** : Prends une clé, récupère la valeur associée et l'affiche. Si la paire n'existe pas, retourne **empty**.
- **WRITE** : Prends une clé et une valeur, tente de les stocker dans l'espace restant de l'EEPROM sans écraser les paires qui n'ont pas été supprimés. Si ça a fonctionné retourne l'adresse en hexadécimal sinon retourne **no space left**. Si la paire existe déjà, retourne **already exists**.
- **FORGET** : Prends une clé, supprime la paire clé/valeur si elle est trouvée. Sinon retourne **not found**.
- **PRINT** : Affiche le contenu de l'EEPROM en format hexdump -C

```
> READ "lol"
empty
```

```
> WRITE "lol" "je sais pas"
done
> READ "lol"
"je sais pas"
> PRINT
00000000 1f4c 6120 7069 7363 696e 6520 656d 6265 | La piscine embel
00000010 6464 6564 2063 2765 7374 2074 726f 7020 | dded c'est trop |
00000020 6269 656e 2021 0a00 0000 0000 0042 0000 | bien !. B |
...
```