

Projet : Persistence of Vision (POV) – affichage d'un message lumineux par Atmega32U4

J.-M Friedt

FEMTO-ST/département temps-fréquence

`jmfriedt@femto-st.fr`

`transparents à jmfriedt.free.fr`

15 mai 2015

Objectifs du projet

Objectifs

Organisation

Stockage de la police

Résultats

Conclusion

- **Objectif** : utiliser la persistance rétinienne pour afficher un message au moyen d'un réseau de LEDs alignées en mouvement
- Concept familier pour l'affichage 7-segments/LCD ou les écrans cathodiques, mais historiquement utilisé avant l'avènement du cinéma (folioscope et zootrope en 1834 ...)
- Domaine d'application : acquisition et traitement de données sur support mobile, nécessité d'alimenter sur pile un dispositif qui sera en rotation (roue, rotor de moteur ou de turbine).

État de l'art

- Principe : une barrette de LEDs en mouvement affiche un motif qui évolue pour afficher un message
- Contraintes : synchroniser sur la vitesse de rotation du support
- Projet classique faisant intervenir du GPIO, input capture, interruptions dans un cadre ludique

Exemples de démonstrations dans la littérature :

- SpokePOV Persistence of Vision for your Bike !¹
- R.P. Paul, G.B. Rathod, V.R. Trivedi, P.V. Thakkar, *Persistence of Vision Control Using Arduino*, I.J. Intelligent Systems and Applications, 2014, **01**, pp.102-111 (10.5815/ijisa.2014.01.11)²
- Spinning 16-led POV display using Teensy 2.0³
- S. Monk, *15 Dangerously Mad Projects for the Evil Genius*, Mc Graw Hill (2011), chap. 8⁴

1. <http://www.ladyada.net/make/spokepov/>

2. <http://www.mecs-press.org/ijisa/ijisa-v6-n1/IJISA-V6-N1-11.pdf>

3. https://github.com/amcinnes/pov_display

4. https://www.youtube.com/watch?v=U1q33B_GnLo

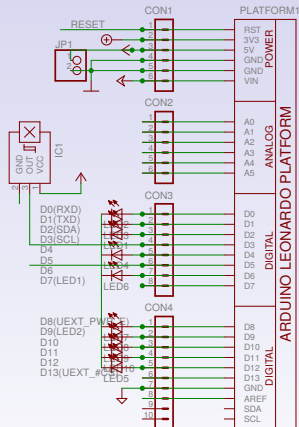
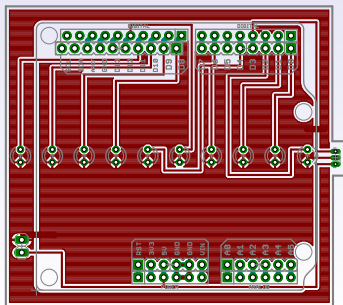
Organisation

10 séances de 3 heures :

- ① comprendre le sujet + bibliographie
- ① choix technologique : prise en main du microcontrôleur et des fonctions de base, planification du travail
- ② conception du circuit (schéma) et commande des composants
- ③ suite de ② (routage), test de la sonde à effet Hall, tension d'alimentation et distance de déclenchement
- ④ réalisation d'un premier prototype permettant de se familiariser avec le concept de POV : un motif périodique (chenillard) est affiché tandis que la plateforme est en rotation – relation entre GPIO et position
- ⑤ programme permettant d'afficher un message – table de caractères (**difficulté : dépassement de RAM**)
- ⑥ déclenchement du motif sur un point fixe de référence (INT0)
- ⑦ affichage sur timer au lieu d'afficher sur délai fixe (boucle vide)
- ⑧ ajustement de la vitesse du timer afin d'étaler le message sur toute la surface
- ⑨ finalisation rapport et présentation orale

Circuit imprimé et première démonstration

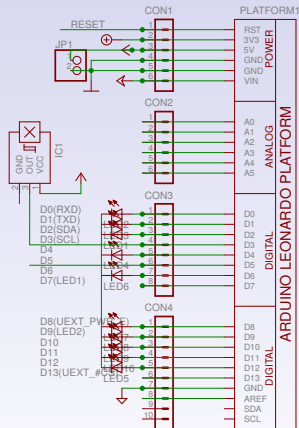
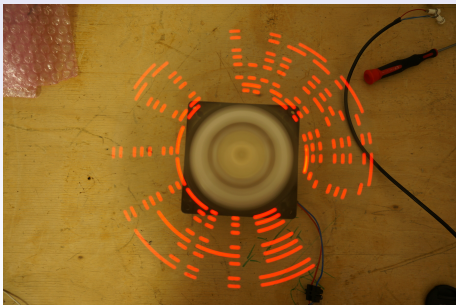
- logiciel s'apparentant au chenillard
- Olimexino-32U4 respecte la nomenclature Arduino et mélange les ports $\Rightarrow$ LUT reliant chaque LED à son port/position
- montage mécanique de mise en rotation



Circuit imprimé : CADSoft Eagle (www.cadsoft.de)

Circuit imprimé et première démonstration

- logiciel s'apparentant au chenillard
- Olimexino-32U4 respecte la nomenclature Arduino et mélange les ports $\Rightarrow$ LUT reliant chaque LED à son port/position
- montage mécanique de mise en rotation



Circuit imprimé : CADSoft Eagle (www.cadsoft.de)

Table d'allocation des caractères

- 10 LEDs (0x000-0x3FF) encodées sur un short
- 10×5 pixels/caractère
- caractères affichables de la table ASCII : 32-127
- 95 caractères×2 octets/colonne×5 colonnes/caractère=950 octets
- travail incrémental : ajouter les caractères si nécessaire

```
short c[475]= // on a 2.5 KB RAM
{0x000,0x000,0x000,0x000,0x000, // space
 0x000,0x3fb,0x3fb,0x000,0x000, // !
 0x000,0x000,0x000,0x000,0x000, // "
 0x048,0x3ff,0x048,0x3ff,0x048, // #
...
0x3ff,0x203,0x206,0x20e,0x3f9, // 0
0x000,0x100,0x3ff,0x000,0x000, // 1
0x107,0x209,0x211,0x221,0x1c1, // 2
0x202,0x221,0x221,0x252,0x18c, // 3
0x3e0,0x020,0x020,0x3ff,0x020, // 4
0x3e1,0x221,0x221,0x212,0x20c, // 5
0x0fe,0x121,0x221,0x211,0x00e, // 6
...
}
```



Problème de gestion de mémoire

Seulement 2,5 KB de RAM, mais table de caractères est statique donc on peut la mettre en flash (32 KB) ... sauf sur une architecture Harvard.

Objectifs

Organisation

Stockage de la police

Résultats

Conclusion

- Sur MSP430 (Von Neumann), sans puis avec const :

```
msp430-gcc -mmcu=msp430f1611 -O0 -o 3pov_memoire.msp430 3pov_memoire.c
msp430-size 3pov_memoire.msp430sansconst
```

text	data	bss	dec	hex	filename
112	966	2	1080	438	3pov_memoire.msp430sansconst

```
msp430-size 3pov_memoire.msp430aveconst
```

text	data	bss	dec	hex	filename
1062	16	2	1080	438	3pov_memoire.msp430aveconst

- Sur Atmega32U4 (Harvard), sans puis avec const :

```
avr-gcc -mmcu=atmega32u4 -O0 -Wall -o 3pov_memoire.avr 3pov_memoire.c
avr-size 3pov_memoire.avrsansconst
```

text	data	bss	dec	hex	filename
232	966	0	1198	4ae	3pov_memoire.avrsansconst

```
avr-size 3pov_memoire.avraveconst
```

text	data	bss	dec	hex	filename
232	966	0	1198	4ae	3pov_memoire.avraveconst

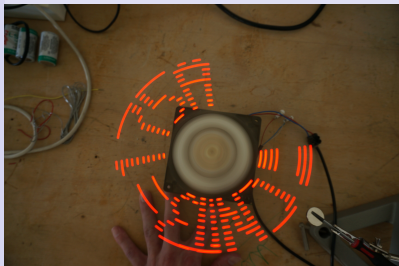
(text va en flash, data et bss en RAM)

Affichage d'un message alphanumérique

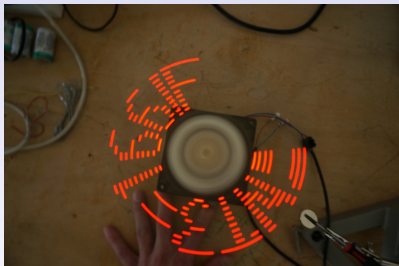
- $3000 \text{ rpm} = 50 \text{ rps} = 20 \text{ ms/tour}$, donc afficher "HELLO WORLD" = 12 chars $\Rightarrow 1.7 \text{ ms/char}$
- 5 colonnes/char et un espace = 6 colonnes/char $\Rightarrow$ chaque colonne s'affiche $1700/6=283\mu\text{s}$
- utilisation du *timer* en mode *Output Compare* pour cadencer l'affichage
- ajustement de la cadence sur vitesse de rotation (*Input Capture* v.s INT0)

Input Capture

- Déclenchement d'une interruption sur une transition sur une broche liée à un *timer*
- Étallonnage préliminaire $\Rightarrow$ cadencement à 62500 Hz=16 MHz/256
50 Hz=0x4E2, 20 Hz=0xC35, 10 Hz=0x186A, 8 Hz=0x1E84, 4 Hz=0x3D09



0x175A=96 ms=10,4 Hz

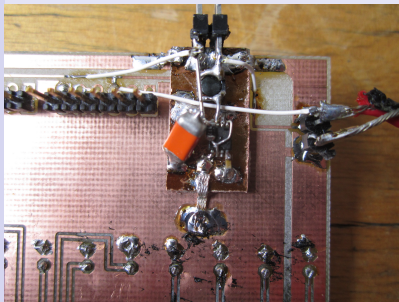


0x166F=92 ms=10,9 Hz

Attention : la sonde à effet Hall⁵ nécessite une alimentation de 5 V/5 mA, elle ne commute pas si elle est alimentée en 3,6 V.

Génération 5 V

- 1 Convertisseur DC-DC en step-up (3,6 V → 5 V)
- 2 MAX1674 : entrée entre 0,7 et 5 V, sortie fixe à 5 V (300 mA)
- 3 ici, MAX631 : entrée entre 2,5 et 5 V, sortie fixe à 5 V (50 mA)
Montage en *dead-bug* pour ne pas perdre de temps à refaire un circuit imprimé



Conclusion

Objectifs

Organisation

Stockage de la
police

Résultats

Conclusion

- 1 Projet permettant d'aborder de nombreux périphériques d'un microcontrôleur (GPIO, timer, interruption matérielle)
- 2 application embarquée nécessairement alimentée sur pile (rotation)
- 3 complexité incrémentale avec validation de chaque étape.

Perspectives

- 1 ajouter la couleur en remplaçant l'unique LED rouge par colonne par le triplet RGB ou une LED 3-couleurs^{6 7}.
- 2 solution 1-wire du WS2812 (Hackable n.6, 2015) au lieu de 10 GPIOs?

6. LED RGB 5 mm, Farnell référence 2290374, 1 euro/p

7. *Panasonic LED driver ICs POV (Persistence of Vision) illumination demo examples* à <http://channel.panasonic.com/contents/10919/>