

www.sylvainmahe.xyz

LE BLOG

de Sylvain Mahé

contact@sylvainmahe.xyz



Article: Sylvain Mahé

contact@sylvainmahe.xyz

[Retour](#)

[Suite](#)

Les outils complémentaires de MODULE

En programmation C++ pour les microcontrôleurs de marque Microchip comme l'ATmega48P, l'ATmega88P, l'ATmega168P, l'ATmega328P, l'ATmega164P, l'ATmega324P, l'ATmega644P ou l'ATmega1284P dont il est question ici, des outils de compilation et de programmation de la puce performants et standards existent.

Ces outils sont disponibles pour les systèmes **Linux** (c'est mon cas), ou **Windows**. Pour Mac OS il faudra effectuer des recherches car je ne connais pas ce système, mais Mac OS étant une base Unix je pense que cela ne doit pas beaucoup différer de Linux.

Outils pour Linux:

Vous devez installer les paquets **gcc-avr**, **avr-libc**, **binutils** et **avrdude** pour programmer le microcontrôleur installé sur mes automates programmables MODULABLE M20 ou M32.

Pour ma part j'utilise le système d'exploitation Linux Ubuntu (qui est une distribution Linux largement utilisée de nos jours), en exemple voici la ligne de commande à écrire dans le terminal pour télécharger et installer les paquets sur Linux Ubuntu:

```
sudo apt-get install gcc-avr && sudo apt-get install avr-libc && sudo apt-get install binutils && sudo apt-get install avrdude
```

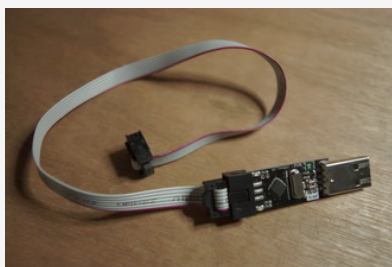
Outils pour Windows:

Vous devez télécharger et installer un programme qui se nomme **WinAVR** pour programmer le microcontrôleur, ce programme contient les mêmes outils que sur Linux mais ici en version pour Windows:

[Site internet de WinAVR](#)

Le programmeur:

Le programmeur est une petite carte qui fait l'interface entre votre ordinateur personnel et le microcontrôleur (implanté sur l'automate programmable), ce qui permet d'envoyer votre programme dans cette puce. Vous pouvez fabriquer un programmeur vous même, ou utiliser une carte toute faite. Pour ma part j'ai essayé les deux.

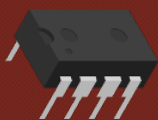


Je vous conseille d'utiliser le programmeur **USBasp** qui fonctionne à merveille, disponible à l'achat ici:

[Site internet de l'USBasp](#)

Le programmeur **USBasp** est comme son nom l'indique équipé d'un **port USB**, à connecter directement à un port USB de votre ordinateur personnel, et à l'autre extrémité d'un port dédié à la programmation du microcontrôleur (port SPI avec alimentation +5V), à connecter via une nappe adaptée (fournie) sur le port nommé **PROGRAMMER** de l'automate programmable (qui lui même alimente en +5V la totalité de l'automate programmable et est relié en SPI au microcontrôleur).





www.sylvainmahe.xyz

LE BLOG

de Sylvain Mahé

contact@sylvainmahe.xyz



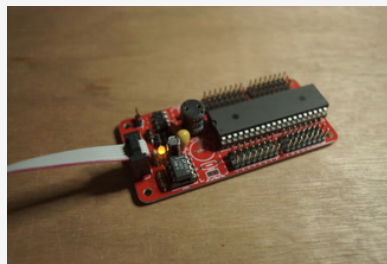
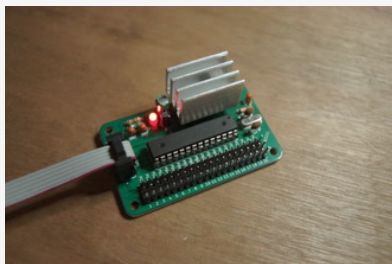
[Retour](#)

[Suite](#)



À noter que la programmation par le SPI du microcontrôleur a plusieurs avantages:

- Évite de contenir dans la mémoire un bootloader.
- De ce fait, évite un délai au démarrage du microcontrôleur.
- Minimise le nombre de composants embarqués sur la carte électronique.
- En conséquence, dispense d'équiper la carte électronique d'un composant FTDI qui sur certains systèmes concurrents, cause des problèmes d'impédance sur les entrées/sorties du microcontrôleur.



Compiler et téléverser votre programme:

Pour compiler et téléverser votre programme dans l'automate programmable via le programmeur USBasp, il vous suffit d'exécuter (de double-cliquer) les routines contenues dans le répertoire **exemple** situé dans l'archive de MODULE (téléchargeable en page d'accueil dans la section "Téléchargements") nommées **Compiler.sh** si vous utilisez un système d'exploitation Linux, ou **Compiler.bat** si vous utilisez un système d'exploitation Windows.

*Pour que cette opération de compilation mais surtout de téléversement fonctionne, sur **Windows** il faut installer le **pilote de l'USBasp**, sur **Linux** ou **Mac OS** en revanche il n'y a pas besoin de pilote. Par contre sur Linux Ubuntu que j'utilise il est tout de même nécessaire d'effectuer une petite opération décrite ci-dessous.*

Procédure pour Linux Ubuntu:

Sur Linux Ubuntu, il convient d'écrire quelques lignes dans le terminal afin d'ouvrir un fichier qui permettra de spécifier quelques paramètres utiles pour l'USBasp:

```
sudo gedit /etc/udev/rules.d/99-USBasp.rules
```

Cette ligne ouvre l'éditeur de texte **Gedit**, ensuite il suffit de copier la ligne suivante dans cet éditeur de texte:

```
SUBSYSTEMS=="usb", ATTRS{idVendor}=="16c0",  
ATTRS{idProduct}=="05dc", GROUP="plugdev", MODE="0666"
```

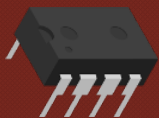
Une fois ceci effectué, vous enregistrez le fichier, et vous pouvez quitter Gedit.

Pilote de l'USBasp pour Windows:

Sur Windows le pilote de l'USBasp n'est pas présent par défaut sur ce système. Par conséquent vous devez le télécharger et l'installer vous même.

L'auteur de l'USBasp recommande d'utiliser un outil pour Windows qui se nomme **Zadig** et qui permet de rechercher ce pilote, personnellement je n'ai jamais essayé (ne possédant pas Windows) mais vu la qualité de son travail je pense que cela est de bon conseil:

[Site internet de Zadig](#)



www.sylvainmahe.xyz

LE BLOG

de Sylvain Mahé

contact@sylvainmahe.xyz

[Retour](#)[Suite](#)

Les fusibles du microcontrôleur:

Quand vous achetez un microcontrôleur neuf n'ayant jamais été programmé, les fusibles qui permettent de définir des paramètres comme la vitesse de téléversement, l'utilisation d'un quartz externe ou encore le verrouillage du SPI (et bien d'autres paramètres fondamentaux pour le microcontrôleur) sont définis par défaut par le constructeur de la puce.

Les routines décrites ci-dessus (Compiler.sh et Compiler.bat) s'occupent de modifier automatiquement les valeurs des fusibles.

*Les valeurs de fusibles que j'ai indiqué dans les routines sont les **plus optimisées pour les automates programmable MODULABLE M20 et M32** que j'ai développés (dont les plans de fabrication sont disponibles dans les sections "Téléchargements" et "Fabrications et diverses réalisations" en page d'accueil).*

Si jamais vous souhaitez programmer avec MODULE sans utiliser les routines que je propose, sachez que cette opération sur les fusibles est à effectuer une seule fois dans deux cas précis, lorsque vous recevez un **microcontrôleur neuf n'ayant jamais été programmé**, ou lorsque vous récupérez un microcontrôleur qui possiblement dans sa mémoire contient un bootloader, ou des paramètres de fusibles inadaptés.

design du blog: sylvain mahé