

Electronics Project

Day 07 : Analogique

contact@42chips.fr

Résumé: L'analogique c'est pas automatique

Chapitre I

Préambule

La colophane (rosine) est le résidu solide obtenu après distillation de la térébenthine.

Substance récoltée à partir des arbres résineux et en particulier les pins (le genre *Pinus*) par une opération que l'on appelle le gemmage.



Le nom vient de Kolophôn, une cité grecque antique de l'Asie mineure où l'on produisait cette substance.

La colophane est solide et cassante à température ambiante.

Sa couleur va du jaune très clair au quasi noir en fonction de la distillation.

Sa couleur, ou grade, est définie par une échelle de lettre allant de D pour le plus foncé à X pour le plus clair.

La colophane ne fond pas mais ramollit à la chaleur, son point de ramollissement varie de 90 à 110 °C.

La colophane est composée à 90% d'un mélange d'acides organiques de la famille des diterpènes appelés acides résiniques, ces acides résiniques sont des isomères.

La colophane peut être utilisée comme flux en brasage tendre.

Les flux servent à réduire la tension superficielle de la brasure fondue et lui permettent de couler plus facilement pour recouvrir plus rapidement et efficacement les surfaces des parties à braser.

Des couches d'oxydes se formant en permanence sur les surfaces quand on les chauffe, le flux a pour but de les dissoudre et les éliminer.

Pour en faciliter l'emploi, on fabrique des fils de brasure tendre creux qui contiennent directement un ou plusieurs cœurs (âmes) de flux.

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

v=p7YXXieghto

| | |
|---|---------------|
|  | Exercice : 00 |
| The nozzle is initializing | |
| Dossier de rendu : <i>ex00/</i> | |
| Fichiers à rendre : main.c | |
| Fonctions Autorisées : avr/io.h , util/delay.h , avr/interrupt.h | |

Lisez la valeur du potentiomètre linéaire RV1 en utilisant le peripherique ADC.

- ADC doit etre configuré avec une résolution de 8 bits et AVCC comme référence.
- Affichez ensuite sa valeur au format hexadecimal toutes les 20ms sur la console.

```
...  
00  
a1  
...
```



Exercice : 01

Please wait while we calibrate, ..., the nozzle

Dossier de rendu : *ex01/*

Fichiers à rendre : **main.c**

Fonctions Autorisées : **avr/io.h**, **util/delay.h**, **avr/interrupt.h**

- Lisez le potentiomètre RV1 + la LDR (R14) + la NTC (R20)
- Affichez ensuite les valeurs au format hexadecimal toutes les 20ms sur la console.

```
...  
00, ef, ff  
00, ef, ff  
...
```

Chapitre IV

10 bit funk



Exercice : 02

Ca fait beaucoup la non ?

Dossier de rendu : *ex02/*

Fichiers à rendre : `main.c`

Fonctions Autorisées : `avr/io.h`, `util/delay.h`, `avr/interrupt.h`

- Lisez le potentiomètre RV1 + la LDR (R14) + la NTC (R20)
- Mais avec l'ADC en 10bit
- Affichez ensuite les valeurs au format hexadecimal toutes les 20ms sur la console.

```
...  
0, 128, 1023  
0, 128, 1023  
...
```



Exercice : 03

Lecture Température

Dossier de rendu : *ex03/*Fichiers à rendre : `main.c`Fonctions Autorisées : `avr/io.h`, `util/delay.h`, `avr/interrupt.h`


L'Atmega328P est capable de mesurer sa température interne. Bon c'est pas ultra précis mais on va la lire quand même.

- Lisez la valeur du capteur interne de température.
- Puis affichez-la sur la console convertie en degré Celsius.

```
...  
20  
22  
25  
...
```

Chapitre V

La vie en couleur, c'est le secret du bonheur

| | |
|---|---------------|
|  | Exercice : 04 |
| Couleur analogique | |
| Dossier de rendu : <i>ex04/</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> | |

Lisez la valeur de RV1 avec votre ADC.

- RV1 doit permettre de changer la couleur de D4 avec la fonction `wheel`
- Aussi les LEDs D1-D4 doivent afficher la valeur de RV1 comme une jauge digitale.
 - LED D1 : 25%
 - LED D2 : 50%
 - LED D3 : 75%
 - LED D4 : 100%



Avec une pointe de `musique` c'est toujours mieux !