

www.sylvainmahe.xyz

## LE BLOG

de Sylvain Mahé

contact@sylvainmahe.xyz



Article: Sylvain Mahé

contact@sylvainmahe.xyz

[Retour](#)

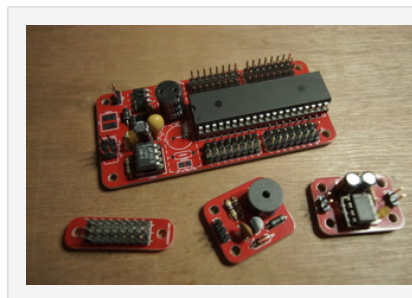
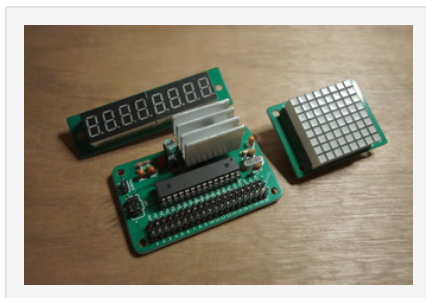
[Suite](#)

# Comprendre ce qu'est MODULE

Partie logicielle, MODULE est une suite de classes écrites en **C++** pur (sans bibliothèque additionnelle) dédiées à la programmation des microcontrôleurs **ATmega48P**, **ATmega88P**, **ATmega168P**, **ATmega328P**, **ATmega164P**, **ATmega324P**, **ATmega644P** et **ATmega1284P** qui équipent mes automates programmables.

Partie matérielle, je vous propose les plans de mise en production (par vos soins ou par une usine de fabrication) de quelques cartes électroniques partie commande avec microcontrôleurs (ce sont les automates programmables associés à la bibliothèque MODULE), mais également de l'électronique de puissance.

Photo de gauche, l'automate programmable **MODULABLE M20** accompagné des afficheurs à digits et matrice, photo de droite, l'automate programmable **MODULABLE M32** avec à ses cotés également diverses cartes, toutes sont proposées en téléchargement:



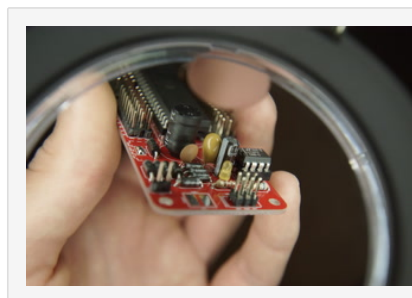
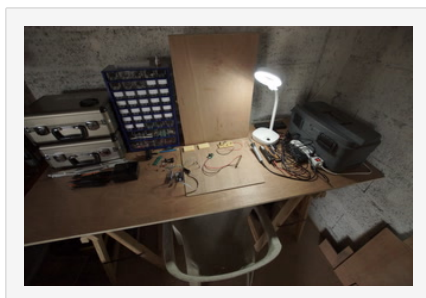
*Les plans de fabrication de mes cartes électroniques sont disponibles dans les sections "Téléchargements" et "Fabrications et divers réalisations" en page d'accueil.*

## Historique du projet MODULE:

Le projet MODULE et les cartes électroniques qui y sont associées a débuté en Août 2014 et aura demandé 3 ans de travail.

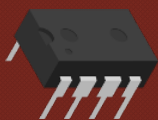
Depuis longtemps je me suis intéressé à l'électricité, mais l'électronique (embarquée) avant ce projet m'est toujours apparu comme obscure et difficile à appréhender. En Août 2014, j'ai donc décidé d'apprendre avec la meilleure volonté l'électronique, parce que c'était un domaine que je ne connaissais (sans même encore parler de maîtriser ou de pratiquer) absolument pas.

Sans formation dans le domaine, il aura fallu acharnement pour ainsi progresser en lisant pendant des mois les documentations techniques des composants, des articles sur internet (dans lesquels il convient souvent de trier et de recouper l'information), mais aussi et surtout à force de pratique et d'expériences personnelles.



## La philosophie de MODULE:

MODULE n'est pas programmé en surcouche d'une autre bibliothèque comme l'est la plupart des bibliothèques dont fait partie Arduino™, de fait le code source C++ est dédié et optimisé pour le microcontrôleur ATmega48P, ATmega88P, ATmega168P, ATmega328P, ATmega164P, ATmega324P, ATmega644P ou ATmega1284P afin d'obtenir le meilleur compromis en C++ en terme de rapidité et d'espace mémoire.



www.sylvainmahe.xyz

## LE BLOG

de Sylvain Mahé

contact@sylvainmahe.xyz



Retour

Suite

*MODULE est une bonne alternative à Arduino™ pour programmer plus efficacement l'ATmega48P, l'ATmega88P, l'ATmega168P, l'ATmega328P, l'ATmega164P, l'ATmega324P, l'ATmega644P ou l'ATmega1284P.*

L'idée de MODULE veut qu'aucun code C++ n'ait été recopié d'un livre ou d'internet, tout est une création personnelle unique, ceci dans le seul but de comprendre le fonctionnement des choses, d'optimiser, et de partager une création personnelle tout simplement.

Aucune bibliothèque même couramment admise (par la communauté de programmeurs) comme faisant partie du standard du langage C++ n'a été utilisée pour programmer MODULE, comme par exemple stdlib.h, stdio.h, ou encore math.h, toujours dans un souci de compréhension du fonctionnement. Cela m'a permis par exemple de faire un peu de mathématiques (sans math.h) et de comprendre comment construire des fonctions comme sinus, cosinus, ou encore arc tangente que l'on retrouve souvent dans les calculatrices modernes sans en connaître et en imaginer réellement le fonctionnement interne.

*Ne pas être lié à d'autres programmes permet à MODULE d'être indépendant, et en outre de ne pas subir les aléas, évolutions, bogues, ou encore obsolescences de fichiers concurrents.*

MODULE qui a débuté avec l'ATmega328P (ce fut le point de départ), c'est aussi redonner tout l'intérêt pour ce petit microcontrôleur 8 bits de plus en plus délaissé au profit de microcontrôleurs 32 bits au fréquences de fonctionnement plus élevées du fait que les bibliothèques concurrentes demandent plus de cycles pour faire les mêmes opérations que MODULE.

### La normalisation de la structure interne et syntaxique de MODULE:

MODULE a été programmé avec rigueur et normalisation du code source (comme tous les programmes que je vous propose en téléchargement ou en exemple).

Une ligne de conduite concernant la logique, la structure, la syntaxe, le nommage des variables, des fonctions et des classes, ou même simplement l'aspect visuel (indentation, espaces, etc...), a été respectée afin d'obtenir le code source le plus lisible, fonctionnel et propre possible.

*Ceci n'est pas plus une réponse à tout ce que je peux constater dans la communauté des programmeurs actuellement, mais plus un souci perfectionniste (même si en soit la perfection est une donnée subjective).*

### Simplicité de programmation avec MODULE:

En interne, MODULE s'occupe de réaliser les tâches les plus laborieuses, ardues et complexes que requiert la programmation brute d'un microcontrôleur. Ceci permet au développeur de réaliser des projets élaborés en un minimum de lignes de programmation, sans pour autant pâtir sur les performances ou brider le nombre de possibilités.

MODULE se veut accessible au débutants, tout en permettant la réalisation d'applications professionnelles.

### Que peut-on faire avec MODULE?

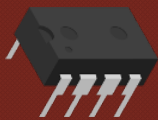
MODULE est comme son nom l'indique modulaire, il n'a en soit pas de limite, la seule est notre imagination. Sachez qu'à l'époque de la conquête spatiale, des ingénieurs et techniciens ont envoyés des fusées dans l'espace avec bien moins comme calculateur qu'un ATmega328P !

Avec MODULE vous pouvez faire fonctionner des gyroscopes, magnétomètres, baromètres, émetteurs/récepteurs 2.4Ghz, faire fonctionner des servos-moteurs et des esc, lire des pwm, créer une interface entre l'utilisateur et l'automate programmable via des boutons, des potentiomètres, des afficheurs, des buzzers et divers haut-parleurs. Il est possible de gérer le temps, l'aléatoire, de filtrer des valeurs, de gérer la veille du microcontrôleur, la communication entre plusieurs automates programmables, ou encore de sauvegarder des données dans sa mémoire EEPROM interne.

*Libre à vous de programmer d'autres fonctions dans MODULE si vous le souhaitez !*

Photo ci-dessous, un exemple de projet réalisé avec MODULE et l'automate programmable MODULABLE M20 équipé de l'ATmega328P:





www.sylvainmahe.xyz

## LE BLOG

de Sylvain Mahé

contact@sylvainmahe.xyz



[Retour](#)

[Suite](#)

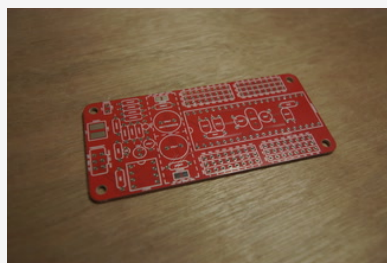


### L'électronique associée à MODULE:

Toute l'électronique que je vous propose en téléchargement (dans les sections "Téléchargements" et "Fabrications et divers réalisations" en page d'accueil) est réalisée avec les mêmes contraintes de normalisation que la programmation, c'est-à-dire avec la plus grande rigueur et dans le souci de prise en compte de ce qui pourrait être considéré comme des détails.

Je réalise les plans de mes circuits imprimés à l'aide du **logiciel KiCad** avec le système d'exploitation Linux Ubuntu, les fichiers originaux de mes projets KiCad ne sont pas disponibles en téléchargement (disponibles cependant sur demande), mais sachez que mes dessins sont réalisés à l'aide de **ma propre bibliothèque de composants en vue schématique et PCB**, bibliothèque directement construite de façon normée (comme le reste) à partir de ma propre expérience concernant les caractéristiques techniques des composants concernant leurs implantation (ergonomie de positionnement, dimensions des perçages pour les broches traversantes, prise en compte des parasites, etc...) et bien évidemment des documentations techniques officielles (constructeurs) des composants électroniques.

Le reste de la conception n'est pas laissée au hasard (calcul des dimensions des pistes en fonction du courant, plans de masse reliés au GND, distances entre les composants, distances au bord des PCB, normalisation de la sérigraphie et du nommage, épaisseur des lettres, des tracés, position et orientation des mots, etc...).



Les plans de fabrication de mes circuits imprimés sont tous disponibles au **format de fichier Gerber**, car c'est un fichier normalisé utilisé par toute l'industrie de l'électronique en général pour la transmission de plans pour la mise en production de carte électroniques.

*Le format de fichier Gerber est le standard directement pris en charge par toutes les usines de fabrication (quel que soit le pays considéré).*