

Raspberry Pi et développement : Apprendre à programmer (épisode 1)

Depuis quelques années, le Raspberry Pi est devenu un élément incontournable des bidouilleurs en informatique, des geeks en quête d'outils pour s'initier à la programmation, mais aussi des développeurs les plus aguerris. Silicon.fr vous dresse un aperçu des capacités de développement avec le pico-ordinateur.

Le **Raspberry Pi** a été créé initialement pour favoriser l'apprentissage de la programmation auprès des plus jeunes. Il représente toutefois aussi une opportunité pour le monde IT.

Les langages de programmation dédiés aux débutants sont en effet à la portée de tous les métiers. Ainsi, un ingénieur travaillant sur un outil d'automatisation centré sur un Raspberry Pi et ses ports d'entrées-sorties sera en mesure de créer lui-même le code validant le bon fonctionnement de son prototype, **sans devoir passer par la DSI**. Même au sein d'équipes informatiques, ces outils – qui sont en général des interpréteurs – permettent de créer des applications en un temps plus court qu'avec des solutions comme le C.

Python...

Sous Linux, **Python** est une des offres de programmation proposées en standard. Il est d'un accès relativement facile pour les débutants, tout en permettant de créer des applications business de qualité professionnelle. Un environnement de développement basique, mais efficace, est de la partie : **Idle**. Le tout est accessible sous licence Open Source et livré gratuitement avec Raspbian.

Autre avantage de Python, c'est un interpréteur. Le code se veut ainsi **plus facile à déboguer et à maintenir** que celui compilé en langage machine. De plus, le code créé sur un PC x86 installé sous Linux devrait fonctionner directement sur Raspbian. La portabilité du code Python adapté à Windows demeure pour sa part plutôt bonne. Une valeur sûre donc, en cinquième position du classement Tiobe des langages de programmation de juillet 2015.

... ou Basic ?

Le **Basic** a longtemps été une référence pour débiter dans la programmation. Il est souvent critiqué pour son manque de fonctionnalités avancées. Mais cela a bien changé avec les dialectes modernes, qui sont finalement assez proches d'offres comme Python.

Pour les traitements lourds, **BBC Basic** est une solution particulièrement intéressante. De suffisamment haut niveau, ce langage **intègre un assembleur** et peut être compilé via diverses offres, dont ABC (un produit commercial proposé au sein de [DDE](#)). De nombreux composants permettent de faciliter la création d'applications desktop ou le pilotage de périphériques. Problème, cette offre ne fonctionne pas sous Linux, mais sous RISC OS, système d'exploitation digne d'intérêt, mais à réserver – en entreprise – aux applications embarquées et non aux usages desktop ou serveur, pour lesquels il n'est plus aujourd'hui adapté.

Les alternatives

Pratiquement tous les interpréteurs demeurent faciles d'accès. En particulier les offres dédiées au monde du web, qui pourront aisément être adaptées à d'autres usages. **Node.js, Perl, Python, Ruby, TCL** sont tous disponibles sous Raspbian.

Certains compilateurs sont eux aussi abordables. [BaCon](#) est méconnu, mais permet de générer du code C depuis un dialecte de type Basic. C'est une des offres les plus prometteuses du marché. Très portable, BaCon fonctionnera sur les machines Linux x86 comme ARM. Enfin, Java ne devra pas être oublié. À bien des égards – et à juste raison –, les professionnels le considèrent comme le Basic du monde des entreprises.

Voir aussi nos quiz :

[Quiz Silicon.fr – Êtes-vous un expert des langages de programmation ?](#)

[Quiz Silicon.fr – Les outils de développement rapide \(RAD\)](#)

[Quiz Silicon.fr – 9 questions sur Java 9](#)

—

[Quiz Silicon.fr – L'informatique britannique, du BBC Micro au Raspberry Pi](#)

[Quiz Silicon.fr – Savez-vous tout sur le Raspberry Pi 2 ?](#)

[Quiz Silicon.fr – 10 questions sur le projet Debian](#)