

Electronics Project

Day 08 : SPI

contact@42chips.fr

Résumé: The SPI Who Loved Me

Chapitre I

Préambule

"James Bond : Casino Royale" est un film d'espionnage sorti en 2006, basé sur le premier roman de la série James Bond écrite par Ian Fleming. Le film est réalisé par Martin Campbell et met en vedette Daniel Craig dans le rôle de James Bond.

L'histoire commence lorsque Bond est envoyé en mission pour affronter Le Chiffre, un banquier de terroristes internationaux. Bond doit empêcher Le Chiffre de remporter un tournoi de poker à haut risque au Casino Royale à Montenegro, afin de priver les terroristes de leur financement.

Bond est aidé par Vesper Lynd, une agente du gouvernement qui est responsable de la gestion de l'argent de la mission. Cependant, Bond et Vesper se retrouvent bientôt confrontés à des dangers mortels alors que leur mission se complique.

Le film est connu pour ses scènes d'action intenses et sa représentation brutale de James Bond, qui diffère des interprétations précédentes du personnage. Le film a été un succès critique et commercial, et a relancé la franchise James Bond avec Daniel Craig dans le rôle principal.

"James Bond : Casino Royale" est le meilleur Bond car il est rempli d'action intense, de scènes captivantes et de la représentation crue de Bond par Daniel Craig.

C'est un film qui a relancé la franchise avec brio et a remporté un grand succès critique et commercial.

#007 #CasinoRoyale #DanielCraig #BestBond

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.
- 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.
- 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

Red Eye

	Exercice : 00
SPI Rouge	
Dossier de rendu : <i>ex00/</i>	
Fichiers à rendre : <code>main.c</code>	
Fonctions Autorisées : <code>avr/io.h</code> , <code>util/delay.h</code> , <code>avr/interrupt.h</code>	

- Écrire un programme qui initialise le SPI en maître et qui allume la LED D9 en rouge.










Exercice : 01

Blue Squadron Centurion

Dossier de rendu : *ex01/*Fichiers à rendre : **Makefile**, **rgb.c**Fonctions interdites : **Aucune**

- Vous vous souvenez de l'exercice 1 du day 06 ? On recommence !
- Vous devrez écrire un programme qui contrôle la LED RGB D6.
- Voici un tableau des couleurs qui doivent s'afficher successivement.

nom	R#	G#	B#	couleur
rouge	ff	0	0	
vert	0	ff	0	
bleu	0	0	ff	
yellow	ff	ff	0	
cyan	0	ff	ff	
magenta	ff	0	ff	
white	ff	ff	ff	

- En boucle et en changeant de couleur toutes les secondes.



Exercice : 02

Cylon Centurion


Dossier de rendu : *ex02/*Fichiers à rendre : **main.c**Fonctions Autorisées : **avr/io.h, util/delay.h, avr/interrupt.h**

Écrivez un programme qui allume successivement les LEDs D6 à D8.

- La période totale d'un cycle devra être de 1 seconde.
- la durée d'allumage de chaque LED devra être de 250ms.

Chapitre IV

You throw quite a party.


	Exercice : 03
Imperious Leader	
Dossier de rendu : <i>ex03/</i>	
Fichiers à rendre : main.c	
Fonctions Autorisées : avr/io.h , util/delay.h , avr/interrupt.h	

Lisez la valeur de RV1 avec votre ADC.

- Les LEDs D6 à D8 doivent s'allumer en fonction de la valeur de RV1 comme l'afficherait une jauge digitale.
 - LED D6 : 33%
 - LED D7 : 66%
 - LED D8 : 100%

Chapitre V

Welcome to the party, pal.

	Exercice : 05
IL-Series	
Dossier de rendu : <i>ex05/</i>	
Fichiers à rendre : <code>main.c</code>	
Fonctions Autorisées : <code>avr/io.h</code> , <code>util/delay.h</code> , <code>avr/interrupt.h</code>	

- Vous devez réaliser un programme qui écoute sur le port série.
- Ce programme devra modifier la couleur sur les LEDs D6, D7 ou D8 en fonction de la valeur envoyée par l'entrée standard.
- Vous devrez afficher une couleur HEX RGB au format `#RRGGBBDX`.
- `DX` correspondra à la valeur de la LED que vous souhaitez modifier (D6/D7/D8).
- Si le programme lit `#FULLRAINBOW` sur l'entrée standard, les LEDs devront s'allumer selon une fonction `wheel()` que vous aurez réalisé pendant l'exercice 2 du day 06.
- Le programme devra afficher un message d'erreur si le paramètre envoyé sur l'entrée standard est illisible avec l'usage.

```
#000000
#35aaaa
#221299
...
#FULLRAINBOW
```




Exercice : 05

Lucifer

Dossier de rendu : *ex05/*Fichiers à rendre : **main.c**Fonctions Autorisées : **avr/io.h, util/delay.h, avr/interrupt.h**

- Écrire un programme qui devra allumer les 3 LEDs D6, D7 et D8, et qui réponde aux problématiques suivantes :
 - le potentiomètre RV1 permet de changer la valeur d'une des 3 couleurs primaires RGB.
 - le bouton SW2 permet de valider la valeur du potentiomètre, et de changer de couleur primaire.
 - le bouton SW3 permet de passer d'une LED à la suivante.