

## ÉLÉMENT DE PORTFOLIO 05



# Logiciel ou bibliothèque logicielle

## 1 DÉFINITION DE CET ÉLÉMENT

**Titre de l'élément :** L'environnement OMicroB pour la programmation de microcontrôleurs

**URL de l'élément :** <https://github.com/stevenvar/OMicroB>

## 2 MOTIVATIONS DU CHOIX DE CET ÉLÉMENT

OMicroB (*OCaml on Microcontroller Boards*) est un logiciel/environnement de programmation de haut niveau pour la programmation de microcontrôleurs à faibles ressources.

Nous l'avons inclus dans ce portfolio car il s'agit d'une approche originale de programmation qui privilégie la sûreté de fonctionnement, la portabilité du code et l'utilisation parcimonieuse de la mémoire.

Cet environnement est issu de premiers résultats du projet OCaPiC [11] d'exécution du langage OCaml par une machine virtuelle implémentée directement en assembleur PIC. OMicroB généralise cette approche « machine virtuelle » en abstrayant plusieurs familles de microcontrôleurs cibles (voir figure 1) pour expérimenter de façon portable des styles de programmation de haut niveau (fonctionnel/impératif et objet/modulaire) en OCaml, dans un cadre confortable de type statique.

OMicroB [9] a été développé pour la thèse de Steven Varoumas [5] pour tester de manière effective une extension synchrone dédiée à la gestion des interactions ainsi que l'analyse générique de code-octet pour le calcul du pire temps d'exécution [6]. Le portage d'OMicroB sur l'architecture LCHIP dans le cadre du projet éponyme [1] (FUI : 2016-2020) a permis de proposer une diversification de voies d'exécution de logiciels critiques.

## 3 PRÉSENTATION DE CET ÉLÉMENT

OMicroB est un environnement de programmation de microcontrôleurs apportant expressivité, sûreté d'exécution et portabilité. Il se compose d'une implantation spécialisée de la machine virtuelle OCaml ainsi que de sa bibliothèque d'exécution (en particulier son gestionnaire automatique de mémoire).

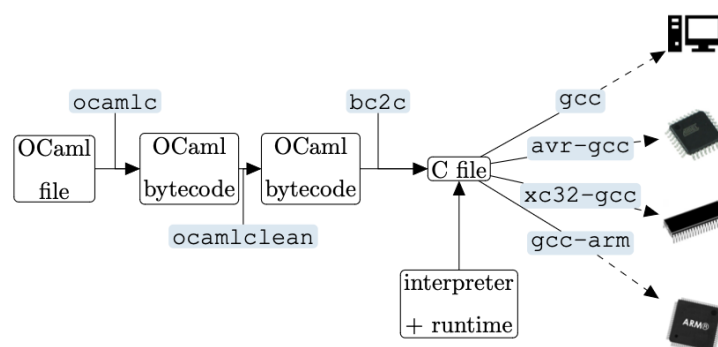


FIGURE 1 – Chaîne de compilation d'OMicroB

La figure 1 montre la chaîne de compilation d'OMicroB sur PC ou ciblant différentes familles de microcontrôleurs. Le programme OCaml est compilé par le compilateur standard OCaml (`ocamlc`). Le fichier de code-octet produit est alors passé à la commande `ocamlclean` qui enlève le code mort, puis au traducteur `bc2c` qui engendre un programme C portable contenant dans un tableau l'ensemble du code-octet ainsi que l'état mémoire obtenu par évaluation partielle (une partie de l'exécution du programme source étant effectuée à la compilation autant que

possible). Ce programme est alors compilé par le compilateur C de la cible et lié avec l'interprète de code-octet et la bibliothèque d'exécution spécifique pour produire un exécutable transférable vers la cible.

Le typage statique fort d'OCaml garantit l'absence d'erreur de typage du programme, et un mode simulateur permet de tester le programme sur ordinateur avant son transfert sur microcontrôleur. L'utilisation parcimonieuse de la mémoire d'OMicroB assure la faible empreinte mémoire des programmes s'exécutant sur des microcontrôleurs à faibles ressources (moins de 32 Ko de code, moins de 4 Ko de mémoire). Différents montages ont été réalisés dans le domaine du divertissement (tempéreuse, jeux à deux joueurs) mais aussi en ciblant le logiciel critique avec l'extension synchrone OCaLustre [8] et l'utilisation dans le projet LCHIP, porté par la société Clearisy, qui propose pour la redondance des calculs cette voie d'exécution spécifique sur une carte munie de deux micro-contrôleurs.

Nous l'utilisons comme un laboratoire pour la montée en abstraction [7] afin de :

- ▶ s'abstraire des microcontrôleurs ;
- ▶ s'abstraire des composants électronique pour les montages ;
- ▶ s'abstraire de la concurrence en proposant un modèle synchrone.

OMicroB est aussi utilisé en cours de « compilation avancée » de M1 du parcours STL (Science et Technologie du Logiciel) de Sorbonne Université, et a été présenté dans un cours invité à la conférence JFLA [10]. L'environnement OMicroB devient mature et propice à une plus large diffusion.

De plus nous utilisons actuellement une version spécifique d'OMicroB appelée O2B [2] pour exécuter OCaml sur un processeur softcore (comme le Nios2) embarqué sur du matériel reconfigurable FPGA dans le but de s'interfaçer avec d'autres circuits [4]. O2B est notamment utilisé en combinaison avec le compilateur Macle pour l'accélération matérielle de programmes OCaml sur FPGA [3].

## 4 RÉFÉRENCES BIBLIOGRAPHIQUES

- [1] Thierry Lecomte, David Déharbe, Denis Sabatier, Etienne Prun, Patrick Péronne, Emmanuel Chailloux, Steven Varoumas, Adilla Susungi, and Sylvain Conchon. Low Cost High Integrity Platform. In *ERTS 2020 - 10th European Congress on Embedded Real Time Systems*, January 2020.
- [2] Jocelyn Sérot and Emmanuel Chailloux. OCaml sur circuit FPGA. In *JFLA 2021 - 32 èmes Journées Francophones des Langages Applicatifs*, en ligne, France, April 2021.
- [3] Loïc Sylvestre, Emmanuel Chailloux, and Jocelyn Sérot. Accelerating OCaml Programs on FPGA. *International Journal of Parallel Programming*, 2023.
- [4] Loïc Sylvestre, Jocelyn Sérot, and Emmanuel Chailloux. A Virtual Machine Approach for High-level FPGA Programming. In *2022 IEEE 30th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, pages 1–1, New York City, United States, May 2022. IEEE.
- [5] Steven Varoumas. *Modèles de programmation de haut niveau pour microcontrôleurs à faibles ressources*. Theses, Sorbonne Université, November 2019.
- [6] Steven Varoumas and Tristan Crolard. WCET of OCaml Bytecode on Microcontrollers : An Automated Method and Its Formalisation. In Sebastian Altmeyer, editor, *19th International Workshop on Worst-Case Execution Time Analysis (WCET 2019)*, volume 72 of *OpenAccess Series in Informatics (OASIS)*, pages 5 :1–5 :12, Stuttgart, Germany, July 2019. Schloss Dagstuhl.
- [7] Steven Varoumas, Basile Pesin, Benoît Vaugon, and Emmanuel Chailloux. Programming microcontrollers through high-level abstractions. In *VMIL 2020 - 12th ACM SIGPLAN International Workshop on Virtual Machine and Intermediate Languages*, pages 5–14, November 2020.
- [8] Steven Varoumas, Benoît Vaugon, and Emmanuel Chailloux. Concurrent Programming of Microcontrollers, a Virtual Machine Approach. In *8th European Congress on Embedded Real Time Software and Systems (ERTS 2016)*, pages 711–720, January 2016.
- [9] Steven Varoumas, Benoît Vaugon, and Emmanuel Chailloux. A Generic Virtual Machine Approach for Programming Microcontrollers : the OMicroB Project. In *9th European Congress on Embedded Real Time Software and Systems (ERTS 2018)*, January 2018.
- [10] Steven Varoumas, Benoît Vaugon, and Emmanuel Chailloux. La programmation de microcontrôleurs dans des langages de haut niveau - Cours invité. In *JFLA 2018, Vingt-neuvièmes Journées Francophones des Langages Applicatifs (JFLA 2018)*, BANYULS, France, January 2018.
- [11] Benoît Vaugon, Philippe Wang, and Emmanuel Chailloux. Programming Microcontrollers in Ocaml : the OCaPIC Project. In *International Symposium on Practical Aspects of Declarative Languages (PADL 2015)*, volume 9131 of *LNCS*, pages 132–148. Springer Verlag, June 2015.