



Alimentation Laboratoire Dirigé par un Microcontrôleur ATMega8535

Hirschmann Vitalis
Microtechnique



Lausanne, 22 février 2007

Table des matières

1	Introduction	2
2	Hardware	2
2.1	Boitier	3
2.2	Alimentation	3
2.3	Bloc relais	4
2.4	Interface utilisateur	5
3	Code	6
3.1	Utilisation	6
3.2	Interruptions	6
3.3	Structuration en modes programme	7
3.4	Programmation	8
4	Conclusion	8
5	Schéma électrique	8

1 Introduction

L'idée de cette Alimentation est d'avoir un outil de travail dans un atelier qui réunit plusieurs fonctions tel que

- source de tension 5V / 12V / et 0..12V réglé
- appareil de mesure
- générateur de signaux
- Interrupteur idéal

Donc l'alimentation laboratoire devrait simplifier le travail et mettre en place les outils les plus utilisés. La précision n'est pas trop importante, les domaines cibles sont le bricolage, la mesure d'ordre de grandeur et le test de composants et capteurs.

Ce rapport devrait donner une idée de ce qu'on peut faire en plus avec le kit robotique du club Robopoly¹

2 Hardware

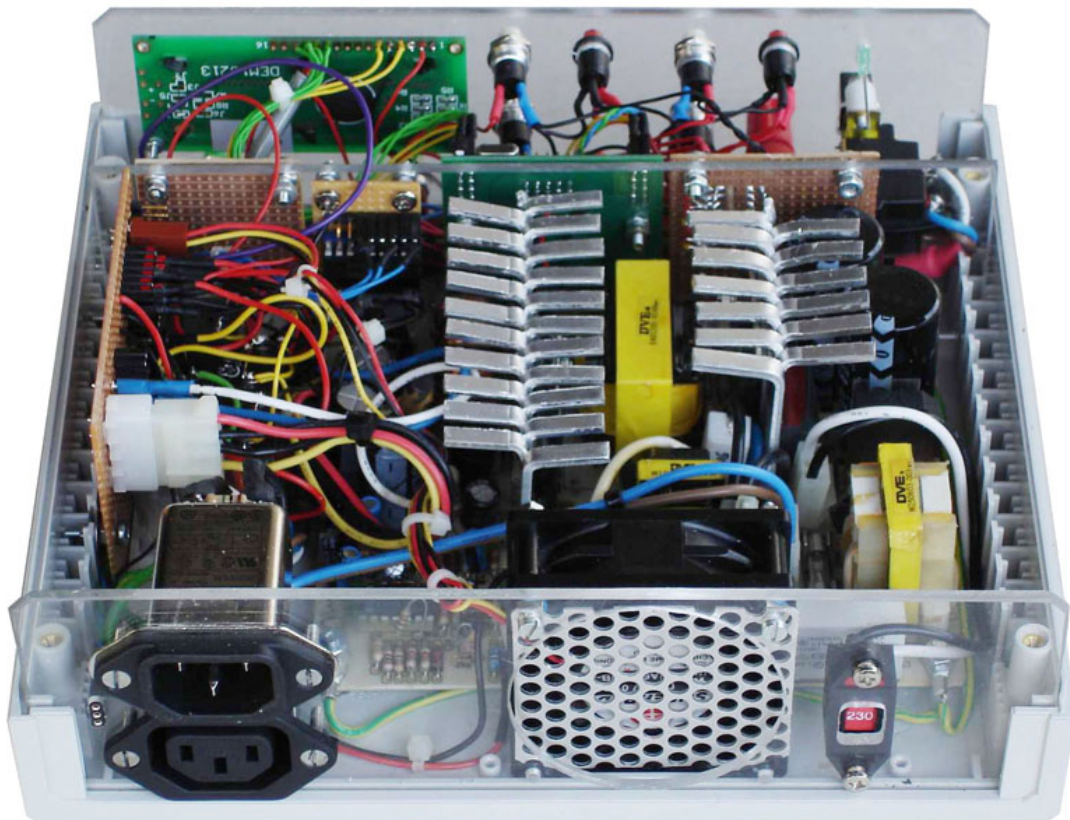


FIG. 1 – L'intérieur de l'alimentation laboratoire

¹robopoly.epfl.ch

2.1 Boîtier

Le boîtier *BOPLA* de Conrad² est un boîtier qui est modulaire en hauteur et il existent des fixation standard pour les PCB typ Euroformat. Il est livré sans plaques avant et arrière. Ceux sont réalisés en plexiglas pour son grand facilité de traitement et l'esthétique.



FIG. 2 – Le boîtier *BOPLA*

2.2 Alimentation ⚡ Danger de mort, haute tension ⚡

Pour les travaux avec les composants haute tension : Il est absolument nécessaire de connecter les points de masse de l'alimentation avec le fil de masse pour des raisons de sécurité. Les vis qui fixent l'alimentation et qui sortent du boîtier (et peuvent être touchés) doivent aussi être mis à la masse. Toutes les soudures doivent être isolés à l'aide de gaine thermoretractable si possible.

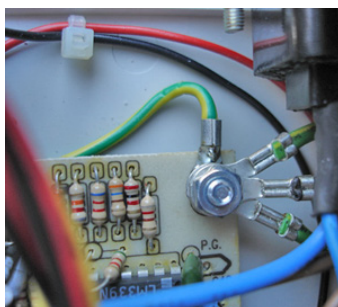


FIG. 3 – Mise à la masse d'une vis qui fixe l'alimentation

Une solution possible pour l'alimentation de base serait un transformateur avec le circuit standard de redressage et stabilisation. Les désavantages principales de cette réalisation seraient le poids élevé et sa taille.

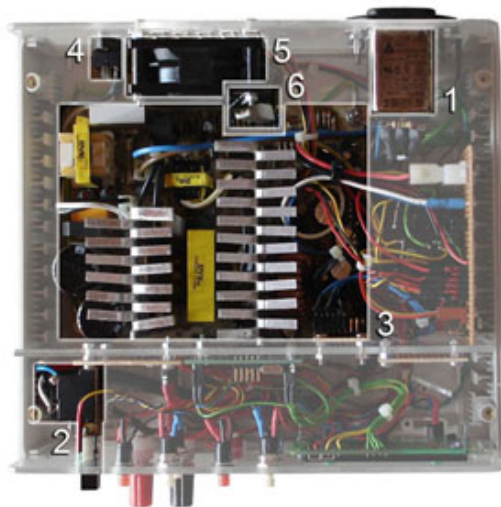
Une autre possibilité est de démonter une alimentation PC et réutiliser ce circuit avec les connections qui vont avec.

Les avantages sont le petit poids et la compatibilité avec notre projet : Sorties de 5V et 12V avec une puissance largement suffisante pour une alimentation laboratoire. Un désavantage est que les alimentations PC ne fonctionnent souvent pas en absence d'une charge. Celle-là doit donc être 'simulée' avec une résistance de puissance. Les alimentations récents possèdent souvent un connecteur de mise en marche qui doit être activé.

La plupart des alimentations PC sont équipés avec un ventilateur qui est commandé par un capteur de température. Au cas échéant le ventilateur peut être commandé à l'aide d'une sortie

²www.conrad.ch

PWM du microcontrôleur ou simplement être branché à 12V ou 7V ce qui est normalement suffisant.



1. Connecteur / Prise 230V
2. Switch principal On / Off 230V
3. Bloc Alimentation PC
4. Switch 230V / 110V
5. Ventilateur 12V
6. Résistance de charge

FIG. 4 – Composants de l'alimentation PC

2.3 Bloc relais

Pour simplifier l'interface utilisateur j'ai réalisé une prise (connecteurs 'banane' 4mm) pour 5V / 12V / *réglé* et une prise pour les sorties du microcontrôleur. Cette configuration oblige de pouvoir changer la source pour la sortie, ce qui est réalisé avec des relais. Le bloc de relais est commandé par un ULN 2803³ qui est branché au microcontrôleur. Le ULN contient déjà les diodes de protection pour les relais.

Le bloc relais contient trois relais en série, un pour changer entre 5V et 12V, le deuxième entre 12V et 0..12V *réglé* et le troisième pour déconnecter la sortie de l'alimentation. Le dernier est déconnecté en état de repos (Mesure de sécurité).

³ULN2803A SGS Thomson, eight darlington arrays



1. PCB de sortie / bloc relais
2. PCB avec ULN 2803

FIG. 5 – Bloc relais et PCB de sortie

2.4 Interface utilisateur

Toutes les éléments de contrôle, le LCD et le microcontrôleur sur son PCB⁴ sont séparés de la partie haute tension par une plaque plexiglas en raison de sécurité.

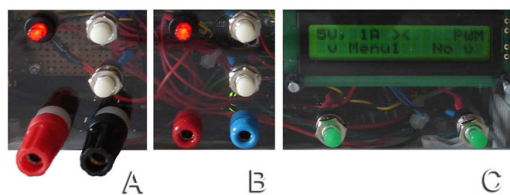
Pour chaque sortie (alimentation et fonctions) il y a trois boutons. Deux avec les fonctions *up* et *down* et le troisième pour activer ou désactiver la sortie. Il est donc possible de brancher complètement un circuit et après allumer la sortie.

Au dessous du LCD se trouvent encore deux boutons pour changer le programme pour les sorties.



1. PRISME avec microcontrôleur
2. Boutons et sorties
3. LCD à deux lignes

⁴ATMega8535 sur le kit PRISME de Robopoly



- A** Sortie *alimentation* avec boutons
blanc : *up* et *down*
rouge : *on / off*
- B** Sortie *fonctions* avec boutons
- C** LCD et contrôle des programmes,
voir 3.1

FIG. 6 – Interface utilisateur et microcontrôleur

3 Code

3.1 Utilisation

Avec le premier bouton programme **P1** on change entre Menu1 et Menu2, ce qui correspond aux sorties *alimentation* et *fonctions* (Fig 6). Avec le deuxième bouton programme **P2** on choisit la fonction pour la sortie active.

Les boutons *up* et *down* (**ud1** et **ud2**) aident à choisir une tension ou à ajuster une fréquence pour les sorties correspondantes. Selon la fonction choisit ils sont actifs ou non.

Les sorties peuvent être calibrés tout en étant éteints. Tant que le bouton rouge **o1** est allumé il n'y a aucune tension ou action sur la sortie *alimentation*. En appuyant le bouton il s'éteint et la sortie devient active.

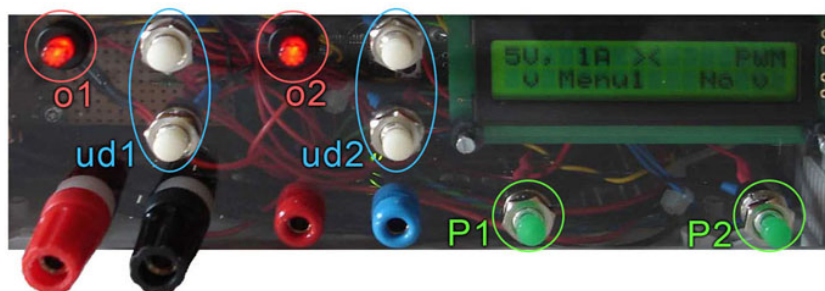


FIG. 7 – L'interface avec les boutons

3.2 Interruptions

Les deux boutons **o1** et **o2** sont connectés aux interruptions externes du microcontrôleur (*int0* et *int1*) pour assurer un fonctionnement absolument fiable. En cas de problème avec le circuit branché on peut donc le débrancher en tout moment en appuyant le bouton rouge.

Dans la routine d'interruption le programme inverse l'état du relais de la sortie, soit ouvert soit fermé.

Listing 1 – Routine d'interruption

```
int_0 :
    in        _sreg , SREG

    INVP      PORTA, Relon           ; on/off Relais de sortie

    rcall     lcd_busy
    sbic      PORTA, Relon
    rjmp      out_on

    cbi       PORTC, LED1           ; allumer LED
```

```

WAITMS 700
rjmp    ret_1

out_on:
sbi      PORTC, LED1          ; éteindre LED
WAITMS 500

ret_1:
rcall    lcd_startup
OUTI     GIFR, (1<<INTF0)     ; pas d'interruption durant la routine
out      SREG, _sreg
reti

```

3.3 Structuration en modes programme

Seulement les deux boutons **o1** et **o2** génèrent des interruptions. L'état des autres boutons est analysé dans une boucle infinie. Le bouton programme **P1** incrémente un compteur (**program1**) qui est argument pour une macro de *lookup table* (**J_TABLE**). Le programme saute ainsi à la partie programme correspondant à la fonction choisie.

Le fanion *T* du registre *SREG* aide à contrôler si la sortie *alimentation* ou *fonctions* est active. Donc le fanion est testé au début de la fonction **Prog_1** pour travailler sur la bonne sortie.

Le nombre de programmes à disposition peut être élargit selon les besoins. Il faut adapter le nombre de programmes **anz_prog1** et la table **menu1_table**, et enfin créer les fonctions correspondants aux extensions.

Listing 2 – Routine du bouton **o1**

```

Prog_1:                                ; Menu
brts     T_1                           ; -> sortie-2

T_0:                                       ; sortie 1 actuel
lds      b0, program1
inc      b0
cpi      b0, anz_prog1

brlo     p_depasse_1
ldi      b0, 0
p_depasse_1:
sts      program1, b0

lookup_1:
J_TABLE b0, menu1_table
rjmp     fin_prog1

T_1:                                       ; sortie 2 actuel
lds      b0, program2
inc      b0
cpi      b0, anz_prog2

brlo     p_depasse_2
ldi      b0, 0
p_depasse_2:
sts      program2, b0

lookup_2:
J_TABLE b0, menu2_table

fin_prog1:

```



```
rcall    lcd_startup  
  
ret
```

Listing 3 – Tables pour la macro J_TABLE

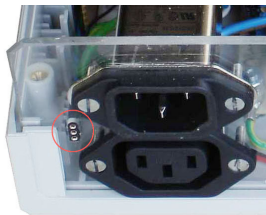
```
menu1_table:  
.dw     prog10, prog11, prog12  
  
menu2_table:  
.dw     prog20, prog21, prog22
```

3.4 Programmation

Entre la programmation série RS232 et USB j’ai choisit le USB pour les raisons suivantes :

- grande vitesse de charge du code sur le chip
- petite dimension du connecteur, possibilité d’intégration du port USB dans le boîtier
- compatibilité du port USB
- possibilité de transmission de données de mesure vers le PC à travers le PrismeMonitor⁵

Pour le programmeur USB⁶ on a besoin d’un *bootloader* qui doit être chargé la première fois à l’aide d’une interface RS232. Le connecteur est intégré dans la plaque arrière donc il n’est pas nécessaire d’ouvrir le boîtier pour la programmation.



Le microcontrôleur est connecté au programmeur USB avec les trois lignes suivantes :

1. Tx
2. Rx
3. GND

FIG. 8 – Le connecteur pour la programmation USB

4 Conclusion

Ce document n’est pas un mode de construction mais veut donner des idées pour la réalisation de projets similaires. En cas d’incendie, mort ou dommages aux tiers à cause de bricolage non contrôlé je dégage toute responsabilité⁷

Pour toute remarque adressez-vous à microbotique@gmail.com

5 Schéma électrique

Le schéma n’est pas tout à fait complet (le microcontrôleur se trouve sur le PCB du PRISME) mais les entrées et sorties correspondent au montage.

⁵© by Gilbert Pascal, disponible sur robopoly.epfl.ch

⁶© by Ambühl Alessandro, robopoly.epfl.ch

⁷Numéros d’urgence : Pompiers 118, Ambulance 144

