

Qt pour le Raspberry Pi

guide pas à pas pour la configuration
du *framework* d'interface graphique utilisateur



Logo du *framework* Qt (www.qt.io).

Mathias Claussen (Elektor)

Notre collègue du labo d'Elektor, Mathias Claussen, spécialiste du logiciel, devait réaliser une commande par écran tactile pour la radio FM sur Raspberry Pi [1]. Pour de telles fonctions, le *framework* d'interface graphique utilisateur Qt est pratiquement devenu un standard de l'industrie, son choix s'est vite imposé. Pour les développeurs confrontés à de tels défis, Mathias a écrit un manuel grâce auquel la compilation de Qt sur le RPi devient rapide et facile. Bien entendu, il inclut une démo simple.

Une commande de développement de logiciel est soumise à un cahier des charges, c'est-à-dire à une liste d'exigences que le produit final doit respecter. Dans le cas présent, la commande émanait de la rédaction d'Elektor. La spécification des collègues était lapidaire : « Débrouille-toi pour que ça marche ! »

Si vous développez des logiciels, vous avez sûrement déjà été confronté à ce genre de critères « artistiques ». Mais après quelques froncements de sourcils, il fut néanmoins possible de définir une spécification plus précise.

Ce que voulaient les collègues était la réalisation d'une application Linux rendant possible la commande d'une radio FM. Il fallait également utiliser et afficher les informations RDS si elles étaient présentes

dans le signal. Et, est-il nécessaire de le préciser, alors que c'est pratiquement toujours le cas pour le logiciel, de préférence, le projet devait être fini avant-hier ;-) !

À ce stade, j'ai décidé de réaliser le projet en utilisant le *framework* Qt [2]. Comme le code source devait être ouvert, il n'y eut pas de problème de choix entre les versions GPL et commerciale de Qt. La version GPL permet même de réaliser des applications au source propriétaire, quoique, dans ce cas, il est recommandé d'examiner les conditions de la licence Qt de plus près.

Qt

Qt [2] est un *framework* multiplateforme qui permet de réaliser des

applications en C++/QML basées sur un code commun et idéalement exécutables sur diverses plateformes. Parmi celles-ci, on compte les suspects habituels, Windows, Linux, Android, iOS, QNX ainsi que certaines variantes embarquées de Linux. L'usage de ce *framework* permet d'économiser des efforts en bien des endroits et de ne pas avoir à se colleter avec les particularités des différents systèmes d'exploitation. Parmi les bibliothèques du *framework*, celles de l'interface graphique sont particulièrement utiles pour un développement rapide. Ce qui est généralement vrai l'est aussi pour le Raspberry Pi. L'utilité de Qt m'était déjà apparue dans des projets antérieurs (de l'application console à un petit outil d'interface graphique). La chose élégante à propos de Qt est que l'expérience acquise dans divers projets n'est pas seulement utile pour le RPi, mais peut également s'appliquer à d'autres plates-formes supportées par le *framework*.

Un framework à jour

On remplirait des livres avec tout ce qu'il y aurait à expliquer et commenter sur le *framework* Qt. Cela concerne de nombreux aspects tels que le fonctionnement des outils, le développement d'applications ou la conception d'interfaces utilisateur graphiques (GUI). Si vous voulez commencer maintenant sur un RPi, vous pouvez utiliser une version Qt prête à l'emploi. C'est confortable, mais présente l'inconvénient que cette version finie a maintenant plus de deux ans. Dans le monde du logiciel, cela équivaut à deux éternités. L'alternative habituelle est « compilez-le vous-même ». Comme on intègre ainsi des correctifs et des mises à jour de sécurité qui ne datent pas de l'antiquité, la version à jour n'en est que plus sympathique.

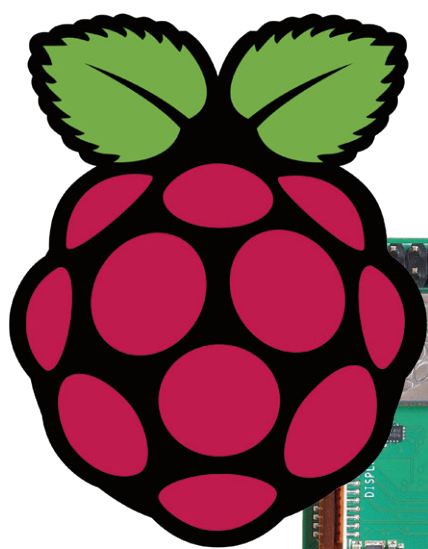
Le *framework* Qt proprement dit inclut également Qt Creator. Cet outil bien utile s'améliore à chaque nouvelle version. De nombreux bogues ont déjà été corrigés dans le passé. S'il est possible d'exécuter Qt Creator sur le RPi, on est toutefois limité par la mémoire RAM et le nombre de ports d'écran disponibles. Si l'on décide de compiler Qt, autant en profiter pour configurer une version pour PC. On a alors la possibilité de compiler et de tester sur le PC des applications

qui ne font pas appel à des particularités matérielles du RPi. Avec les réglages adéquats, il suffit alors d'un clic pour copier et exécuter sur le RPi l'application testée sur le PC. Cela inclut la possibilité de déboguer le code, de déclarer des points d'arrêt et d'afficher des valeurs de variables.

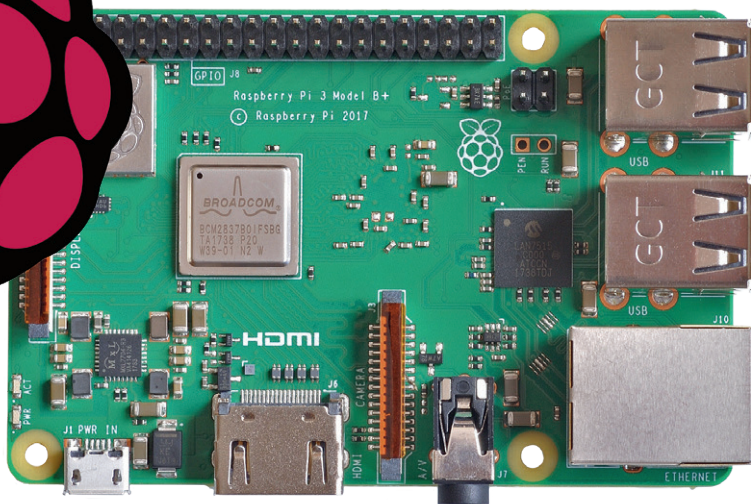
Manuel utilisateur (pour son bonheur)

Si l'on parvient à se procurer un mode d'emploi, on se heurtera tôt ou tard à un problème, soit qu'on oublie un quelconque réglage, soit que ce mode d'emploi concerne une version qui pose problème avec le système d'exploitation actuel du RPi. C'est pourquoi j'ai écrit un manuel [3] qui décrit les différentes étapes de l'installation sur le RPi, et de même pour l'environnement de développement sur le PC.

Cette dernière partie étant plus simple sous Linux, l'environnement PC a été optimisé pour Ubuntu. Mais, pas de souci, vous n'aurez pas à dédier un PC supplémentaire rien qu'au développement pour le RPi (ce qui poserait sans doute un problème de place plus que de coût). Il suffit d'une machine virtuelle (Virtual Box) et d'environ 60 Go d'espace disponible sur le SSD de votre PC, que vous consacrez de toute façon à des usages techniques.



Logo de la Fondation
Raspberry Pi
(www.raspberrypi.org).



Outre l'installation proprement dite, le manuel décrit avec précision un petit programme de démonstration. De plus, on peut télécharger le projet de radio FM dans Qt et observer l'interaction des différentes parties. Ce manuel doit être un point de départ et montrer comment utiliser les différents outils et ne pas se laisser dérouter par des détails importants dès la mise en place. ◀

(180477-04 – VF : Helmut Müller)

LIENS

- [1] « Radio FM avec RDS », Elektor 1-2/2019 : www.elektormagazine.fr/180119-04
- [2] Framework Qt : www.qt.io
- [3] Manuel d'utilisation de Qt (PDF, 41 pages, en anglais) : <https://github.com/ElektorLabs/160520-QT-compile-and-install-manual>