

1 But

Mon but était d'avoir une carte minimaliste avec son petit PCB permettant des connexions directes. Évidemment elle doit être équipée d'un microcontrôleur de la série AVR.

Les premières cartes adaptées que j'ai trouvées sont les Arduino dans la version pro. Certains, sur le web, pestent contre l'appellation pro pour une carte si simple, mais je considère que c'est réellement fait pour celui qui sait s'en servir et qui rajoutera seul ce dont il a besoin : je n'ai besoin, ni que le travail soit mâché, ni de fonctions inutiles. Les non-pros comme moi, ont aussi le droit de s'en servir, m'enfin !

L'arduino pro trouvé sur IB est parfois qualifié de mini ou de nano.

C'est une petite carte qui ne comprend que ce dont j'ai besoin :

- processeur AT mega328p
- son horloge : résonateur 16MHz 0,5 %, certaines versions ont un vrai quartz, au même prix, il faut bien regarder la photo IB
- son alimentation avec un régulateur low drop-out (50mV @ 150mA), précision 1 %, coef temp 4ppm
- bouton et fonction reset
- 22 ports disponibles
- une LED de signalisation sur PB5
- une LED alimentation
- alimentation possible par Vcc direct (utilisable sur le port de programmation) ou via le régulateur

2 Achat

La version minimaliste se trouve à 11,36€ (Lx) ou 9,19 (mnx) avec port USB il se trouve à 24,95 (cd), 37,10(sc)

Dès la première page de gogolisation, I.B. pointe son nez et on le trouve à 3\$ port gratuit (il arrive 3 semaines après par un courrier de Swiss Post, Deutsche Post, ou autre poste du Liechtenizistan). L'avantage d'I.B. c'est qu'on est quasiment sûr qu'il parvient au destinataire, IB éliminant sans pitié les mauvais fournisseurs (le coincoin n'offre lui, aucune garantie anti-arnaque).

Je ne suis pas parfaitement sûr que les arduinos plus chers, soient meilleurs, ou exempts du bug de la diode.

3 La chose !

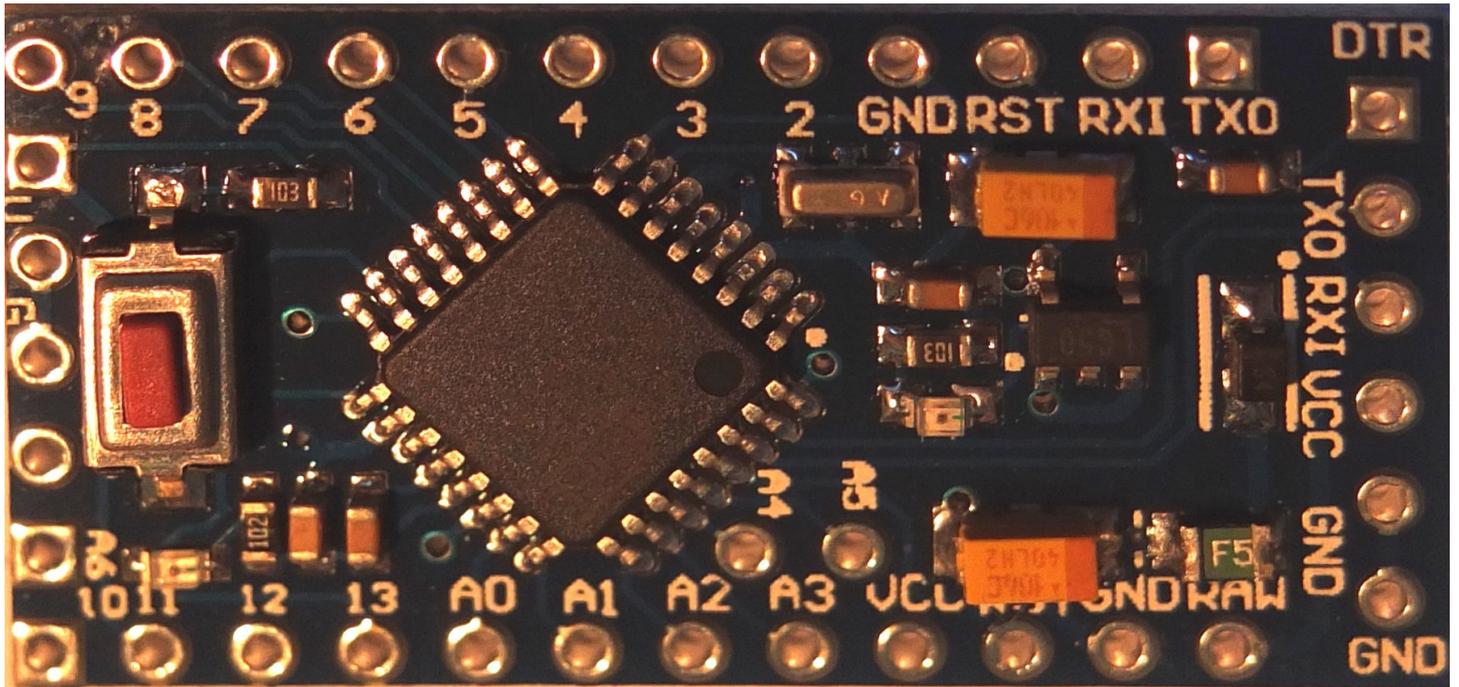
C'est une petite carte de 34 x 18 mm, un timbre-poste quoi, livré avec les broches à wrapper nécessaires.



Taille réelle

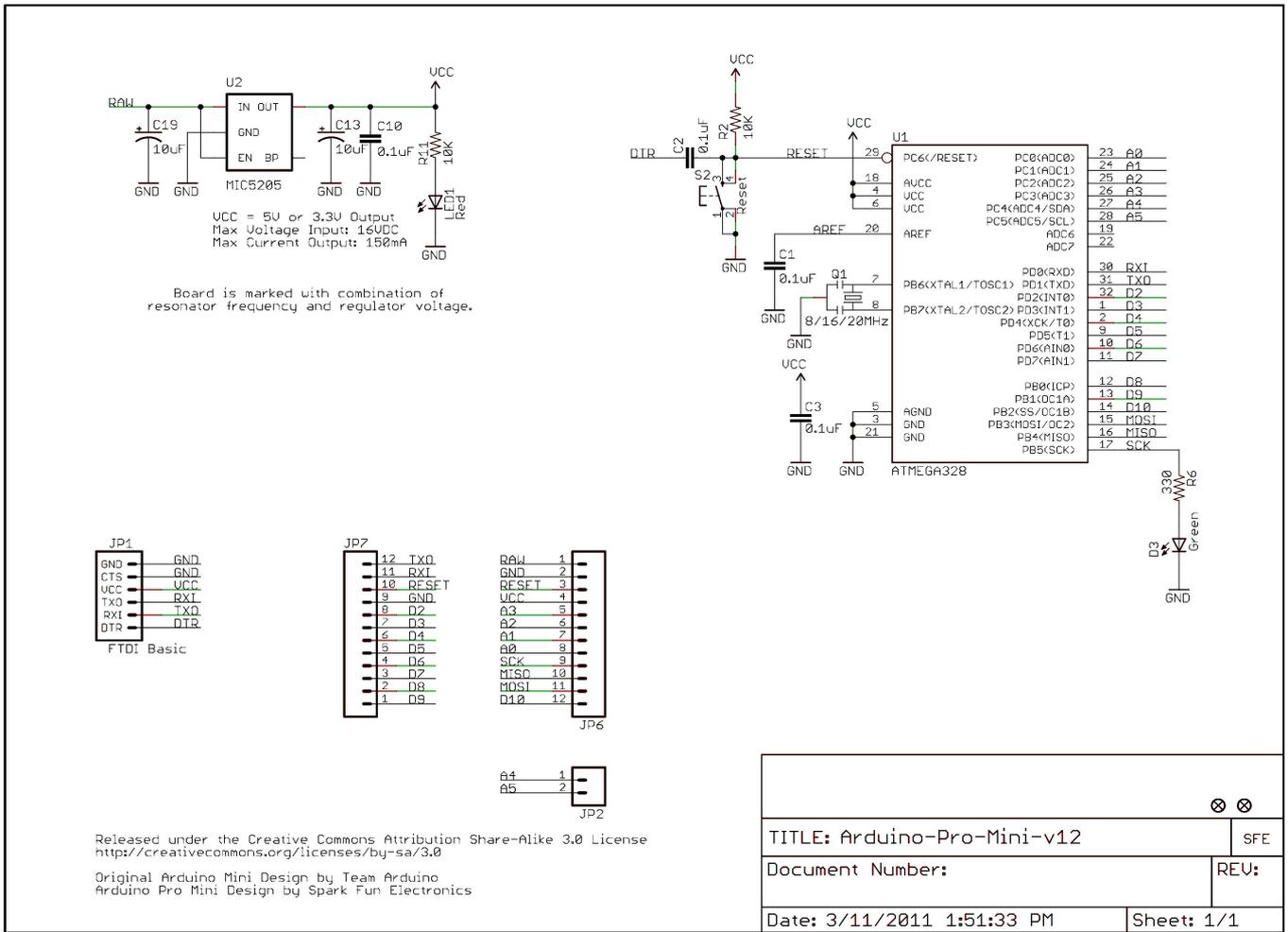


les dessous !



Il y a des différences d'implantation et de composants entre chaque série que j'ai reçue, Ce qui reste constant est l'interface extérieure : dimensions, position et fonction des broches,

Le schéma se trouve facilement sur le Web

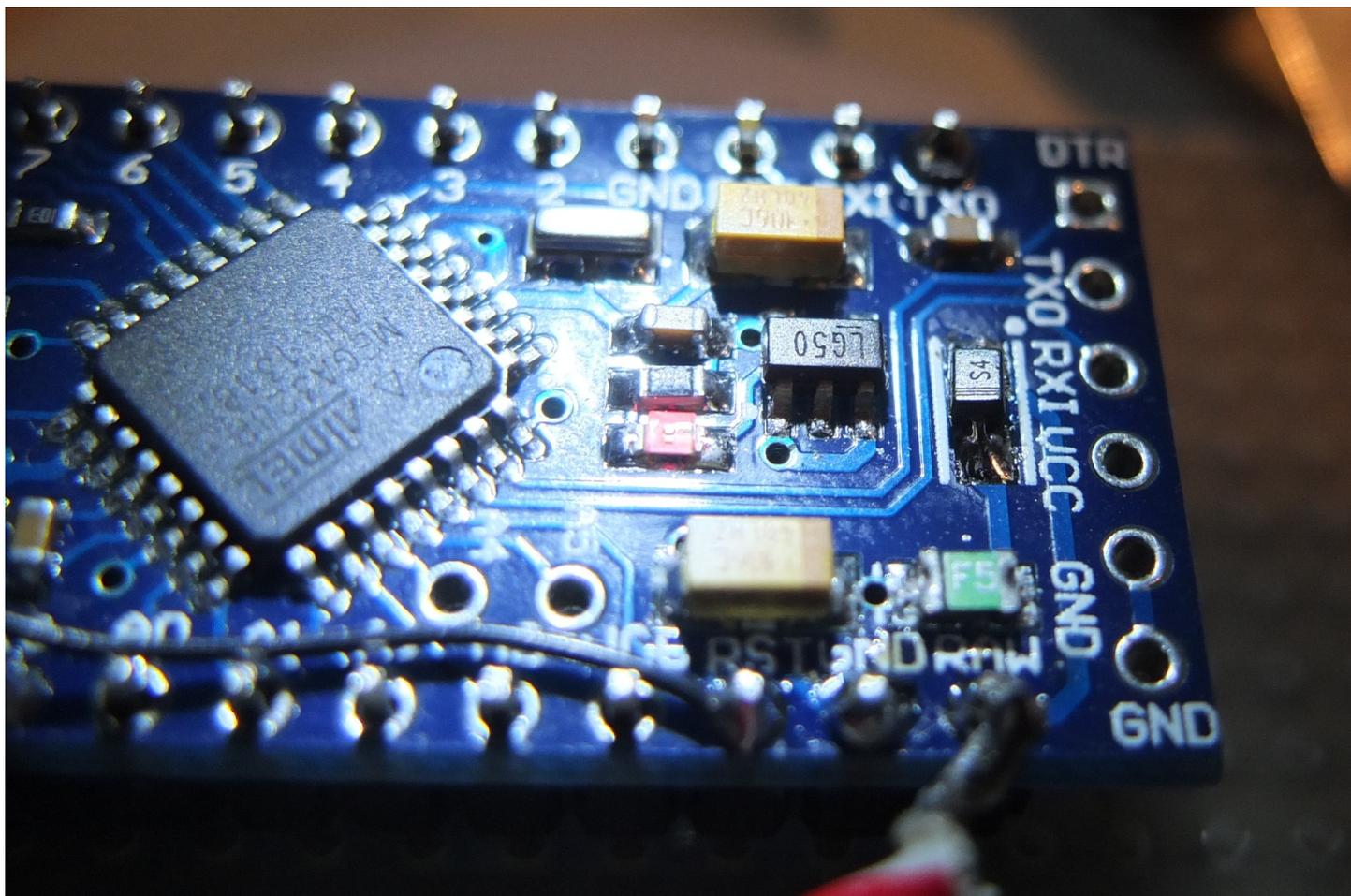


JP6	silk signal	A6	A7	GND	Vcc	Silk signal	JP7
12	10 PB2					9 PB1	1
11	11 PB3					8 PB0	2
10	12 PB4					7 PD7	3
9	13 PB5					6 PD6	4
8	A0 PC0					5 PD5	5
7	A1 PC1	A4				4 PD4	6
6	A2 PC2	A5				3 PD3	7
5	A3 PC3					2 PD2	8
4	Vcc					GND	9
3	RST					RST PC6	10
2	GND					RXI PD0	11
1	RAW					TXO PD1	12



Mon arduino ne suit pas complètement ce schéma, car ma carte possède une variante dans le circuit d'alimentation : l'entrée RAW passe au travers d'une diode marquée S4 (diode Schottky tension de chute 0,2V) et d'un élément marqué F5 (évidemment un fusible réarmable, sans doute 500mA, de résistance série 1Ω). Le hic, c'est que **la diode est montée à l'envers !** Et c'est voulu, au moins par le dessinateur : la sérigraphie a un point d'orientation de montage, et le montage de la diode, qui respecte bien ce point, ne fonctionne pas. Il suffit de redresser le redresseur (ancienne appellation des diodes) pour que tout rentre dans l'ordre. C'est juste un peu délicat, la diode mesure 1,6 x 0,8mm (boîtier taille 0603).

Les livraison suivante étaient correctes.



La diode « redressée » (marquée S4)

4 Utilisation

Pour utiliser la carte, on peut en utiliser le firmware déjà incorporé. Quand on met sous tension, la diode D3 clignote, il y a donc un programme qui tourne à bord ! C'est le « bootloader » du monde arduino. Pour utiliser ce bootloader, il faut entrer des commandes soit par port USB (d'où l'intérêt des cartes arduino avec port USB) soit par port série, disponible sur cette carte (TXO, RXI accessibles par port série interne). Ces commandes sont émises par les systèmes de développement arduino, dont je ne me sers pas. Je n'ai pas réussi à trouver la syntaxe en mode TTY.

Mon choix est d'utiliser le processeur « brut » et de le programmer par son port SPI (lignes RST, SCK, MISO, MOSI).

Si je veux me servir des bonnes performances du régulateur interne, suffisant pour l'utilisation des convertisseurs A/D 1 bits en mode ratiométrique, en tous cas meilleur que la référence bandgap interne, il faut alimenter par l'entrée RAW. J'ai rajouté une Schottky entre l'entrée de programmation et le Vcc,

ce qui permet d'utiliser les deux alimentations sans qu'elles n'interfèrent entre elles. L'entrée RAW sortira normalement une tension supérieure à la prise USB et sera prioritaire.

5 Programmation

Le programmeur USBASP V2.0, trouvé aussi sur IB à 3\$, utilise un port USB de l'ordinateur, nécessite pour fonctionner avec l'arduino, une reprogrammation de son firmware (avec un autre programmeur identique, acheté en backup, juste en cas de fausse manip destructrice sur le premier). L'autre programmeur, avec son firmware d'origine, me sert à programmer des microcontrôleurs Attiny13. **Les dernières versions que j'ai reçues fonctionnent directement !**

5.1 Programmeur

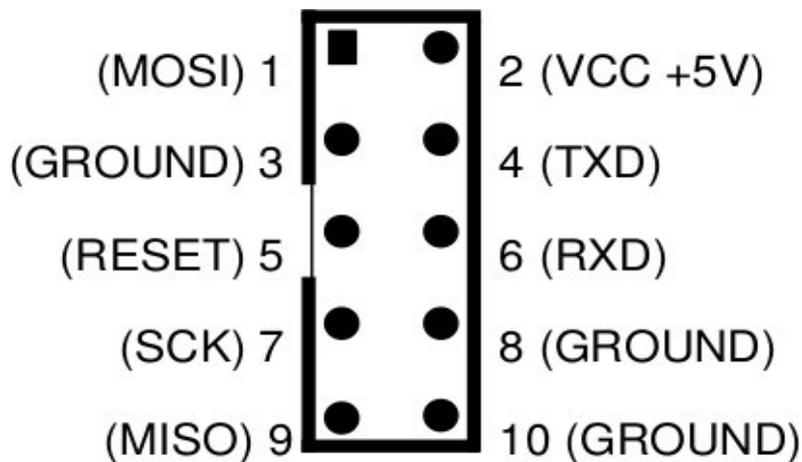
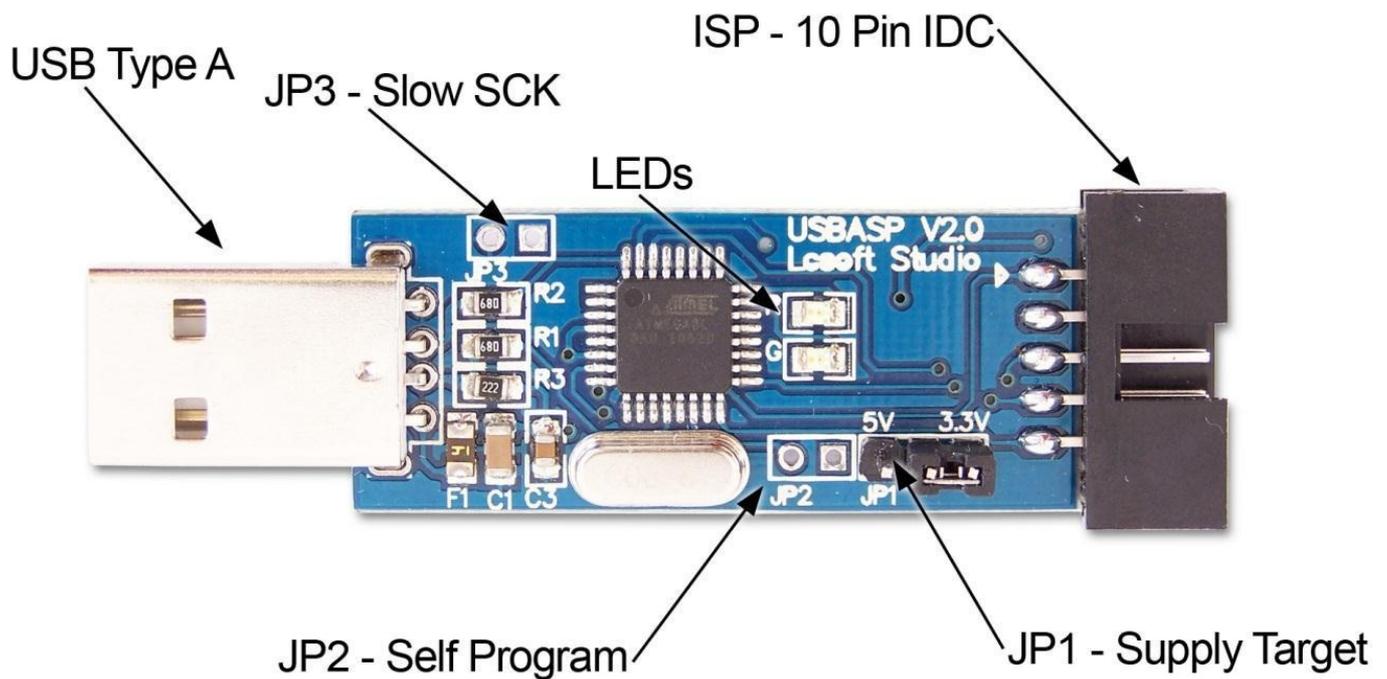
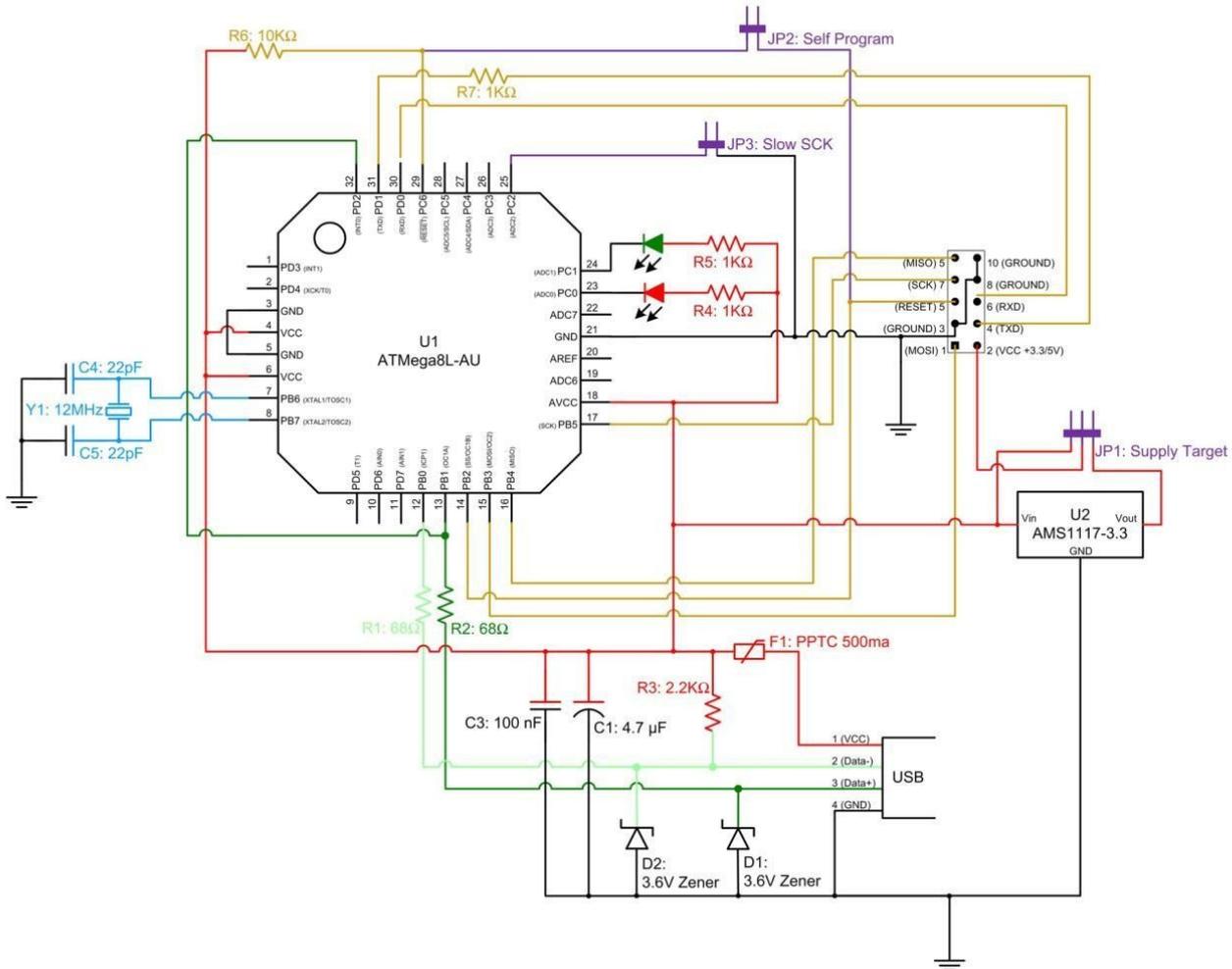


Figure 2. 10 Pin ISP pinout



Le patch se trouve chez <http://www.fischl.de/usbasp/>

Le programmeur dispose d'une sortie permettant d'alimenter le processeur à programmer, la carte arduino consommant moins une bonne nonantaine de mA avec un « hello world » minimum). J'ai rajouté une diode Schottky entre le programmeur et l'arduino, diode non nécessaire si la carte ne tourne que via le programmeur, mais c'est une sécurité car mon montage va être soumis à des configurations différentes. Un intérêt de cette configuration, surtout par rapport à la série Attinyyxx, est que, si on n'a pas absolument besoin de tous ses ports, le programmeur peut rester connecté pendant la mise au point.

Sous Minux (flavour Linuxlint 15 à 17 Mate), une fois le code développé, avec le compilateur (gratuit) avr-gcc, la commande de programmation d'un fichier compilé (*.hex) est : **avrdude -p m328p -c usbasp -U flash:w:main.hex**

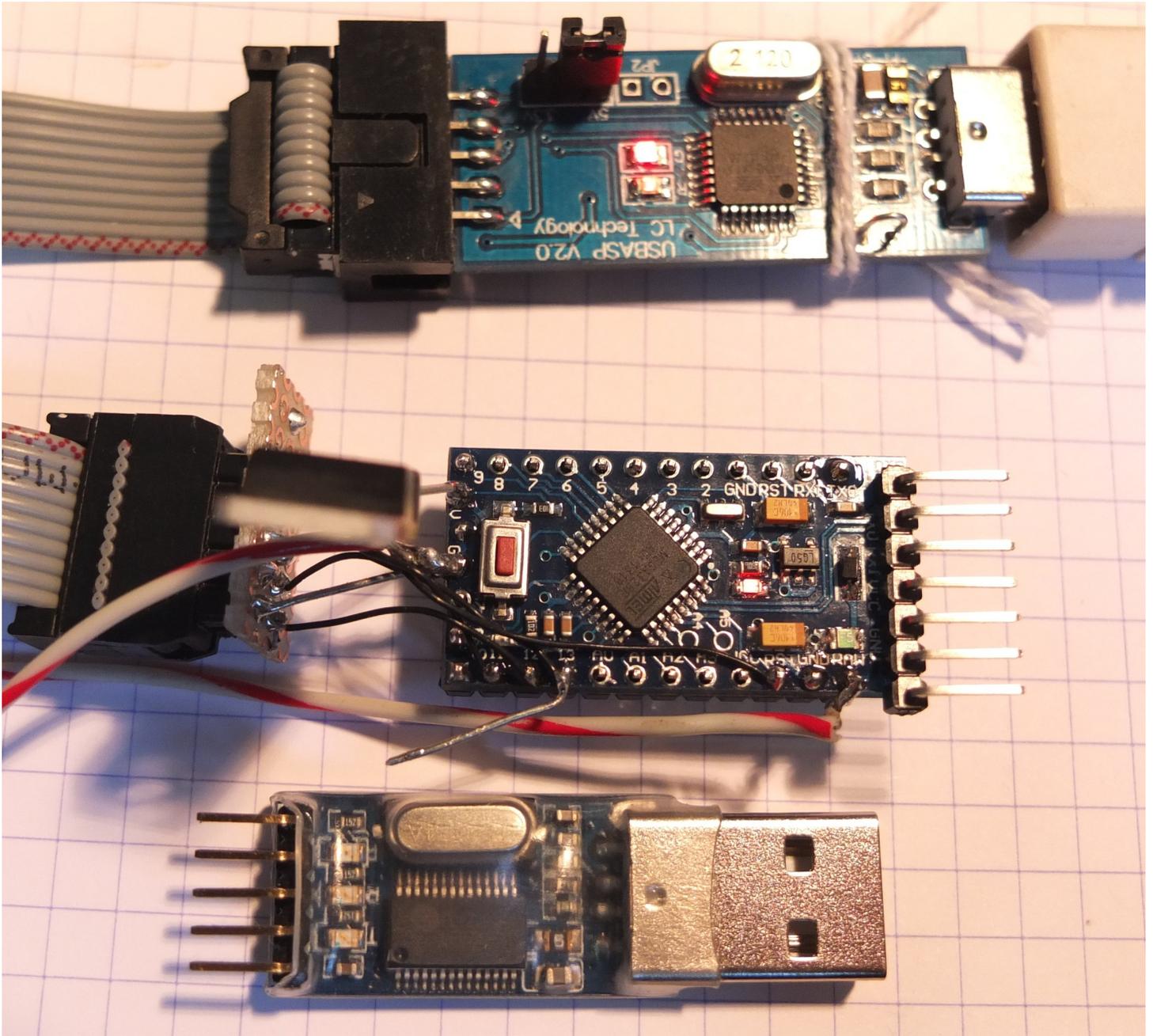
la sauvegarde du bootloader (au cas où je voudrais y revenir) est : **avrdude -p m328p -c usbasp -U flash:r:bootloader.hex:i**. C'est la même commande pour sauvegarder tout programme déjà contenu dans la puce.

Le mode terminal permet, entre autres, de lire et d'écrire les fusibles **avrdude -p m328p -c usbasp -t**
Écriture directe des fusibles (ici lfuse et 6a hex) est **avrdude -p m328p -c usbasp -U lfuse:w:0x6a:m**

Les sorties du bus SPI (SCK, RST, MISO, MOSI) viennent directement du microcontrôleur de la carte programmeur : elles peuvent donc être abîmées par une mauvaise utilisation. De plus ces sorties affectent les broches du mega 328 de l'arduino (mais uniquement pendant une programmation). Pour conserver le double usage de ces broches, j'évite d'y mettre des circuits à résistance inférieure à 4kohms, le programmeur peut alors rester connecté en permanence.

Les programmes avr-gcc et avrdude existent sous WinDaube : je n'ai pas essayé.

6 Chaîne de développement



En bas l'adaptateur USB-TTy D-SUN

Les photos sont importantes, elles sont souvent LE moyen d'identification sur IB

L'adaptateur USB-Tty (USB-RS232) est très pratique, il permet de dialoguer entre l'arduino (port RS232) et le PC (port COM : via USB)

7 Conclusions

Cette carte déjà câblée, est devenue pour moi un composant à bon prix, que je peux employer à toutes les sauces. Sa datasheet est essentiellement constituée de son schéma et de la datasheet du mega328p. J'ai grand'fun !