

[Programmez!]

Le magazine des développeurs

programmez.com

220 JUILLET-AOÛT 2018

Enfants • ados • parents

CODER EN S'AMUSANT

 **Se lancer
dans le
minage**
de crypto-monnaies

**Migrer
d'AngularJS
à Angular**

**J'automatise
la maison**

Ma première application avec



LE SEUL MAGAZINE ÉCRIT PAR ET POUR LES DÉVELOPPEURS

Apprendre à programmer au lycée avec Texas Instruments



TI-Innovator™ Rover

Le robot programmable par la calculatrice graphique pour une multitude d'expériences dynamiques



TI-Innovator™ Hub

Le boîtier de commande doté d'une carte professionnelle MSP 432 pour contrôler le Rover et réaliser des projets simples



Les calculatrices graphiques

Les calculatrices graphiques TI-83 Premium CE et TI-Nspire™ CX/CX CAS pour se familiariser avec la programmation



Livres

Les livres Eyrolles "Algorithmique et programmation" pour le lycée général et professionnel afin de développer des projets pour tous les niveaux



TEXAS INSTRUMENTS



EDITO

LES ANDROÏDES RÊVENT-ILS DE LA « DÉVELOPPEUR NATION » ?*

Nous sommes inondés de développeurs. Si l'informaticien est parfois mal compris en France, étonnement, le développeur est le sauveur. Les politiques parlent de développeurs, les grandes entreprises parlent des développeurs et les médias parlent aussi des développeurs.

Les développeurs vont donc sauver le monde. Enfin, si on arrive à en trouver

Plus sérieusement, pour ce numéro d'été, nous avons voulu vous proposer un dossier pour les parents et les enfants & ados. On va parler robotique, cartes de prototypages, programmation, etc. De quoi s'occuper durant (tout) l'été. Et la programmation n'a rien d'une entité obscure mais est quelque chose de bien réel.

Nous parlerons aussi de Kotlin, Angular, Java, .Net et de JavaScript. Les grands ont aussi le droit de s'amuser un peu...

Les androïdes rêvent-ils de la startup nation ?**

Nous sommes inondés de startups. Si l'entreprise est parfois mal vue en France, étonnement la startup est le paradis sur Terre. Les politiques parlent de startups, les grandes entreprises parlent de startups et les médias parlent aussi de startups.

Les startups vont donc sauver le monde.

Oui il y a des milliers de startups en France. Certaines sont rachetées. Mais ces réussites, oui il y en a, ne doivent pas cacher la réalité, la vraie : le poids réel dans l'économie (actuellement), le nombre de salariés et la pérennité. Il faut être réaliste, sur 10 startups combien survivent après 5 ans (si elles arrivent à durer aussi longtemps) ? 50 % si on veut être optimiste, 30 % si on est pragmatique.

Ce n'est pas choquant. Car une startup, comme toute entreprise, est une aventure sur une idée qu'il faut développer mais elle doit surtout trouver son marché, avoir des utilisateurs. Et c'est un problème qui n'est pas toujours compris par la startup. Trouver les millions d'Euros c'est bien, mais pour quoi faire ? Parfois, on se le demande.

L'autre difficulté est de se démarquer des autres startups qui font parfois peu ou prou la même chose. Eh oui, manque de chance pour toi, la disruption intéresse aussi d'autres startups que la tienne.

L'excès de startups nuit-il à la santé ? Je ne sais pas mais le réveil à la réalité sera difficile pour certains.

Mais bon heureusement, nous avons un super plan Intelligence Artificielle qui sauvera la France et qui nous aidera à rester au top de l'IA. L'Europe n'est pas en reste avec un comité IA. Nous avons eu un si beau succès avec les clouds souverains (P.S. : ça se voit que je suis un peu sarcastique ?).

François Tonic
ftonic@programmez.com

* voir le **

** vous l'aurez toutes et tous reconnus "Les androïdes rêvent-ils de moutons électriques" de Philip. K. Dick qui a servi de fondation à Blade Runner (euh, le premier, pas 2049).

SOMMAIRE

Agenda4

Tableau de bord6

Kotlin par JetBrains8

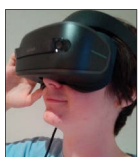
GitHub :
stop ou encore ?9

Guide des cartes 201811

De la réalité augmentée
avec Vuforia18



Faites du minage !
Partie 121

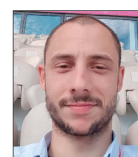


Dossier spécial :
programmation
parents-enfants
& ados - Partie 125

Abonnez-vous !42



C++48



Apprentissage
Kotlin51



J'automatise
la maison - Partie 155

POO en Java - Partie 165



Progressive
Web Application
Partie 168

Global .net tools71



Migrer vers Angular73



Infra as code79

Dans le prochain numéro !

Programmez! #221, dès le 31 août 2018

SQL Server sur Linux : une version explosive
Blazor : le nouveau framework de Microsoft
Au coeur de C++ 17

CONFÉRENCES

Xebia organise 2 conférences techniques

- FrenchKit, la première conférence française dédiée aux développeurs iOS et macOS. 20 et 21 septembre 2018 - frenchkit.fr/
- XebiCon, la conférence qui vous donnera les clés pour tirer le meilleur des dernières technologies : Data, Architecture, mobilité, DevOps, etc. 20 novembre 2018 - xebicon.fr

JUILLET – AOÛT

Nantes Maker Campus 2018

Nantes accueille le gros événement maker de cet été : Nantes Maker Campus. L'événement réunira des artistes, des dizaines de makers dans un grand village dédié avec des ateliers et des conférences. Des milliers de personnes sont attendues du 6 au 8 juillet au cœur de Nantes ! Site : <http://nantesmakercampus.com>

Kansas Fest (USA)

Kansas Fest 2018 se tiendra à Kansas City du 17 au 22 juillet. Il s'agit de la 30e édition de cette conférence réunissant les fervents de l'Apple II venant du monde entier. La session plénière sera animée par Roger Wagner rendu célèbre par sa série d'articles Assembly Lines <https://www.kansasfest.org/>

Apple II Festival France

La quatrième édition du petit frère français de Kansas Fest aura lieu au 1er au 5 août 2018 au Maska à Castéra-Verdun dans le Gers. Au programme ici aussi des sessions sur le matériel et le logiciel, des ateliers et beaucoup de convivialité. <https://www.apple2festivalfrance.fr/>

NOVEMBRE

DEVFEST TOULOUSE 2018

08 Novembre / Toulouse

Le DevFest Toulouse est un événement organisé par les communautés de développeur.se.s de Toulouse, et porté administrativement par le GDG Toulouse. Pour rendre tout cela possible, une équipe de bénévoles sur-vitaminée s'active en coulisse. Forte des deux dernières éditions, l'équipe remet le couvert pour une troisième année. Les nouveautés de l'édition 2018 : plus de talks, de speakers, dans un lieu plus grand : le centre des Congrès Pierre Baudis, 600 participants attendus ... et un super thème : le rétro gaming/rétro computing !

Billetterie ouverte : <https://www.billetweb.fr/devfest-toulouse-2018>

Plus d'infos sur : <https://devfesttoulouse.fr>

BAROMÈTRE

Baromètre HIRED recherche d'emploi

En mai la demande pour les profils spécialisés en Ruby a explosé. Go, React et Python restent des valeurs sûres tandis que les profils DevOps reviennent en grâce.

Comme en avril, le baromètre Hired de la recherche d'emploi de ce mois de mai met en évidence une certaine disparité entre les technologies attendues par les recruteurs IT et celles maîtrisées par les candidats mais aussi des évolutions conséquentes. Ainsi, le langage qui a suscité le plus de demandes d'entretiens par les recruteurs était le mois dernier le Ruby, avec pas moins de 9,3 demandes par candidat. Le mois dernier, ce nombre était de 6,3. Avec respectivement une moyenne de 7,7, 6,5 et 5,8 entretiens proposés par candidat, Go, DevOps et React complètent le haut du classement des technologies les plus demandées. Le Devops fait d'ailleurs un retour en grâce vu qu'il n'avait suscité que 3,7 demandes d'entretien en moyenne en avril. Ceci est expliqué d'après Hired par le phénomène du RGPD qui a fait prendre conscience du besoin de profils Devops ainsi que leur recrutement.

Java, PHP et Node qui représentent à eux trois plus de la moitié des technologies mises en avant par les candidatures restent des valeurs sûres pour les recruteurs. En mai 2018, elles ont respectivement permis aux candidats les ayant mises en avant d'être conviés à 4,9, 5,2 et 5,7 entretiens en moyenne. Ces chiffres sont toutefois en baisse d'environ 0.2 points depuis le mois d'avril.

Au cours des 6 derniers mois la tendance est quasiment identique. Les demandes pour DevOps et Vue étant revenues à des niveaux normaux, il n'existe plus de disparité flagrantes entre les chiffres de 1 mois à 6 mois comme nous avons pu en constater le mois dernier.

Mai 2018			
Technologies demandées	Pourcentage de candidatures développeurs	Technologies demandées	Nombre moyen de demandes d'entretiens
Java	18.72%	Ruby	9.3
Node	16.23%	Go	7.7
PHP	13.63%	DevOps	6.5
Python	12.83%	React	5.8
Angular	11.62%	Python	5.5
React	9.62%	PHP	5.5
Android	5.01%	Node	5.2
.NET	3.81%	Vue	5.2
Go	3.01%	iOS	5.2
Vue	2.40%	Java	4.9
Ruby	2.20%	Angular	4.7
iOS	1.60%	.NET	4.6
DevOps	1.60%	Android	3.2

Décembre 2017 à Mai 2018			
Technologies demandées	Pourcentage de candidatures développeurs	Technologies demandées	Nombre moyen de demandes d'entretiens
Java	17.63%	Go	10
Node	14.85%	React	9.5
PHP	14.10%	DevOps	8.8
Python	11.78%	Node	7.7
React	10.65%	Python	7.4
Angular	10.43%	Ruby	7.3
Android	5.40%	Vue	7,1
.NET	3.98%	Angular	6.7
Ruby	3.68%	PHP	6.5
Go	2.78%	iOS	6
iOS	2.40%	Java	5.8
Vue	2.33%	.NET	5.3
DevOps	1.35%	Android	4.6

A propos de HIRED

HIRED.com est une plateforme technologique de recrutement qui attire et sélectionne les meilleurs profils techniques du marché afin de permettre aux entreprises de recruter des candidats qualifiés rapidement et efficacement. Chaque semaine, entre 50 et 100 nouveaux candidats de la communauté française HIRED en recherche active (développeurs, data scientists, designers, etc.) sont triés sur le volet grâce à des algorithmes et prêts à être contactés.

COMMUNAUTÉS

Annoncez vos meetups,
et conférences
sur Programmez ! :

ftonic@programmez.com

Prochain numéro !
Programmez! #221,
dès le 31 août 2018

OPÉRATION POUR 1 EURO DE PLUS

Pour bénéficier de cette offre exceptionnelle, il suffit de commander WINDEV Mobile 23 (ou WINDEV 23, ou WEBDEV 23) chez PC SOFT au tarif catalogue avant le 13 Juillet 2018. Pour 1 Euro de plus, vous recevrez alors le ou les magnifiques matériels que vous aurez choisis. Offre réservée aux sociétés, administrations, mairies, GIE et professions libérales, en France métropolitaine. L'offre s'applique sur le tarif catalogue uniquement. **Voir tous les détails sur : WWW.PCSOFT.FR ou appelez-nous au 04.67.032.032** Le Logiciel et le matériel peuvent être acquis séparément. Tarif du Logiciel au prix catalogue de 1.650 Euros HT (1.980,00 TTC). Merci de vous connecter au site www.pcsoft.fr pour consulter la liste des prix des matériels. Tarifs modifiables sans préavis.

**Aucun
abonnement
à souscrire.**
Compatible
tous opérateurs

CHOISISSEZ :

- iPhone X 64GB
ou
- iPhone 8 256GB
ou
- iPhone 8 Plus 64GB
ou
- MacBook Air 13,3" 128GB
ou
- lot de 2 iPad New 9,7" 128GB

(Détails et autres
matériels sur
www.pcsoft.fr)

PROLONGATION JUSQU'AU 13 JUILLET

COMMANDEZ WINDEV MOBILE 23 OU WEBDEV 23 OU WINDEV 23 ET RECEVEZ LE NOUVEL Apple iPhone X

Choix de la couleur
sur le site



Atelier de
Génie Logiciel
Professionnel
cross-plateformes

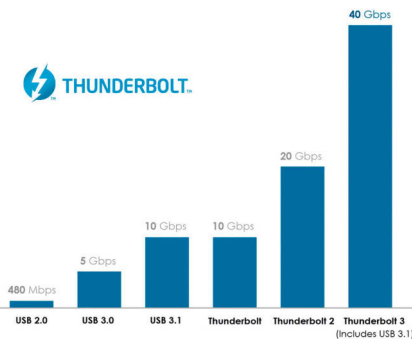


WWW.PCSOFT.FR

Tesla licencie 9 % de son personnel pour relancer la machine.

Surface. Microsoft devrait lancer de nouveaux modèles début 2019.

Tous les disques **USB-C / Thunderbolt** ne se valent pas. Regardez les débits théoriques de votre matériel. S'il affiche 40 Gb/s (toujours théorique), il s'agit d'un Thunderbolt 3. Mais attention à bien choisir les câbles qui vont avec. Il faut impérativement qu'il



soit certifié 40 Gb/s (eh oui pourquoi faire simple ?). Généralement, on trouve surtout du 10 Gb/s bien plus répandu chez les constructeurs et moins cher. Si vous connectez un écran 4K, faites-le impérativement en Thunderbolt 3.

Apple continue de développer son système pour voiture autonome (?) : carOS. Les voitures tests vont bientôt atteindre la centaine sur les routes. Les contours du projet demeurent inconnus et nous ne connaissons rien de l'ampleur des équipes ni des ambitions réelles d'Apple.

Quels frameworks JavaScript allez-vous utiliser ou utilisez-vous ?

Sans trop de surprise, vous utilisez Angular à 27 %, React suit à 20 %. Vue atteint tout de même 13 %. En revanche Vanilla n'est pas très connu et encore moins Meteor. Vous êtes 27 % à affirmer coder JavaScript à la main. Belle abnégation ! (source : programmez.com)

L'Europe veut de l'Intelligence Artificielle

et devenir une puissance qui compte dans ce domaine. Un comité de 52 experts a été mis en place pour définir les grandes lignes éthiques, anticiper les défis et bien dépenser le budget alloué (+2,6 milliards). Après la France qui va nous refaire un cloud souverain sauce IA et l'Europe qui nous sort un magnifique comité, c'est sûr, nous serons leaders de l'IA (// mode troll activé).

Le recyclage est un art surprenant.

Après un faux départ, l'idée d'un Netflix à la française a été ressortie comme par magie : Salto. Le projet regroupe TF1, France Télévisions et M6... L'ambition est simple : sortir une plateforme de streaming dès 2019 avec du sport, de l'infos, des séries et des films. Netflix, Amazon et prochainement Disney et Apple, jouent la catégorie (largement)

supérieure : ils ont une infrastructure énorme au niveau mondial, un catalogue de contenus capables de capter tous les publics et surtout des productions de qualité. Les géants du streaming pensent au niveau mondial, seule condition de réussite. Et petit détail tout de même : le budget. Netflix c'est +6 milliards d'investissements, Amazon 1 milliard pour la future série Seigneur

des Anneaux, Apple +1 milliard... Et il faudra trouver un équilibre entre 3 partenaires... Et rien ne dit que Salto sera bien au rendez-vous en 2019...

600 000 \$ c'est la modique somme demandée pour l'Apple I avec la carte mère d'origine. L'Apple I est aujourd'hui un des micro-ordinateurs les plus rares.

L'ordinateur quantique

futur cauchemar des responsables de sécurité ? Le patron d'IBM Research prédit que le quantique pourra casser toutes techniques de cryptage connues et que rien ne serait à l'abri. Heureusement pour nous, l'ordinateur quantique n'est pas encore prêt à sortir des laboratoires, mais les grandes institutions publiques, et certaines entreprises peuvent installer ce type de machines, si le budget suit. D-Wave, constructeur quantique historique, estime que l'ordinateur quantique n'est pas encore capable de rivaliser avec l'ordinateur actuel dans tous les usages et domaines. Mais la recherche avance vite, très vite.

Tu ne sais quoi regarder cet été ? Révisé tes classiques !



Eh oui pas de Games of Thrones, ni de Mr Robot et encore moins de Man in the High Castle. Une bonne occasion pour regarder l'excellente série Star Trek Discovery même si le premier épisode n'est pas indispensable. Juste pour se rappeler ce qu'est une série de SF digne de ce nom, on regardera de nouveau Battlestar Galactica (celle de 2004). Toujours aussi énorme ! Et surtout, on regardera une série totalement passée inaperçue en France : Halt and Catch Fire. Une pure merveille comme on en voit rarement. 4 saisons jouissives, même si la 2e est pour nous la moins réussie. On (re)découvre les grandes étapes de l'informatique et de la technologie : la secte Macintosh, les costumes IBM (si, si), l'arrivée d'Internet et des chatrooms, la découverte de l'obscur WWW sur une NeXT Cube... Pour les plus hardcores : Wargames.

Le projet Marzipan existe bel et bien

La conférence développeur Apple, la WWDC, a livré un premier élément de réponse sur un projet que la rumeur a évoqué depuis quelques mois : Marzipan. L'idée est de proposer un modèle unique de développeur pour les apps iOS et macOS. Même si Apple n'a pas tout dévoilé ni tout dit, le travail est en cours. Et plusieurs frameworks et librairies sont d'ores et déjà sur les deux systèmes : AppKit et UIKit. Pour le moment, Apple porte les composants systèmes et teste avec ses propres applications. Apple a d'ores et déjà prévenu que la WWDC 2019 sera l'occasion d'en savoir plus. C'est sans doute en juin prochain que les développeurs auront accès à ce nouveau modèle de développement.

BATTLEKART

WINDEV au coeur d'une attraction unique au monde !

Vous aimez le kart ? Vous aimez le laser game ? Vous aimez les jeux vidéo ? Sébastien Millecam a eu l'idée originale de réunir le meilleur de ces mondes dans une seule activité : BattleKart.

Sébastien Millecam a fondé BattleKart avec l'idée originale de mixer du karting électrique et de la réalité augmentée. Le résultat est unique avec un condensé de technologies où cohabitent un ERP, un outil de supervision temps réel conçus avec WINDEV et le moteur de jeu Unity. Sensations inédites garanties !



Sébastien Millecam, fondateur de BattleKart.

Kart et réalité augmentée, BattleKart c'est quoi ?

Vous êtes au volant d'un véritable kart, vous roulez réellement sur une piste dessinée au sol via 96 projecteurs positionnés au plafond. Sur votre volant, vous disposez de boutons pour lancer des missiles (virtuels) sur les autres pilotes, vous cherchez à rouler sur les bonus mobiles affichés au sol, vous évitez les flaques d'huile (virtuelles elles aussi) lâchées par vos concurrents...

Bref, vous êtes dans le jeu et vous vivez pleinement les interactions entre les joueurs. Les sensations sont extraordinaires avec ce mélange d'objets réels (les karts) et d'objets virtuels (missiles, bonus...).

Du POC à la mise en production : WINDEV et Unity font la paire

Fraîchement sorti de son école d'ingénieur, ce passionné d'informatique et de jeux fonde sa société du même nom : « l'idée de BattleKart est née dans le train entre Mouscron et Tournai en rentrant dans mon logement universitaire. Après une première

maquette et un POC réalisé avec WINDEV en dernière année d'études, j'ai pu convaincre des investisseurs de m'accompagner dans l'aventure du projet grande nature », raconte le fondateur. Et d'ajouter : « l'AGL WINDEV a toujours été au cœur du développement du système de supervision et de l'ERP mis en place. J'ai utilisé Unity en complément pour la partie moteur de jeu et graphique ».

Supervision en temps réel

L'architecture de la conception de ce jeu inédit est étonnante. Chaque kart est équipé de microcontrôleurs, d'une tablette affichant le tableau de bord. Chaque projecteur suspendu au plafond est couplé à un PC qui génère les images visualisées au sol. Le tout est piloté par un serveur connecté en réseau qui sait où se trouvent tous les objets réels et virtuels. Ce serveur reçoit les informations du système de géolocalisation et calcule les interactions entre les objets réels et virtuels. Si un kart croise un missile, le système génère une animation d'explo-

sion et un son. Le système de géolocalisation s'appuie sur 336 capteurs alimentés en PoE et connectés au réseau. Il envoie 270 fois par seconde au serveur la position de chaque kart au centimètre près. Et sur la base de ces informations, le serveur calcule les interactions.

WINDEV omniprésent dans l'aventure

Sébastien Millecam ne tarit pas d'éloges sur l'AGL de PC SOFT : « WINDEV est partout dans le développement de nos solutions. Que ce soit dans la conception des bornes d'accueil avec le scan du QR code, les bornes photos pour le podium des participants, le monitoring des scores et des karts, le contrôle des lumières, la gestion de la caisse, la CRM, ou encore la supervision des projecteurs ». Et de poursuivre : « si j'ai choisi WINDEV, c'est pour sa rapidité de développement. Je peux tout programmer en WLangage, en orienté objet ou pas, je peux gérer des flux multi-cast par exemple. Et en plus mon code est cross-plateformes ».

Sur la piste, les opérateurs présents sont équipés d'un smartphone Android attaché sur l'avant-bras, pour lancer ou arrêter une course, recevoir les alertes provenant de capteurs ou du pilote demandant une pause en urgence. « L'application native Android provient d'un premier module développé en WINDEV que j'ai porté dans WINDEV Mobile en une demi-journée. C'est toujours du WLangage », précise Sébastien Millecam.

L'aventure BattleKart rencontre un franc succès avec déjà plus de 70 000 joueurs venus vivre « en live » des émotions inédites. Des franchises sont en négociation pour déployer le concept innovant hors d'Europe.





La rédaction



IdeaJ, ReSharper, PhpStorm, nous connaissons bien vos outils mais nous savons peu de choses sur JetBrains.

Pouvez-vous nous présenter l'éditeur ?

JetBrains a été créée en 2000 par trois développeurs et a grandi jusqu'à employer aujourd'hui plus de 800 personnes dans le monde, ceci sans aucun financement externe. Notre objectif est le même depuis le départ : aider les développeurs à être plus efficaces et productifs en leur épargnant les tâches répétitives, grâce à nos IDE, à nos solutions pour les équipes, et à notre langage de programmation, Kotlin.

Comment Kotlin est né chez JetBrains ?

Comment Google l'a choisi ?

Kotlin est un langage de programmation que nous avons lancé en 2010, d'abord pour répondre à un besoin interne. En effet, avec Java, nous avions une base de code très vaste et nous souhaitions travailler avec quelque chose de plus concis, sécurisé, qui supporte des fonctionnalités bien précises. Après un temps de recherches infructueuses, nous avons finalement décidé de créer notre propre langage. Kotlin est ensuite devenu populaire au sein de la communauté des développeurs Android. En ce qui concerne Google, le mieux serait de leur demander directement pourquoi ils ont choisi Kotlin. Mais je pense qu'ils ont opté pour ce langage car il était déjà largement utilisé et apprécié par la communauté.

Le langage est multiplateforme.

Où est-il disponible en dehors d'Android ?

Kotlin a été développé pour créer des applications desktop et côté serveur ciblant la JVM. Le support d'Android est venu peu de temps après, de même

Questions – réponses autour de Kotlin

Hadi Hariri est développeur et VP of Developer Advocacy chez JetBrains. Il explique et présente Kotlin partout dans le monde. Il a bien voulu répondre à nos questions.

que le support d'autres plateformes, notamment JavaScript. Depuis quelques années, nous proposons le support natif via LLVM, ce qui signifie qu'aujourd'hui on peut non seulement créer des applications Kotlin multiplateformes, car fonctionnant sur la JVM, mais aussi via Kotlin/Native, qui couvre entre autres macOS, Windows, Linux, RaspberryPi, iOS.

Certains parlent de Kotlin comme d'un langage moderne. Okay mais on disait la même chose de Java en 1995 ou de C#.

Qu'en pensez-vous ?

D'un point de vue marketing, il serait étrange de lancer un nouveau langage de programmation sans le qualifier de "moderne" !

Blague à part, tout langage comporte des caractéristiques propres, qu'il s'agisse d'erreurs faites lors de son processus de conception ou des innovations qu'il apporte par rapport aux solutions existantes. Cela fait partie du cycle de vie d'un langage. Kotlin n'échappe pas à la règle, mais nous faisons tout pour proposer le meilleur produit possible.

Lorsque les gens parlent de Kotlin comme d'un langage moderne, je pense qu'ils font référence au fait qu'il a d'emblée offert le support de nombreuses fonctionnalités, comme les lambdas, ce qui n'a été proposé par le langage de programmation Java qu'à partir de sa version 8. L'objectif principal de Kotlin est d'aider les développeurs à résoudre les problèmes de programmation contemporains.

Pourquoi Kotlin est non-nullable par défaut pour les types ?

Parce que null est l'erreur à un milliard de dollars. Il ne s'agit pas seulement d'éviter les *null pointer exceptions*, mais d'inciter les développeurs à mieux anticiper en réfléchissant aux erreurs à éviter en amont. Les développeurs se retrouvent souvent à écrire du code juste pour vérifier si certaines valeurs ne sont pas null, en ignorant les cas dans lesquels les valeurs sont null et en traitant ces va-

leurs comme non existantes, ce qui est souvent renforcé par le compilateur.

Quel est l'intérêt des coroutines ?

La philosophie qui a sous-tendu la conception de Kotlin a été de réduire la surface du langage lui-même et de déléguer un maximum aux bibliothèques. Notre approche de programmation sans blocage est en phase avec cette philosophie. Nous avons ajouté le support pour les coroutines, ou *suspendable computations* au langage, en plus de fournir des bibliothèques qui intègrent des approches différentes de l'asynchrone et de la programmation concurrente.

Voici un exemple de lancement de 100 000 threads vs 100 000 coroutines

```
fun threads() {
    val jobs = 1..100000
    jobs.forEach {
        thread {
            Thread.sleep(1000L)
            print(".")
        }
    }
}

fun coroutines() = runBlocking {
    val jobs = List(100000) {
        launch(CommonPool) {
            delay(1000L)
            print(".")
        }
    }
    jobs.forEach { it.join() }
}
```

Dans le cas des threads (selon la machine), il est plus que probable que cela ne fonctionne pas. Cela va au-delà des possibilités de mémoire et crée littéralement 100 000 threads. Avec les coroutines cela ne pose pas de problème, notamment parce que cela ne crée pas 100 000 threads. L'idée der-

rière les coroutines est que les fonctions sont suspendues et reprises. On passe alors de multitâche préemptif à coopératif.

Avec Kotlin 1.2, il est possible de compiler le code en JavaScript et de l'utiliser dans un navigateur. Comment les équipes ont-elles conçu cette fonction ?

Bien que JavaScript soit l'un des langages actuellement les plus populaires, certains développeurs préfèrent travailler avec d'autres langages, notamment avec des langages statiques. Nous pensons que Kotlin est un langage statique de qualité, et, plus important, un langage concis. Kotlin offre une alternative aux développeurs ayant besoin de cibler le navigateur ou le JavaScript côté serveur (Node). Il existe également TypeScript, mais il propose une approche différente en termes de comportement statique et de nombreux développeurs trouvent que Kotlin est plus concis et plus agréable à utiliser.

Le langage Swift a cassé à plusieurs reprises la rétrocompatibilité pour différentes raisons. Comment gérez-vous l'évolution du langage et cette compatibilité ?

Notre but était, et reste, de conserver la compatibilité binaire et nous faisons tout pour y parvenir. En principe, cela ne devrait pas avoir d'impact sur l'évolution des fonctionnalités. De plus, lors de la phase de développement de Kotlin, nous avons proposé un plan de migration par étapes et délivré des avertissements concernant l'obsolescence d'une version, en indiquant que les versions suivantes supprimeraient un aspect spécifique (en le considérant comme une erreur). Tout ceci en fournissant les outils correspondants via IntelliJ IDEA pour simplifier la migration. Cela fonctionne bien et nos utilisateurs ont beaucoup apprécié cette démarche.

Réalisez-vous des benchmarks entre Kotlin et d'autres langages ?

En ce qui concerne la vitesse du compilateur, Java est toujours plus rapide que Kotlin, mais nous nous sommes considérablement améliorés et ne sommes plus très loin de la vitesse de Java. Chaque nouvelle version (y compris 1.2.50, version la plus récente) marque une avancée dans ce domaine. En termes d'environnement d'exécution, il n'y a absolument aucun impact de performance et plusieurs comparaisons ont été faites par des tiers externes qui ont validé cela. •

GitHub : stop ou encore ?

Début juin, la rumeur d'un week-end est devenue réalité : Microsoft rachète GitHub pour 7,5 milliards ! Immédiatement, les réactions se sont multipliées et souvent contrastées : les pour, les contre, les "on verra bien".

GitHub n'est pas la seule plateforme pour faire des dépôts Git mais c'est la plus connue et la plus utilisée.

Est-ce un rachat si étonnant ?

Microsoft avait tenté l'aventure avec CodePlex, une plateforme de partages de codes et de projets. Mais la plateforme resta confidentielle. L'éditeur a par la suite mis de plus en plus de codes sur GitHub.

Depuis quelques mois, Microsoft veut s'adresser de nouveau aux développeurs et pas seulement avec Visual Studio ou C#. L'expérience Windows Mobile a laissé des traces chez certains dévs ainsi que le flou sur son avenir, durant de nombreux mois, alors que les apps et services pour Android et iOS sortaient régulièrement.

Microsoft avait du mal sur le développement mobile ? Le rachat de Xamarin permettait à l'éditeur de renforcer son offre et d'attirer les développeurs mobiles Android et iOS. Ce n'est pas parce que Windows Mobile est mort qu'il ne faut pas garder les développeurs sur les outils Microsoft.

Finalement, le rachat s'est plutôt bien déroulé et les outils continuent à évoluer à bonne cadence. Ils sont intégrés à l'offre Visual Studio. La partie tarifaire, parfois considérée comme un frein, a été largement remaniée pour rendre gratuits la plupart des services et outils de la plateforme.

GitHub est un énorme morceau, de la taille de Nokia, grosso modo racheté au même prix. Et cela concerne des dizaines de millions de développeurs, de projets et de lignes de codes. GitHub est LA plateforme référence pour de nombreuses entreprises. Pour Microsoft, comme dit durant la conférence de presse qui a suivi l'annonce du rachat, il y a des opportunités à développer de nouveaux services pour les développeurs mais surtout pour les entreprises.

Pour Microsoft, il y a donc une logique à ce rachat : renforcer le contact avec les développeurs

tout en proposant de nouveaux services aux entreprises, sans liens directs avec ses propres plateformes.

Cependant, 7,5 milliards, il faudra bien les rentabiliser et GitHub doit rester une structure indépendante. Et ce dernier point est très important pour rassurer les utilisateurs et les entreprises. Même s'il faudra rapidement clarifier la stratégie, la roadmap et voir comment les équipes seront impactées. Microsoft a procédé à plusieurs changements, notamment sur le patron de la société. Un ancien de Xamarin a pris la tête de GitHub.

Ne pas brusquer les développeurs

Dès l'annonce officielle, les commentaires ont commencé à tomber ici et là. Quelques jours après le rachat, les statistiques de GitLab, un des principaux concurrents, montrent une migration de 250 000 projets... A voir si la tendance se poursuit. Si le million de projets est dépassé, oui on pourra commencer à parler de fuites.

Mais quand on regarde l'immensité de la base GitHub c'est finalement un minuscule grain de sable, car la plateforme pèse :

- 28 millions d'utilisateur
- +58 millions de répertoires

GitLab n'a pas connu une histoire de tout repos. En février 2017, une erreur d'un sysadmin avait provoqué la perte de 6 heures de données.

SourceForge, eh oui l'historique des dépôts, est toujours là, même s'il a perdu depuis bien longtemps sa suprématie. Une alternative crédible, pourquoi pas ?

A Microsoft de montrer que GitHub restera GitHub. Mais on peut aussi s'interroger sur la pérennité du service de référentiels de projets : Team Foundation Server. Pour le moment, rien n'indique une fermeture ou un passage à GitHub.

Pas de problème selon la fondation Linux

Pour la fondation Linux, qui a réagi rapidement, ce rachat n'est pas un problème. Le clivage propriétaire – open source est dépassé. Le temps où Ballmer, alors patron de MS, criait que Linux était un cancer, est bien loin. Aujourd'hui, Microsoft est membre de la fondation Linux, collabore à de nombreux projets ouverts, travaille avec les communautés, ouvre de nombreux codes et projets. Et ce rachat est l'illustration de l'importance de l'open source pour Microsoft. C'est un point souvent mis en avant par celles et ceux qui ne critiquent pas ce rachat.

Et n'oublions pas que Microsoft avait activement aidé GitHub à développer Electron.



VALENTIN MARIETTE : GITHUB AVANT TOUT

Valentin travaille chez Artifakt. Il est passionné de nouvelles technos et particulièrement de Cloud et de l'automatisation du déploiement des architectures.

Comment utilises-tu GitHub au quotidien ?

Au quotidien, je me sers de Github pour héberger ou partager mes sources de mes projets personnels. Je n'utilise que le côté gratuit de Github. Aucune grosse fonctionnalité ni de repository privé.

Avant de passer à Github as-tu regardé les alternatives ?

Oui, Gitlab en instance privée et un peu de svn pendant mes études.

Que t'inspire le rachat de Github ?

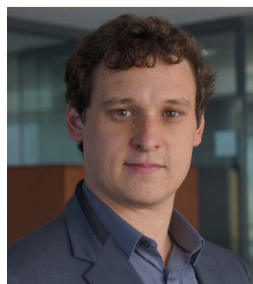
J'utilise Github depuis quelques temps et qu'il appartienne à Microsoft ou non ne

change rien. La politique de Microsoft a beaucoup changé avec Satya Nadella. De nombreuses décisions ont été prises dans ce sens, notamment l'ouverture de certaines technologies qui ne l'auraient pas été il y a quelques années. Une grande majorité des projets commu-

nautaires ou non de Microsoft sont hébergés sur Github.

Si la situation change, es-tu prêt à migrer ?

À moins que l'offre gratuite disparaisse, je ne migrerai pas.



JÉRÉMIE ROBERT DE CCI FRANCE

Avec les enjeux juridiques actuels tels que le Cloud Act, nous avons opté pour un hébergement en France, géré par nos soins. La version conteneurisée de Gitlab est simple à faire évoluer. Docker est aussi exploité grâce à Gitlab Runner, qui permet de lancer les jobs pour l'intégration continue.

Je suis Jérémie Robert, Directeur technique chez CCI France. C'est donc avec un réel intérêt que j'ai suivi les dernières actualités concernant le monde des forges logicielles.

Chez CCI France, nous avons fait le choix d'arrêter Github en faveur de la solution Gitlab. Depuis la version 10 de la solution open source, sortie en septembre 2017, nous avons vu une véritable évolution qui nous a fortement convaincus.

Concernant les fonctionnalités, Gitlab n'a rien à envier à Github, surtout concernant la philosophie devops avec un ensemble d'outils prêt à l'emploi native-ment (Gitlab CI/CD, intégration Kubernetes, Auto Devops...).

Cette philosophie revêt une importance particulière au sein du réseau CCI. Par exemple, afin de rester proche des équipes opérationnelles, je facilite et j'automatise au maximum les process grâce aux bonnes pratiques du mouvement devops.

GitLab est ainsi un outil socle sur lequel je peux m'appuyer en tant que Directeur et devops. Je peux ainsi bénéficier des différents avantages apportés par ce courant : une maîtrise de la chaîne de A à Z, la possession de toutes les clés pour résoudre rapidement les blocages et des échanges plus ouverts avec les équipes.

Cette migration nous a donc permis non seulement de retrouver le contrôle sur nos données, mais aussi d'améliorer nos processus de déploiement. Seule la communauté reste un point fort en faveur de la solution rachetée par Microsoft.

J'espère donc que Gitlab gagnera en popularité, ce qui sera à mon sens bien mérité. Les projets open source pourraient ainsi enfin être hébergés sur une forge qui respecte l'esprit d'ouverture.

Sondage express Programmez!

Le rachat de Github ne vous laisse pas indifférent.

Vous êtes :

33 % à continuer à utiliser Github mais à surveiller les éventuels changements ;

33 % à penser à migrer vers GitLab ou équivalent ;

28 % à dire que cela ne va rien changer à vos projets mis sur Github.

Ninjacat n'est pas très populaire : vous êtes **6 %** à le préférer à Octocat.





François Tonic

CARTES

guide

Guide 2018

des cartes makers, IoT et pour mini-PC

mise à jour

Avertissement : toutes les cartes présentées ne sont pas forcément faciles à trouver ou à des prix trop élevés. Toutes les cartes ne sont pas prises en compte.

Dans le n°206, nous avons dressé un panorama des cartes de prototypage, makers, IoT ainsi que de celles pour construire des barebones et mini-PC. Il est parfois difficile de choisir une carte de prototypage / IoT pour son projet ou même pour créer un mini-PC. Il en existe des dizaines et chaque mois, de nouvelles apparaissent. Comment choisir ? Quelle carte pour quel usage ? Oui, Arduino et Raspberry Pi sont toujours les références mais en réalité, de nombreuses alternatives existent et sont souvent plus intéressantes.

Comme nous l'avons dit en introduction, de nouvelles cartes sortent chaque mois. La plupart d'entre elles ne sont pas disponibles en France, ou de manière confidentielle, ou à des prix élevés. Les ODR01D commencent à être bien distribuées depuis 18 mois ce qui n'est pas le cas des cartes telles que les C.H.I.P., les Pine64, VoCore ou encore Omega2. Les quelques boutiques spécialisées et Internet vous aideront. N'hésitez pas à comparer les prix.

Ce guide n'a pas été simple à construire. Comment catégoriser certaines cartes ? Elles peuvent être présentes dans plusieurs usages. Nous avons volontairement créé trois catégories.

Il n'est pas simple créer des catégories ou des usages types. Certaines cartes sont très généralistes et peuvent s'adapter à tous les usages ou presque. Certaines seront présentes dans toutes les catégories.

Nous avons, arbitrairement, établi 4 catégories :

- Prototypage, DIY (Faire soi-même), maker ;
- Mini-PC, mini-serveur ;
- IoT finalisé, industrialisation et wearable.
- Boards + capteurs intégrés.

1 Les cartes de prototypage rapide/maker

Pour créer rapidement un POC, un petit IoT, nous avons l'embaras du choix. Les cartes les plus courantes sont les multiples déclinaisons de l'Arduino avec des caractéristiques variées. Les clones sont particulièrement intéressantes mais pas toujours de qualité.

En dehors de l'Arduino, pourquoi ne pas opter pour une ESP8266 ? Peu chère, puissante, WiFi par défaut. Les ESP proposent une plateforme matérielle mature et stable. Et il existe de multiples firmwares. Vous pouvez les coder à la manière d'une Arduino. Ces cartes sont moins permissives qu'une Arduino, le code et le montage doivent être propres, et les GPIO sont moins nombreux.

Ces cartes sont parfaitement adaptées aux petits projets / IoT, pour automatiser des objets, etc. Inutile de mettre une carte surdimensionnée. A vous de trouver le bon design de carte pour votre projet.

Les dimensions et la chaleur dégagée sont deux critères de choix, ainsi que la consommation. Vous pouvez opter pour une ATtiny même si les ressources matérielles sont bien plus limitées.

Si vous avez besoin d'une carte plus puissante, vous devrez opter pour des modèles premium. La Raspberry Pi reste la référence. Il existe de nombreuses programmes.com



alternatives, parfois plus ou moins compatibles sur les GPIO et le code. Vous pouvez aussi opter pour des plateformes très différentes telles qu'une Gadgeteer, une Beaglebone ou la récente Asus Thinker Board.

Attention tout de même aux dimensions de la carte selon les contraintes du design de votre objet. N'hésitez pas à regarder les formats « nano » : Pi Zero, Nano Pi, Onion, etc.

Comment choisir la bonne carte ? De nombreux critères peuvent être utilisés :

- Le prix ;
- La notion d'open hardware de la carte ;
- La communauté ;
- Les outils et le modèle de développement ;
- Disponibilité en France ;
- Les tutoriels et projets facilement trouvables ;
- La facilité à connecter et à coder les capteurs ;
- Consommation, dimension, poids.

Pour des montages ou pour des objets du quotidien (automatiser un jardin, une lumière, etc.), dans la plupart des cas, une carte Arduino suffira.

Vous pourrez, si besoin, utiliser une carte plus puissante telle qu'une Raspberry Pi en serveur pour récupérer les données ou manipuler les fonctions de votre objet.

La gamme Arduino & compatibles

La gamme Arduino est très large : de la simple Uno à la Leonardo en passant par la Nano, Pro Mini, MKR1000, Zero, Due, ou encore la Mega et la Yun. Sans oublier toutes les cartes compatibles telles que la Tinyduino, la Gemma, la Flora, LilyPad, etc. Toutes ces cartes ne sont pas forcément adaptées à tous les projets. Les Nano, Pro Mini, Tinyduino sont idéales pour des objets à dimensions réduites. Elles permettent d'optimiser la taille de votre montage électronique, mais vous aurez des contraintes avec des ressources très limitées et un nombre limité de broches (GPIO).

	Uno	Nano	Mega	Tinyduino	LilyPad
Microcontrôleur/CPU	Atmega328p	Atmega128 Atmega328p	Atmega2560	Atmega328p	Atmega128v Atmega328p
Vitesse	16 MHz	16 MHz	16 MHz	8 MHz	8 MHz
GPIO analogiques/ digitaux	6/14	8/14	16/54	6/14	6/14
Réseau	Non	Non	Non	Non	Non
Flash	32 ko	16 ou 32 ko	256 ko	32 ko	16 ko
USB	USB	Micro	USB	Micro	-
Voltage (volt)	5/12	5/9	5/12	2,7/5,5	2,7/5,5
Programmation	C	C	C	C	C
Prix	7-9 €* *****	-2 €* ****	-10 €* *****	14,95 \$ *****	3 €* ***
Notre avis**	*****	****	*****	*****	***

* tarif clone et non de la carte officielle

** Un avis est toujours subjectif. La notation sur 5* se base sur notre expérience avec la carte.

L'Arduino est idéale pour le prototypage rapide, la réalisation d'objets, débiter en programmation et l'électronique. Mais la plateforme a plusieurs limites :

- consommation ;
- code qui tourne en boucle : 1 seul code à la fois ;
- puissance limitée du microcontrôleur.

Mais l'Arduino suffira dans de nombreux montages. Son modèle de programmation est simple et très accessible. Et la communauté est très riche. Le fait que la plateforme soit open source est un avantage certain car il existe de nombreux clones, plus ou moins fiables mais à des tarifs imbattables.

Attention tout de même : certains modèles sont spécifiques à Arduino et sans clone telles que les cartes de la série MKR. Ces cartes sont souvent dédiées à des usages précis mais leur utilité dans un projet reste à définir et elles sont souvent vendues à des tarifs élevés.

Arduino IDE, l'outil de développement référence d'Arduino, supporte des dizaines de cartes et sa prise en main est rapide. Certaines cartes demandent un peu de manipulation pour les installer, mais dans la plupart des cas, c'est une histoire de 3-4 clics de souris. Et le code est en général facilement repris d'une carte à une autre. Parfois, il faut tout de même modifier le code ou changer de librairies. Si vous cherchez un modèle de développement de type JavaScript, Python ou Java, vous devrez regarder sur des cartes de types ESP, Raspberry, Nano Pi, etc. Il existe des systèmes très simples pour connecter les capteurs. Seeed Studio est le fabricant le plus connu avec sa plateforme Grove. Il suffit de connecter le shield Grove. Les capteurs se branchent en Plug&Play. Tout est intégré et sans soudure ! Grove est un peu cher (un kit complet est vendu env. 40-50 €), mais vous

gagnez en simplicité. Il existe aussi l'excellente alternative à Grove : les composants Tinker du Français Hack&Invent que nous avons testé il y a quelques mois.

Si vous souhaitez construire votre propre carte Arduino, rien de plus facile... ou presque. Il faudra acquérir le microcontrôleur et flasher le bootloader. Bref, vous pouvez vous amuser !

Les autres boards

Arduino n'est pas l'unique possibilité, vous pouvez utiliser bien d'autres boards des plus puissantes telles que les Raspberry Pi à la minuscule ESP8266. Tout dépend de vos envies et exigences de ressources. Pour notre part, si une Arduino suffit pour notre projet, nous choisirons une Arduino. Si vous avez besoin d'un peu plus de puissance ou tout simplement de pouvoir exécuter plusieurs programmes ou utiliser d'autres langages, vous devrez utiliser d'autres plateformes.

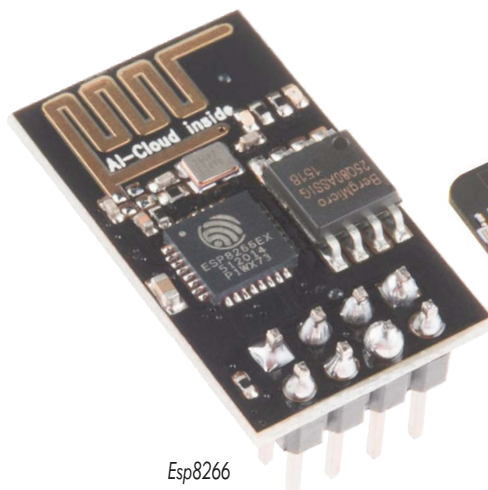
	Pi Zero W	C.H.I.P.	Omega2+	ESP8266	Carte Attiny85*
Microcontrôleur/CPU	ARM	ARM	MIPS	RISC	Attiny85
Vitesse	1 GHz	1 GHz	580 Mhz	80 MHz**	20 Mhz
GPIO analogiques/ digitaux	40 en tout	80 en tout	18	selon modèle	6
Réseau	WiFi Bluetooth	Bluetooth	WiFi	WiFi	-
Flash	-	4 Go	32 Mo	512 ko à 16 Go**	8Ko
USB	2x micro	USB	USB	Micro/rien**	connecteur
Voltage (volt)	3,3/5	3/5	3,3	3,3**	5
Programmation	modèle Pi	très ouvert	très ouvert	très ouvert	compatible Arduino
OS par défaut	Linux	Linux	Linux	-	-
Prix	10 \$	9 \$	13 \$	-2 à 10 \$**	env. 2 €***
Notre avis	****	**	****	*****	*****

* Modèle pris en compte : Digispark, nano-board complète avec connecteur USB

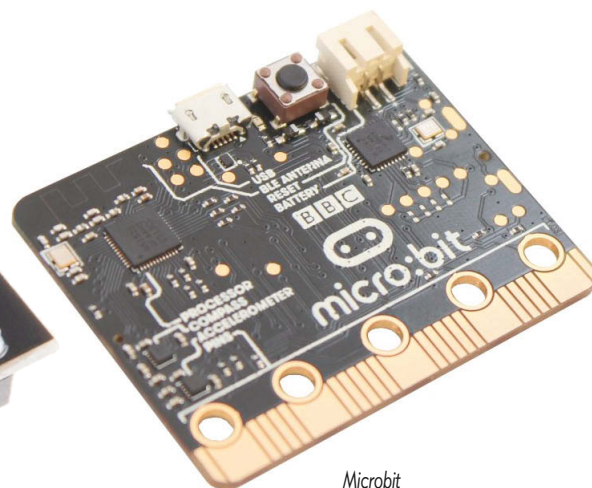
** selon les modèles

*** par lot de 25 minimum. A l'unité, le tarif sera un peu plus élevé

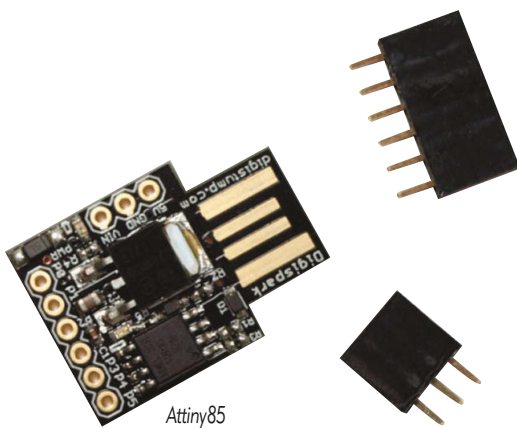
La Pi Zero est une version minimaliste de la Raspberry Pi. Elle peut être une bonne alternative pour un objet ou un projet un peu complexe. La version la plus récente embarque Wifi et Bluetooth, mais le tarif est à doubler. La C.H.I.P. est plus complète et offre un excellent ratio prix/ressources. Elle est encore jeune, mais elle est déjà une alternative très crédible.



Esp8266



Microbit



Attiny85



Odroid

Les ESP8266 sont des nano board WiFi créées par Espressif System. Aujourd'hui, il en existe de très nombreux modèles. Les prix sont particulièrement intéressants. Le domaine de développement est très souple et une multitude de firmwares sont disponibles. Pour nous, il s'agit aujourd'hui de la meilleure alternative aux Arduino et surtout, les ESP sont taillées par la production en masse. Les défauts des ESP pour un maker sont principalement le manque de GPIO, même si on peut trouver des solutions et un voltage de la carte pouvant être handicapant : 3,3 V. Vous serez capable d'optimiser les dimensions de votre projet et la consommation sera elle aussi largement améliorée. Et surtout, la connectivité WiFi par défaut est un gros avantage.

Les cartes Attiny peuvent être intéressantes : peu chères, dimensions réduites, compatibles Arduino. Oui, ces cartes possèdent peu de GPIO mais elles sont idéales pour débiter dans l'univers des IoT et maker.

Pour une découverte très facile et intuitive des IoT et de l'électronique, nous avons très récemment testé la carte micro:bit, carte anglaise. Certes, elle est plus limitée qu'une Arduino ou une ESP, mais tellement sympathique. Autre kit intuitif, Wio Link. Il s'agit d'une plateforme Plug&Play basée sur une ESP8266. En 5 minutes vous pouvez interagir entre la carte, des composants et une application mobile. La programmation est très ouverte : JavaScript, Java, Python, etc. La carte proprement dite est vendue 14 \$.

2 Projets avancés, wearable, IoT industriel, bornes

Sauf à créer sa propre plateforme matérielle, ce qui est toujours possible et pas forcément aussi cher qu'on le pense, même en série limitée, quand vous souhaitez créer des objets de dimensions réduites, wearables ou un IoT en série ou des bornes interactives, le choix ne manque pas !

	Petite série	Wearable	Borne	Domotique	IoT grande série
ESP8266	*****	****	*	***	*****
Raspberry Pi, ODROID, C.H.I.P., etc.	****	*	*****	*****	*
LilyPad, TinyDuino, Xadow, etc.	****	****	**	**	**
AirBoard	****	***	***	****	****
Omega2+	***	**	*	***	**
ATTiny	** / ***	***	*	***	*
NanoPi-NEO 2	***	*	***	***	*
MediaTek MT3620	*	*	***	***	***



Tinyduino

Pour des projets IoT, de bornes ou de domotique, le choix n'est pas forcément très simple. Pour une borne ou au cœur d'un distributeur, les Raspberry Pi et équivalents, sont bien adaptés. Car il faut de la ressource matérielle, un système d'exploitation, une connectivité, sans pour autant alourdir la facture. Même si ces cartes sont assez volumineuses, la taille des bornes permet leur usage.

En IoT, au sens large, la taille de l'objet, son interactivité, sa connectivité, sa consommation, son prix seront des critères de choix. Des cartes de type Intel Edison ou Curie sont taillées pour les IoT et pour la phase industrielle, pour de petites séries, pas forcément, à cause des contraintes techniques et du prix de la carte. Un ESP8266 pourra s'installer à peu près partout, à condition de ne pas surestimer sa puissance. Pour des objets de domotique, des IoT du quotidien, les ESP sont parfaits. Pour des projets et objets nécessitant un peu plus de ressources, par exemple en pins de branchement, des cartes comme la AirBoard sont intéressantes. Un avantage est sa compatibilité avec les infrastructures réseau de type SigFox.

Pour des projets en grande série, vous pouvez aussi créer votre propre carte ou vous tourner vers des fournisseurs industriels tels que NXP.

Pour des projets wearable ou domotiques, AirBoard, Xadow, TinyDuino, LilyPad, ESP, etc. conviennent très bien. Là, tout dépend de vos affinités. Si vous êtes habitué à Arduino et que la plateforme vous suffit, les variantes embarquées sont nombreuses. Au-delà, des plateformes plus puissantes comme ESP ou Onion Omega sont des solutions à considérer.

L'Onion Omega2+ est une carte récente qui peut se comparer facilement à un C.H.I.P., fonctionne sur un processeur MIPS, embarque 128 Mo de RAM, 32 Mo de stockage, de l'USB 2, du WiFi, des GPIO. Le potentiel est là, à voir comment la carte évoluera dans le temps et si sa communauté va croître.

Sur la partie IoT on notera l'arrivée dans l'année des premières cartes équipées de Microsoft Azure Sphere OS. Il s'agit d'un système Linux taillé pour les IoT. MediaTek a annoncé les premières cartes développeurs pour septembre

mais à un prix largement supérieur (+80 \$) aux quelques dollars annoncés lors de l'annonce, pour la production en grande quantité. Les premiers IoT devaient être déployés à la fin de l'année.

3 Cartes + capteurs intégrés

Pour démarrer encore plus rapidement, comprendre le fonctionnement des cartes et éviter tout montage, il existe des solutions intégrant la board et des capteurs. Nous avons limité notre panorama à 4 cartes. Elles sont toutes de qualité. Selon le modèle les capteurs disponibles sont plus ou moins nombreux. La Matrix Creator est la plus complète mais aussi la plus chère.

Elle nécessite une Raspberry Pi pour fonctionner. Le Sensor Hat est la carte la plus connue. Il possède de nombreux capteurs, elle est moins chère que la Matrix Creator et de petite taille.

	Matrix Creator	SensorHat	Adafruit Circuit Playground Express	Micro:bit
CPU / microcontrôleur	Cortex-M3 Spartan-6	-	Cortex-M0	Cortex-M0
Capteurs	ultraviolet accéléromètre magnétomètre température couronne LED Humidité Gyroscope pression	Gyroscope accéléromètre magnétomètre température pression humidité matrice LED mini-joystick	accéléromètre température magnétomètre luminosité 25 LED 7 zones tactiles 2 boutons NeoPixel 10 LEDs	accéléromètre magnétomètre 25 LED 2 boutons
Carte supplémentaire	Raspberry Pi	Raspberry Pi	-	-
Microphone	8	-	1	-
Protocoles communications supportés	zigbee Z-Wave NFC	-	-	BLE
GPIO	17	-	14	19
Prix	env. 84 €	35 €	env. 25 €	19-22 €

Après il s'agit d'une histoire d'affinité. Nous aimons beaucoup l'Adafruit Circuit Playground pour sa compatibilité Arduino et sa facilité de programmation. Idem pour la Micro:bit qui est une carte facile d'utilisation. Elle a été massivement distribuée dans les écoles anglaises.

4 Construire un mini ou nano-PC ou un mediacenter

On parle beaucoup de mini-PC, de barebone, de fracture numérique, de rétro-console, de nanoserveur, etc. Leur point commun se base sur des cartes de type Raspberry, ODROID, C.H.I.P., pcDuino, etc. Elles coûtent généralement entre 40 et 80 € et proposent une puissance suffisante pour beaucoup de fonctions pour une utilisation non professionnelle et hors gaming. Oui beaucoup d'utilisateurs n'ont pas besoin d'un PC ou d'un Mac coûtant au moins 600 ou 700 € minimum et dont la puissance sera sous-exploitée. Éventuellement des ordinateurs de type ChromeBook, mais il faut une connexion réseau et le tarif débute à 250 – 290 €. Avec des cartes, ayant un excellent ratio prix/puissance, on peut construire un vrai PC à -100 € ! Même pour la partie serveur, aujourd'hui, des solutions particulièrement excitantes existent et nous impressionnent par leur puissance et surtout par un tarif défiant toute concurrence.

	Raspberry Pi 3+	ODROID-XU4	Banana Pro	NanoPi Neo 2
CPU	Cortex A53 1,4 GHz	Exynos-544(1)	Cortex A7 1 GHz	Cortex A53
CPU 64 bits	oui	oui	Non	oui
GPU	VideoCore IV	Mali T628 MP6	Mali400MP2	Mali-450
Mémoire vive	1 Go	2 Go	1 Go	512 Mo
Stockage	SD	SD eMMC	SD	SD
Port SATA	non	non	oui (v2)	non
Ethernet	1 Gb	1 Gb	1 Gb	1 Gb
Sans-fil	Bluetooth WiFi	non	WiFi(2)	oui
USB	4	2xUSB 3 USB 2 Host	2	1
Vidéo	HDMI 1.4	HDMI	HDMI 1.4	non
OS supportés	Linux Windows 10 IoT Android	Ubuntu Android	Linux Android	Linux
Ecosystème/ communauté en France	****	*	**	**
Tarif (carte seule)	40-45 €	env. 70 €	45-50 €	25 €

	pcDuino3B	Orange Pi Prime	C.H.I.P.	VoCore2 Ultimate(3)
CPU	ARM Cortex A7 1 GHz	Cortex A53	ARMv7 1Ghz	MT7628AN MIPS
CPU 64 bits	Non	oui	Non	oui
GPU	Mali 400 Dual Core	Mali-450	Mali 400	aucun
Mémoire vive	1 Go	2 Go	512 Mo	128 Mo
Stockage	4 Go	SD	4 / 8 Go	16 Mo
Port SATA	Oui	non	non	non
Ethernet	1 Gb	1 Gb	non	100 mb
Sans-fil	non	oui	WiFi Bluetooth	WiFi
USB	2	3	2	micro USB
Vidéo	HDMI 1.4	HDMI	non	non
OS supportés	Linux Android	Linux Android	Linux	Linux OpenWRT
Ecosystème/ communauté en France	*	**	*	*
Tarif (carte seule)	60 €	-35 €(4)	9 \$	44,99 \$

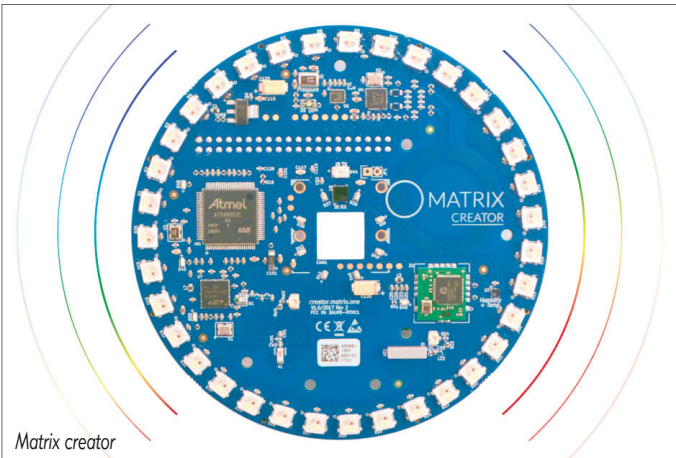
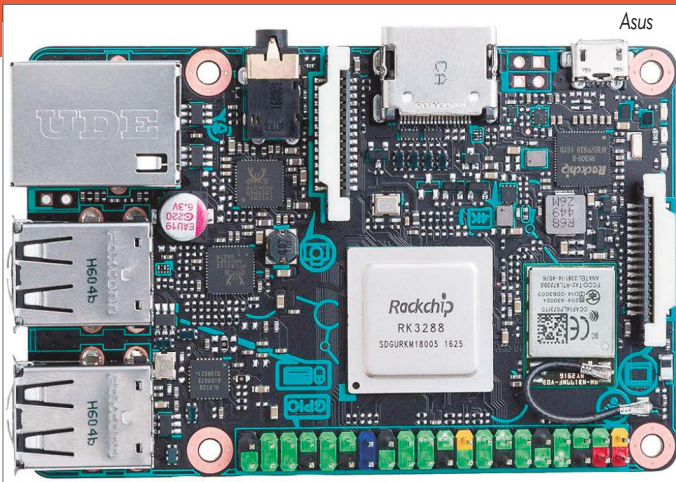
	Asus Thinker Board
CPU	4 coeurs RK3288
CPU 64 bits	oui
GPU	Mali T764
Mémoire vive	2 Go
Stockage	SD
Port SATA	non
Ethernet	1 Gb
Sans-fil	WiFi / BT 4
USB	4 x USB 2
Vidéo	HDMI
OS supportés	ThinkerOS, Android Armbian, DietPi Yocto...
Ecosystème/ communauté en France	**
Tarif (carte seule)	+60 €

(1) Processeur Samsung avec Cortex A15 et Cortex A7

(2) Bluetooth en option

(3) Edition complète avec boîtier

(4) uniquement sur des sites spécialisés. Sinon, vous trouverez la carte à 40-45 €



Le choix ne manque pas pour créer un vrai PC et pas uniquement en classique Pi. Avec la Pi vous jouez la sécurité, la carte est facilement trouvable, elle est mature, stable, avec une communauté. Depuis 2 ans, les alternatives se multiplient. Malheureusement, elles sont parfois difficiles à trouver, avec une faible communauté, ou pire, une faible documentation. Par exemple, le pcDUINO et les CHIP sont de bonnes cartes mais les ressources disponibles sont peu nombreuses. Les ODROID se diffusent mieux en France. Elles sont particulièrement intéressantes en terme de puissances et de ressources matérielles. Autres valeurs sûres : Nano pi et Orange pi.

Plus récemment, le constructeur Asus a sorti sa propre carte : la Tinker Board. Bonnes performances, facilement trouvable, elle est encore un peu chère mais prometteuse malgré un dégagement de chaleur relativement élevé (pas propre à cette carte) et une stabilité à améliorer. Elle peut se transformer en mini-PC. La C.H.I.P. avait l'ambition de proposer une alternative à la Pi à un prix très bas, 9 \$. Elle a souffert d'une stabilité douteuse à son lancement et il fallait rajouter des shields supplémentaires pour avoir la vidéo. Problème : le shield coûtait plus cher que la carte ! Dommage. La version Pro propose des ressources améliorées mais garde les mêmes lacunes que la v1. Les cartes pcDuino peuvent être un autre choix même si nous la considérerons plus pour des montages et bornes, mais pourquoi pas ? La version 3B propose des ressources intéressantes, un Ethernet 1Gb.

Toutes ces cartes fonctionnent sur des processeurs ARM, donc il faudra utiliser des systèmes et des logiciels ARM et non x86. Mais l'offre se développe rapidement, notamment sur la partie serveur. Pour la bureautique, Internet, vous n'aurez aucune difficulté pour trouver des logiciels adaptés même si l'offre est plus limitée et tous les logiciels ne sont pas disponibles en version ARM.

Côté cartes x86, la situation bouge finalement très peu. Quelques projets ont vu le jour ici et là tel que la JaguarBoard mais elle se révéla décevante et instable. Aujourd'hui, les cartes UP sont sans doute les plus prometteuses. Elles

sont vendues à partir de 99 \$. La version haut de gamme atteint 339 \$. Mais à ces tarifs élevés, on dispose d'un processeur Celeron ou Atom ou Pentium, de 2 à 8 Go de RAM, d'une GPU Intel embarquée, stockage eMMC, 2 ports HDMI, USB 2 & 3, port SATA, 2 Ethernet 1 Gb, compatible avec Windows / Linux / Android ! Vous pouvez sans aucun problème créer votre PC !

Comment faire son choix ? Plusieurs critères sont à considérer :

- Le stockage interne, les interfaces de connexion ;
- L'ajout de stockage et les performances de celui-ci ;
- La connectivité vidéo ;
- La connectivité réseau ;
- Processeur vidéo ;
- Mémoire vive et processeur ;
- Les ports USB disponibles ;
- Les OS supportés ;
- Le dégagement de chaleur.

Sur le stockage quelques cartes proposent un stockage directement sur la carte, mais la plupart nécessitent un rajout. Ce point est intéressant : est-ce uniquement une carte SD ou une autre connectivité ? La carte SD reste un support lent. Un port eMMC ou Sata sera le bienvenu. Mais n'espérez pas obtenir les performances d'un PC standard. Vous pouvez ajouter un disque dur via le port USB, mais, là-encore, les performances seront médiocres, comme nous l'avons constaté avec le PiDrive sur une Raspberry Pi. Malgré tout, pour un usage basique, le stockage ne sera pas le critère bloquant. Par contre, la connectivité vidéo et la puce graphique seront à regarder surtout si vous souhaitez avoir un usage en rétroconsole ou en mediacenter. Les Pi3+ et ODROID s'en sortent plutôt bien.

La connectivité réseau est bien entendu un autre point essentiel. Le WiFi peut être le critère discriminant dans votre choix. Le port Ethernet est une interface classique et largement présente sur les cartes. Honnêtement, un port 10/100 suffira sauf si vous avez de gros débits à faire passer ou en mode serveur, dans ce cas, opter obligatoirement pour un Ethernet 1 Gb.

Pour l'alimentation, les claviers, l'écran, le boîtier, aucun souci à avoir ! Rien que du standard ! Malheureusement certaines utilisent des connecteurs d'alimentation peu répandus. Vous devrez acheter en plus un câble adapté.

Il existe des kits complets pour monter un ordinateur de bureau ou un portable. L'un des plus connus est le Pi-Top. Il inclut un châssis, un écran 13,3", un clavier. Une version Tout-en-un desktop est disponible, à un tarif intéressant (-160 €). Ou vous pouvez regarder le projet Pi-to-go : <http://blog.parts-people.com/2012/12/20/mobile-raspberry-pi-computer-build-your-own-portable-rpi-to-go/>

Si vous souhaitez créer un serveur, pour une petite entreprise ou personnelle, c'est possible. Nous vous conseillons des cartes assez véloces pour le faire, une Pi3 ou une ODROID, éventuellement une Pine64 si les besoins restent modestes. N'oubliez pas que vous êtes en architecture ARM et non x86. Et vous pouvez même créer des clusters sans aucun problème. En Pi, il existe plusieurs solutions matérielles et logicielles pour les monter et les gérer.

Plus audacieux, l'usage de la VCore. Impressionnante par sa taille ridicule, une pièce de 2 €, elle offre des ressources matérielles très intéressantes. Pour créer un nanoserveur (Web, serveur de fichiers, etc.) ou monter un NAS personnel, le VoCore2 est une solution à ne pas négliger même si cela peut vous surprendre. Par défaut, la carte fonctionne avec OpenWRT, un Linux largement répandu dans les routeurs. Pour un prix agressif, l'ensemble complet, édition Ultimate, propose une solution complète. La carte seule est vendue - 18 \$.

À vous de jouer maintenant !

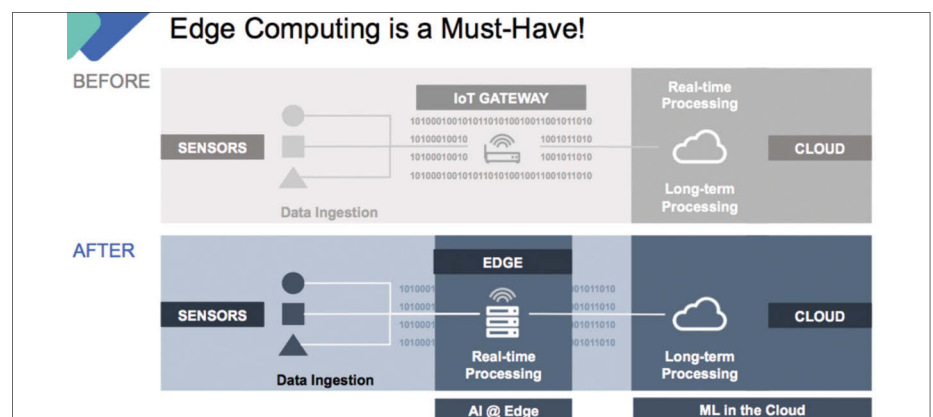
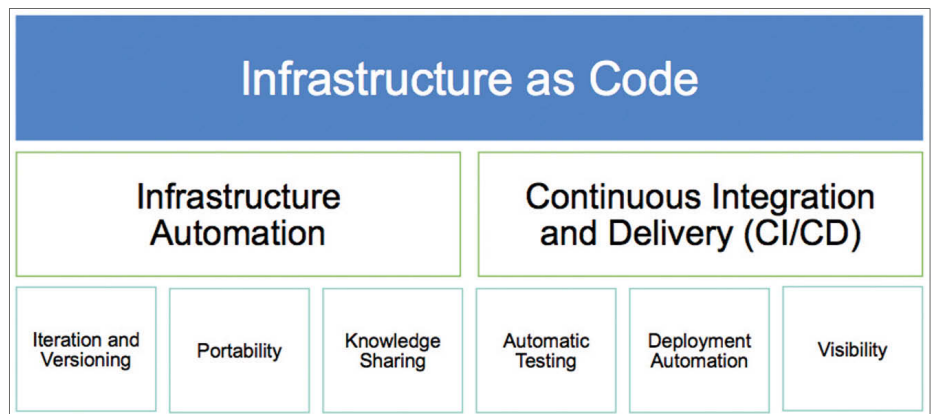
Nutanix : le développeur et le code au cœur de l'infrastructure nouvelle génération

Nutanix ne vous dit sans doute pas grand chose. Jusqu'à présent, cette société était connue pour ses solutions d'infrastructure hyperconvergée : la proposition de tout ce qu'il faut dans une seule boîte pour déployer, administrer son infrastructure. Aujourd'hui, l'infrastructure est en pleine mutation : approche Software Defined, cloud privé / public / hybride, les IoT, les conteneurs et l'Infrastructure As Code.

L'architecture applicative bouleverse elle aussi l'infrastructure telle que l'entreprise l'utilisait. Dans toute cette effervescence, plusieurs évidences s'imposent : le rôle du développeur dans l'infrastructure, la généralisation des API, des frameworks et le fait que l'infrastructure devient totalement flexible et éclatée avec les IoT et les notions de Edge Computing. Nutanix, depuis un an, fait un grand virage. Son activité historique reste d'actualité mais la R&D regarde au-delà. L'infrastructure, qu'elle soit convergée ou non, se code. Cela peut paraître surprenant mais dans les architectures actuelles, l'automatisation des infrastructures (serveurs, allocations des ressources, déploiement des applications) se réalise sous forme de script et de flux.

Sherlock : un framework IoT

Le projet a été annoncé à l'occasion de la dernière conférence Nutanix en Mai 2018 à la Nouvelle Orléans. Next. Avant de parler du projet Sherlock en détail, essayons de comprendre pourquoi Nutanix souhaite aborder la question de l'IoT. Avec les IoT, on change d'échelle. On parle de l'échelle planétaire avec 100 fois plus de composants à gérer. Les entreprises commencent à peine à maîtriser la gestion dans les datacenters ; la gestion des devices IoT va être un défi pour ces entreprises. De plus au vu de la quantité de données à traiter, pour des raisons de latence ou simplement de régulation, il ne sera par exemple pas possible de déplacer toutes les données récupérées dans un point central ou un data lake dans le cloud public. L'information devra être traitée dans les edge (magasin, usine, plateforme pétrolière, ...). Autre problème avec l'IoT, le nombre de composants nécessaires pour traiter l'information.



On entend parler de TensorFlow, MXNet, Caffe, DynamoDB, Kafka, Hive, Spark, Grafana, HBase, ..., à tout cela on ajoute les containers, les fonctions, la persistance de la donnée, ...

Le projet Sherlock veut accomplir la mission de prendre ces composants, les maintenir dans le temps, avec la possibilité d'administrer simplement l'infrastructure pour les devices IoT que ce soit dans un cloud public ou un cloud privé. En plus, Sherlock veut fournir une plateforme micro-PaaS pour que les entreprises puissent développer facilement et rapidement leurs applications sans se soucier de l'infrastructure sur laquelle elles tournent. Ce projet sera également l'occasion de qualifier du matériel adapté basé sur du Xeon D ou Core i7 par exemple, des accélérations matérielles, un format différent du format traditionnel des datacenters du type 2U, supportant le Wifi, le Bluetooth 5, Thread ou ZigBee.

Nutanix veut continuer son expansion au-delà des datacenters car la société est convaincue que le monde sera hybride et que les Edge, avec l'IoT, vont être une extension du datacenter des entreprises.

Flow

Nutanix Flow est l'offre de Software-Defined Networking de Nutanix pensée pour un monde hybride (cloud public et privé). Il s'agit d'offrir de la sécurité de type application-centrique permettant de détecter des attaques qui ne seraient pas interceptées par des produits de sécurité paramétrique. Nutanix Flow est intégré à Acropolis, et est non intrusif. Il permet de créer en quelques clics des règles de sécurité et de les gérer.

A ce jour l'outil permet déjà :

- une visualisation du réseau en fonction des applications, ainsi que de la performance et de la disponibilité du réseau, tout cela sans plugin ou agent ;
- du service chaining ou Insertion qui permet de rajouter des fonctions de type firewall dans Flow ;
- d'automatiser la configuration des VLANs des switch ToR (Top of Rack) ou de modifier automatiquement les règles d'un load balancer externe ;
- de créer des règles de sécurité permettant de micro-segmenter les applications.

De plus avec l'acquisition récente de Netsil,

Nutanix va aller plus loin. Netsil a été pensé et développé pour l'ère des micro-service et va être par exemple capable de créer une cartographie des applications déployées dans un cloud public, sans agent et en faisant de l'inspection de paquets au niveau 7 grâce à une technologie développée par l'équipe actuelle de Netsil lorsqu'ils étaient encore à l'université de Pennsylvanie.

La compagnie pense que les outils d'APM (Application Performance Monitoring) traditionnels sont plus difficiles à mettre en œuvre dans cette génération de micro-service et containers. Ces outils nécessitent de rajouter du code dans les applications mais les applications évoluent très vite et les APIs ou requêtes RPC sont les moyens de communiquer avec les applications. Un changement de code pour faire de l'APM est simplement lourd et complexe à gérer. La durée de vie des applications et l'échelle nécessitent également une approche différente et beaucoup plus dynamique. Enfin les nombres de langages (Python, Go, R, ...), frameworks (Node.js, Mesos, Kubernetes, Spark, Hadoop, ...) et bases de données (Cassandra, MongoDB, InfluxDB, ...) qui explosent nécessitent de prendre une approche différente là où Netsil va être capable de faire des analyses de plusieurs milliers d'interactions sur le réseau en une fraction de seconde grâce à des techniques avancées d'analyse des flux. Netsil va, sans code spécifique ni dépendance avec des frameworks, capturer en temps réel et analyser les interactions réseaux entre les différents services, à partir desquels l'outil pourra créer une cartographie applicative mais également les dépendances avec les services internes et externes.

Nutanix Developer ou comment faire aimer l'infrastructure aux développeurs

Les API sont partout et servent à tout. Sur la partie infrastructure et services d'infrastructure, on en trouve chez tous les fournisseurs de cloud et plateformes d'infrastructure. L'objectif est toujours le même : exposer les services, aider le développeur à intégrer les services, accéder aux fonctionnalités de la plateforme. Pour cela, Nutanix propose une page dédiée : Nutanix Developer.

interview

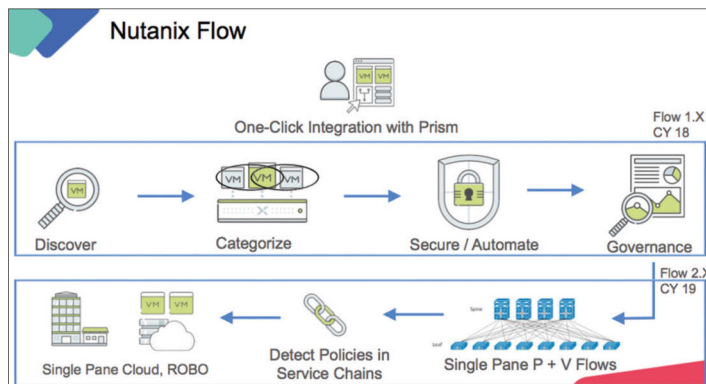
Questions – réponses avec James Karuttykaran, Directeur SE Europe du Sud, Nutanix

En quoi l'infrastructure as code est-elle un élément important pour un acteur de l'infrastructure comme Nutanix ? Et en quoi cela impacte-t-il votre vision de l'infrastructure ?

Le mouvement DevOps et l'adoption plus importante du cloud public mettent en évidence un manque d'agilité des infrastructures traditionnelles. L'agilité est le sujet que nous avons adressé depuis les débuts de Nutanix et c'est pour nous une nécessité dans les entreprises. L'infrastructure as code fait partie de l'ADN de Nutanix avec un environnement complètement exploitable en API Rest. Les acquisitions des sociétés telles que Calm.io vont permettre d'aller encore plus loin : les administrateurs systèmes pourront automatiser leur infrastructure sans avoir à taper une seule ligne de code.

Depuis 1 an, vous parlez de plus en plus aux développeurs ? Est-ce une simple distorsion visuelle ou une réalité profonde ? Parler aux développeurs est-ce plus difficile que de parler au sysadmin et aux responsables d'infrastructures ?

Oui c'est bien une réalité, durant ces dernières années nous nous sommes attelés à régler les problèmes d'infrastructure en la rendant plus agile jusqu'à être invisible pour les utilisateurs. Nos clients nous ont fait comprendre qu'il y a un vrai problème entre les équipes de développement et les équipes d'infrastructure. L'infrastructure est perçue comme un frein à l'agilité et les développeurs sont tentés de consommer du cloud public. Nous souhaitons avec Calm proposer un outil commun qui réconcilie les 2 équipes, aussi bien les dev que les ops, et qui simplifie la mise en œuvre d'applications ainsi que la gestion du cycle de vie de l'application (build,



test, deploy, maintain), et ce quelle que soit la plateforme, publique ou privée.

En infra as code, les outils sont cruciaux (Salt, Ansible, Terraform, Kubernetes, Puppet / Chef, Docker, etc.), comment travaillez-vous avec ?

Nous avons créé des partenariats technologiques avec certains acteurs qui nous ont permis de développer des intégrations et des livres blanc. Par exemple Docker, Kubernetes et Terraform sont maintenant considérés comme des standards que nous supportons nativement. De plus, nous souhaitons laisser la liberté à nos clients d'utiliser la plateforme Nutanix avec le maximum d'outils du marché, là où certains acteurs de l'infrastructure peuvent imposer leur stack complète.

L'autre tendance forte est la partie serverless que l'on voit un peu partout (AWS, Google, Microsoft, IBM, etc.), qu'est-ce que cela change pour vous ?

C'est un sujet en constante évolution que nous suivons attentivement en attendant que le marché se standardise car les framework serverless sont nombreux (OpenFaaS, Kubeless, Fission, Nuclio, Iron Functions, ...). Les clients utilisant Calm ou Acropolis Container Services (ACS) peuvent s'en servir pour automatiser le déploiement d'une plateforme serverless. Un des

premiers produits Nutanix qui devrait proposer des fonctions as a service est le projet Sherlock.

Les IoT et le Edge Computing impactent beaucoup le datacenter. Quelle est votre approche chez Nutanix ?

Ces deux sujets sont étroitement liés pour nous. Jusqu'à maintenant une grande partie des vendeurs d'infrastructures publiques ou privées conseillent de récupérer les données dans un point central comme un data lake par exemple pour ensuite les analyser. Mais si on regarde dans le détail, en 2017 il existe 35 millions de noeuds dans les datacenter publics et privés en comparaison avec 3,1 milliards de devices IoT dans les entreprises. En 2020 il devrait y avoir 7,5 milliards de devices IoT et le trafic généré par les devices IoT vers le cloud sera de 600 ZB. L'IoT va être confronté à de nombreux problèmes comme la latence, l'autonomie en cas de coupure, les règles de conformité et la difficulté de gestion des devices. Pour résoudre ces problèmes, nous pensons qu'une grande partie des données peut être traitée dans les edges. C'est dans ce but que nous avons récemment annoncé notre projet Sherlock qui consistera à fournir une plateforme intégrée qui permettra de développer facilement et rapidement des applications pour l'IoT sans se soucier des problèmes d'infrastructure. •

La réalité augmentée by Vuforia



La réalité augmentée (aussi appelée AR) permet d'inclure en temps réel des informations qu'elles soient textuelles, vidéos, photos ou objets en trois dimensions, en surimpression de notre vision de la réalité. Nous changeons de part le fait, plus notre perception de la réalité que la réalité elle-même.

Note

Pour installer l'application, il vous faudra au minimum un android 4.4+ "Kit kat" ou un IOS 9+. Ces contraintes sont principalement imposées par le Framework Vuforia.

Vous trouverez dans le numéro du mois de juin dernier un article traitant plus longuement de la réalité augmentée. En le lisant, vous en saurez plus sur les paradigmes de l'AR, ses champs d'applications et les différents acteurs présents aujourd'hui sur ce marché.

Mon but

Cet article n'a pas pour but de présenter en détails l'AR, ni même de proposer une démarche pour construire sa propre application. Pour cela il existe des tutoriaux très détaillés, des démonstrations vidéo etc. très bien faites. Non, il a pour objectif de vous donner mon retour sur ma montée en compétence sur le sujet. Pour vous permettre d'éviter de tomber dans le même travers et vous permettre de constituer au mieux votre équipe de développement. L'idée de base est de proposer une App (compatible Android et iOS) qui permettra à un smartphone (en mode caméra), de montrer du contenu 3D dès lors qu'une image prédéfinie est reconnue. Cette application

est développée pour un magazine spécialisé sur l'Egypte, elle est à caractère publicitaire et vous le verrez, elle est très portée sur la culture de l'égypte antique. Elle va permettre aux utilisateurs de visualiser des modèles 3D et de se rendre compte des éléments architecturaux égyptiens en 3D directement sur une simple page de magazine avec votre téléphone. Extra, non ?

Vuforia me permettra de modéliser les items à reconnaître ; de gérer la reconnaissance d'objets et d'images ; de réaliser le positionnement du contenu supplémentaire sur l'écran. Enfin Je devrais être en mesure de faire apparaître des boutons virtuels, pour diriger l'utilisateur vers d'autres sections de l'App ou afficher d'autres contenus.

Vuforia

Vuforia est un fork d'openCV, LA librairie principalement utilisée pour le traitement et la reconnaissance d'images. Elle permet d'aller de la reconnaissance faciale à l'extraction de modèle 3D en passant par du

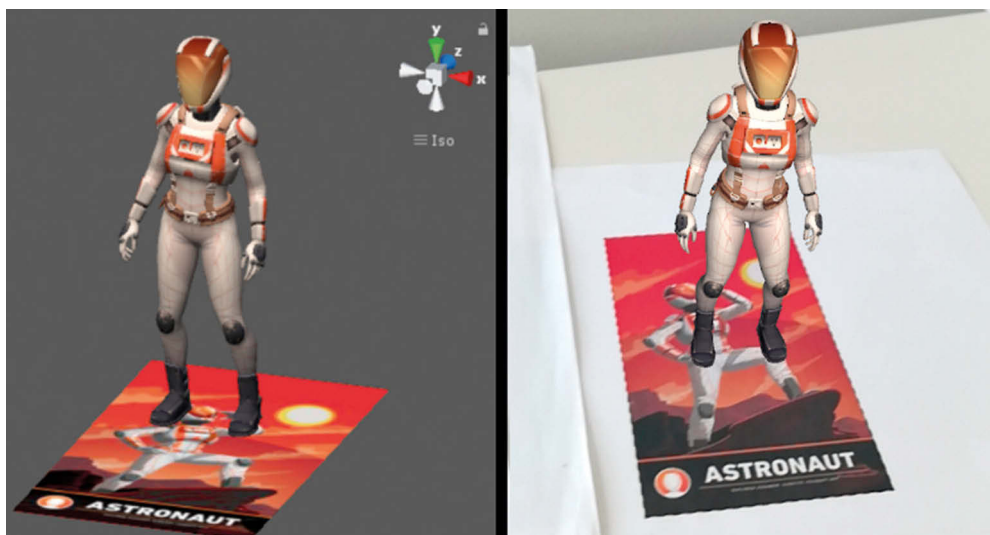
tracking vidéo, et encore ce n'est qu'un échantillon des algorithmes proposés par openCV. Elle compte également plus de 47.000 personnes actives dans sa communauté. L'avantage de Vuforia SDK réside dans le fait qu'il va s'occuper du calibrage entre le monde réel et le monde virtuel. Ainsi il va pouvoir positionner le contenu avec précision en fonction de la perspective visuelle de l'utilisateur à travers son téléphone. Cela évitera pas mal de calculs mathématiques relativement complexes.

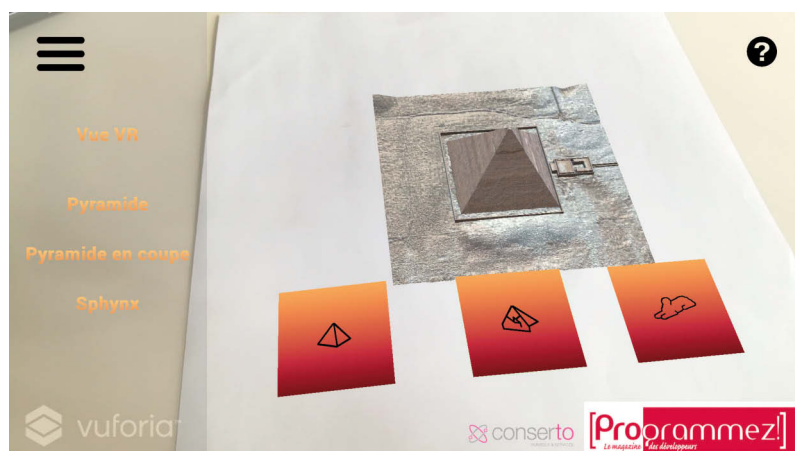
Même si ce framework peut être porté nativement que ce soit sur Android ou iOS, notre socle technique repose principalement sur le moteur Unity car il permet de créer des applications multiplateformes sans modifications du code. Nous l'avons choisi principalement pour sa simplicité d'exécution car très peu de code sera nécessaire pour des fonctionnalités de "base". Notez bien que l'utilisation du SDK natif nécessite cependant plus de temps et il est recommandé d'avoir de solides bases en programmation mobile.

La reconnaissance des images ou autres se fera par le biais d'une base de données directement sur la plateforme Vuforia que vous viendrez alimenter par des marqueurs de tous types.

Le POC sur Vuforia

Au départ j'avais imaginé une application dans laquelle les utilisateurs toucheraient l'écran là où ils désireraient voir apparaître, par exemple, une pyramide. C'est pourquoi je voulais utiliser ARCore pour Android et son équivalent iOS ARKit. J'ai cependant rapidement été confronté à un problème de taille : le peu d'appareils compatibles, que ce soit avec ARCore ou ARKit. Avec ARCore, les téléphones devaient disposer





de la version Android 7 minimum, et ce n'est pas le seul critère requis. Les appareils compatibles se comptent sur les doigts d'une main, on peut vous assurer que ce ne sera que perte de temps de tenter de l'installer sur un appareil qui n'est pas inclus dans la liste de ceux compatibles. Pour ARKit, tous les iPhone possédant iOS 11+ sont compatibles ce qui peut potentiellement représenter un parc plus grand que son concurrent pour l'instant. Ce ne sont bien sûr que des restrictions temporaires car Google veut ouvrir d'ici à 2 ans ARCore au plus grand nombre.

J'ai donc décidé de passer à une technologie plus accessible par la majorité et j'ai choisi pour cela Vuforia. Vuforia est largement plus permissif en termes de compatibilité, la grande majorité des smartphones actuels fonctionnent avec, cependant dans tous les cas sachez que la performance de votre caméra sera étroitement liée à la qualité du tracking des modèles en AR. Le site de Vuforia étant relativement bien documenté sa prise en main a été facile et rapide. J'ai rapidement pu prendre mon envol pour le développement de l'application. J'ai commencé par créer ma propre base de données, directement sur le portail développeur de Vuforia, avec les images qui allaient me servir de "Marker" (c'est ce qui va être le socle pour la génération de mes modèles 3D), le tout sur la thématique égyptienne.

J'ai également importé les modèles 3D (ceux qui sont affichés sur les markers) sur Unity. Il ne restait plus qu'à associer chaque modèle à son marqueur sur le moteur de jeu. J'avais alors tous les éléments pour développer sereinement mon application.

Un point assez important pour le souligner est que Vuforia ne nécessite pas d'être un expert en programmation puisque l'on peut faire énormément de choses sans avoir besoin de coder. Si vous connaissez les B.A.BA d'Unity, vous pourrez largement vous en sortir. Le SDK étant étroitement lié au moteur de jeu, cela évite d'être bloqué à cause d'un manque de connaissance sur Unity. Si vous souhaitez ajouter de l'interaction via une interface cela se fera uniquement via les fonctionnalités d'Unity et des scripts. Au final Vuforia va enrichir Unity en vous permettant de rajouter de l'AR à vos applications, n'oubliez cependant pas de bien configurer le SDK dans les settings ! Rapidement vous pourrez avoir une application fonctionnelle, en particulier si vous désirez uniquement afficher des modèles 3D en AR.

Les interactions plus poussées nécessitent de la programmation, l'ajout de boutons ou les interactions avec les objets en AR passeront forcément par du code / des scripts. Les virtual buttons, qui sont les boutons AR interactifs proposés par Vuforia, bien qu'un peu tricky, sont intéressants et peuvent apporter un plus à votre application à condition d'arriver à les faire fonctionner. Cependant les virtual buttons pour être facilement utilisables doivent être placés sur une image ou une partie d'image relativement grande et surtout riche en détails et contrastes. Et quand je parle de contraste, ce sont des contrastes forts, noir sur blanc. Pour les détails, un damier par exemple ne fonctionnera pas, le motif étant trop répétitif, privilégiez plutôt une image comme celle d'une foule. Plus l'image sera grande et détaillée, plus les virtual buttons seront facilement activables. Toutefois, ce n'est

pas la seule condition à leur activation, qui sera plus ou moins rapides dépendamment de la lumière, de la définition de l'image utilisée et de la qualité de votre caméra. Vous pouvez régler leur sensibilité, mais je n'ai pas vu de grande différence à l'utilisation entre haute et basse sensibilité. Enfin vous l'aurez compris, les virtual buttons sont assez capricieux et pas forcément fiables. L'application propose même une partie en VR, toujours grâce à Vuforia, afin de découvrir dans notre cas l'intérieur d'une pièce de la pyramide. Implémenter la VR a été légèrement plus complexe que pour l'AR mais reste à portée tout de même d'un développeur novice.

Le SDK Vuforia est également disponible sur Android Studio, mais là pour le coup, il va falloir être habitué à la programmation sur mobile, ou alors vous serez bloqué une fois le tutoriel Vuforia fini.

Mais on ne s'arrête pas là ...

Vuforia permet aussi l'utilisation de la VR avec votre téléphone, par exemple avec un cardboard. Tout est déjà pensé afin que vous puissiez passer, au sein d'une même application, de l'AR à la VR, il faudra tout de même mettre la main à la pâte afin de coder le changement entre les deux modes. Tous les paramètres sont de base inclus à Vuforia pour vous permettre de faire de la VR pour mobile, de la configuration du cardboard que vous allez utiliser à la façon dont est tenu votre téléphone. Et il n'y a pas d'installation supplémentaire à faire pour ça puisque le support pour le CardBoard et Google VR est également built-in dans le SDK. Il y a également un support pour utiliser l'holoLens à la place d'un téléphone.

Vuforia propose des façons diverses et variées de faire de l'AR, dans cet article j'ai principalement parlé du classique "Marker et modèle" mais le SDK propose beaucoup plus d'options :

- Détection de modèles : va permettre de reconnaître des objets grâce à leur forme en comparaison avec des modèles 3D (que vous pouvez imprimer via une imprimante 3D ou tout simplement utiliser des objets du quotidien, comme une chausure).
- Détection de plans : va détecter des surfaces planes sur lesquels vous pourrez positionner des objets.

- L'utilisation des marqueurs : il y a plein de types de marqueurs possibles, du QRCode à vos images de vacances en passant par la cannette de soda ou votre radiateur. Tous sont éligibles au rôle de marqueur.

Cette magie a quand même un coût

En effet, Vuforia propose une plateforme de développement totalement gratuite. C'est au moment du choix du mode de déploiement de votre application que les modes payants apparaissent. Parmi ces modes, citons le mode "Classique", le mode "Cloud" et le mode "Pro".

Les différences principales entre ces différents modes résident dans la possibilité d'utiliser des "markers" plus ou moins sophistiqués et dans le volume et le type des objets traités.

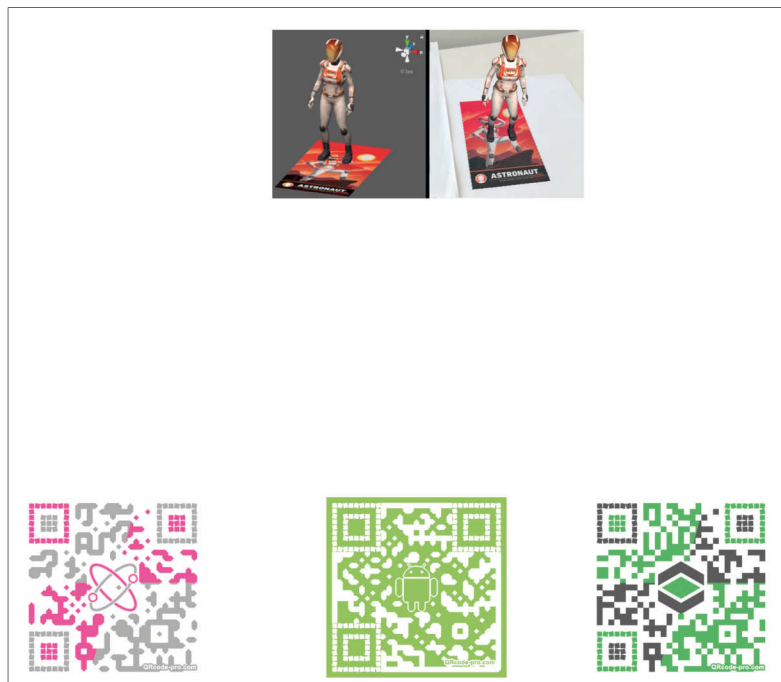
Tous ces modes permettent d'utiliser des images, des "QR Codes", des modèles 3D (créés via une App Android fournie par Vuforia : "Vuforia Object Scanner") comme "markers". Par contre l'importation de "markers" 3D depuis un modèle 3D pré-existant est réservée au mode "Pro".

Le mode "Cloud" propose une fonctionnalité (le Cloud database) permettant de concevoir des applications qui vont utiliser un nombre important (jusqu'à 100.000) de cibles et autoriser des modifications et des ajouts de contenus dynamiquement (nous sommes là proche d'un fonctionnement en time to market), tous les objets étant stockés dans une base de données sur le Cloud. Ce mode inclut un nombre maximum de 10.000 reconnaissances sur jusqu'à 100.000 images par mois.

A noter que le mode "Pro" embarque lui aussi cette fonctionnalité, mais sans limitation. Il reste un mode de déploiement gratuit, mais celui-ci ne vous permettra de déployer que les applications développées sur Unity. Et vous permettra uniquement de "poser" un contenu virtuel sur une surface plane (horizontale) basique (telle qu'un sol...) grâce à une détection de plan comme peut le faire ARKit ou son homologue ARCore.

Testez par vous-même !

OK alors tout ça c'est bien beau mais j'imagine que vous voulez tester tout cela par vous-même non ? Et bien afin que vous ayez de quoi vous faire une idée au com-



plet de ce que donne le système de détection Vuforia, je vous propose de venir récupérer notre application et même son code source sur le Github de CONCERTO ! Pour cela rendez-vous sur la page Github du projet :

https://github.com/LabsConcerto/Magazine-Programmez-AR_Pyramid

Vous y trouverez le code source du programme mais également les assets et un readme qui vous expliquera pas à pas comment monter le projet afin de pouvoir l'ouvrir dans Unity.

Enfin si vous voulez récupérer l'application, flashez le QRCode se trouvant en haut dans le readme et il vous mènera tout droit vers son téléchargement sur le playstore.

Pour les utilisateurs iOS il vous faudra malheureusement vous-même compiler le projet sur un Mac et transférer le programme dans votre appareil car le délais de validation d'une application sur l'AppleStore dépassait considérablement le délais de publication.

Il ne reste plus qu'à lancer le programme et pointer l'application en direction de la page du magazine contenant les différents QRCode de couleur.

Ceux-ci vont directement se transformer en bouton (virtuelle) de navigation et un modèle 3D sera d'ores et déjà visible au-dessus des 3 boutons.

Un menu situé en haut à gauche vous permettra également de naviguer et de passer sur le mode Réalité Virtuelle et une aide en

haut à droite vous récapitulera les différentes fonctionnalités.

Pour finir avec Github, je vous rappelle que toutes suggestions sont bonnes à prendre (dans le cadre où elles restent constructives bien sûr) et vous invite à laisser commentaires issues ou même idées si vous le souhaitez.

Conclusion

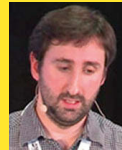
Au final, la plus grande conclusion que nous pouvons apporter est que Vuforia est un SDK très rapide à prendre en main pour les fonctionnalités de base, on peut très facilement créer une application d'AR basique (juste des modèles affichés en 3D sur des markers), il faudra compter un peu plus de temps si vous souhaitez rajouter des interactions. Le Vuforia SDK nécessite toutefois des connaissances en Unity ou en développement mobile (si vous développez avec Android Studio). Le SDK regorge d'options qui permettent d'améliorer le tracking (ex: garde le modèle même si le marker est hors champs). Il faudra cependant un petit temps de prise en main pour se familiariser avec toutes les possibilités qu'offre ce framework. Une expérience concluante intégrable sans soucis dans un projet de plus grande envergure !

Retrouvez le projet sur Github et Playstore : https://github.com/LabsConcerto/Magazine-Programmez-AR_Pyramid ainsi que tous nos autres projets Open Source sur : <https://github.com/LabsConcerto>

Faites du minage !



Arnaud BREL
arnaud.brel@sogeti.com



Laurent GRANGEAU
laurent.grangeau@gmail.com

Partie 1

Dans ce numéro, nous vous proposons de revenir sur les grands principes de la blockchain et son utilité. Nous vous proposons aussi à la fin de cet article de construire votre propre rig (plateforme) de minage afin de miner vos premières crypto-monnaies : de l'assemblage du hardware, à la configuration du software et un comparatif des différents logiciels de minage.

La blockchain est une technologie de stockage et de transmission d'information, transparente, sécurisée et fonctionnant sans organe central de contrôle. Dans ce sens, la blockchain fonctionne de manière décentralisée, et nous verrons dans la suite de l'article qu'il est difficile, voire impossible, de falsifier une information présente sur la blockchain.

Sécurisation de la blockchain

Pour sécuriser les principales blockchains, on a recours à ce que l'on appelle une preuve de travail - Proof of Work en anglais.

Le principe de la preuve de travail est d'effectuer des calculs complexes, coûteux en termes de temps et d'énergie, afin de valider les transactions d'un bloc, qui, associé à la signature du bloc précédent, permettront de l'ajouter à la blockchain.

Dans la mesure où un bloc est créé à intervalle régulier, la difficulté pour trouver la solution au chiffrement est ajustée en fonction du nombre de participants du réseau à l'instant du calcul mais aussi en fonction du nombre de transactions contenues dans le bloc et la chaîne de bloc précédente.

L'ordinateur ou le groupe d'ordinateurs qui trouve en premier la solution du chiffrement diffuse le résultat aux autres participants du réseau qui peuvent facilement valider sans requérir de la puissance de calcul. Lorsque la solution est validée, elle est diffusée à l'ensemble du réseau. Le mineur ayant trouvé la solution est récompensé en monnaie nouvelle selon les modalités définies par le protocole de la crypto-monnaie.

Chaque bloc contient le hachage du bloc précédent, ainsi chaque bloc a une chaîne de blocs qui contiennent tous les deux une grande chaîne de travail. Changer un bloc N (qui n'est possible qu'en faisant un nouveau bloc N+1 contenant le chiffrement de la chaîne de bloc précédente N-1) requiert un travail considérable car il faut recalculer d'abord le chiffrement du bloc N avant de chiffrer le bloc N+1. La falsification est donc difficile voire impossible.

Cet algorithme étant très coûteux en énergie, les blockchains principales recherchent d'autres alternatives pour pouvoir continuer à sécuriser la blockchain sans dépenser autant d'énergie. Un de ces algorithmes est le proof-of-stake.

L'algorithme du proof-of-stake, ou algorithme de preuve d'enjeu, est une méthode par laquelle une chaîne de bloc d'une crypto-monnaie vise à atteindre un consensus distribué. Alors que la preuve de travail (PoW) demande aux utilisateurs d'exécuter plusieurs fois les algorithmes de hachage ou de calculer des puzzles mathématiques selon des algorithmes pour valider les transactions électroniques, la preuve d'enjeu demande à l'utilisateur de prouver la possession d'une certaine quantité de crypto-monnaie (leur « participation » dans la crypto-monnaie) pour prétendre à pouvoir valider des blocs supplémentaires dans la chaîne de bloc et de pou-

voir toucher la récompense, s'il y en a une, à l'addition de ces blocs. Les incitations de comportement des mineurs fonctionnent aussi différemment : dans la méthode de validation par preuve de travail, le mineur a intérêt à vendre la récompense obtenue pour chaque bloc miné notamment pour payer ses frais de consommation énergétique ; alors que dans le système de validation par preuve d'enjeu, le mineur a intérêt à conserver ses pièces pour maximiser ses chances futures de pouvoir valider les blocs. Une monnaie utilisant la validation par preuve d'enjeu n'a pas d'inflation (elle est déflationniste). La masse monétaire est créée au début et les mineurs validant les blocs supplémentaires sont récompensés en touchant le montant des frais de transfert.

Architecture CPU vs GPU

Comme on peut le voir, l'algorithme principal utilisé est un algorithme qui nécessite une grosse quantité de calcul. Plus il y a de ressource disponible pour faire ces calculs, plus rapide est la récompense du mineur.

Le CPU, dans la plupart des cas, est rapidement limité par son nombre de coeurs. Par exemple, un coeur CPU peut exécuter 4 instructions 32 bits par horloge (en utilisant une instruction SSE 128 bits) ou 8 via AVX (256 bits). Pour les CPU les plus récents, il existe au maximum jusqu'à 18 coeurs (pour le Core i9 par exemple), soit 144 instructions exécutées par cycle d'horloge.

La technologie SMT, pour Simultaneous multithreading, permet de multiplier ce nombre d'instructions par cycle d'horloge. Pour Intel, la technologie est baptisée Hyperthreading, et permet de doubler le nombre d'instructions, soit 288 instructions exécutées par cycle d'horloge.

Le GPU possède beaucoup plus de coeurs. Si nous prenons l'exemple de la NVIDIA GeForce 1080 Ti, cette carte possède 3584 coeurs spécialisés dans le traitement vidéo. Elle peut donc traiter environ 12 fois plus de calcul à la seconde que les CPU.

Les logiciels de minage s'appuient donc majoritairement sur des cartes graphiques afin d'augmenter le nombre de calculs à la seconde, et donc le nombre de blocs minés.

Le cas des ASICs

Dans le cas du Bitcoin, s'appuyant sur l'algorithme proof-of-work, les mineurs utilisent des cartes de minage spécialisées, appelées ASICs (pour Application-specific integrated circuit, littéralement « circuit intégré propre à une application ») afin d'augmenter drastiquement la quantité de Bitcoin miné, quelquefois jusqu'à 300 fois plus rapide qu'un GPU classique.

L'algorithme proof-of-work utilisé dans le Bitcoin s'appuyant principalement sur un double hash SHA-256, les ASICs sont optimisés pour ne savoir calculer que cet algorithme de manière matérielle,

Définition

Mh/s : La capacité de minage est mesurée par le nombre de tentatives pour trouver un bloc qu'un mineur peut effectuer. Chaque tentative consiste à créer un candidat bloc unique, et à créer un résumé du candidat bloc au moyen de la fonction de hash cryptographique SHA-256. Cette tentative s'appelle un hash. Puisqu'il s'agit d'un effort continu, on parle de hashes par seconde ou h/s. Un Mh/s (ou méga hashes par seconde) est l'équivalent de 1 million de hashes par seconde.

et non plus logicielle, d'où les gains de performance.

L'Ethereum, quant à lui, a été codé spécifiquement pour être résistant à ce type de circuit spécialisé, en nécessitant notamment d'avoir le graphe complet d'Ethereum en mémoire pour calculer le prochain bloc.

La méthode encore aujourd'hui la plus répandue pour miner de l'Ethereum est d'utiliser des GPUs traditionnels. L'avantage d'avoir des GPUs est aussi de pouvoir miner d'autre crypto-monnaies par rapport à la rentabilité des cours d'échange de ces crypto-monnaies.

Nous verrons dans la suite de cet article comment échanger ces monnaies sur des places d'échange dédiées, et comment adapter automatiquement le minage par rapport à la rentabilité de ces monnaies.

Minage solo

Comme vu précédemment, le but de miner des crypto-monnaies est d'aider à valider les transactions, en résolvant des calculs complexes pour créer le prochain bloc de la blockchain, et donc d'être récompensé pour ces calculs.

Cependant, le but de la blockchain est de "trouver" un bloc dans un temps constant, par exemple toutes les 10 secondes. Cela veut dire que toutes les 10 secondes, tous les mineurs sont avertis du nouveau bloc et de nouveaux calculs sont lancés afin de valider le prochain.

Il est possible de miner en solo, c'est à dire sans l'aide de personne, mais dans ce cas, il faut avoir une puissance de calcul énorme afin d'être sûr de trouver un bloc dans un temps court, et donc d'être souvent récompensé.

Malheureusement, la difficulté des différentes blockchains est telle qu'il n'est plus rentable de miner en solo.

Minage en pool

Le minage en pool quant à lui se différencie par la mise en commun des différentes puissances de calcul de chacun des mineurs, afin d'augmenter la probabilité de trouver un bloc, et donc d'être récompensé.

Une fois le bloc trouvé, chacun des mineurs est récompensé par rapport au pourcentage de puissance apporté.

Par exemple, si un bloc rapportant 5 Bitcoins est trouvé par le pool, et que ma puissance totale fournie dans le pool contribue à hauteur de 10%, je serai rémunéré 10% de 5 Bitcoins, c'est à dire 0,5 Bitcoin.

Il existe beaucoup de pools de minage, chacun avec ses avantages et ses inconvénients. Nous verrons dans la suite de l'article comment se créer un compte sur un de ces pools de minage et comment paramétrer le logiciel de minage afin de pouvoir récupérer ses récompenses.

Les différentes places de marchés

Une fois les premiers blocs minés, et donc les premiers versements effectués sur notre adresse blockchain, nous avons la possibilité de les convertir en d'autres crypto-monnaies, mais aussi en monnaies fiduciaires, c'est à dire en monnaie ayant cours dans les différents états du monde.

Nous verrons à la fin de l'article comment s'inscrire sur une plateforme d'échange de crypto-monnaies afin d'échanger, d'acheter

ou de vendre ces monnaies.

Dans la suite de cet article, des choix ont été faits quant à la crypto-monnaie à miner, le pool de minage choisi, le matériel et le logiciel choisis ainsi que la place de marché afin de garder une certaine lisibilité.

Il est tout à fait possible de choisir d'autres éléments que ceux cités ci-dessous.

Comment monter un RIG de minage

Abordons maintenant la partie la plus intéressante de cet article, la construction du rig de minage. Nous allons détailler ci-dessous tous les éléments nécessaires afin de construire notre plateforme, ainsi que leur prix, ce qui permettra de choisir les composants les plus adaptés par rapport à la puissance voulue, et au prix souhaité.

Comme expliqué précédemment, afin d'être le plus efficace et le plus rentable, il est nécessaire de construire son rig à base de GPU. Voici un tableau comparatif de la performance de quelques cartes, ainsi que leur consommation électrique et leur prix.

Carte	Puissance de mining	Consommation électrique	Prix
Nvidia GeForce GTX 1070	30 Mh/s	150 Watts	440 €
AMD Radeon RX580	29 Mh/s	185 Watts	300 €
Nvidia GeForce GTX 1060	20 Mh/s	120 Watts	275 €
AMD Radeon RX Vega 56	31 Mh/s	210 Watts	455 €
Nvidia GTX 1080 Ti	32 Mh/s	250 Watts	830 €
Nvidia GTX 1070 Ti	30 Mh/s	180 Watts	450 €

Plus de cartes sont présentes sur le rig, plus rapide est le minage, et donc l'envoi de récompense. Cependant, il est nécessaire de mettre des cartes vidéo avec au minimum 3Go de mémoire embarquée afin de charger le fichier DAG qui est le fichier de donnée de la blockchain.

CPU

Le choix du CPU lors de la construction d'un rig de minage n'a que peu d'importance. Effectivement, les calculs seront bien plus performants lorsqu'ils seront résolus par le GPU que le CPU. Il n'est donc pas nécessaire de dépenser de l'argent dans le dernier CPU. Un simple Intel Celeron fera l'affaire. Voici un tableau de quelques CPUs

CPU	Puissance de mining	Consommation électrique	Prix
AMD Ryzen Threadripper 1950X	16 coeurs / 32 threads	180 Watts	680 €
AMD Ryzen 7 1800X	8 coeurs / 16 threads	95 Watts	205 €
Intel Pentium G4400	2 coeurs / 2 threads	54 Watts	45 €
Intel Core i5-7600K	4 coeurs / 4 threads	91 Watts	190 €
Intel Pentium G4560	2 coeurs / 4 threads	54 Watts	50 €

Carte mère

Le choix de la carte mère est important. En effet, afin de maximiser le nombre de cartes que contiendra le rig, il faut que la carte mère possède beaucoup d'emplacement PCI. La plupart des cartes mères détiennent maintenant 6 emplacements PCI, ce qui est une bonne base pour monter une plateforme avec 4 ou 6 GPUs.

Voyant l'engouement du minage, certaines marques n'hésitent plus à proposer des cartes mères optimisées pour le minage, avec des emplacements PCI pouvant monter jusqu'à 19 (!)

Carte mère	Emplacement PCI	Socket CPU	Prix
Asus B250 Mining Expert	19	Socket 1151	155 €
ASRock H110 Prix BTC+	13	Socket 1151	105 €
Gigabyte GA-H110-D3A	6	Socket 1151	60 €
MSI Z170A Gaming Pro Carbon	7	Socket 1151	300 €
Asus ROG Strix Z270E	7	Socket 1151	155 €

Mémoire

Pour la mémoire, il n'est pas nécessaire d'en avoir une grosse quantité, cette mémoire servant principalement au système d'exploitation, et non au minage lui-même. Une simple barrette de 4Go devrait suffire largement à mettre en place le système d'exploitation, ainsi que le logiciel de minage associé. Dans le cas du minage d'Ethereum, pour ne pas que le système s'arrête lors de la création du fichier DAG, 8 Go de mémoire sera plus confortable

Mémoire	Type	Vitesse	Prix
Crucial 4Go	DDR4	2400 Mhz	50 €
Ballistix Elite 4Go	DDR4	3200 Mhz	70 €
Crucial 8Go	DDR4	2400 Mhz	90 €
Ballistix Elite 8Go	DDR4	3000 Mhz	110 €

Stockage

Comme la mémoire, il n'est pas nécessaire d'avoir une grosse capacité de stockage. Cependant, si le rig de minage ne sert qu'à faire du minage solo, il sera alors nécessaire d'augmenter la capacité de stockage afin de récupérer l'intégralité de la blockchain à miner. Dans le cas d'Ethereum, la blockchain complète pèse tout de même 667 Go à l'écriture de cet article. Pour le cas du Bitcoin, la blockchain est quand même moindre, avec une taille de 163 Go. Dans le cas d'un minage en pool, un simple disque dur de 60 Go ou 120 Go suffit.

Disque dur	Stockage	Interface	Prix
Sandisk DDS Plus SLC	120 Go	SATA-3	42 €
Crucial MX300	1050 Go	SATA-3	240 €
Intel SSD 545s Series	512 Go	SATA-3	150 €
Kingston KC1000	240 Go	PCI	150 €

Riser PCI-cable

La plupart des cartes graphiques citées précédemment prennent souvent deux slots d'emplacement sur la carte mère, ce qui empêche le plus souvent de mettre plus de 2 ou 3 cartes graphiques

sur la même plateforme.

Afin de palier cela, il existe des câbles permettant de déporter les GPUs en dehors de la plateforme, par exemple sur un cadre en bois ou en métal permettant d'aligner les cartes graphiques et d'optimiser le refroidissement de ces cartes.

Ces câbles sont généralement vendus par lot de 6 et coûtent environ 50 €

Alimentation

L'alimentation est importante. Le choix d'une bonne alimentation pouvant supporter le CPU ainsi que les cartes graphiques n'est pas à prendre à la légère. Il existe 4 niveaux de certification de ces alimentations : 80 Plus bronze, 80 Plus argent, 80 Plus or et 80 Plus platinum. Ces certifications permettent respectivement :

- Alimentation 80 Plus Bronze : 82, 85 et 82% de rendement minimum à respectivement 20, 50 et 100% de charge.
- Alimentation 80 Plus Argent : 85, 88 et 85% de rendement minimum à respectivement 20, 50 et 100% de charge.
- Alimentation 80 Plus Or : 87, 90 et 87% de rendement minimum à respectivement 20, 50 et 100% de charge.
- Alimentations 80 Plus Platinum : 90, 94 et 91% de rendement minimum à respectivement 20, 50 et 100% de charge.

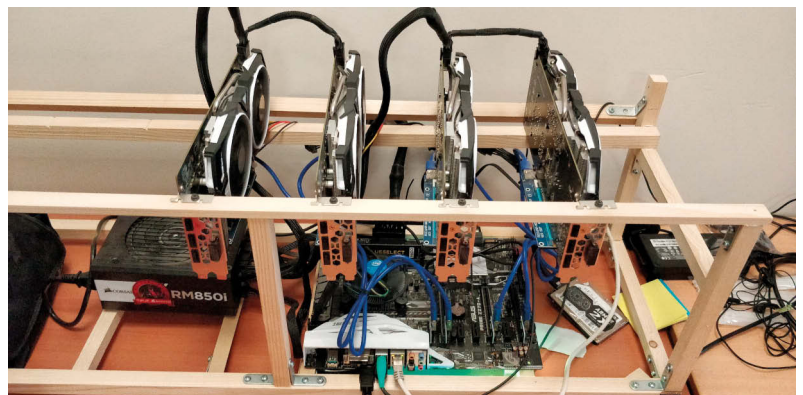
Il faut aussi garder à l'esprit que notre plateforme fonctionnera 24/7, ce qui est à prendre en compte dans le choix de la certification.

Montage de la plateforme

Dans la suite de cet article, nous allons partir sur une configuration assez puissante afin de produire environ 80 Mh/s, mais aussi économique pour ne pas exploser le budget. Vous retrouvez dans le tableau ci-dessous la configuration choisie.

Composant	Marque	Prix
CPU	Intel BX80677 G3930	35 €
Carte mère	Asus Prime Z270-A	145 €
Mémoire	Corsair CMV4GX4M	50 €
Stockage	Intel SSD 545s Series	150 €
Alimentation	Corsair CP-9020083-EU RMi	150 €
PCI-Riser	4x Marktoll PCIE 1x to 16x	16 €
GPU	4x MSI GeForce GTX 1060 Armor 6G	335 €
Total		1934 €

Pour - 2000 €, nous obtenons une plateforme capable de miner à une puissance de 80 Mh/s. Les gains estimés sont d'environ 0,15 ETH par mois.



Maintenant que la plateforme est montée, il faut installer toute la partie logicielle afin de pouvoir participer au minage de la blockchain choisie.

Il existe des systèmes d'exploitation dédiés au minage, par exemple EthOS (payant), ou Simplemining OS (gratuit, avec fees sur votre hashrate, et on ne peut pas mélanger les RX series avec les R series et/ou les Nvidia). Ces OS sont dédiés au minage, fonctionnent souvent avec Linux et sont livrés avec tous les outils nécessaires pour miner automatiquement, ce qui évite l'installation et la configuration souvent complexe des outils de minage. Ces systèmes d'exploitation ne sont pas limités pour le nombre de GPU, tant que votre CM les reconnaît, ils les gèrent.

Pour ne pas ajouter de coût supplémentaire, il est possible de miner avec une distribution Linux, la plus simple étant Ubuntu. Cependant, il faut faire attention à la version de distribution. Les logiciels de minage étant principalement codés pour tirer parti du matériel via OpenCL ou CUDA, la performance du minage dépend très fortement de la version des drivers installés. Sur Ubuntu, la dernière version propriétaire des drivers AMD date de 2015. Depuis, l'effort s'est porté sur une écriture open source, qui est bien souvent moins performante que les drivers propriétaires. En ce qui concerne Nvidia, les performances ne sont pas impactées quelle que soit la version du driver installée.

Il est devenu plus simple depuis peu de miner avec Windows 10, qui fonctionne parfaitement avec les derniers drivers Nvidia ou AMD. C'est ce que nous allons utiliser dans la suite de cet article pour miner, l'installation et la configuration des drivers Nvidia étant plus simple que sous Linux.

Logiciels de minage

Il existe plusieurs logiciels de minage, avec chacun ses spécificités, par exemple Ethmine.

Pour cet article, le logiciel Claymore a été choisi. Très stable, il peut s'utiliser indifféremment avec des cartes AMD et/ou NVIDIA. On peut paramétrer tout un tas de choses et on peut même miner 2 blockchains en même temps (par exemple Ethereum et Zcash), mais attention car la puissance de calcul disponible est répartie entre les différentes blockchains minées en parallèle.

Pour lancer le minage, rien de plus simple :

```
EthDcrMiner64.exe -epool URL_POOL:PORT_POOL -ewal WALLET_ADDRESS -eworker WORKER_NAME -epsw x"
```

Il faut évidemment remplacer les valeurs de URL_POOL, PORT_POOL, WALLET_ADDRESS et WORKER_NAME par vos informations.

Cette ligne de commande utilisera toute la puissance de calcul disponible pour effectuer le minage et vous récompenser.

Il existe quelques options utiles pour ce logiciel qui peuvent influencer sur la puissance de calcul

- di x Index des GPU (si on ne met pas ce paramètre, le logiciel démarre avec tout ce qui peut miner). Permet de dire quels GPU minent.
Par exemple -di 023 va permettre de miner avec le 1er, le 3ème et le 4ème GPU (oui, 0=1 je sais c'est bizarre :))
- esm x Mode Stratum (x=0 par défaut) :
0 eth-proxy mode (dwarppool.com, ethermine.org,...)

- 1 qtmminer mode (ethpool.org)
- 2 miner-proxy mode (coinotron.com)
- 3 nicehash mode

- etha x Type d'algorithme Ethereum à utiliser pour les cartes AMD (si on ne met pas ce paramètre, le logiciel détecte automatiquement la valeur)
0 optimisé pour les cartes performantes
1 optimisé pour les cartes à Hashrate bas
2 optimisé pour les pilotes Linux
On peut spécifier l'algorithme pour chaque carte en séparant par une virgule chaque valeur par exemple : -etha 0,0,1,1
- ethi x Définit l'intensité pour le calcul (x=8 par défaut). On peut diminuer volontairement cette valeur si on souhaite pouvoir avoir en parallèle une activité de bureau ou si on a des problèmes de stabilité. La valeur la plus basse est « -ethi 0 ». Attention, par expérience on constate souvent que ça n'est pas parce que l'intensité est la plus haute que le Hashrate sera le meilleur.
On peut spécifier l'intensité pour chaque carte en séparant par une virgule chaque valeur par exemple : -ethi 7,7,8,8
- eres x Ce paramètre est à mettre si votre miner crashe lors de changement d'Epoch. Lors d'un changement d'Epoch, le logiciel recharge le DAG dans la mémoire GPU. Mais ce nouveau DAG étant légèrement plus gros que l'ancien, et parfois ça crashe. Pour éviter ça, ce paramètre réserve la mémoire qui sera nécessaire pour le DAG qui sera chargé dans x Epoch (par défaut x=2).
- li x Low intensity mode. Pour réduire l'intensité de calcul, histoire que ça chauffe moins ou que ça freeze moins (si vous voulez avoir une activité de bureau en même temps). -li 10 mine moins que -li 1 (0 par défaut)
- nofee Enlève la « taxe » instaurée par le développeur pour se rémunérer, mais réduit de 4% environ le hashrate.
- tt x Cible de température du GPU. -tt 80 signifie une température cible de 80°C pour le 1erGPU. -tt 70,80,75 par exemple s'utilise pour spécifier la température cible pour les 3 premiers GPU. -tt -50 (négatif donc) permet de fixer la vitesse des ventilateurs (en %), ici 50%. Par défaut, le logiciel est configuré sur -tt 1, qui permet juste d'avoir l'info sur la température du GPU et la vitesse des ventilos. Attention, pour les cartes NVIDIA on ne peut que monitorer et pas gérer la température.
- ttli x Baisse l'intensité du minage pour rester à la température x. -ttli 70,80,75 par exemple s'utilise pour que les 3 premiers GPU restent aux températures cibles (70°, 80° et 75°)
- fanmax x Spécifie la vitesse max des ventilos (en %). -fanmax 80 signifie une vitesse max de 80%. On le spécifie également par GPU : -fanmax 70,80,75 par exemple
Non supporté avec NVIDIA
- fanmin x Spécifie la vitesse min des ventilos (en %). -fanmin 30 signifie une vitesse min de 30%. On le spécifie également par GPU : -fanmin 40,50,40 par exemple
Non supporté avec NVIDIA

La suite dans le N°221

Programmez en famille

Partie 1



© Y. Serra



François Tonic

Le code comme 3^e langue vivante

Dire qu'il faut faire découvrir la programmation aux enfants c'est une chose. Mais faire comprendre de quoi ça parle aux parents et aux enfants, c'est parfois une autre histoire. Et le fossé est parfois très grand. Une étude menée par Institut Juniors & Co pour Texas Instruments fait le point(1).

Le premier constat de l'étude montre le manque d'informations des parents envers le code. La compréhension s'améliore, mais reste largement perfectible : 76 % des parents ont entendu vaguement parler, ou pas, de la programmation. En 2017, le chiffre était de 84 %. Dans le détail, 39 % des parents répondants disent ne pas savoir ce qu'est le coding. 25 % disent qu'ils savent très bien ce que c'est.

Un réel problème de compréhension

Au-delà de cette question, est-ce que l'intégration de programmation au programme scolaire est bien connue ? Là encore, il y a un progrès, mais il reste encore beaucoup à faire. Ainsi, à l'école, 87 % des parents ne savent pas que le code est intégré au programme du primaire, 73 % au collège et tout de même 66 % au lycée. Il y a donc un réel problème de communication. Ce manque d'information des parents pose une réelle question sur comment les écoles et l'Éducation nationale informent et diffusent l'information. C'est un travail de terrain qu'il faut mener.

L'étude met en évidence une question sensible : les programmes scolaires intègrent-ils trop ou assez de nouvelles technologies ? C'est un débat important. Ce n'est pas à l'école de former des informaticiens, mais plutôt de sensibiliser les enfants et ados à la technologie, ce qui n'est pas la même chose. Si l'enfant veut continuer, il le fera de lui-même si cela est possible. Il ne faut pas tomber dans le tout technologie. En revanche, renouveler la pé-

dagogie actuelle, intégrer les outils actuels, cela pourrait être intéressant. À vouloir trop en faire, n'est-ce pas aussi le risque de saturer les enfants et de faire l'effet inverse : provoquer une certaine réticence à la technologie.

Car finalement, tout le monde n'a pas vocation à travailler dans les technologies.

Les enfants plus en avance que les parents

Sans surprise, les élèves sont souvent mieux informés que les parents. Pour 2018, 44 % disent en avoir entendu parler (même vaguement), 37 % connaissent très bien. 19 % affirment ne pas savoir ce qu'était le coding avant de lire la définition.

Par contre, les lycéens ont conscience de l'importance du code et de la programmation en général. 43 % disent qu'il est probable que le code fera partie des compétences à maîtriser pour un futur métier. À égalité avec ceux qui pensent que c'est peu ou très probable. Cette question est intéressante, car cela montre une certaine sensibilité des lycéens et qu'ils voient l'importance des technologies et du code.

Et les enseignants

L'étude parle aussi des enseignants. Les répondants se disent de mieux en mieux familiarisés. 90 % estiment en savoir plus que leurs élèves. Ce chiffre nous laisse sceptiques. Peut-être faut-il y voir une grande sensibilité à la programmation des répondants. En pratique, la nuance est importante, car 45 % disent enseigner la programmation que quelques fois. 42 % di-

sent souvent. Reste à savoir ce qu'ils entendent par souvent. Mais une chose est claire : les enseignants savent que la programmation fera partie de l'avenir des élèves. Mieux vaut la connaître, même un peu, que pas du tout.

Se pose aussi la formation à la programmation des enseignants. 73 % disent avoir fait de l'auto-formation. Cela révèle un problème, car seuls 26 % disent avoir eu une formation proposée par l'Éducation Nationale. La question est de savoir ce que l'on attend par auto-formation : quels outils ? Quels langages ? Quelles bases sont apprises ?

En conclusion l'étude met en avant la nécessité d'avoir les bons outils pour apprendre et expérimenter la programmation. Il est important de montrer concrètement des résultats, car faire que de la théorie ce n'est pas une bonne chose.

L'étude permettait aussi d'avoir des retours sur les outils que Texas Instruments propose aux écoles et lycéens comme les calculatrices programmables, le TI-Innovator Hub et le Innovator Rover. Cet ensemble permet de concrétiser les lignes de codes et comprendre l'impact du code, comment on développe des comportements. Ce n'est pas un hasard si les concours de robotique remportent un joli succès auprès des Lycéens. •

(1) Etude menée en ligne sur 477 parents et 201 lycéens.



Pour bien préparer l'été et la future rentrée !

Quels conseils donner pour cet été et pour préparer la prochaine rentrée ? Clément Fantoli, professeur de technologie, nous donne quelques conseils.

Les vacances d'été sont là, la rentrée c'est déjà demain :-). Quels conseils ou activités les enfants et ados peuvent-ils faire pour découvrir la programmation ou se perfectionner de manière ludique ?

Il existe beaucoup de sites internet comme code.org où les enfants et ados trouveront de petits exercices sous forme de jeux pour se familiariser ou se perfectionner sur la programmation. Pour les plus créatifs les logiciels Scratch ou Mblock leur permettront de créer des lutins (petits personnages) et de les faire interagir. Car si la rentrée approche, c'est que les vacances sont encore là et qu'il faut garder le côté récréatif de la programmation.

Quand on parle de l'apprentissage du code à l'école, on parle de théorie, d'algorithmes. Comment ne pas dégoûter les enfants du code, notamment en primaire et collèges ? Faut-il miser sur la pratique, le ludique avant tout et réserver les notions avancées au lycée ?

J'aime comparer la programmation à un gâteau, pour que cela marche, il faut que les élèves goûtent à tous les ingrédients. Certes le côté ludique de la programmation d'un robot permettra de mieux capter l'attention des élèves mais n'oublions pas qu'ils seront évalués sur des notions théoriques tel que l'algorithme ou l'algorithmique. Nous nous devons de conserver une partie théorique à partir du moment où elle est utile et comprise par l'élève. Par exemple lorsqu'un élève a du mal à décrire le comportement d'un système, on introduit la notion d'algorithme en lui expliquant que c'est une succession d'étapes en fonction de conditions.

Finalement c'est quoi apprendre le code à l'école ? Comment voyez-vous cela dans les classes ? Quelle est l'idée derrière ça ?

Apprendre le code à l'école c'est avant tout démystifier la programmation, trop souvent considérée comme un domaine réservé à des experts en informatique. C'est également un excellent moyen de créer des séquences en interdisciplinarité entre la technologie, les mathématiques et la physique-chimie.

Nous pouvons ainsi mener des projets intéressants qui demanderont aux élèves des compétences plus ou moins poussées et les élèves sont très volontaires. Nous allons même jusqu'à créer des applications pour smartphone, je vous laisse imaginer l'entraî qu'il y a derrière un tel projet...

Il existe de nombreuses initiatives : Minecraft, concours robotiques, hackatons, Hour of Code, etc. Existe-t-il une formule magique pour intéresser les enfants et les ados à la programmation ?

Chaque enfant est unique, certains s'amuseront à programmer des robots, d'autres vont adorer taper des lignes de codes ou créer des personnages et des histoires sur Scratch. Il n'y a donc pas de formules magiques. Toutefois, pour intéresser le plus d'enfants en classe, il faut tout miser sur une situation déclenchante accessible et intéressante pour les élèves.

Avez-vous des enfants qui se découvrent une vraie passion pour le code et qui en demandent toujours plus ? Est-ce plus facile à « gérer » ?

Oui et même plus d'un. Roman, élève de sixième, a appris tout seul à lire des lignes de code. J'ai une élève de cinquième, Kalista, qui passe son temps libre à créer des systèmes avec une carte Arduino UNO. Le collège finance un club « programmation robotique » et un autre « création de jeux vidéo » et c'est un réel succès, plus de 20 élèves inscrits sur les deux clubs qui ont lieu pendant la pause méridienne.

Pour ce qui est de la gestion des élèves passionnés il faut prévoir du travail

supplémentaire ou mettre en place du tutorat entre pairs car en règle générale, là où certains auront besoin de 40 minutes pour créer un petit programme, eux le feront en 5 minutes. Mais même s'ils demandent un peu de travail supplémentaire à créer, ces élèves sont, pour l'enseignant que je suis, une réelle source de motivation.

En Angleterre, il y a quelques années, la carte Micro:bit avait été largement distribuée dans les écoles, même chose aux Etats-Unis avec d'autres cartes. Qu'en est-il en France ?

Malheureusement nous ne sommes pas en Angleterre... En France c'est très différent d'un collège à l'autre. Dans le mien, impossible de négocier l'achat de cartes et de modules programmables. J'ai donc financé moi-même le matériel à hauteur de 600 euros. Le lycée d'à côté qui a davantage de moyens a également financé du matériel qu'il met à disposition dans une des salles du Lycée. Le collège a tout de même financé l'achat de 5 Mbots mais cela reste insuffisant pour une étude à mener pendant 4 ans. J'espère que le gouvernement prendra un jour exemple sur nos voisins et commandera du matériel fiable pour l'ensemble des collèges de France. Je ne crois pas que ce soit à l'ordre du jour...

Est-ce que les parents s'investissent aussi ou regardent-ils la programmation comme une entité étrange ?

Les parents qui suivent la scolarité de leur progéniture voient la programmation autrement et essaient souvent de suivre en même temps que leur enfant. Je me rappelle d'un papa avec qui j'ai parlé de la domotique liée au google home pendant plus d'une heure.

Mais certains continuent de diaboliser les écrans et la programmation alors que la solution se trouve plus dans un accompagnement pour un usage raisonné, mais cela est un autre débat.



Damien Nicolet
Développeur Java dans
une PME le jour, et
« bidouilleur » le reste
du temps.

NumWorks : la calculatrice pour développeurs !

NumWorks, fabricant français, est le petit nouveau dans le monde un peu fermé de la calculatrice. Le premier modèle a été introduit pour la rentrée scolaire 2017 et vise pour l'instant les élèves de lycée, toutes filières confondues, avec pour mot d'ordre : la simplicité d'utilisation.



NumWorks 1



Coloration syntaxique Python 2

Elle dispose du mode examen et de toutes les fonctionnalités utiles en lycée, comme les suites, les probabilités, les statistiques, les matrices, les régressions, le solveur d'équations et surtout la programmation en langage Python. Pour l'essayer, il existe un émulateur gratuit en ligne : <https://www.numworks.com/fr/simulateur/>.

Chaque notion du programme de mathématiques affichable sur une calculatrice est présente sous la forme d'une application dédiée : 1

Sur l'écran d'accueil, on a une icône pour un accès simplifié à chaque application :

- **Calculs** : cette application est composée de toutes les fonctionnalités classiques disponibles sur les autres modèles étudiés pour le lycée ;
- **Fonctions** : l'application met à disposition toutes les fonctionnalités permettant d'étudier des fonctions. Les différents outils disponibles permettent de manipuler la tangente en un point, la fonction dérivée et l'intégrale entre autres ;
- **Suites** : la NumWorks dispose d'une application dédiée aux suites, qui permet de les étudier facilement et de les comparer ;
- **Statistiques** : une fois les données saisies, le diagramme en boîte s'affiche ainsi que les paramètres de la série sans manipulation supplémentaire. La comparaison de deux séries est très simple et très utile au lycée ;
- **Probabilités** : l'implémentation des lois de probabilités est vraiment un plus par rapport à ce qui existe chez la concurrence. La manipulation aisée des seuils, avec l'affichage sur la fonction à densité n'a pas son pareil en termes de pédagogie ;
- **Régressions** : l'application permet d'étudier des nuages de points de données ;
- **Python** : l'application permet de saisir des scripts et de les interpréter dans une console, exactement comme cela est demandé dans le Bulletin Officiel.

Cette dernière est une vraie nouveauté et permet à la calculatrice d'être conforme à l'aménagement des programmes en vigueur au lycée concernant la programmation. La coloration syntaxique finit de mettre en valeur cette application. 2

Une de ses principales caractéristiques de cette nouvelle calculatrice est d'être ouverte, aussi bien au niveau du hardware que du software.

La section "engineering" du site officiel est une mine d'informations. Globalement, tout le nécessaire pour modifier la calculatrice est directement disponible (schémas, spécifications des composants, ...).

Un point à noter : la partie matérielle est sous licence Creative Commons BY-NC-ND qui interdit les dérivés. La licence du logiciel est récemment passée sous Creative Commons BY-NC-SA, ce qui autorise maintenant les dérivés. En effet, auparavant, la licence BY-NC-ND interdisait de modifier le code source, et donc les contributions, mais NumWorks acceptait les Pull-Request sur GitHub. Il est possible d'essayer la calculatrice directement en ligne. Un simulateur est disponible sur le site de NumWorks : <https://www.numworks.com/fr/simulateur/>

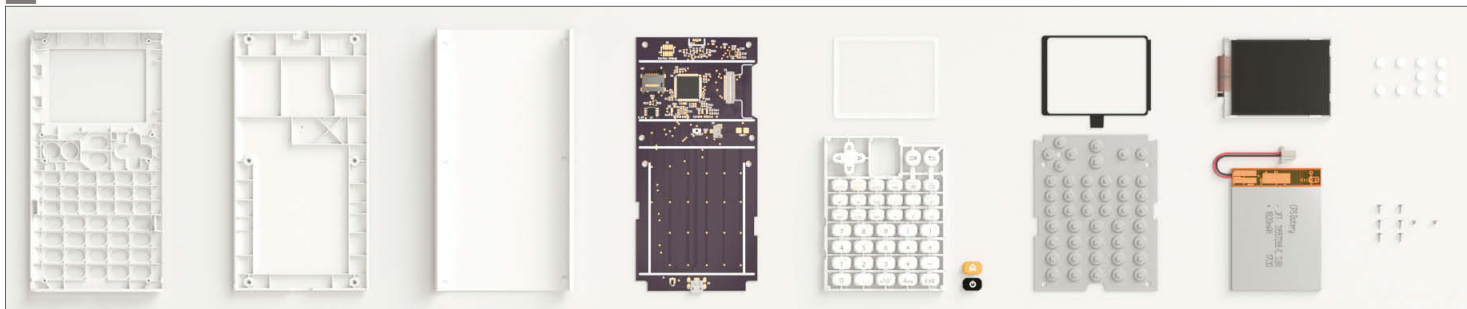
Présentation du matériel

La carte électronique est très dépouillée. Une face ne contient que les contacteurs du clavier, et l'autre assez peu de composants. 3

Les principaux périphériques "utilisateurs" de la calculatrice sont les suivants :

- Un microcontrôleur STM32F412, le cerveau de la calculatrice ;
 - Un écran LCD couleur rétroéclairé de 324x240 pixels basé sur un contrôleur ST7789V ;
 - Une LED RGB, utilisée pour le mode examen ;
- On peut aussi trouver des connecteurs supplémentaires qui ne sont pas utilisés actuellement :
- Un port JTAG ;

3 Composants



- Un UART ;
- Un port SPI ;
- Un emplacement pour carte Micro SD ;
- Un emplacement pour une puce de flash externe.

Le microcontrôleur

Il fait partie de la grande famille des microcontrôleurs à cœur ARM. Son architecture est parente de celle utilisée sur la quasi totalité des smartphones, mais adaptée à l'embarqué (puissance, consommation, ...). Nous avons donc affaire à un processeur 32 bits, couplé à 256 Ko de RAM et avec une flash intégrée de 1 Mo. Cela peut paraître un peu limité, mais il se situe plutôt dans le "haut de gamme" en termes de microcontrôleur.

Ce qui va distinguer un microcontrôleur d'un processeur traditionnel est l'intégration d'un certain nombre de périphériques à l'intérieur de la puce. En plus de la RAM et de la flash déjà citées, on peut trouver, entre autres :

- Un contrôleur mémoire (nommé « FSMC ») ;
- Un contrôleur USB ;
- Des timers / compteurs ;
- Un générateur de nombres aléatoires.

Beaucoup d'autres périphériques sont intégrés, mais ceux-ci seront les principaux utilisés par le firmware. Nous allons détailler leur utilité par la suite. Il est à noter que les connecteurs internes non utilisés sont chacun reliés à un contrôleur qui leur correspond.

L'utilisation d'un microcontrôleur plutôt que d'un CPU traditionnel permet de grandement simplifier la conception de l'électronique. Un CPU va nécessiter des composants supplémentaires : mémoire, flash et contrôleurs divers. De plus les microcontrôleurs sont généralement très peu consommateurs en énergie et déjà optimisés pour un fonctionnement sur batterie.

Les périphériques

Le FSMC

Le « Flexible Static Memory Controller » est un périphérique permettant de connecter des mémoires externes au microcontrôleur, et d'y accéder de manière transparente depuis le firmware.

Dans le cas de la NumWorks, il est légèrement détourné pour accéder à l'écran LCD. L'écran est "vu" depuis la calculatrice comme deux adresses mémoires, une permettant d'envoyer des commandes et l'autre permettant d'envoyer des données à l'écran.

L'interface de bas niveau avec l'écran est assez simple, et est globalement réalisée par deux primitives : définir un rectangle dans lequel l'écran va limiter son affichage (via l'adresse de commande), et envoyer séquentiellement les pixels à afficher (via l'adresse de donnée).

Le contrôleur USB

Il est utilisé dans les versions récentes du firmware, pour envoyer des scripts Python ou mettre à jour la calculatrice.

La programmation de l'USB étant relativement complexe, elle ne sera pas décrite dans cet article.

Timer / compteurs

Un des timers du microcontrôleur est utilisé pour générer la couleur de la LED RGB.

Dans le monde des microcontrôleurs, un timer est un périphérique qui fournit d'une part un registre s'incrémentant à une fréquence déterminée, et d'autre part un ou plusieurs comparateurs permettant de déclencher une action quand le compteur atteint une certaine valeur.

Dans le cas de la LED RGB, ce timer permet de générer un signal PWM (Pulse Width Modulation) qui permet de moduler l'intensité de chacune des composantes rouge, verte et bleue de la LED de façon individuelle.

Un signal PWM est un signal périodique qui vaut donc alternativement 1 ou 0. Sa fréquence est constante, mais le temps passé à 1 est variable. En anglais, on parle de « duty cycle ». Un « duty cycle » de 70 % veut dire que le signal vaut 1 pour 70 % du temps, et 0 pour 30 % du temps. Dans le cas de la LED, le signal PWM la fait clignoter assez rapidement pour que l'œil humain ne perçoive qu'une variation d'intensité lorsque le « duty cycle » change.

Le générateur de nombre aléatoires

Le générateur de nombres aléatoires permet d'obtenir un nombre aléatoire de façon matérielle. Il présente l'avantage de ne pas être déterministe, contrairement aux générateurs logiciels, qui, à partir de la même valeur d'initialisation, retournent la même séquence de nombres.

Dans le cadre d'une calculatrice, cela permet d'éviter que tous les élèves d'une classe ne se retrouvent avec la même séquence de nombres aléatoires lors d'un exercice par exemple.

Présentation du logiciel

Le firmware nommé "epsilon", est disponible sous GitHub : <https://github.com/numworks>.

Il est écrit en C++, et ne dépend d'aucune librairie externe. NumWorks fait le choix de ne pas utiliser de librairies extérieures. On peut donc retrouver dans les sources plusieurs éléments. Pour commencer, une implémentation du runtime "C", c'est-à-dire tout ce qui se passe avant l'appel de la fonction main : initialisation de la pile, appel des constructeurs statiques, initialisation des variables statiques, ... Il y a également une implémentation limitée de la libc, appelée "liba" qui contient un nombre restreint de fonctions. Une implémentation de l'émulation logicielle des nombres flottants complète le microcontrôleur qui contient bien une unité de calcul flottant, mais elle est seulement "simple précision". Tous les calculs en double précision sont donc réalisés par le CPU. Enfin, une implémentation de la libgcc contient des appels qui peuvent être faits par le compilateur vers des fonctions de bas niveau.

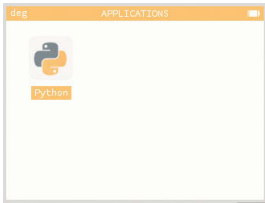
Le firmware est décomposé en plusieurs parties, chacune contenue dans un dossier à la racine du projet.

ION

Ion est la couche d'abstraction du matériel. Tous les accès au matériel sont réalisés par ce composant (écran, clavier, niveau de la batterie, ...). Cela permet de porter facilement le firmware sur différentes plateformes.

Il existe d'ailleurs plusieurs implémentations de cette partie. Une est destinée à la calculatrice physique, une au simulateur ou bien une pour le simulateur en ligne.

C'est ce qui permet au firmware de s'exécuter facilement sur diff-



4

rents environnements, et même sur un autre modèle de calculatrice (<https://github.com/numworks/epsilon/pull/534>). Les différents composants sont :

- Kandinsky : Kandinsky est le moteur graphique, de bas niveau. Il permet de tracer du texte, de remplir des rectangles, ...
- Poincaré : Poincaré est en charge de la partie mathématique. Il s'occupe d'interpréter et d'évaluer les expressions mathématiques.
- Escher : Escher et une boîte à outils graphique. Il fournit des composants de haut niveau (boutons, onglets, ...) et se base sur Kandinsky pour effectuer le tracé.
- Apps : Apps contient l'ensemble applications de la calculatrice, une application par dossier, ainsi que l'écran principal permettant de sélectionner une application.

Le SDK

Installation

Plusieurs SDK sont disponibles en fonction de la plateforme : « device », « simulator » ou « emscripten ». Les deux qui vont nous intéresser sont le « device » pour compiler le firmware qui s'exécute sur la calculatrice, et le « simulator » pour construire un exécutable natif, permettant de tester et débayer epsilon sans la calculatrice. « emscripten » permet de compiler epsilon en JavaScript, pour l'exécuter dans un navigateur. C'est cette version qui est utilisée dans le simulateur sur le site de NumWorks.

L'installation des différents outils est décrite sur : <https://www.numworks.com/resources/engineering/software/sdk/>

Compilation du firmware

Dans cette partie, je me place sous linux, et avec les outils de base installés (git) et le SDK fonctionnel.

Il faut commencer par cloner le dépôt :

```
$ git clone https://github.com/boricj/numworks-hello-world.git
```

Nous allons maintenant compiler une version « simulator » et le lancer. Cela se fait assez simplement :

```
$ make PLATFORM=simulator
$ ./epsilon.elf
```

Si tout se passe correctement, le simulateur de la calculatrice devrait s'afficher à l'écran.

Si cela ne se passe pas correctement, à ce stade la plupart des erreurs proviennent soit d'un problème de « PATH », soit qu'une des bibliothèques nécessaire est manquante.

Par défaut, le Makefile compilera une version « device ». Ici nous utilisons l'option « PLATFORM » pour spécifier la construction de l'exécutable natif.

Il existe d'autres options qu'il est possible de lister en lançant :

```
$ make info
EPSILON_VERSION = 1.5.0
EPSILON_ONBOARDING_APP = 1
EPSILON_SOFTWARE_UPDATE_PROMPT = 1
EPSILON_APPS = calculation statistics graph probability sequence regression settings
code solver
EPSILON_I18N = en fr es de pt
```

« EPSILON_ONBOARDING_APP » et « EPSILON_SOFTWARE_UPDATE_PROMPT » permettent d'inclure ou non la page d'accueil de la calculatrice (notification de mise à jour du firmware et choix de la langue). L'autre option intéressante est « EPSILON_APPS » qui permet de lister les applications à inclure.

Il est ainsi possible de compiler le firmware par exemple avec :

```
$ make PLATFORM=simulator EPSILON_APPS="code"
```

Pour obtenir : 4

Hello world

Nous allons donc ajouter une nouvelle application. Pour cela, il existe un modèle qui peut être utilisé :

<https://github.com/boricj/numworks-hello-world>

```
$ git clone https://github.com/boricj/numworks-hello-world.git apps/hello_world
```

Nous avons donc les sources de l'application « Hello World » dans le sous dossier « apps/hello_world ». Compilons tout de suite avec :

```
$ make PLATFORM=simulator EPSILON_APPS="hello_world settings"
```

Nous obtenons alors : 5

Intéressons-nous un peu aux sources de cette application. Nous trouvons d'abord des fichiers « base.xx.i18n » qui contiennent les locales de l'application. Ces fichiers sont transformés en C++ par un script Python dans le fichier « apps/i18n.cpp ». Le fichier « helloworld_icon.png » contient l'icône de l'application (elle aussi sera traduite en C++, dans le fichier « apps/hello_world/helloworld_icon.cpp »).

Viennent ensuite les sources à proprement parler. Le point d'entrée de l'application est dans le fichier « App.cpp ». L'application est une sous-classe de App de Escher. Elle doit donc implémenter un certain nombre de méthodes. Une partie est liée au « Descriptor » de l'application et couvre les ressources évoquées au paragraphe précédent.

Un des composants importants est le « Snapshot ». Il permet de stocker l'état de l'application lorsqu'elle devient inactive (par sa méthode « pack ») et de le restaurer (par sa méthode « unpack »). Ensuite, nous retrouvons deux fichiers, « hello_controller.cpp » et « hello_view.cpp ». Comme leurs noms l'indiquent, le premier est le contrôleur, qui est chargé de recevoir les événements et mettre à jour le second (la vue) en conséquence.

Les événements sont reçus dans la vue par la méthode « handleEvent » :

```
bool HelloController::handleEvent(lon::Events::Event event)
```

```
if (event == lon::Events::OK || event == lon::Events::EXE) {
    m_helloView.changeColor();
    return true;
}
return false;
}
```

Ici, le contrôleur fait changer la couleur de fond de la vue lors de l'appui sur les touches « OK » ou « EXE ». Retourner VRAI indique que l'événement a été traité dans ce contrôleur, FAUX veut dire qu'il n'a pas été traité et qu'il faut donc parcourir le reste de la hié-

rarchie de l'application.

Le fonctionnement décrit ici est le même pour toutes les applications. Il existe aussi un grand nombre de composants réutilisables qui sont définis dans Escher : boutons, tables, champs texte, ... Le plus simple pour aller plus loin est de s'inspirer de ce qui est fait dans les autres applications.

Installation du firmware sur la calculatrice

Une fois la mise au point terminée, il suffit de lancer :

```
make clean
make epsilon_flash
```

et de connecter la calculatrice à l'ordinateur. Sur les nouvelles versions du firmware, un écran indique que la calculatrice est connectée et le flashage se fait tout de suite. Dans tous les autres cas (version précédente, calculatrice « briquée », ...), il faut appuyer avec un trombone sur le bouton reset au dos de la calculatrice lorsqu'elle est connectée au port USB. Elle exécute alors le bootloader du constructeur de la calculatrice, qui permet toujours de la récupérer.

Quelques mots supplémentaires

La « boucle principale » de la calculatrice prend sa source ici : "escher/src/run_loop.cpp".

Tant qu'on ne modifie pas la partie « ion », il est normalement impossible d'endommager la calculatrice. On peut toujours la récupérer avec la manipulation décrite précédemment.

La liste des événements se trouve ici : "ion/include/ion/events.h".

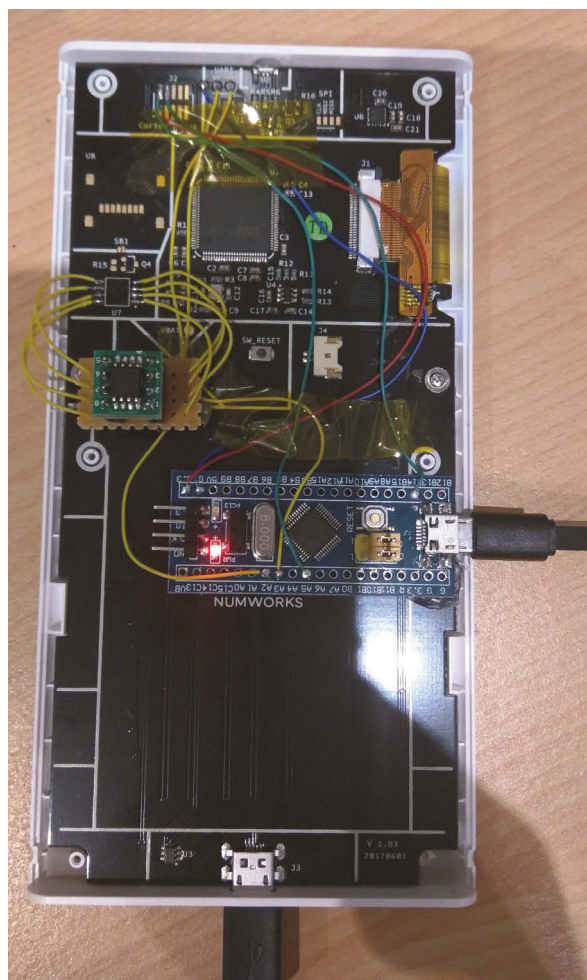
Hacking

Avant toute chose, il ne faut pas oublier que la calculatrice utilise une tension interne de 2,8 V, la connecter à un Arduino ou autre ne doit se faire qu'en prenant en compte cette contrainte. Il y a un risque non négligeable de la griller. (Ce qui m'est personnellement arrivé, et qui s'est soldé par un remplacement du microcontrôleur). L'ouverture de la calculatrice est très simple. Il suffit de décoller les patins blanc, et de dévisser les six vis qui se trouvent en dessus (prévoir un tournevis Torx T4).

Un des ports les plus intéressants est le port JTAG (Joint Test Action Group). Il permet de connecter un débogueur (gdb) directement sur le microcontrôleur. C'est une aide inestimable pour s'aventurer dans la suite. En plus des points d'arrêts et autres débogages classiques, cela permet de manipuler directement les registres des périphériques du microcontrôleur et facilite grandement la compréhension de leur fonctionnement.

Il est possible de souder ce genre de connecteur : <https://1bitsquared.com/products/jtag-swd-smd-connector>. L'emplacement de l'ouverture est prévu dans le boîtier de la calculatrice. Cela permet d'utiliser par exemple une sonde <https://1bitsquared.de/products/black-magic-probe>.

Un exemple de montage utilisé pour tester l'intégration d'une puce de flash externe. La sonde utilisée est en fait le firmware « Black Magic Probe » sur une « Blue Pill » (une carte de développement STM32F103 que l'on peut trouver pour quelques euros en ligne)



6 Exemple de hacking

Il est aussi possible d'exploiter les connecteurs suivants, avec pour certains des travaux plus ou moins aboutis existants :

- UART : <https://github.com/zhaofengli/epsilon/tree/uart>.
- SPI : permet de connecter des circuits externes qui utilisent ce protocole.
- QSPI : permet d'interfacer une puce flash externe pour « étendre » la flash de la calculatrice : <https://github.com/zardam/epsilon/tree/extflash>.
- Carte SD.

Pour comprendre leur fonctionnement et modifier le firmware pour les exploiter, les deux références sont le « datasheet » et le « reference manual » du microcontrôleur, ainsi que le schéma de la carte de la calculatrice, disponibles ici : <https://www.numworks.com/resources/engineering/hardware/electrical/>

Conclusion

La calculatrice NumWorks est très certainement la seule calculatrice open-source sur le marché. C'est un pari assez audacieux, mais qui a déjà permis à une petite communauté de se créer autour de la calculatrice et qui n'attend plus que vous !



Aurélien Vannieuwenhuyze
Dirigeant - Junior Makers Place
www.juniormakersplace.fr

« Junior Makers Place est une académie qui initie, éduque et perfectionne les digital natives de 7 à 18 ans au codage informatique afin de leur donner les clefs de compréhension du monde de demain. »

Un microcontrôleur propice au monde de l'éducation



Arduino, Raspberry et pourquoi pas Micro:Bit comme nouveau compagnon de jeu durant ces prochaines vacances d'été ? Pour vous occuper durant les périodes chaudes de début d'après-midi avant de rejoindre la plage, nous vous proposons de réaliser avec vos enfants un oeil animatronique piloté à l'aide du microcontrôleur Micro:Bit.

Dès sa conception par la société BBC, le microcontrôleur Micro:Bit a été pensé pour un usage éducatif. Doté d'une interface de programmation en ligne, basée sur l'assemblage de briques de code (façon Scratch) ou via le langage Javascript, les plus jeunes peuvent rapidement créer leur premier projet alliant programmation et électronique. On note également la présence d'un simulateur permettant de valider le bon fonctionnement du code et de son interaction avec partie électronique.

Autre point important. Nul besoin de cartes complémentaires, car le microcontrôleur comporte un ensemble de fonctionnalités très appréciables :

- Un accéléromètre,
- Une boussole numérique,
- Le Bluetooth,
- Un capteur de température,
- Un afficheur de 25 leds.

Simplicité, fonctionnalités et prix abordable (environ 20€) font de ce microcontrôleur un point d'entrée intéressant dans le monde des Makers.

Le projet

Nous allons réaliser l'animation d'un oeil animatronique à l'aide du microcontrôleur Micro:Bit. Dans cet article, nous n'allons pas décrire la partie mécanique de l'oeil que vous pourrez retrouver sur notre site (www.juniormakersplace.fr), mais allons nous attarder sur le code permettant à deux servomoteurs de faire bouger l'oeil de gauche à droite et de haut en bas.

Pour réaliser ce projet, il convient d'acquies le matériel suivant :

- 1 Microcontrôleur Micro:Bit,
- 2 Servomoteurs,
- 1 coupleur de piles AAA 4,5 v avec interrupteur,
- 3 cosses ou pinces crocodiles pour connecter les câbles sur le microcontrôleur,
- Du câble et des dominos. **1**

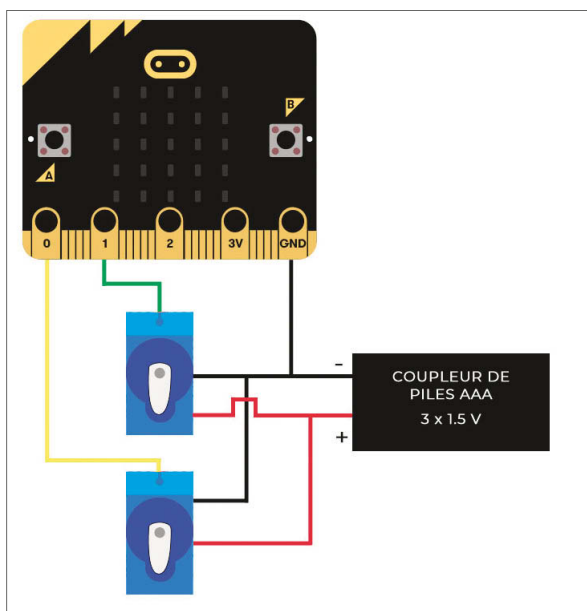
Le câblage

Pour tout montage nécessitant l'emploi de servomoteurs, il est conseillé d'utiliser une alimentation externe pour ces derniers, car la carte Micro:Bit ne délivre que 3V en sortie. Cette tension étant insuffisante au bon fonctionnement des servomoteurs.

Nous utiliserons donc un coupleur de piles délivrant une tension de 4,5 V pour alimenter les servomoteurs le Micro:Bit sera quant à lui alimenté par le port USB de notre PC. Seuls les câbles de commande de position des servomoteurs seront raccordés aux bornes P0 et P1 du microcontrôleur Micro:Bit. **2 3**

Plages de positions

Nous souhaitons faire bouger l'oeil de façon aléatoire de droite à gauche et de haut en bas selon une plage de positions prédéfinies pour chaque servomoteur. Ces plages dépendent de la structure mécanique que vous aurez réalisée et peuvent être adaptées librement suite aux tests que vous aurez effectués.



Servo	Borne du Micro:Bit	Position Min	Position Max
1	P0	80	100
2	P1	80	100

Réaliser la mise en rotation des servomoteurs

Afin de valider le câblage et le bon fonctionnement des servomoteurs, nous allons coder via l'interface de programmation Micro:Bit (<https://makecode.microbit.org/>) leur rotation.

Deux options s'offrent à nous pour programmer le microcontrôleur :

- Une interface graphique via l'assemblage de briques de fonctionnalités,
- Un éditeur de code JavaScript.

Dans cet article, nous choisissons d'utiliser la première option, car ce projet doit pouvoir être réalisé par des novices en programmation.

Utilisons les briques de code suivantes et téléversons le code sur le Micro:Bit

(les briques permettant de commander les moteurs se trouvent dans la rubrique Avancé > Broches de l'éditeur) **4**

Lorsque le bouton A ou le bouton B est pressé, les moteurs connectés aux bornes P0 et P1 du micro:bit doivent se mettre en rotation. Si tel est le cas, nous pouvons alors passer au codage de la mise en mouvement de notre oeil animatronique. Dans le cas contraire, il convient de vérifier le câblage.

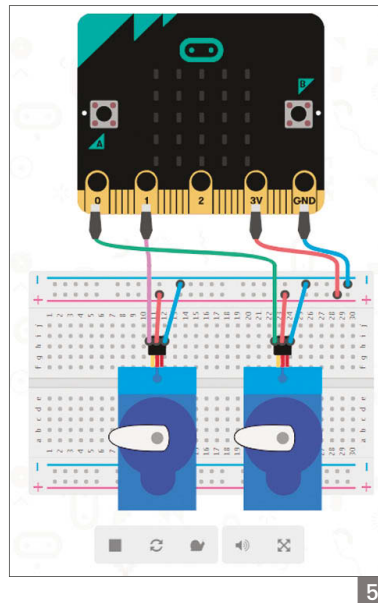
Il est également possible de tester le code avec le simulateur fourni par l'interface graphique. Cela peut s'avérer fort pratique pour éviter toute détérioration du matériel et être utilisé comme outil de débogage.

5

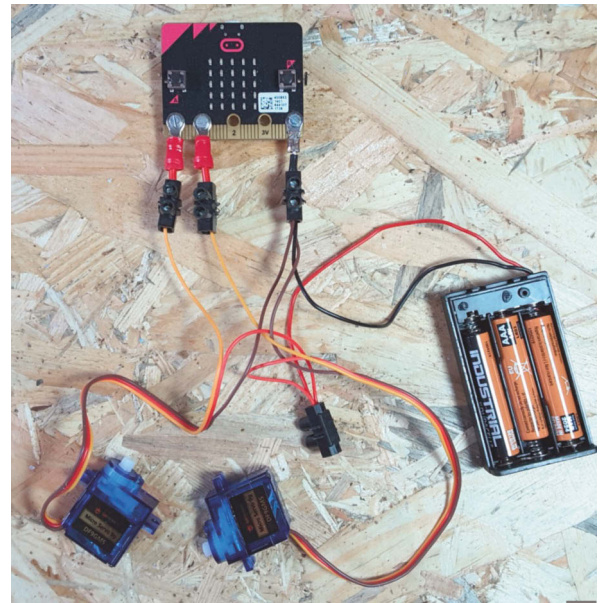
Un défi de taille: générer un nombre aléatoire entre deux valeurs avec le Micro:Bit

Nous allons avoir besoin de générer pour chaque servomoteur, un nombre compris entre 80 et 100 correspondant aux positions minimales et maximales dans le but de rendre l'animation de notre animatronique aléatoire.

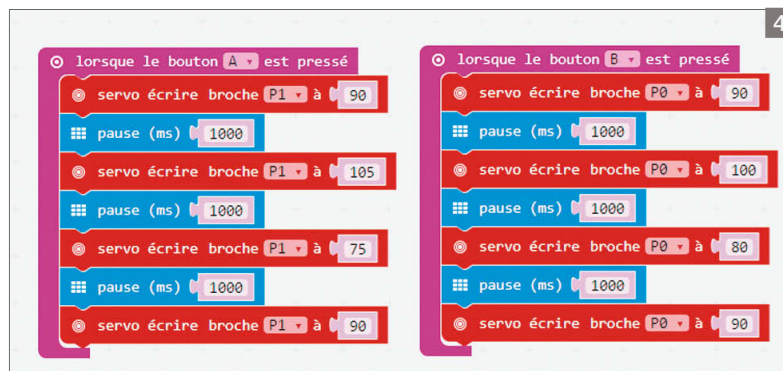
Sur l'interface de programmation Micro:Bit, il ne nous est possible que de générer un nombre aléatoire compris entre 0 et une autre valeur, or ce dont nous avons besoin



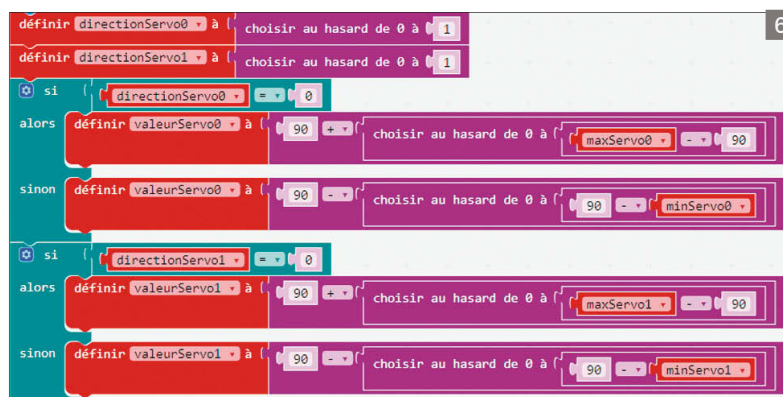
5



3



4



6

est de choisir un nombre entre 80 et 100. Pour résoudre ce souci, nous allons choisir un nombre compris entre 0 et 10 que nous allons ajouter ou retirer à la valeur de centrage du servomoteur (90).

Cet ajout ou cette suppression se fera par un choix également aléatoire. **6**

Explication de l'algorithme :

Choix de la direction du servomoteur : 1 ou 0

Si la direction est égale à 0

Alors

On calcule l'angle de déplacement en ajoutant à la valeur de centrage (90) une valeur comprise entre 100-90 correspondant à la valeur maximum du servo moteur auquel on retranche sa valeur de centrage

Sinon

On calcule l'angle de déplacement en re-tranchant à la valeur de centrage (90) une valeur comprise entre 90-80 correspondant à la valeur de centrage du servo à laquelle on retranche sa valeur minimum.

Nos moteurs sont à présent capables d'évoluer de façon aléatoire sur leurs plages de positions définies.

Déplacement en pas-à-pas

Lorsque la nouvelle position est affectée au servomoteur celui-ci se déplace immédiatement vers celle-ci.

Afin de rendre le déplacement de l'oeil plus lent, nous avons mis en place un déplacement en pas à pas suivant cet algorithme :
Prise en compte de l'ancienne position du moteur : anciennePosition

Prise en compte de la nouvelle position du moteur : valeurServo

Calcul du nombre de positions d'écart :

Si valeurServo > anciennePosition alors
nombreDePositions
= valeurServo-anciennePosition

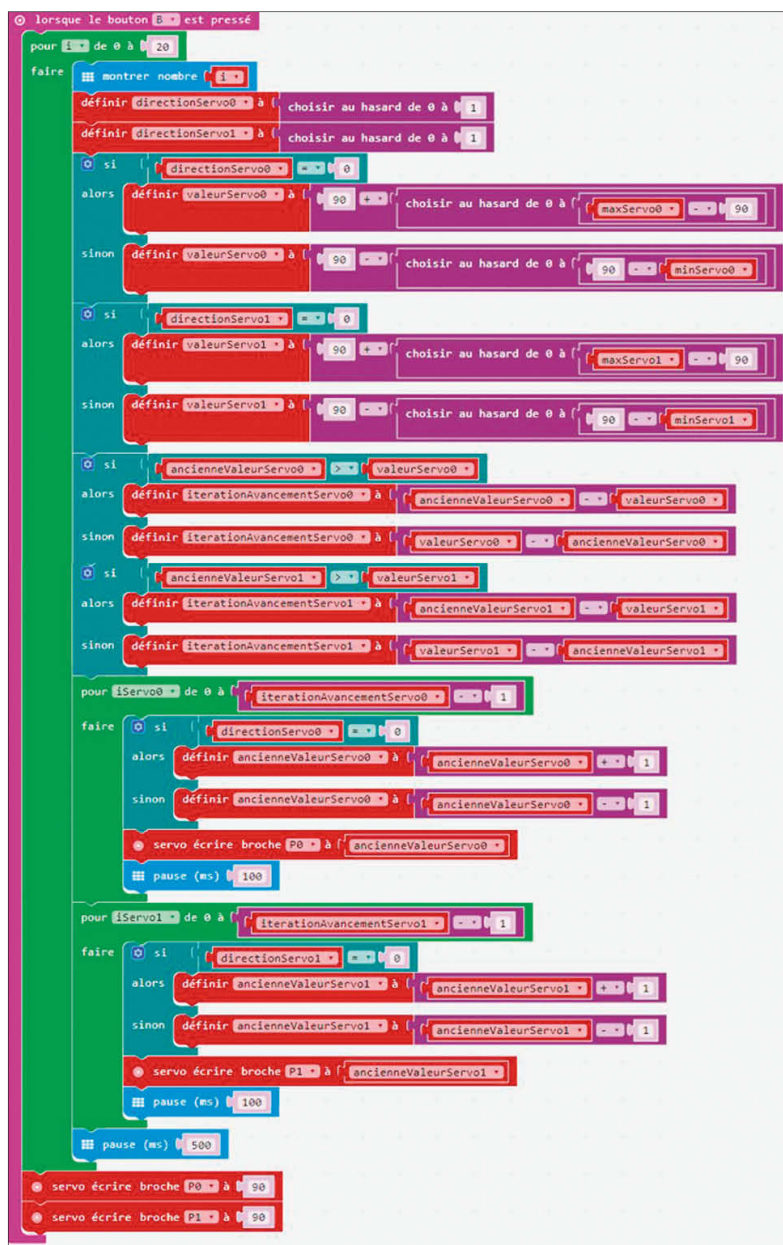
Sinon

nombreDePositions
= anciennePosition-valeurServo

On itère ensuite sur ce nombre de positions en réalisant une rotation du moteur en fonction de la direction tirée au sort préalablement et en faisant une pause entre chaque rotation.

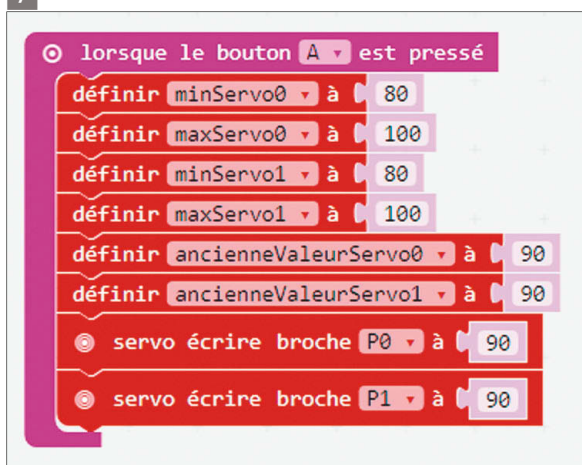
Le code complet

Lorsque l'on appuie sur le bouton A, on initialise les variables et on positionne les



8

7



servomoteurs à leur point de centrage.

C'est également le moment de vérifier que la partie mécanique de l'animatronic est en bonne position (les éléments doivent se trouver en position centrée). 7

Une fois le bouton B appuyé, l'oeil se met en mouvement pendant une itération de 20 mouvements. Ce nombre peut bien entendu être modifié à souhait. Chaque numéro d'itération est affiché sur l'afficheur LED du Micro:Bit 8

Pour aller plus loin

Pour ce projet, nous avons utilisé l'interface de programmation simplifiée. Pour aller plus loin il est possible d'utiliser le langage

JavaScript voire le langage Python : <https://python.microbit.org/v/1>

Bien que ce microcontrôleur offre de nombreuses possibilités de création de projets, le nombre de broches disponibles peut devenir un frein à la créativité.

Pour notre projet, il serait impossible de piloter un second oeil animatronic, car nous ne pourrions connecter qu'un seul servomoteur sur les deux requis. Un passage au microcontrôleur de type Arduino peut alors s'avérer utile.



Nicolas Decoster
Informaticien et scientifique chez Magellium
Co-fondateur et animateur à la Compagnie du Code

Scratch 3 bientôt en ligne

Scratch est un grand classique de la programmation pour les enfants. Il a été conçu et développé par le MIT (Massachusetts Institute of Technology) sur les bases solides des idées de Seymour Papert et de son langage Logo, comme outil d'expérimentation et de création par excellence.

Le but étant de programmer des choses aussi variées que des jeux, des quizz, des histoires animées, etc. Il s'accompagne d'une communauté impressionnante et favorise le partage et l'échange des projets développés. Mais malgré ce succès, et bien que les bases conceptuelles soient très solides, d'un point de vue technique la version courante (Scratch 2) repose sur une souche technique mourante. Elle a en effet été développée en Flash à une époque où partir sur cette techno était un choix raisonnable. Mais depuis, les standards web se sont renforcés et offrent de plus en plus de possibilités, au point qu'Adobe a déclaré fin 2017 la fin programmée du support de Flash pour 2020. Mais le MIT n'a pas attendu cette annonce : il a décidé d'abandonner Flash pour le développement de Scratch, et a lancé il y a presque deux ans un vaste chantier de réécriture complète de Scratch, en n'utilisant cette fois que les standards HTML5, CSS et JavaScript. Une première version de test est déjà en ligne pour ceux qui veulent commencer à l'expérimenter avant la sortie officielle imminente, a priori d'ici la fin 2018.

Nouvelle apparence

Première et principale nouveauté, le look a été complètement revu. On retrouve bien tous les blocs, les catégories, les espaces dessin et son, la scène, etc., mais l'interface a été rajeunie. Le MIT a revu l'aspect des blocs : leurs couleurs sont un peu moins vives et ils sont plus gros pour une meilleure manipulation sur tablette.

Une particularité de cette nouvelle version est que les catégories stylo et musique sont maintenant vues comme des extensions, et pour les utiliser il

faut les charger explicitement depuis le volet extension (accessible par un bouton en bas à gauche). Au passage, dans ce volet extension on peut voir sur quoi le MIT est en train de travailler pour ouvrir Scratch à d'autres possibilités : des extensions Micro:bit et Lego (WeDo, Mindstorms et Boost) sont prévues pour interagir avec le monde matériel, une extension de traduction est déjà disponible (que nous présentons un peu plus loin) et une autre offrira des capacités de reconnaissance vocale aux scratcheurs. **1**

Un petit ajout, mineur dans la mise en oeuvre mais qui est très pratique, est la partie réglage des propriétés du lutin, telles sa position ou sa direction. Dans Scratch 2, ces informations n'étaient accessibles qu'après un clic droit sur le lutin et n'étaient pas toutes modifiables. Maintenant, elles apparaissent au-dessus de la liste des lutins et on peut changer directement les valeurs. **2**

Indiquons également qu'un nouvel outil permet de sélectionner les directions de manière plus visuelle et interactive pour le bloc "s'orienter à" ("point in direction" dans la version anglaise). **3**

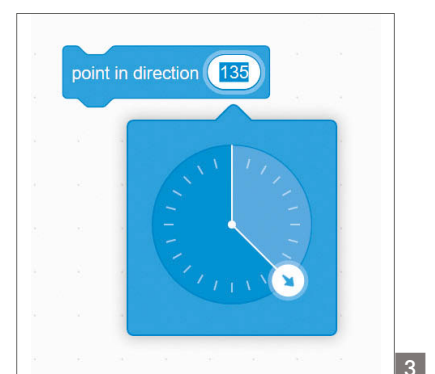
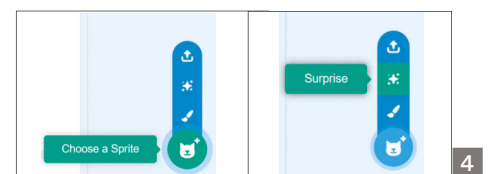
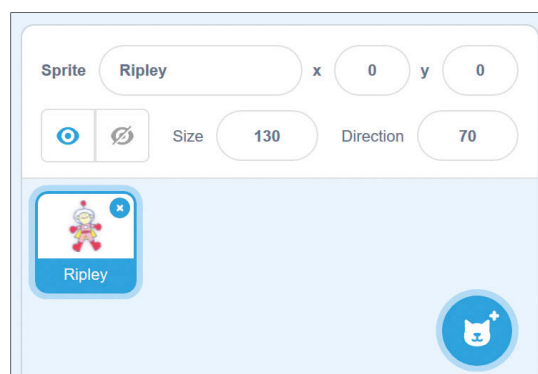
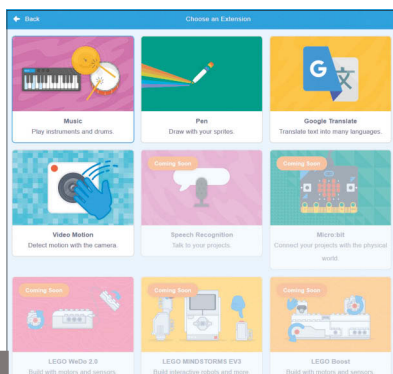
Le choix de nouveaux lutins, sons et arrière-plans a été simplifié et est plus accessible. Une option "surprise" permet de laisser Scratch choisir pour nous aléatoirement dans la bibliothèque disponible. Cela permet de pallier les manques d'inspiration (qui sont assez courants lors d'ateliers d'informatique créative) en se laissant guider par le hasard. **4**

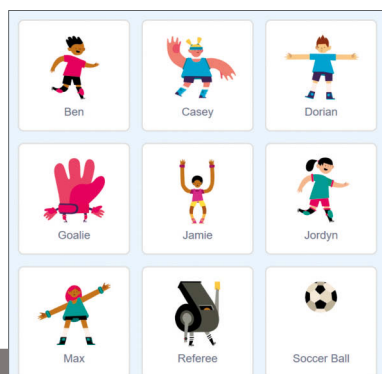
Pour cette nouvelle version, il y a également un effort réalisé sur le contenu des différentes bibliothèques. Le MIT travaille avec des graphistes et des musiciens pour produire de nouveaux lutins,

sons et arrière-plans de qualité. Pour l'instant, certains éléments présents dans Scratch 2 ont disparu et de nouveaux lutins et arrière-plans sont apparus, en particulier sur les thèmes de l'espace, du fantastique, du sport et des dinosaures. Avec Scratch 2, il est laborieux de visualiser les costumes associés à un lutin. Pour cela, il faut ouvrir la bibliothèque, sélectionner un lutin, et une fois celui-ci importé dans le projet, consulter l'onglet "costumes". Si cela ne convient pas, il faut supprimer le lutin et recommencer l'opération avec un autre. Cette sélection de lutins en fonction des costumes est fortement simplifiée avec Scratch 3 : dans la bibliothèque, il suffit de passer la souris au-dessus d'un lutin pour voir défiler tous ses costumes. **5**

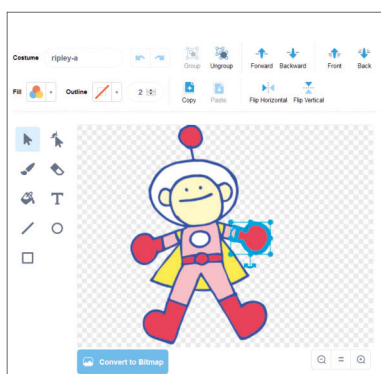
Scratch 3 propose un nouveau système d'aide en ligne. Pour l'instant, le contenu est loin d'être aussi riche qu'avec Scratch 2, mais on peut déjà en apprécier le potentiel. Il s'agit tout simplement d'une petite fenêtre de démonstration qui s'affiche mais qui ne bloque pas l'interface : on peut suivre les vidéos des différentes étapes tout en exécutant les consignes en simultané.

L'outil de dessin a également été retravaillé. Il ne supporte pas encore l'édition en mode bitmap, mais il offre déjà plus de contrôles en mode vecteur ; entre autres pour travailler sur les noeuds de manipulation des formes, les couches d'affichage,





5



6

la sélection et le réglage des couleurs. 6

L'outil d'édition de son a été simplifié et quelques effets ont été ajoutés, comme "accélération", "ralentissement", "écho" ou "robot".

Nouveautés pour la programmation

Comme déjà évoqué, cette nouvelle version se concentre essentiellement sur deux objectifs : abandon de flash pour des standards web et refonte esthétique. Cependant, le MIT a profité de cette réécriture pour ajouter quelques blocs et en modifier quelques autres afin de faciliter la vie des développeurs. Un nouveau bloc de type événement est apparu, "when timer > (10)", qui permet de lancer un script s'exécutant dès que le chronomètre dépasse une valeur. En effet, il est courant que l'on demande à l'utilisateur d'effectuer une action dans un temps imparti (réponse pour un quizz, par exemple), contrôlé avec un chronomètre. Avant l'arrivée de ce bloc, il fallait coder cela avec une boucle infinie qui surveillait le chronomètre, ce qui était moins élégant. Lorsqu'on développe un projet qui manipule fréquemment des mots et des phrases, ou plus généralement des chaînes de caractères, on a vite besoin de détecter si un mot en particulier est présent dans une phrase (réponse d'un utilisateur, par exemple). Avec Scratch 2, cela est possible mais laborieux : il faut parcourir les deux chaînes caractère par caractère pour effectuer la comparaison. Scratch 3 introduit un nouveau bloc de type question qui fait exactement cela : "(apple) contains (a) ?".

Dans le même ordre d'idée, avec Scratch 2, il est compliqué de savoir à quel indice se situe un élément donné d'une liste (comme lorsqu'on cherche un élément dans un sac à dos pour l'en retirer). Pour ce faire, il faut utiliser une boucle sur les éléments de la liste en incrémentant une variable pour l'indice, comparer l'élément recherché à l'élément courant et sélectionner l'indice s'ils sont égaux. Tout cela est laborieux, surtout si l'on doit faire cette opération régulièrement. Scratch 3 introduit un nouveau bloc qui fait exactement cela :

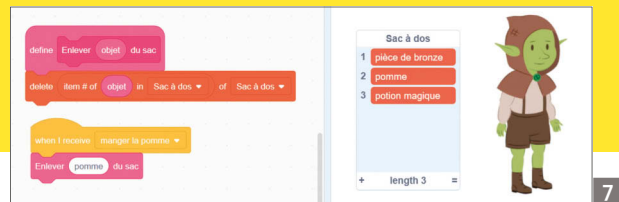
"item # of (thing) in (items)". Rappelons que pour les Anglo-Saxons, le caractère "#" est souvent utilisé pour "number" ou, ici, "indice". 7

Comme nous l'avons dit, Scratch 3 propose une nouvelle extension de traduction. Elle se base sur Google Translate et s'utilise simplement en donnant une phrase et en indiquant la langue dans laquelle on veut la traduire. Pour l'instant, on ne peut pas choisir la langue de départ et la phrase doit être en anglais, mais on peut raisonnablement espérer que dans une version future on pourra traduire des phrases provenant de n'importe quelle langue. À noter qu'il est possible de récupérer la langue courante de l'utilisateur, stockée dans une variable. 8

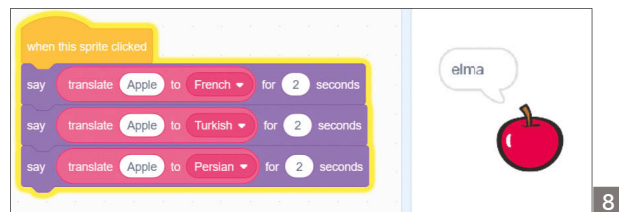
En ce qui concerne les possibilités graphiques, cette version de Scratch offre une avancée importante : il est maintenant possible de choisir plus précisément les couleurs du stylo. D'une part, la présence d'une palette permet la sélection de la couleur pour le bloc "mettre la couleur du stylo à", contrairement à Scratch 2 où l'on ne peut choisir des couleurs qu'avec la pipette (par conséquent, celles qui ne sont pas affichées à l'écran sont exclues). D'autre part, la sélection de la couleur à afficher de manière programmatique (c'est-à-dire par bloc et avec une valeur) est enrichie : avec Scratch 2, on ne peut positionner que la teinte (appelée tout simplement "color") et la luminosité ; avec Scratch 3, il est également possible de positionner la saturation et la transparence, ce qui ouvre très largement la palette des couleurs que l'on peut dessiner avec le stylo.

Adoption

Bien entendu, malgré une réécriture complète, la compatibilité avec Scratch 2 est un objectif important. Cependant, sachant que le moteur d'exécution et le moteur de rendu sont complètement différents (on passe de Flash à HTML5/JavaScript), la compatibilité totale est très difficile à obtenir. Pour tester cela, il est possible d'importer un projet Scratch 2 présent sur le site officiel de Scratch à partir de son identifiant. Les tests



7



8

que nous avons effectués montrent que pour la plupart des projets simples tout fonctionne correctement. Lorsque les projets sont un peu plus compliqués, il peut y avoir des petits dysfonctionnements, voire une incompatibilité importante pour de très gros projets (comme le fameux Pac-Man d'aoliver2). Étant donné que le MIT se donne comme objectif de remplacer définitivement Scratch 2 par Scratch 3 pour le rendu des millions de projets présents sur le site officiel, on peut supposer que la compatibilité sera la plus complète possible. Étant basé sur des standards récents, Scratch 3 supporte tous les navigateurs modernes : Chrome (63+), Edge (15+), Firefox (57+) et Safari (11+) ; par contre, Internet Explorer n'est pas supporté. En ce qui concerne les plateformes mobiles, Scratch 3 supporte Chrome (62+) et Safari (11+). Cependant, l'édition n'est possible que sur tablette ; les smartphones peuvent quant à eux afficher des projets Scratch 3, mais sans possibilité de les modifier. Pour l'instant, la version de test est en anglais mais d'autres langues, dont le français, sont prévues pour la sortie officielle, de même qu'un certain nombre de fonctionnalités encore en cours de développement, comme le sac à dos ou le dessin en mode bitmap. Bien sûr, les fonctions en lien direct avec l'aspect communautaire de Scratch, comme le stockage en ligne, le partage ou le remix, ne seront branchées que lors du lancement sur le site officiel. Comme nous l'avons évoqué, Scratch 3 propose un mécanisme d'extension. Celui-ci va être ouvert aux développeurs désirant fournir de nouvelles extensions pour étendre les possibilités de Scratch. Les anciennes extensions ScratchX ne seront pas compatibles, mais la rédaction d'un guide de migration est prévue.

Une fois Scratch 3 officiellement lancé, la version 2 ne sera plus disponible en ligne. Toutefois, il sera possible de télécharger les versions offline de Scratch 2 et Scratch 1.4 pendant encore plusieurs années. Enfin, indiquons que l'ensemble est développé comme une application web statique et qu'il est donc très facile d'héberger soi-même une version de l'éditeur Scratch 3 (ce que l'Académie de Grenoble a déjà fait). De plus, la fin de la dépendance avec Flash en fait un outil utilisable partout où un navigateur web est présent, favorisant ainsi encore un peu plus son adoption. •



Maxime Ellerbach
Lycéen



Laurent Ellerbach
Software Engineer Manager, Microsoft
laurelle@microsoft.com

Automatiser un bras robotique avec un ESP8266

L'idée du projet initial était d'automatiser un bras robotique simple pour apprendre à Maxime à programmer dans un contexte ludique. Le but étant de remplacer le montage de base consistant au pilotage manuel avec des manettes reliées directement aux moteurs par une page web avec un dessin du bras et un pilotage en cliquant sur les éléments du bras.

Ce projet permet d'apprendre plusieurs technologies comme le fonctionnement de http, HTML, coder en C++ et faire un peu d'électronique. C'est donc un projet assez complet sans être non plus trop compliqué.

J'avais déjà acheté le bras robotisé depuis un moment et il prenait la poussière. Jusqu'au jour où Maxime a voulu se remettre à faire un peu de développement sur Arduino. Je lui ai donc naturellement proposé ce projet qu'il a accepté avec grand plaisir.

Il est bien sûr possible de remplacer le bras robotisé par n'importe quel matériel possédant plusieurs moteurs et pilotable depuis de simples interrupteurs.

J'avais une idée précise de l'architecture complète. Ce n'était pas mon premier projet de ce type. J'ai donc découpé le projet en plusieurs éléments. A chaque partie terminée, je lui donnais donc le travail suivant à effectuer.

Maxime : j'ai pris plaisir à refaire de l'Arduino, cela faisait longtemps que je n'avais pas codé du hardware. Ce projet était donc l'occasion pour moi de programmer quelque chose de concret tout en m'améliorant en programmation.

Première étape : paramétrage de l'ESP8266

Laurent : j'ai opté pour un ESP8266 de type nodeMCU pour ses nombreuses entrées-sorties. J'ai indiqué à Maxime d'installer l'application Arduino depuis le Windows Store. Ce qu'il a fait. Il fallait ensuite installer les drivers pour l'ESP. Pour cela, Maxime n'ayant pas les droits administrateurs, j'ai dû effectuer l'installation.

Maxime : après l'installation des drivers, nous avons téléchargé une librairie nous permettant de connecter l'ESP au wifi. Le reste était plutôt simple, il suffisait juste d'inclure la librairie et de définir le ssid et le mot de passe du wifi.

Deuxième étape : l'électronique

Laurent : le bras est simple et possède 5 moteurs et une lampe. Le principe de base pour piloter un moteur est d'utiliser un Hbridge. Il s'agit d'un montage tout à fait classique. Dans un premier temps, je n'avais que des composants bruts à base de L298N. Il a donc fallu à Maxime qu'il fasse un peu de câblage comme indiqué dans

le circuit avec 2 moteurs pour les premiers tests. J'ai ensuite acheté un montage intégré facilitant le câblage. **1**

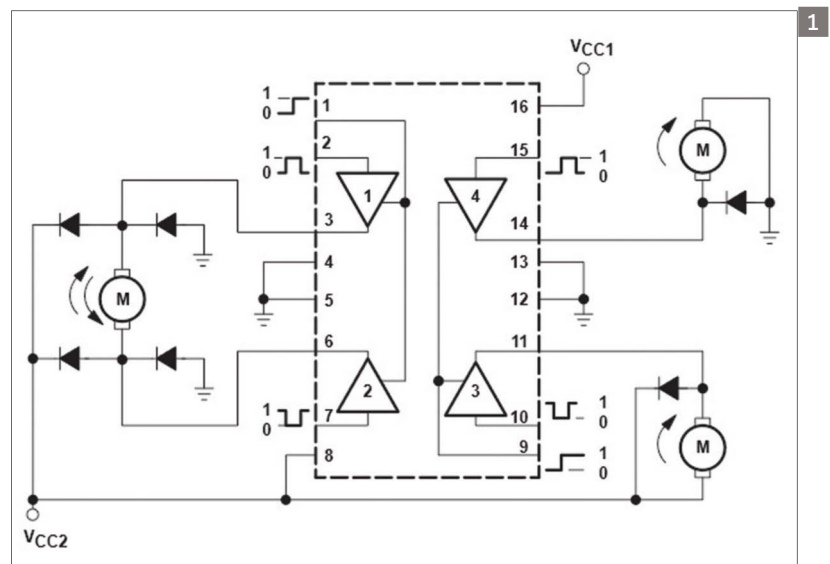
Maxime : les premiers Hbridges étaient compliqués à mettre en place et prenaient beaucoup de place. Le plus dur était le placement des diodes qui étaient un peu fragiles donc très facile à tordre sinon, cela m'a permis de réviser quelques notions en électricité !

2 3

Troisième étape : commander les moteurs à partir d'une URL

Laurent : pour faciliter la tâche, je suis parti sur l'utilisation d'une librairie REST existante (<https://github.com/marcoschwartz/aREST>). L'intérêt est de pouvoir piloter simplement tous les IO directement depuis le browser sans avoir à coder quoi que ce soit dans un premier temps. La librairie offre aussi la possibilité d'ajouter ses propres fonctions de call back ce qui sera utilisé plus tard pour piloter les moteurs. De plus la documentation est bien faite et il y a un exemple qui permet de commencer directement avec un ESP8266.

Maxime : ici, les fonctions sont définies :



Le matériel

Bras robotisé :
environ 50€

ESP8266 :
4€ sur Banggood.com (on peut trouver – cher, NDLR)

3 Hbridge :
12€ sur Banggood.com

Board de prototypage et câbles 7€ sur Banggood.com

Soit environ 80 €


```
rest.function("MU", MUp);
rest.function("MD", MDown);
rest.function("MS", MStop);
```

Lorsque l'adresse ip est suivie de /MU, la Fonction MUp est appelée

```
int MUp(String command) {
  int state = command.toInt();

  if (state == 1) {
    digitalWrite(D0, HIGH);
    digitalWrite(D1, LOW);
  }
}

if (state == 2) {
  digitalWrite(D2, HIGH);
  digitalWrite(D3, LOW);
}
// etc
}

int MStop(String command) {
  int state = command.toInt();
  if (state == 1) {
    digitalWrite(D0, LOW);
    digitalWrite(D1, LOW);
  }
  if (state == 2) {
    digitalWrite(D2, LOW);
    digitalWrite(D3, LOW);
  }
  // etc
}
```

La variable state est utile pour éviter qu'il y ait une fonction par moteurs. State permet de déterminer quelle sortie (D0, D1, D2, D3 etc...) mettre à high ou low. La fonction MDown est l'inverse de MUp (lorsque MUp met D0 en HIGH et D1 en LOW, MDown met D0 LOW et D1 en HIGH).

Exemple : <http://adresseip/MU?p=2>

Cet URL appellera la fonction MUp et met la fonction « state » à 2 qui va mettre la sortie D2 à high et la sortie D3 à low.

Exemple : <http://adresseip/MS?p=2>

La fonction MStop est appelé, le paramètre est 2 ce qui arrête le moteur 2.

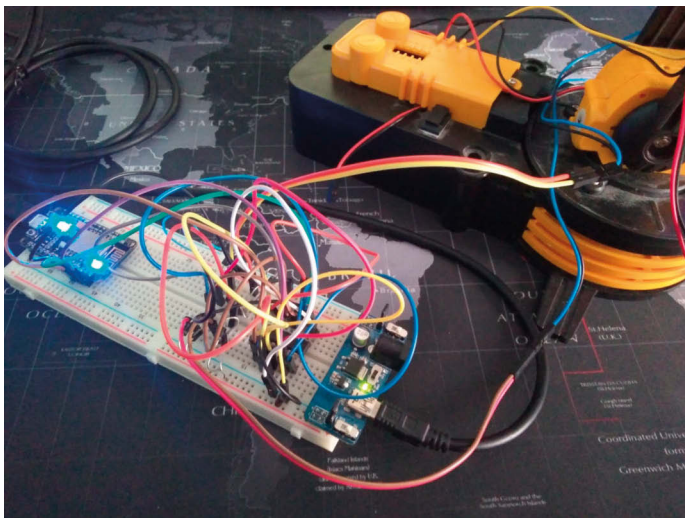
Quatrième étape : commander les moteurs depuis une page web

Laurent : j'ai fourni à Maxime une page web vraiment simple contenant un bouton, un script permettant d'appeler une URL. Je lui ai ensuite demandé de créer une page incluant l'appel à tous les moteurs.

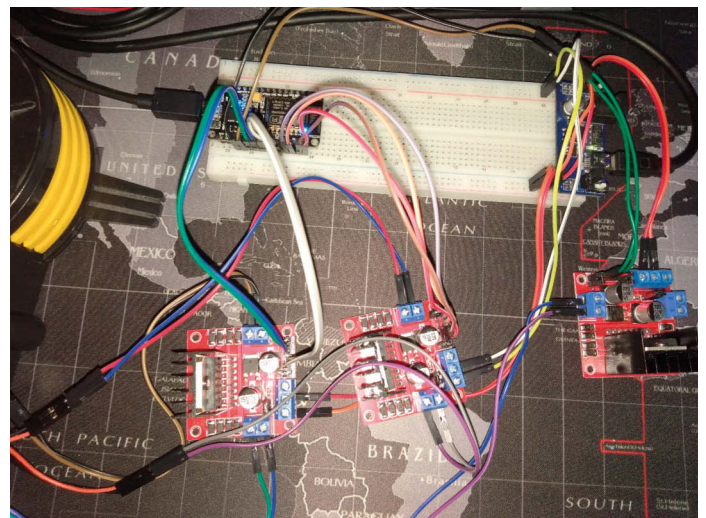
```
<html>
  <head>
    <title>Exemple de page web</title>
  </head>
  <body>
    <button onclick="btn('http://adresseip/MU?p=1')">Moteur 1 marche</button>
  </body>
  <script lang="javascript">
    var xhr = new XMLHttpRequest();
    function btn(cmdSend) {
      xhr.open('GET', cmdSend + '&_=' + Math.random());
      xhr.send(null);
    }
  </script>
</html>
```

Maxime : mon premier réflexe était de copier-coller la ligne de code puis de modifier la fonction appelée :

```
<button onclick="btn('http://adresseip/MU?p=1')">Moteur 1 marche</button>
<button onclick="btn('http://adresseip/MD?p=1')">Moteur 1 marche arrière</button>
```



2 Premiers H Bridges (seulement pour deux moteurs)



3 Hbridges intégrés pour tous les moteurs

```
<button onclick="btn('http://adresseip/MS?p=1')">Moteur 1 stop
</button>
```

Puis reproduire cela pour le moteur 2, puis 3, etc. Le tout a parfaitement fonctionné et m'a permis de comprendre comment cela fonctionne.

Cinquième étape : pilotage des moteurs depuis un graphique en SVG

Laurent : les ressources de la board étant limitées, il faut réduire au maximum ce qui est stocké et envoyé à l'utilisateur. Du coup plutôt que d'opter pour une image, je me suis dit que j'allais utiliser un SVG. Ce format vectoriel même si un peu bavard car basé sur du XML permet quand même de créer des objets assez simplement. J'ai donc utilisé un éditeur de SVG pour dessiner le bras. J'ai beaucoup optimisé le XML généré pour qu'il prenne le moins d'espace minimum. ⁴

La difficulté a surtout été de rajouter du script et que l'objet soit visible depuis une page web. Le principe est le suivant :

```
<svg width="640" height="480" xmlns="http://www.w3.org/2000/svg" xmlns:svg="http://www.w3.org/2000/svg">
<g>
<line y2="99" x2="283" y1="99" x1="195" stroke-width="5" stroke="#000000" id="s18"
onclick="btn('http://ip/MA&quot;p=1')"/>
</g>
<script xlink:href="script.js" xmlns:xlink="http://www.w3.org/1999/xlink"/>
</svg>
```

Le principe est de référencer le script par la balise `xlink:href` sans oublier la référence `xmlns:xlink="http://www.w3.org/1999/xlink"`. Sans cela il n'est pas possible d'appeler le script. Le script doit ensuite être positionné dans une balise `<![CDATA[]]>`. Et de façon à ce que le script soit utilisable pour d'autres parties du code, attention à bien avoir un retour chariot après le dernier crochet ouvrant et avant le premier fermant. Le reste du code javascript pouvant se trouver intégralement sur la même ligne pour gagner de la place.

```
// <![CDATA[
var xhr = new XMLHttpRequest();
function btn(cmdSend) {
  xhr.open('GET', cmdSend + '&_=' + Math.random());
  xhr.send(null);
}
// ]]>
```

Ensuite l'appel se fait simplement depuis la page web à l'aide du tag `object`. Attention à ne pas utiliser le tag `image`, le script du SVG ne fonctionnera pas.

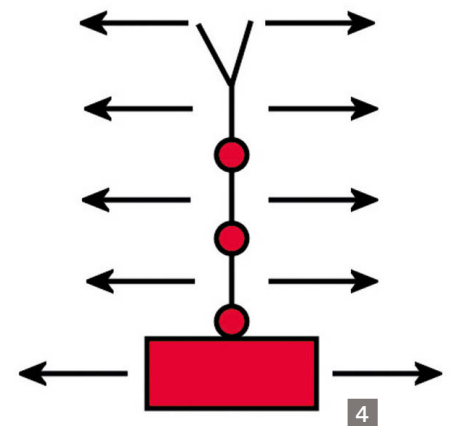
```
<html><head></head><body>Joli bras : <br>
<object data="graph.svg" type="image/svg+xml"></object>
</body></html>
```

Maxime : pour le coup, dans cette partie, je n'ai pas fait grand chose... A part donner quelques conseils visuels. Il m'a ensuite fallu mettre les « onclick » dans tout le SVG en appelant les bonnes fonctions.

Sixième étape : intégration de la page, du SVG et du script dans l'ESP8266

Laurent : la librairie REST ne permet que de retourner un format Json prédéfini. Or l'idée est de retourner des fichiers html, du script, du SVG. J'ai du coup modifié la librairie pour permettre de renvoyer n'importe quel élément de type texte. Je ne rentre pas dans le détail, un pull request a été effectué sur le projet principal et le code est également disponible sur mon propre GitHub. Une fois cette modification effectuée, il était possible de renvoyer du texte sans aucun problème. J'ai donné un exemple de base comme à chaque fois pour permettre à Maxime de l'intégrer dans son propre code.

Maxime : j'ai bien observé comment mon père utilisait les nouvelles fonctions qu'il a implémentées dans aRest dans son exemple puis j'ai essayé d'intégrer les nouvelles fonctionnalités à mon code existant. J'ai eu quelques problèmes à rajouter du code dans le code en html et SVG du fait de la limite de caractères assez faible. J'ai donc dû optimiser mon code et simplifier le nom de certaines variables en quelques lettres seulement.



Septième étape : modes avancés de gestion des moteurs

Laurent : une fois lancé, Maxime a intégré durant les différentes étapes plusieurs éléments comme un timer permettant de piloter les moteurs pendant un temps donné. Il a aussi implémenté une fonction permettant de ne pas arriver en butée avec les moteurs et de les arrêter avant. Il a également ajouté des fonctions avancées permettant de faire fonctionner plusieurs moteurs en même temps afin de déplacer des objets horizontalement ou verticalement.

Maxime : pour un meilleur contrôle des moteurs j'ai rajouté quelques fonctionnalités. J'ai tout d'abord voulu piloter les moteurs pendant une durée spécifique, pour cela, j'ai téléchargé la librairie « Timer-2.1 ». Elle offre plusieurs fonctionnalités avec des timers comme « every » qui appelle une fonction à intervalles de temps réguliers ou « after » qui appelle une fonction après un laps de temps. Après avoir bien examiné toutes les possibilités offertes par la librairie, j'ai finalement choisi d'utiliser « after ».

Voici le principe : j'appelle la fonction MU avec comme paramètre 1, ce qui fait tourner le moteur 1 et dès lors qu'il tourne, je déclenche un timer qui après 1 seconde va appeler la fonction MS toujours avec le paramètre 1 ce qui arrête uniquement le moteur 1 (sans paramètre, la fonction MS arrête tous les moteurs).

```
#include <Timer.h> // déclaration de la librairie
Timer tim; // déclaration du timer, variable « tim »
void loop() {
  tim.update();
```

```
// reste du code pour le serveur web, etc.
}
int MUp(String command) {
  int state = command.toInt();

  if (state == 1) {
    digitalWrite(D0, HIGH);
    digitalWrite(D1, LOW);
    tim.after(1000, StopMotor, (void*)state);
  }
}
// etc
}
int MStop(String command) {
  int state = command.toInt();
  if (state == 1) {
    digitalWrite(D0, LOW);
    digitalWrite(D1, LOW);
  }
  // etc
}
```

J'ai refait les mêmes choses pour chaque moteur puis j'ai voulu aller plus loin dans ma démarche de simplification d'utilisation. Je voulais éviter que lorsqu'un moteur arrive en bout de course, il continue à tourner et risque de se bloquer ou d'être endommagé. J'ai donc pensé à faire une variable qui est incrémentée ou décré- mentée à chaque fois que le moteur tourne et lorsque la variable dépasse une certaine valeur, elle appelle la fonction MS pour arrê- ter le moteur. Cette fois ci, on utilisera « every »

Voici mon code pour le moteur 1 :

```
#include <Timer.h> // declaration de la librairie
Timer M1; // declaration du timer, variable « M1 »
int tickEvent1;
int mo1 = 0; // la variable qui va être incrémentée tous les 10 ms

void loop() {
  M1.update();
  // reste du code pour le serveur web, les autres moteurs, etc.
}
int MUp(String command) {
  int state = command.toInt();

  if (state == 1) {
    if (mo1 < 7100) {
      M1.stop(tickEvent1); //cette ligne est uniquement ici pour déboguer
      tickEvent1 = M1.every(10, m, (void*)1); //appelle la fonction m toute les
      10ms avec le paramètre 1
      digitalWrite(D0, HIGH);
      digitalWrite(D1, LOW);
    }
  }
  // etc
}
```

```
void m(void* motor) { int state = (int)motor;

  if (state == 1) {
    Serial.println(mo1); // me permet de garder un œil sur les variables depuis le
    moniteur série Arduino
    mo1 += 10; // incrémente la variable de 10
    if (mo1 > 7000) { // 7000 est le temps que j'ai choisi comme limite
      MStop(String(1)); // arrête le moteur 1
    }
  }
  // etc.
}
```

Dans la fonction MStop, il suffit de rajouter « M1.stop(tickEvent1); » qui arrête « tickEvent1 ».

Aperçu depuis un téléphone de la page web : 5

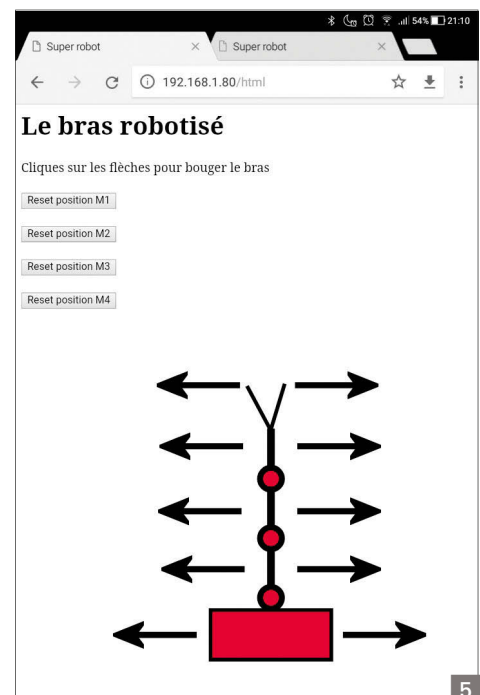
Conclusion

Laurent : ce projet était vraiment très sympa à réaliser et très amu- sant. J'en ai profité pour découvrir le fonctionnement de SVG. Pour moi, c'est un super projet parent-enfant qui nécessite un accompa- gnement minimum mais qui laisse beaucoup d'autonomie à l'enfant. Maxime m'a même demandé de pouvoir piloter son bras robotisé depuis Internet, ce que j'ai paramétré sur le routeur principal.

Maxime : j'ai adoré travailler sur ce projet, c'était pour moi très complet, il m'a fait travailler tant le coté hardware que le coté soft- ware. Le résultat est très satisfaisant. Peut être pour le futur rajouter une webcam pour piloter le Bras robotique à distance et en avoir l'aperçu. J'ai aimé le fait de partir de zéro et de rajouter des fonc- tionnalités au fur et à mesure du temps, partir d'un code assez simple faisant tourner un moteur à partir d'une url et arriver à faire bouger 5 moteurs en simultanée depuis une page web !

Les sources sont disponibles sur mon GitHub :

<https://github.com/Maximellerbach/Robotic-arm>





Nicolas Lethiec,
étudiant à l'ESME Sudria

Défi robotique des écoles pour les élèves de CM2

Le 24 mai dernier s'est déroulé le DREC, ou Défi de Robotique des Ecoles de la Circonscription. Ce projet est une coupe de robotique destinée aux élèves de CM2. Autour de ce projet étaient réunies la circonscription, l'association Sudrianoïde, l'école d'ingénieurs ESME Sudria de Lille, ainsi que l'école e-artsup de Lille. Ce projet est né en 2017, lorsque la circonscription de Roubaix-Hem a acheté des robots LEGO à différentes écoles. Cependant les enseignants ne sachant pas les utiliser, elle demanda la participation de l'ESME Sudria, et ainsi fut réalisé le premier DREC.

Suite à la réussite de cette première édition, 3 élèves de l'ESME Sudria décident de créer l'association de robotique Sudrianoïde, qui, cette année, a participé à l'organisation du DREC. Le DREC a pour objectif de faire découvrir la robotique à des enfants qui n'ont pas forcément la possibilité d'avoir accès à ce genre de technologies, tout en regroupant plusieurs matières qui leur sont enseignées telles que les mathématiques, le français et les arts plastiques. En partageant notre passion pour la robotique, l'objectif est de faire naître cette même passion, et peut-être même une vocation pour la robotique.

Lors de cette coupe nous apprenons aux enfants à se servir du robot LEGO Mindstorms NXT. Cette année, huit classes de CM2 y ont participé, soit près de 250 élèves. Le DREC se déroule sur 1 an de la manière suivante : lors des premiers mois nous organisons une rencontre entre les professeurs et les étudiants de l'ESME Sudria afin de former ceux-ci à l'utilisation du logiciel Lego Mindstorms. A partir de début février, les membres de Sudrianoïde vont une fois par semaine rendre visite à deux écoles participantes afin d'expliquer aux élèves ce qu'ils doivent réaliser pour la coupe, leur expliquer comment fonctionne le logiciel, et les aider à réaliser leurs programmes. Ces rencontres ont pour objectif de leur fournir des pistes de réflexion, de les faire réfléchir afin qu'ils trouvent la solution par eux-mêmes : nous leur mettons à disposition des outils, les aiguillons en leur posant des questions, mais nous évitons au maximum de leur donner la solution. Avec ces interventions nous leurs faisons découvrir comment fonctionne un robot à partir d'un logiciel très visuel et leur expliquons le principe des capteurs, ainsi que leur fonctionnement.

Pour cette coupe, nous leurs demandons de réaliser 3 programmes différents, chacun de difficultés différentes. Le premier est un programme assez simple qui consiste simplement à faire avancer, tourner et attendre le robot : ce programme nous permet vraiment de leur faire comprendre les fonctions de base. Le second programme est un programme qui met en place le concept de condition, c'est-à-dire le robot doit avancer jusqu'à une ligne noire et s'arrêter quand il la détecte avec un capteur infrarouge. Le dernier programme met en place le système de boucle et de choix : le robot doit suivre une ligne noire à l'aide d'un capteur infrarouge ; quand celui-ci détecte une ligne noire, il doit tourner vers la gauche et quand il détecte du blanc il doit tourner vers la droite, cela lui permettant de suivre une ligne noire de toute forme.

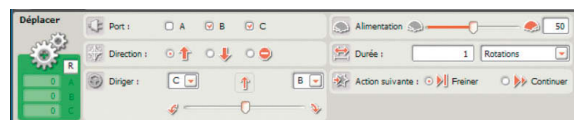
Le jour de la coupe, les élèves doivent présenter leurs travaux devant les autres écoles et un jury. Lors de cette présentation, ils expliquent comment s'est déroulé leur apprentissage, les difficultés rencontrées, ainsi que l'histoire qu'ils vont raconter à travers le parcours de leurs robots. Cela leur fournit également une opportunité de développer leur capacité à faire une présentation orale devant une assistance. Pendant qu'une partie des élèves passe devant le jury, l'autre moitié a accès à différents ateliers tenus par des élèves de l'ESME Sudria. Ils peuvent ainsi découvrir le fonctionnement d'une imprimante 3D, de cartes Arduino, de bras robotisés, ou encore le fonctionnement d'un Segway.

Cette journée a pour objectif de clôturer plusieurs mois de travail de manière agréable pour les enfants, en leur faisant acquérir de nouvelles connaissances de manière ludique et éducative, récompensant le travail fourni au cours des derniers mois. A l'issue de la journée, chaque classe repart avec un robot articulé imprimé en 3D. Le premier prix remporte un robot plus grand. De plus, lors de leur présentation, chaque classe peut valider jusqu'à 8 compétences. A chaque compétence acquise, les enfants reçoivent un badge. Pour nous, membres de l'association Sudrianoïde, cette deuxième édition est une réussite. Les enfants ont montré un grand enthousiasme par rapport à cette journée et au parcours, et pour nous ce fut extrêmement plaisant et enrichissant de former et partager notre passion avec les enfants.

Voici deux petits tutoriels qui expliquent comment réaliser deux programmes du DREC.

1er Tutoriel : Se déplacer.

Après avoir créé un nouveau fichier vous arriverez sur une interface avec sur la droite une colonne d'icônes qui seront vos principaux outils durant ces deux tutoriels. Dans un premier temps nous allons nous intéresser au premier bloc de la colonne, le bloc « déplacer ». Après avoir fait glisser ce bloc sur la partie centrale de votre écran vous pourrez observer en bas de votre écran un encadré comme ci-dessous :



Dans cet encadré vous trouverez plusieurs onglets. Le premier étant l'onglet « Port » qui vous montre sur quel moteur vous allez agir, ici

Abonnez-vous à **Programmez!** Abonnez-vous à **Programmez!** Abonnez-vous à **Programmez!**

Programmez!
Le magazine des développeurs

Nos classiques

1 an
11 numéros
49€*

2 ans
22 numéros
79€*

Etudiant
1 an - 11 numéros
39€*

* Tarifs France métropolitaine



Abonnement numérique

PDF **35€**

1 an - 11 numéros

Souscription uniquement sur www.programmez.com

Option : accès aux archives **10€**

Offres printemps 2018

1 an **59€**

11 numéros
+ 1 vidéo ENI au choix :

- **Symfony 3***
Développer des applications web robustes (valeur : 29,99 €)
- **Raspberry Pi***
Apprenez à réaliser et piloter une lumière d'ambiance (valeur : 29,99 €)

2 ans **89€**

22 numéros
+ 1 vidéo ENI au choix :

- **Symfony 3***
Développer des applications web robustes (valeur : 29,99 €)
- **Raspberry Pi***
Apprenez à réaliser et piloter une lumière d'ambiance (valeur : 29,99 €)

* Offre limitée à la France métropolitaine

Toutes nos offres sur www.programmez.com



Oui, je m'abonne

ABONNEMENT à retourner avec votre règlement à :
Service Abonnements PROGRAMMEZ, 4 Rue de Mouchy, 60438 Noailles Cedex.

- ☐ **Abonnement 1 an : 49 €**
- ☐ **Abonnement 2 ans : 79 €**
- ☐ **Abonnement 1 an Etudiant : 39 €**
Photocopie de la carte d'étudiant à joindre

- ☐ **Abonnement 1 an : 59 €**
11 numéros + 1 vidéo ENI au choix :
- ☐ **Abonnement 2 ans : 89 €**
22 numéros + 1 vidéo ENI au choix :

- ☐ Vidéo : Symfony 3
- ☐ Vidéo : Raspberry Pi

☐ Mme ☐ M. Entreprise : _____ Fonction : _____

Prénom : _____ Nom : _____

Adresse : _____

Code postal : _____ Ville : _____

email indispensable pour l'envoi d'informations relatives à votre abonnement

E-mail : _____ @ _____

☐ Je joins mon règlement par chèque à l'ordre de Programmez !

☐ Je souhaite régler à réception de facture

* Tarifs France métropolitaine

PROG 220
Offre limitée, valable jusqu'au 30 août 2018

Abonnez-vous à **Programmez!** Abonnez-vous à **Programmez!** Abonnez-vous à **Programmez!**

Offres 20^e anniversaire !*

1 an - 11 numéros

79,99 €

(au lieu de 147,99 €)

2 ans - 22 numéros

99,99 €

(au lieu de 177,99 €)

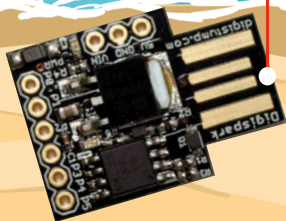


+
1 clé USB contenant
tous les numéros depuis le n°100

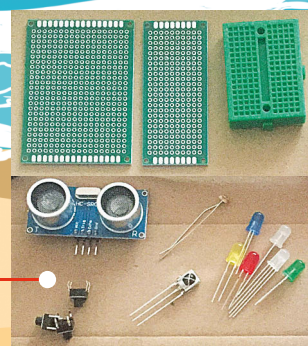


+
4 numéros vintage
(selon les stocks)

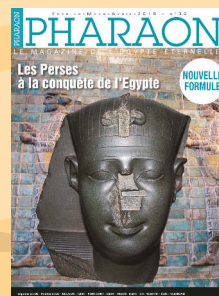
+
1 carte maker Digispark ATtiny84
compatible Arduino



+
1 lot de composants / capteurs
(selon arrivage du jour)



+
1 an de Pharaon Magazine soit 4 numéros :
pour découvrir l'Egypte des Pharaons !



* Offre réservée à la France métropolitaine

Sur notre site web uniquement : <https://www.programmez.com/catalog/20eme-anniversaire>

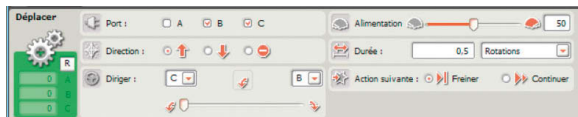
il s'agit des moteurs B et C. Le deuxième est l'onglet « Direction », vous indiquant si vos moteurs vont avancer, reculer ou s'arrêter. Le troisième est l'onglet « Diriger » qui permet de répartir l'alimentation entre 2 moteurs. L'onglet « Alimentation » indique la puissance que la brique fournit au moteur. Enfin l'onglet « Durée » qui indique combien de rotations vont effectuer les moteurs.

Nous allons donc réaliser un premier programme pour que le robot réalise un parcours simple. Ce parcours se déroule ainsi : le robot avancera, tournera à gauche, avancera, tournera à droite, avancera, puis fera un demi-tour, et avancera une dernière fois. Pour cela vous aurez besoin de 4 configurations du bloc « Déplacer ».

1er Bloc : Avancer



2ème Bloc : Virage à gauche



3ème Bloc : Virage à droite



4ème Bloc : Demi-tour



Maintenant que vous avez vos quatre configurations de bloc vous pouvez les placer afin de réaliser le parcours, ce qui donnera : 1er Bloc, 2ème Bloc, 1er Bloc, 3ème Bloc, 1er Bloc, 4ème Bloc, 1er Bloc

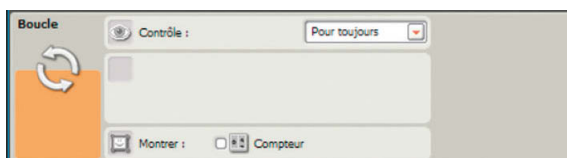
Vous obtiendrez ainsi ce cheminement-ci sur votre écran :



2ème Tutoriel : suivre une ligne noire

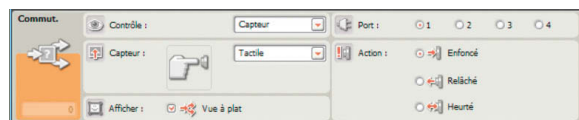
Dans ce tutoriel nous introduirons deux nouveaux blocs, le bloc Boucle et le bloc Commut.

Le bloc Boucle est l'avant dernier bloc de la colonne. De la même façon que pour le bloc Déplacer une interface va apparaître en bas de votre écran.

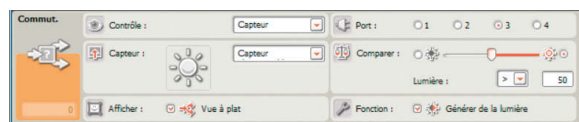


En cliquant sur le menu déroulant vous pourrez voir plusieurs modes tels pour toujours, la boucle se répètera pour toujours, le mode capteur, le mode Temps, le mode compteur, et le mode logique. Nous nous intéresserons ici au mode pour toujours.

Le bloc Commut est le dernier bloc de la colonne. Ce bloc est un bloc de choix, vous réaliserez deux parcours différents au sein de ce bloc et selon la condition remplie un des deux parcours s'exécutera.



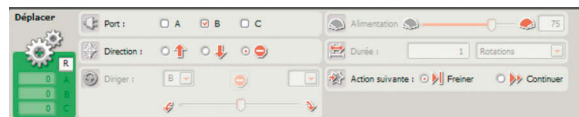
Le bloc Commut possède plusieurs configurations. Pour cet exercice nous serons intéressés par un bloc qui dépend du capteur photosensible ; pour cela il faudra le sélectionner dans le deuxième menu déroulant.



Pour que notre robot suive une ligne noire nous souhaitons qu'il exécute un choix à l'infini, ce choix étant que si le robot capte du noir il tourne vers la gauche et s'il capte du blanc il tourne vers la droite. Pour cela vous devrez commencer par placer une boucle comme celle ci-dessus. Vous devrez ensuite placer le bloc commut configuré comme ci-dessus. Vous aurez ensuite besoin de 2 blocs Déplacer par ligne. Pour la ligne du haut vous aurez besoin d'un bloc réglé uniquement sur le moteur C et dans l'onglet durée cliquer sur le menu déroulant pour sélectionner le mode Illimité.

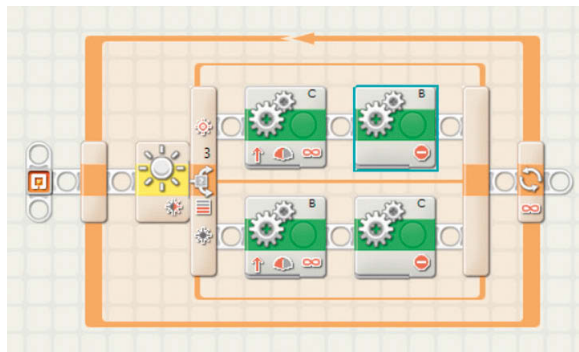


Et un second bloc bloquant le moteur B, pour cela il suffit de sélectionner le 3ème bouton dans l'onglet direction.



Pour la ligne du bas il suffit de réaliser les mêmes blocs que ci-dessus mais en remplaçant C par B et B par C.

Vous obtiendrez donc un programme tel que ci-dessous.





Aurélie Vache

Cloud Dev(Ops) chez Continental

Duchess France & DevFest Toulouse Leader

Toulouse Data Science core-team & Mairaine Elles bougent

@aurelievache

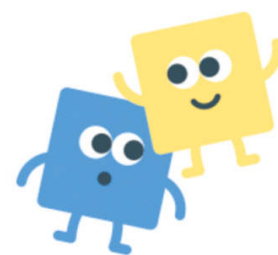


STEAM

Kids

SteamDecks : une plateforme pour apprendre en s'amusant

Depuis deux ans, Duchess France participe à un projet éducatif européen, en collaboration avec des professeurs d'écoles d'Espagne, d'Allemagne et de Roumanie, en tant que partenaire, sur un projet nommé STEAM Decks.



Tout a commencé il y a deux ans, quand l'Université de Deusto, située à Bilbao au pays basque Espagnol, fondée en 1886, nous a contactés pour rejoindre, en tant que partenaire européen dans le programme Erasmus+, un projet éducatif européen qui s'articulerait autour des STEAM. Le projet avec Erasmus+ a débuté le 01/09/2016, se termine le 31/08/2018 et rassemble :

- l'université de Deusto à Bilbao, en Espagne ;
- la fondation éducative Fatima à Bilbao, en Espagne ;
- l'école Bertha-Von-Suttner-Gymnasium Babelsberg à Potsdam, en Allemagne ;
- l'école Ferdinand I à Bucarest, en Roumanie ;
- l'association Duchess France, en France.

STEM, STEAM, c'est quoi ?



Commençons par le commencement. **STEM** est un acronyme qui désigne 4 disciplines : **science**, **technology**, **engineering**, et **mathematics**. Le A fut rajouté plus tard et il désigne les **Arts**. **STEAM** est donc un acronyme qui intègre la science, la technologie, l'ingénierie, les arts créatifs et les mathématiques. Il s'agit d'un mouvement soutenu par la Rhode Island School of Design (RISD) et largement adopté par les institutions, les entreprises et les particuliers. L'objectif de ce mouvement est, entre autres, d'encourager l'intégration de ces 5 disciplines dans l'éducation.

Deck ?

Un deck désigne un ensemble de cartes sur un thème particulier. Les jeux de cartes Pokemon ou Magic The Gathering comportent des decks et chaque joueur combat un autre joueur avec son deck. Le but étant de gagner grâce aux connaissances de son deck et en faisant preuve de stratégie. Les decks ont aussi des caractéristiques/propriétés spécifiques. Par exemple, pour un deck sur les dinosaures, on pourra avoir le poids, la taille, la dangerosité...

STEAM Decks



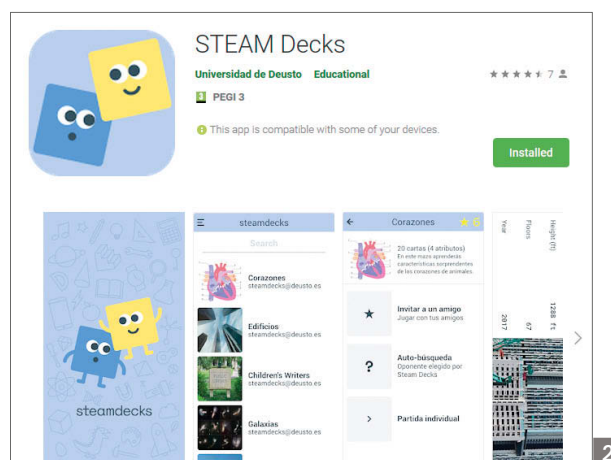
STEAM Decks: "Learning STEAM through a playful online platform, social learning and content cocreation" a pris forme et nous avons le plaisir de pouvoir vous en parler :-).

STEAM Decks est une plateforme de jeux de cartes multijoueur, multilingues, gratuite, en ligne, pour apprendre les STEAM aux enfants, pour les élèves du primaire et du secondaire. Le but de la plateforme est :

- d'apprendre en s'amusant ;
- de stimuler la créativité ;
- d'améliorer les compétences scientifiques, artistiques, mathématiques et numériques.

La plateforme

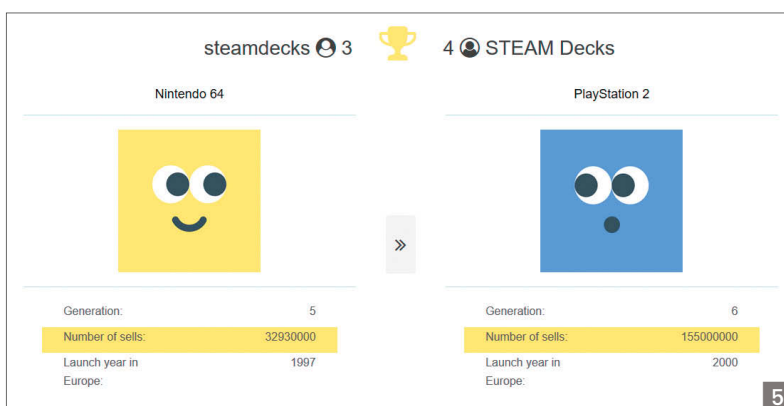
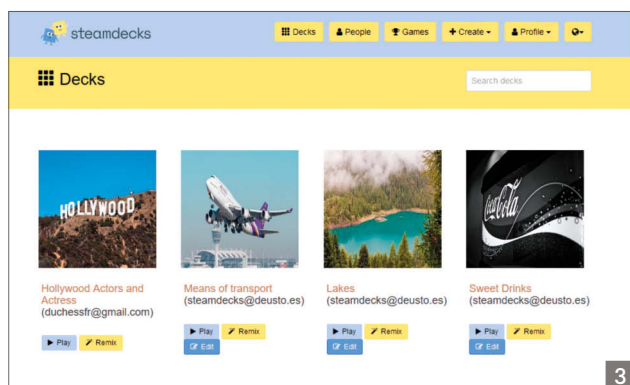
Si j'ai utilisé le mot plateforme, ce n'est pas pour rien ;-). Grâce à la plateforme, un enfant peut donc jouer et apprendre en même temps, mais il peut également créer ses propres decks qui seront joués par les



autres utilisateurs/ joueurs de la plateforme. La plateforme a été développée dans un contexte internationale, de ce fait elle est disponible dans plusieurs langues : Anglais, Espagnol, Français, Allemand, Roumain et Basque. 1

Trêve de blabla, afin de tester et de découvrir la plateforme voici la marche à suivre :

Connectez-vous sur le site <https://steamdecks.deusto.es/> et/ou bien téléchargez l'application STEAM Decks,



uniquement disponible pour Android (<https://play.google.com/store/apps/details?id=es.learninglab.steamdecks>) pour le moment. **2**

Vous devez tout d'abord vous créer un compte. Ensuite dans votre profil, vous pourrez sélectionner votre langue préférée. Une fois ceci fait, vous pouvez aller dans la section Decks afin de pouvoir voir la liste des jeux de cartes (decks) existants, créés par vous ou par les autres joueurs : **3**

- seul,
- contre un autre joueur qui est en ligne,
- jouer un match aléatoire,
- ou bien imprimer le deck pour jouer de manière déconnectée.

Jouons maintenant au deck sur les consoles de jeux vidéo que nous (Duchess France) avons créées. **4**

Sous forme de bataille de cartes, le joueur doit trouver quelle caractéristique est plus haute ou plus basse que celle de la carte de l'autre joueur. Dans mon cas, j'ai choisi de dire qu'il y a eu moins de ventes de Nintendo 64 que de Playstation 2 : **5**

Le bonhomme jaune content indique que j'ai gagné cette bataille, en cliquant sur la flèche de droite on peut passer à la bataille suivante et ainsi de suite, vous avez compris le système: carte après carte, en jouant avec les caractéristiques d'un deck, on

apprend en s'amusant sur de multiples sujets divers et variés : des éléments chimiques, aux galaxies, en passant par les pays européens et les acteurs & actrices hollywoodiennes.

Vous voulez contribuer et créer votre deck ? Pas de soucis, pour cela il vous suffit d'aller sur la partie *Create>Create a new deck*. Vous devez modifier le nom en anglais "deck" par le nom de votre deck, rajouter la traduction du nom de deck dans les langues que vous souhaitez, puis vous devrez ajouter une description, des attributs, ainsi que des cartes.

A noter que pour que le deck soit jouable, il faudra créer au minimum deux cartes dans votre deck.

Il est également possible d'afficher la liste des joueurs existants, de rechercher par nom un joueur, afin de l'ajouter dans votre liste d'amis et de pouvoir jouer contre lui.

Retour d'expérience

La plateforme a été testée dans des écoles en Allemagne et en Roumanie et également par les élèves de l'école Fatima de Bilbao. Les enseignants de ces établissements ont été formés à la plateforme et STEAM Decks.

Nous sommes allés dans l'école Fatima de

Bilbao et avons pu constater que les élèves, durant leurs heures d'écoles, pouvaient jouer à STEAM Decks sur leurs PC portables. Nous avons parlé avec eux et avons pu récolter quelques feedbacks qui étaient très positifs. Outre le fait que STEAM Decks leur permet d'apprendre les connaissances déclaratives, il est plus facile et fun d'apprendre comme en s'amusant, en jouant plutôt que dans un contexte du "par cœur". Nous avons organisé un meetup sur Paris, le 11 Juin dernier, afin de présenter la plateforme. Les précieux feedbacks que nous avons récoltés vont permettre à l'université de Deusto d'améliorer STEAM Decks.

Et maintenant ?

Je vous invite à vous connecter au site et/ou à l'application disponible sur Google Play, de jouer, de tester vos connaissances et de créer un deck si cela vous dit pour vous faire une idée de cette initiative.



Il s'agit d'une première version de la plateforme, STEAM Decks est encore en cours de développement. Grâce à vos feedbacks, la plateforme sera améliorée et des enfants pourront apprendre tout en s'amusant. Mêler l'utile à l'agréable, c'est cool, non ? ;-).

Apprendre à programmer en s'amusant

C'est l'une des grandes nouveautés des programmes du collège; l'algorithme apparaît maintenant dans les compétences de plusieurs disciplines. Il existe de nombreux robots qui permettront de comprendre ce qu'est un algorithme et d'apprendre à programmer. Tout cela à un coût et dépendra surtout de l'âge de l'enfant.



A PARTIR DE 4 ANS : attention aux écrans

A 4 ans il est essentiel d'éloigner les enfants des écrans, mais comment faire pour aborder la programmation ? Deux robots existent et sont adaptés à cet âge, ils sont pré-programmés et l'enfant pourra agir sur son comportement sans utiliser d'ordinateur, de tablette ou de smartphone.

	Le Beebot	L'ozobot
		
Avantages	<ul style="list-style-type: none"> • Programmation facile par simple appui sur les touches directionnelles. • Possibilités de créer des parcours et de programmer le robot ensuite. • L'enfant comprendra rapidement la notion basique d'algorithme. 	<ul style="list-style-type: none"> • Programmation simple, le robot suit une ligne et agit en fonction d'un code couleur au sol (tourner, freiner, accélérer...). • Système innovant ; possibilité de faire des courses sur différents parcours que l'enfant peut lui-même dessiner.
Le prix	90€	60€



A PARTIR DE 6 ANS : vers un usage raisonné et limité des écrans

A 6 ans, l'enfant est capable de créer sur un écran (attention à limiter le temps d'utilisation à une heure par jour) et d'interagir ensuite avec le robot, ainsi il commencera doucement à comprendre la notion d'algorithme.

	Sphero	Le thymio
		
Avantages	<ul style="list-style-type: none"> • Très ludique et technologiquement intéressant. • Il existe plusieurs versions (dont plusieurs Star Wars) mais cela fait fortement varier le prix. • Programmation avec Sphero SPRK : très graphique et simple d'utilisation cette application pour smartphone permet de créer un algorithme en ligne de code par déplacements de blocs. 	<ul style="list-style-type: none"> • Modes pré-programmés (suiveur de ligne, suiveur, peureux...). Ces modes ne demandent pas d'interaction avec les écrans et l'enfant pourra jouer, identifier et comprendre le fonctionnement du robot. • Le robot reconnaît les claquements de mains, le contact de la main, les obstacles, le vide. • Première programmation avec Aseba : interface graphique simple et intuitive, le programmeur en herbe pourra ensuite téléverser son programme dans le Thymio qui l'exécutera.
Le prix	60€ dans sa version basique	130€

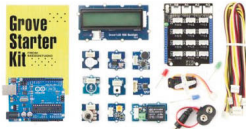

A PARTIR DE 9 ANS : on complexifie

En plus des robots cités précédemment, à cet âge, les enfants sont capables de prendre en compte les informations des capteurs et de modifier le comportement du robot en conséquence.

	Mbot	Lego Mindstorm EV3
		
Avantages	<ul style="list-style-type: none"> • Evolutifs : des modules (pince, écran) peuvent être ajoutés. • Programmable depuis le logiciel Mblock : des bibliothèques ont été spécialement conçues pour le Mbot. • Dans sa version basique, il est capable de mesurer une distance, suivre une ligne, détecter du vide, etc. 	<ul style="list-style-type: none"> • Assez coûteux les robots Mindstorm offre l'avantage ou l'inconvénient (comme tous les packs Lego) d'occuper les enfants avec le montage intégral. • La programmation reste simple, les applications sont très graphiques et bien conçues.
Le prix	90€	+ 300€

A PARTIR DE 12 ANS : vive le modulaire

A cet âge, il est essentiel que les systèmes soient modulaires et que les enfants puissent les modifier à leur guise, il est donc temps de passer aux cartes Arduino.

	Le système Grove	La carte Arduino UNO
		
Avantages	<ul style="list-style-type: none"> • Elle est conseillée pour les programmeurs débutants et a l'avantage de simplifier les branchements. • Des bibliothèques très complètes et spécifiques à sa programmation existent sur Mblock et Ardublock. 	<ul style="list-style-type: none"> • Elle est conseillée pour les programmeurs débutants étant à l'aise avec les notions relatives aux branchements de composants électroniques (VCC, GND, OUTPUT etc.), elle peut se programmer avec différents logiciels (Mblock, Ardublock...). • Il existe des clones disponibles pour une poignée d'euros sur internet, elles fonctionnent tout aussi bien que les vraies.
Le prix	Pack Grove sans la carte arduino : 50€	Carte Arduino : 20€ Carte clone : 3€ Pack clone (carte + câbles + modules) : à partir de 20€

Pas de budget ? Pas de souci. La compréhension de la programmation peut très bien se faire sans matériel, avec le logiciel Scratch par exemple. Il repose sur la programmation par Blocs et l'enfant pourra créer des petits jeux ou histoires. Code.org permet également d'apprendre le codage progressivement tout en s'amusant.



Christophe PICAUD

Consultant sur les technologies Microsoft

christophepichaud@hotmail.com | www.windowsepp.com

Créer un service Windows

Partie 1

LE SQUELETTE TECHNIQUE

Sous Windows, nous développons des applications EXE ou des modules partagés DLL. Il existe une variante ou classe d'application que l'on nomme service Windows qui s'exécute en arrière-plan même si personne n'a ouvert de session sur la machine. Il peut être démarré en mode automatique, au boot de la machine, en mode manuel ou hors de service. Il existe deux types de services : des drivers et des services qui ne sont pas des drivers. C'est le type non-drivers que nous allons étudier ici.

Le SCM : Service Control Manager

Les services sont gérés dans une base de données (la base de registres) et c'est le SCM, démarré au démarrage de la machine, qui se charge des fonctionnalités suivantes :

- Gestion de la base des services ;
- Démarrage des services au démarrage du système ou à la demande ;
- Gestion des informations de statut de tous les types de services ;
- Verrouillage de la base des services.

La base de données du SCM est dans la base de registres sous : HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services. Malgré le fait que la base de registres puisse être modifiée à la main, nous allons passer par les API dédiées aux services Windows. Pour créer un service, il faut utiliser la fonction Win32 CreateService :

```
SC_HANDLE WINAPI CreateService(
    _In_      SC_HANDLE hSCManager,
    _In_      LPCTSTR lpServiceName,
    _In_opt_  LPCTSTR lpDisplayName,
    _In_      DWORD dwDesiredAccess,
    _In_      DWORD dwServiceType,
    _In_      DWORD dwStartType,
    _In_      DWORD dwErrorControl,
    _In_opt_  LPCTSTR lpBinaryPathName,
    _In_opt_  LPCTSTR lpLoadOrderGroup,
    _Out_opt_ LPDWORD lpdwTagId,
    _In_opt_  LPCTSTR lpDependencies,
    _In_opt_  LPCTSTR lpServiceStartName,
    _In_opt_  LPCTSTR lpPassword
);
```

Vous remarquerez que le premier paramètre est un handle sur le SCM. Pourquoi ? En fait, il est nécessaire d'ouvrir le SCM avant d'utiliser cette fonction. Voici le code de création d'un service :

```
bool CServiceModule::Install()
{
    if (IsInstalled())
    {
        std::wcout << _T("Service is already installed") << std::endl;
        return TRUE;
    }

    SC_HANDLE hSCM = ::OpenSCManager(NULL, NULL, SC_MANAGER_ALL_ACCESS);
    if (hSCM == NULL)
    {
        std::wcout << _T("Couldn't open service manager") << std::endl;
    }
}
```

```
return FALSE;
}

// Get the executable file path
TCHAR szFilePath[_MAX_PATH];
::GetModuleFileName(NULL, szFilePath, _MAX_PATH);

SC_HANDLE hService = ::CreateService(
    hSCM,
    SERVICE_NAME,
    SERVICE_NAME,
    SERVICE_ALL_ACCESS, SERVICE_WIN32_OWN_PROCESS,
    SERVICE_DEMAND_START, SERVICE_ERROR_NORMAL,
    szFilePath, NULL, NULL, NULL, NULL, NULL);

if (hService == NULL)
{
    ::CloseServiceHandle(hSCM);
    std::wcout << _T("Couldn't create service ") << SERVICE_NAME << std::endl;
    return FALSE;
}

::CloseServiceHandle(hService);
::CloseServiceHandle(hSCM);
return TRUE;
}
```

La fonction possède quelques subtilités que je n'expliquerai pas ici mais cherchez dans le MSDN Library si vous êtes curieux. Il est ainsi possible de créer un service qui dépend d'un autre service, etc. La routine IsInstalled() est appelée pour que le code soit propre :

```
bool CServiceModule::IsInstalled()
{
    bool bResult = false;

    SC_HANDLE hSCM = ::OpenSCManager(NULL, NULL, SC_MANAGER_ALL_ACCESS);

    if (hSCM != NULL)
    {
        SC_HANDLE hService = ::OpenService(hSCM,
            SERVICE_NAME,
            SERVICE_QUERY_CONFIG);

        if (hService != NULL)
        {
            bResult = true;
        }
    }
}
```

```

{
    bResult = true;
    ::CloseServiceHandle(hService);
}
::CloseServiceHandle(hSCM);
}
return bResult;
}

```

Pour que ce code soit exécuté correctement, il faut être en mode administrateur. Pour gérer les services Windows, il existe un outil sc.exe. Il est aussi possible de créer, détruire, lister des services juste avec cet outil sc. Voyons la base de registres où le SCM gère les services et cherchons la clé « MyService » : **1**

Maintenant que le service est créé, il ne nous reste plus qu'à lui envoyer une commande de type :

- net start MyService
- net stop MyService

Vous allez me dire, on fait start de quoi ? Oui c'est vrai, il nous faut le code qui gère cette commande. Le SCM va nous transmettre les ordres de START ou de STOP (il existe aussi PAUSE). Il faut donc s'enregistrer auprès du SCM pour gérer ces événements. Dans le wmain(), on appelle une méthode Start() :

```

void CServiceModule::Start()
{
    SERVICE_TABLE_ENTRY st[] =
    {
        { SERVICE_NAME, _ServiceMain },
        { NULL, NULL }
    };

    if (m_bService && !::StartServiceCtrlDispatcher(st))
    {
        m_bService = FALSE;
    }
    if (m_bService == FALSE)
    {
        Run();
    }
}

```

La fonction Win32 StartServiceCtrlDispatcher() permet de se connecter au SCM en lui fournissant le point d'entrée du service ; ici : _ServiceMain() :

```

void CServiceModule::_ServiceMain(DWORD dwArgc, LPTSTR * lpszArgv)
{
    _Module.ServiceMain(dwArgc, lpszArgv);
}

```

Nous donnons la main à l'instance du module qui est une variable globale.

```

void CServiceModule::ServiceMain(DWORD dwArgc, LPTSTR * lpszArgv)
{
    m_status.dwCurrentState = SERVICE_START_PENDING;
    m_hServiceStatus = RegisterServiceCtrlHandler(SERVICE_NAME, _Handler);
    if (m_hServiceStatus == NULL)

```

```

{
    //LogEvent(_T("Handler not installed"));
    return;
}
SetServiceStatus(SERVICE_START_PENDING);

m_status.dwWin32ExitCode = S_OK;
m_status.dwCheckPoint = 0;
m_status.dwWaitHint = 0;

// When the Run function returns, the service has stopped.
Run();

SetServiceStatus(SERVICE_STOPPED);
//LogEvent(_T("Service stopped"));
//_Trace.LogDebug("CServiceModule::ServiceMain", "Service stopped");
}

```

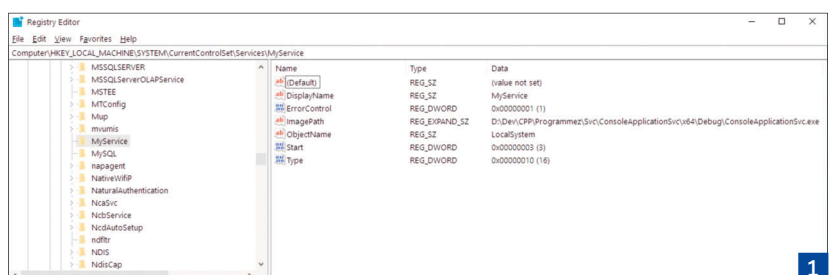
Ici, nous faisons une autre liaison avec le SCM en utilisant la fonction RegisterServiceCtrlHandler et nous appelons Run() :

```

void WINAPI CServiceModule::_Handler(DWORD dwOpcode)
{
    _Module.Handler(dwOpcode);
}

void CServiceModule::Handler(DWORD dwOpcode)
{
    switch (dwOpcode)
    {
        case SERVICE_CONTROL_STOP:
            m_bStop = TRUE;
            SetServiceStatus(SERVICE_PAUSED);
            break;
        case SERVICE_CONTROL_PAUSE:
            break;
        case SERVICE_CONTROL_CONTINUE:
            break;
        case SERVICE_CONTROL_INTERROGATE:
            break;
        case SERVICE_CONTROL_SHUTDOWN:
            break;
        default:
            //LogEvent(_T("Bad service request"));
            break;
    }
}

```



Ici, on gère les états du service. Il faut absolument gérer le stop. Voyons le corps de méthode Run() dont nous avons parlé :

```
void CServiceModule::Run()
{
    _Module.dwThreadId = GetCurrentThreadId();

    //LogEvent(_T("Service started"));
    //_Trace.LogError("CServiceModule::Run", "Service started");

    if (m_bService)
        SetServiceStatus(SERVICE_RUNNING);

    if (m_bService)
    {
        //CString str = "AfxBeginThread";

        m_pThread = AfxBeginThread(AutomateThread, 0, THREAD_PRIORITY_
NORMAL, CREATE_SUSPENDED, NULL);
        if (m_pThread == NULL)
        {
            //LogEvent("Creation du thread AutomateThread impossible");
            SetServiceStatus(SERVICE_STOPPED);
            goto quit_now;
            return;
        }
        m_pThread->ResumeThread();
    }
    else
    {
        //AfxMessageBox("Utilisez NET START pour lancer le service.");
        //PostThreadMessage(_Module.dwThreadId, WM_QUIT, 0, 0);
    }

    MSG msg;
    while (GetMessage(&msg, 0, 0, 0))
        DispatchMessage(&msg);

quit_now:
    return;
}
```

Cette méthode lance un thread : c'est le corps principal du service : un thread en arrière-plan. Dans ce thread, on y ajoute ce que l'on veut. Un serveur TCP/IP, une fabrique de composants COM, etc. On gère juste un test sur la variable _Module.m_bStop pour savoir si on doit terminer et alors on positionne _Module.SetServiceStatus(SERVICE_STOP_PENDING) pour dire que le service s'arrête. On remarquera que les états du service sont explicitement déclarés dans le code via la fonction Win32 SetServiceStatus :

```
void CServiceModule::SetServiceStatus(DWORD dwState)
{
    m_status.dwCurrentState = dwState;
    ::SetServiceStatus(m_hServiceStatus, &m_status);
}
```

Dans notre exemple, le code est tiré d'un service qui fonctionne, c'est un ensemble de composants COM qui sont accessibles au

sein de service Windows donc le thread ne fait rien si ce n'est d'attendre la notification du stop de service via une commande : net stop MyService. Pour être complet, affichons le code complet de la classe (le header) :

```
#pragma once

#define SERVICE_NAME_T("MyService")

class CServiceModule
{
public:
    CServiceModule();
    ~CServiceModule();

public:
    bool IsInstalled();
    bool Install();
    void Start();

public:
    void ServiceMain(DWORD dwArgc, LPTSTR * lpszArgv);
    void SetServiceStatus(DWORD dwState);
    void Handler(DWORD dwOpCode);
    void Run();

public:
    static void WINAPI _Handler(DWORD dwOpCode);
    static void WINAPI _ServiceMain(DWORD dwArgc, LPTSTR * lpszArgv);

public:
    SERVICE_STATUS_HANDLE m_hServiceStatus;
    SERVICE_STATUS m_status;
    bool m_bService;
    bool m_bStop;
    DWORD dwThreadId;
    CWinThread * m_pThread;
};

extern CServiceModule _Module;
UINT AutomateThread(LPVOID pParam);
```

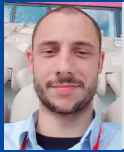
On notera que le code est assez technique. D'un autre côté, une fois qu'on a fait un service, c'est toujours la même chose donc on peut réutiliser les classes et méthodes. Le nom du service est une constante.

Le logger système

Historiquement depuis Windows NT 3.51, il y a un service log des informations dans le journal des événements. Vous pouvez remarquer que dans le code, il y a des _Module.LogEvent() en commentaires...

Conclusion

Construire un service nécessite un code technique, il faut l'avouer. Ce code est réutilisable donc le faire une fois permettra de le réutiliser. Dans cette partie on a vu comment faire la liaison avec le SCM et comment développer les points d'entrées.



Sylvain SAUREL
Développeur Java / Android
<https://www.ssaurel.com>
sylvain.ssaurel@gmail.com

Création d'une application Android ToDo List avec Kotlin

Langage de référence pour le développement d'applications Android depuis le lancement de la plateforme par Google, Java présente certains inconvénients qui nuisent à la productivité des développeurs au quotidien. Afin de remédier à cette problématique, Google a décidé de promouvoir Kotlin en tant que langage de développement officiel pour Android lors de la Google I/O 2017. Dans cet article, nous vous proposons de découvrir Kotlin en réalisant une application ToDo List minimaliste.

Créé par JetBrains, Kotlin est un langage de programmation orienté objet et fonctionnel avec typage statique pouvant être compilé pour la machine virtuelle Java mais également en JavaScript. En tant que développeur Android ou Java, vous avez sûrement déjà entendu parler de JetBrains puisqu'il s'agit de la société derrière l'éditeur de code IntelliJ IDEA qui a su s'imposer comme un concurrent de taille d'Eclipse pour le développement d'applications Java.

La qualité et la popularité d'IntelliJ IDEA ont ensuite convaincu Google de s'appuyer sur cet environnement pour développer Android Studio, l'IDE de référence du monde Android depuis fin 2014. Le choix du géant de Mountain View aura permis aux équipes de JetBrains de rendre de plus en plus populaire leur plugin spécifique permettant de réaliser des applications Android en Kotlin. La montée en puissance de Kotlin auprès des développeurs combinée au lancement de Swift par Apple pour le développement d'applications iOS aura finalement convaincu Google de choisir Kotlin comme le second langage officiel pour la réalisation d'applications Android.

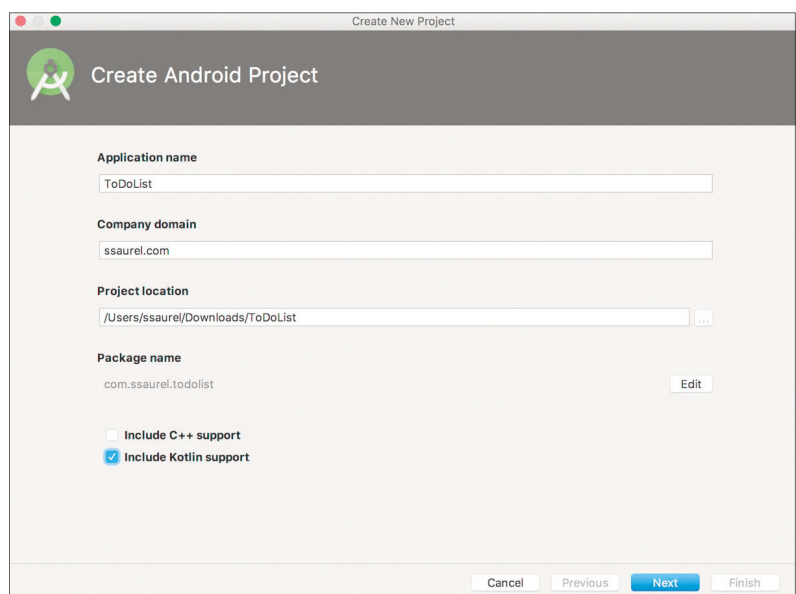
Pourquoi Kotlin ?

Les partisans de langages de scripts plus anciens comme Groovy ou Scala, basés sur la JVM, peuvent se demander pourquoi Kotlin aura finalement été choisi. La pleine intégration de Kotlin au sein d'Android Studio explique en grande partie ce choix. Pour en revenir au choix de supporter un langage comme Kotlin, il faut bien avouer qu'il s'agit d'un langage fun, expressif et moderne. Tout ce que Java n'était plus depuis longtemps.

Comparé à Java, Kotlin est bien plus concis avec une réduction drastique du code boilerplate qui nuit à la lisibilité de bon nombre de classes Java. Il offre une sécurité bien plus forte contre les `NullPointerException`, préservant ainsi le développeur d'un grand nombre de crashes potentiels. Enfin, il propose une pleine interopérabilité avec Java ce qui constitue un point essentiel. A la lecture de ces quelques lignes, vous aurez compris que Kotlin s'annonce clairement comme le langage du futur pour le développement d'applications sous Android et nous allons le mettre en œuvre tout de suite pour réaliser une application ToDo List.

Création d'un projet supportant Kotlin

Le support pour Kotlin est installé par défaut à partir de la version 3.0 d'Android Studio. Il faudra donc utiliser, à minima, cette version pour le développement de notre application. Nous commençons



1 Création d'un projet avec le support Kotlin

par créer un nouveau projet sous Android Studio en veillant bien à inclure le support pour Kotlin sur l'écran de création (figure 1).

Une fois le projet créé, il faut ouvrir le fichier `build.gradle` au niveau de l'application et vérifier que les 2 plugins Kotlin sont bien appliqués au début du fichier comme suit :

```
apply plugin: 'com.android.application'

apply plugin: 'kotlin-android'

apply plugin: 'kotlin-android-extensions'

// ...

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])

    implementation "org.jetbrains.kotlin:kotlin-stdlib-jre7:$kotlin_version"
    // ...
}
```

Il est intéressant de souligner que le plugin Kotlin Android Extensions est en charge de transformer les vues des layouts en propriétés au sein des activités afin d'abstraire le développeur des appels aux méthodes *findViewById*. La configuration de notre projet se termine au sein du fichier *build.gradle* du projet afin de vérifier la présence du plugin Gradle pour Kotlin :

```
buildscript {
    ext.kotlin_version = '1.1.60'

    repositories {
        google()
        jcenter()
    }

    dependencies {
        classpath 'com.android.tools.build:gradle:3.0.0'
        classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:$kotlin_version"
    }
}
```

Ces vérifications faites, nous pouvons passer à la suite et au code à proprement parler.

Modélisation d'une tâche

Une tâche de notre application *ToDo List* sera modélisée au sein d'une classe de données *Task*. On parle de classe de données puisqu'il s'agit d'une classe représentant l'état d'un objet et ne réalisant aucune opération. Notre tâche aura un identifiant, un titre, une description ainsi qu'une date de création. Afin de mieux mettre en évidence la concision apportée par Kotlin, voici l'implémentation classique de la classe *Task* en Java :

code complet sur www.programmez.com & github

Bien que les éditeurs de code actuels permettent la génération de l'ensemble du code de la classe de données *Task*, la maintenance reste à la charge du développeur et constitue une tâche fastidieuse voire inutile.

Passons maintenant à la version Kotlin de l'objet *Task* qui tiendra en une seule ligne de code au sein d'un fichier *Task.kt* :

```
data class Task(val id: Long, val title: String, var description: String, val creationDate: Date)
```

Cette unique ligne de code suffit à obtenir une classe *Task* équivalente à la version Java présentée précédemment puisque tout le code boilerplate est généré automatiquement et demeure transparent pour le développeur. Aucun travail de maintenance n'est donc à prévoir sur ce code.

Création d'un Adapter pour les tâches

Les tâches de notre application seront affichées au sein d'une liste qui s'appuie sur la classe *Adapter* pour le rendu de chacun des éléments la composant. Nous créons donc une classe *TaskAdapter*, héritant de la classe *RecyclerView.Adapter*, qui sera générée avec l'allure suivante par Android Studio :

```
class TaskAdapter : RecyclerView.Adapter<TaskAdapter.TaskViewHolder>() {

    override fun onBindViewHolder(holder: TaskViewHolder?, position: Int) {
        TODO("not implemented")
    }

    override fun getItemCount(): Int {
        TODO("not implemented")
    }

    override fun onCreateViewHolder(parent: ViewGroup?, viewType: Int): TaskViewHolder {
        TODO("not implemented")
    }

    class TaskViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView)
}
```

On remarque ainsi la présence de la classe *TaskViewHolder* qui étend la classe *RecyclerView.ViewHolder*. Ceci est dû au fait que la classe *Adapter* a besoin d'une implémentation de cette classe marquée comme abstraite en son sein. De même, vous aurez remarqué la présence du caractère "?" dans les paramètres de certaines fonctions signifiant qu'une valeur peut être nulle. Ce caractère est ajouté ici car chaque référence provenant d'un objet Java parent peut potentiellement être nulle. En Kotlin, cela devient impossible puisque le langage est null-safe par conception.

Ces observations faites, nous pouvons écrire le code à l'intérieur de notre classe *TaskAdapter*. Nous ajoutons deux paramètres afin de prendre en entrée la liste de tâches à afficher mais également un objet *OnClickListener* permettant de réagir aux clics réalisés sur chaque élément de la liste. On associe ensuite chaque objet *Task* à ce listener dans la méthode *onBindViewHolder*. Dans la méthode *onCreateViewHolder*, on va instancier l'objet *TaskViewHolder* de notre *TaskAdapter* en y passant en entrée la vue *Root* du layout que nous utilisons pour rendre à l'écran un élément de la liste.

Enfin, on finit par implémenter la classe *TaskViewHolder* en associant chaque objet *Task* de la liste aux éléments graphiques afin d'afficher le titre et le texte de chaque tâche. On obtient ainsi le code suivant :

```
class TaskAdapter(val taskList: List<Task>, val listener: (Task) -> Unit) :
    RecyclerView.Adapter<TaskAdapter.TaskViewHolder>() {

    override fun onBindViewHolder(holder: TaskViewHolder, position: Int) =
        holder.bind(taskList.get(position), listener);

    override fun getItemCount() = taskList.size

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): TaskViewHolder {
        val rootView = LayoutInflater.from(parent.context).inflate(R.layout.item_task, parent, false)
        return TaskViewHolder(rootView)
    }

    class TaskViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {
        fun bind(task: Task, listener: (Task) -> Unit) = with(itemView) {

```



```
tvTitle.text = task.title;
tvDescription.text = task.description;
setOnClickListener { listener(task) }
}
}
}
```

Création du Layout XML

L'étape suivante consiste à créer les différents layouts XML pour notre application comme cela se fait habituellement pour les projets Android écrits en Java. Nous définissons en premier le layout d'un élément de la liste de tâches au sein du fichier *item_task.xml* :

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <android.support.v7.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:cardCornerRadius="5dp"
        app:cardElevation="1dp"
        app:cardUseCompatPadding="true"
        app:contentPadding="5dp">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <TextView
                android:id="@+id/tvTitle"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:padding="5dp"
                android:text="Titre d'une tâche"
                android:textAppearance="@style/Base.TextAppearance.AppCompat.Large" />

            <TextView
                android:id="@+id/tvDescription"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:padding="5dp"
                android:text="Description pour une tâche"
                android:textAppearance="@style/Base.TextAppearance.AppCompat.Medium" />
        </LinearLayout>
    </android.support.v7.widget.CardView>
</LinearLayout>
```

Rien de transcendant avec ce layout où l'on a simplement défini 2 composants *TextView* afin d'afficher le titre et la description associés à une tâche. On passe ensuite à la définition du layout de l'activité principale de notre application au sein d'un fichier

activity_main.xml : code sur www.programmez.com & github

Ici, on notera l'importance de bien découper ses fichiers layouts XML pour gagner en lisibilité et donc en maintenance. Pour ce faire, on définit le contenu à proprement parler de l'IHM au sein d'un layout séparé nommé *content_main.xml* :

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:showIn="@layout/activity_main">

    <android.support.v7.widget.RecyclerView
        android:id="@+id/recyclerView"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        tools:layout_editor_absoluteX="8dp"
        tools:layout_editor_absoluteY="8dp" />

    <TextView
        android:id="@+id/tvNotTask"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:gravity="center"
        android:text="Aucune tâche pour le moment."
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        tools:layout_editor_absoluteX="8dp"
        tools:layout_editor_absoluteY="8dp" />
</android.support.constraint.ConstraintLayout>
```

Dans ce dernier layout, nous définissons le composant *RecyclerView* qui assure le rendu des éléments de notre liste ainsi qu'un *TextView* rendu visible uniquement lorsqu'aucune tâche n'est présente dans la liste. Le tout positionné au sein d'un layout parent de type *ConstraintLayout*.

Code de l'activité principale

Il nous reste à écrire le code Kotlin de l'activité principale de l'application. Nousinstancions la liste des tâches afin d'y ajouter quelques tâches avant de configurer le composant RecyclerView en définissant son adapter qui sera un objet *TaskAdapter*.

Nous passons ensuite la liste de tâches en entrée ainsi qu'une implémentation du listener qui sera appelé lors du clic sur un élément de la liste.

Nous gérons l'ajout de nouvelles tâches via un *OnClickListener* positionné sur le bouton flottant.

Enfin, il est nécessaire de rafraîchir à l'écran le contenu de la liste en appelant la méthode *notifyDataSetChanged* du *TaskAdapter* utilisé ce qui nous donne le code suivant :

```
class MainActivity : AppCompatActivity() {

    var id = 0L;

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        setSupportActionBar(toolbar)

        val tasksList = ArrayList<Task>();

        val task1 = Task(id++, "Tâche 1", "Description Tâche 1", Date());
        val task2 = task1.copy(id++, "Tâche 2");
        val task3 = task1.copy(id++, title="Tâche 3", description="Il s'agit de la 3ème tâche");
        tasksList.add(task1)
        tasksList.add(task2)
        tasksList.add(task3)

        recyclerView.layoutManager = LinearLayoutManager(this);
        recyclerView.adapter = TaskAdapter(tasksList) {
            toast("${it.title} Clicked")
        };

        fab.setOnClickListener { view ->
            tasksList.add(Task(id++, "Tâche ${id}", "Description de la tâche ${id}", Date()))
            recyclerView.adapter.notifyDataSetChanged()
        }

        if(tasksList.size > 0) tvNotTask.visibility = View.GONE;
        else tvNotTask.visibility = View.VISIBLE

    }

    private fun toast(msg: String) = Toast.makeText(this, msg, Toast.LENGTH_LONG).show()
}
```

Le développeur Android attentif aura remarqué avec plaisir l'absence d'appels à la méthode *findViewById* pour la récupération des références des vues contenues dans le layout de l'activité principale. Un gain de productivité sympathique au quotidien que l'on doit au plugin Kotlin Android Extensions.



2 Notre ToDo List avec Kotlin en action !

Test de notre ToDo List en Kotlin

Le code de notre application ToDo List en Kotlin terminé, il est temps de passer à la phase de test. Pour ce faire, nous lançons l'application, nous ajoutons plusieurs tâches avant de cliquer sur l'une de ces tâches. La figure 2 nous permet de constater que notre ToDo List minimaliste est pleinement fonctionnelle. **2**

Conclusion

La réalisation d'une application Android ToDo List en Kotlin avec Android Studio nous aura permis de mettre en évidence certaines de ses qualités majeures qui ont poussé Google à en faire un langage officiel de la plateforme. Bien évidemment, cet article n'aura fait qu'effleurer les immenses possibilités de Kotlin. Il est cependant clair que tout développeur Android doit s'investir dans Kotlin dès à présent car ce nouveau langage constitue l'avenir du développement d'applications sous Android. Si migrer des applications Android existantes faites en Java vers Kotlin n'est pas toujours pertinent, le choix de Kotlin pour le démarrage de nouveaux projets s'avère quant à lui plus que justifié.



Sébastien Warin
CEO/CTO Constellation
Creative Technologist
<https://developer.myconstellation.io>
<https://sebastien.warin.fr>

Automatiser sa maison de A à Z

Partie 1

A l'heure de l'intelligence artificielle et de l'internet des objets (IoT), découvrons dans ce dossier les technologies pour concevoir pas à pas une « maison intelligente » en dépassant le cas simple de la domotique largement proposé à travers les box du marché. Amis développeurs, sortez les tournevis et votre IDE et donnez une âme à votre logement !

Aujourd'hui âgé de 2 ans ½, mon fils aîné grandit dans une maison qui lui parle, qui allume ou éteint les lumières, les volets, le chauffage selon les conditions. Il trouve cela normal et cela suscite quelques interrogations de la part du personnel de la crèche quand il parle de la « dame qui parle dans le plafond de la salle de bain ».

Même si cela prête à sourire, il s'agit bien d'une réalité et d'ici quelques années ce sera le standard dans nos logements. Pensez aux GAFA(M) qui proposent tous maintenant des assistants vocaux, des enceintes connectées ou autres protocoles, objets domotique et plus encore.

ATTENTION AU IA WASHING

Attention toutefois aux abus de langage. Dans le monde de la domotique et de l'IoT (voire plus généralement dans le monde des nouvelles technologies), on utilise de manière abusive le terme « intelligent » : capteur intelligent, thermostat intelligent, maison intelligente, enceinte intelligente, etc. Bref tout est intelligent alors que dans la plupart des cas il ne s'agit que de solution « connectée ». C'est ce qu'on appelle l'« IA washing ». On tentera dans ce dossier de faire la part des choses entre la maison connectée et la maison intelligente.

Avant de parler de l'intelligence de la maison, il faut la rendre « connectée ». Pour cela on va mettre en œuvre les principes de base de la domotique. Nous découvrirons dans une première partie les différentes technologies à votre disposition selon les contraintes d'exploitation. Nous compléterons ensuite cette domotique d'objets connectés du commerce ou home-made pour assurer des tâches spécifiques.

L'accent sera mis sur la mixité des technologies. Concevoir une maison connectée sur une seule technologie peut être très limitant mais à l'inverse si plusieurs technologies sont utilisées, il faudra s'assurer de l'interopérabilité de ces dernières pour ne pas tomber dans un patchwork sans nom. Une plateforme d'interconnexion des objets connectés, des applications et des services comme la plateforme Constellation prendra alors tout son sens dans de telles solutions.

D'ailleurs, sur la route de la maison intelligente, il n'y a pas que les objets connectés, n'oublions pas non plus les « services » : la météo, l'agenda, le trafic routier, les réseaux sociaux, les comptes en banque, etc. Les interconnexions de ces services avec les objets connectés de la maison permettront l'émergence de l'intelligence ambiante.

Pour finir nous aborderons les grands principes dans la mise en

place d'une intelligence artificielle pour la maison connectée avec notre plateforme Constellation.

Connecter sa maison

Connecter sa maison consiste à installer des équipements permettant à la fois de percevoir l'environnement (détecteur en tout genre) et d'agir sur celui-ci (pilotage des lumières, des chauffages, etc.). Les éléments de base à connecter dans la maison sont les éclairages et si vous en avez les volets.

Dans un logement les éclairages sont généralement des points lumineux au plafond ou sur vos murs ou bien des prises contrôlées.

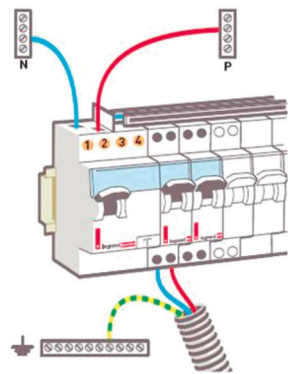
L'électricité générale

Le courant 220V arrive au tableau électrique juste après le disjoncteur de branchement (que l'on nomme plus communément le disjoncteur général), lui-même derrière le compteur ERDF/Enedis. Ce courant alimentera en parallèle une ou plusieurs rangées selon la taille de votre tableau. En principe en tête de chaque rangée on trouve le disjoncteur différentiel par lequel le courant arrive et qui en sortie alimente les disjoncteurs de la rangée.

Un disjoncteur différentiel utilise le principe de l'électromagnétisme qui veut qu'un courant électrique génère une force magnétique proportionnelle à l'intensité du courant qui le traverse. Ainsi on compare l'intensité du courant entre la phase et le neutre : s'il y a une différence d'intensité même minime, l'équilibre des forces magnétiques sera rompu ce qui provoquera la fermeture du circuit. Si l'intensité du courant n'est pas la même entre la phase et le neutre c'est qu'il y a une fuite de courant vers la terre qui si elle transverse un homme peut lui être fatale d'où la nécessité absolue de fermer le circuit. On appelle ce type de dispositif des « protections des personnes ».

1

Une fois que l'on protège les personnes en vérifiant qu'il n'y a pas de fuite, on protège les biens avec des disjoncteurs. Ainsi sur chaque rangée, on alimente en sortie du différentiel plusieurs disjoncteurs. Ces disjoncteurs laissent passer le courant jusqu'à une intensité donnée. Si l'intensité du calibre du disjoncteur est dépassée, le circuit se coupe. On retrouve généralement des disjoncteurs 10A (ampères) pour les circuits d'éclairage, 16A pour les prises, volets ou encore chauffages électriques, 20A pour le four, plaques de cuisson, etc. 2A pour les VMC, sonnette ou autre dispositif à faible consommation, des 32A pour équipement de puissance comme par exemple des plaques de cuisson à induction qui consomment énormément. Bien entendu la section des câbles doit être adaptée au



1 Protections du tableau électrique

calibre du circuit. Par exemple une plaque de cuisine protégée par un disjoncteur 20A utilisant un câble d'une section de $1,5 \text{ mm}^2$ est un non-sens. Le câble aura le temps de fondre avant que le disjoncteur ne saute ! Selon la norme électrique actuellement en vigueur, vous pouvez utiliser du câble électrique d'une section de $1,5 \text{ mm}^2$ jusqu'à 16A, $2,5 \text{ mm}^2$ pour 20A, 6 mm^2 pour 32A, 10 mm^2 au-delà. Attention également à prendre en compte la chute de tension qui est proportionnelle à la longueur du câble, sa section et l'intensité du courant qui le traverse. Ainsi plus le câble est long, plus il faudra augmenter la section pour qu'à une intensité donnée la chute de tension ne dépasse pas les 3%. Par exemple une plaque de cuisson 32A protégée par un disjoncteur 32A au tableau est reliée avec un câble de 30 mètres d'une section de 6 mm^2 comme indiqué précédemment. A pleine puissance (32A) la chute serait de 7,2V soit 6,9% sur une tension monophasée de 230V ce qui dépasse largement les 3% imposés par la norme. Donc même si en soi la section du câble est suffisante pour supporter en toute sécurité 32 ampères, la longueur du câble est trop grande (ou la section trop petite) au regard de la chute de tension pour une telle intensité. En employant un câble d'une section de 10 mm^2 , la chute de tension descend à 4,32V soit environ 1,9%.

Nous ne rentrerons pas plus dans le détail des normes électriques, veuillez toujours vous conformer à la réglementation en vigueur pour votre sécurité.

Pour reprendre notre cheminement, en sortie de chaque disjoncteur, des câbles partent pour alimenter les différents circuits de la maison : éclairages, prises, équipements de la cuisine, volets, chauffe-eau, etc.

Dans le cas de l'éclairage, des volets ou mêmes des prises, un circuit alimentera généralement plusieurs équipements. Typiquement on va retrouver par exemple un disjoncteur « lumières salon » sur lequel part un câble qui alimentera plusieurs points lumineux du salon. Parfois même un seul et même câble en sortie d'un disjoncteur alimente plusieurs pièces (ex : « Lumières Chambres »).

Dans ce cas il est impossible de piloter ces dispositifs depuis le tableau autrement vous piloterez toutes les lumières/prises ou volets

du circuit en même temps ! On devra alors connecter les dispositifs individuellement le plus souvent avec une technologie sans fil s'il n'est pas possible/souhaitable de recâbler.

De plus concernant l'éclairage, le neutre et la phase n'arrive pas directement au point lumineux (ou à la prise contrôlée). En effet, généralement seul le neutre partant du tableau arrive au point lumineux. La phase elle arrive

jusqu'à un interrupteur où un fil de retour repart jusqu'au point lumineux. On dit alors que la phase est coupée au point de contrôle (l'interrupteur). Et en d'autres termes, la « configuration des contrôleurs » est fixée dans les murs. Si vous faites la rénovation de votre habitation ou mieux que vous construisez, ne faites pas ça !

Les technologies domotique

Comme en informatique avec le modèle OSI, la 1ère couche est la couche physique. En domotique c'est le même principe, il existe plusieurs technologies domotique qui au niveau de la couche physique seront soit filaire ou soit sans fil sachant qu'en filaire on

pourrait distinguer deux types : celles utilisant leur propre réseau ou celles qui utilisent les câbles électriques existants (CPL).

Les technologies filaires

Comme nous l'avons vu, dans une maison « classique » l'électricité est relativement simple. Elle est distribuée depuis le tableau électrique qui assure la protection des différents circuits et alimente l'ensemble des équipements au moyen de boîtes de dérivation. Les éclairages sont généralement contrôlés par des interrupteurs, simples dispositifs mécaniques qui « coupent » la phase de la prise/lampe contrôlée. Cette approche ne permet pas de contrôler les équipements et font proliférer le 230V dans toute la maison. De plus elle rend difficile l'évolution dans le temps (les câbles étant ancrés dans les murs, le lien entre le contrôleur et l'action peut être difficilement changé !).

Pour résoudre cela des technologies par « bus de commande » sont arrivées il y a plus de vingt ans essentiellement pour les bâtiments industriels et immeubles de bureau. Elles s'implantent aujourd'hui de plus en plus dans les habitats hauts de gamme et commencent à apparaître dans les appartements de standing. L'idée est de séparer le réseau électrique du bus de commande sur lequel seront connectés les différents éléments de contrôle (interrupteur, thermostat, télécommande, écran tactile, etc.) sur des tensions à faible voltage. Grâce à cela, il nous est possible de contrôler l'ensemble des équipements de la maison depuis n'importe quel type d'interface et nous offre également la possibilité d'associer n'importe quelle commande à une action individuelle, groupée ou générale, et cela à n'importe quel moment (sans modifier le câblage).

La technologie filaire la plus connue et reconnue comme un standard est le KNX. On retrouve aussi ce qu'on appelait anciennement les « Web Relays », des cartes de relais connectées comme l'IPX800, Loxone, domot-MX, etc.

Le KNX est devenu en 2006 un standard normalisé au niveau international et a été choisi par 90% des constructeurs d'équipements électriques en Europe comme Hager, Schneider Electric ou encore Siemens.

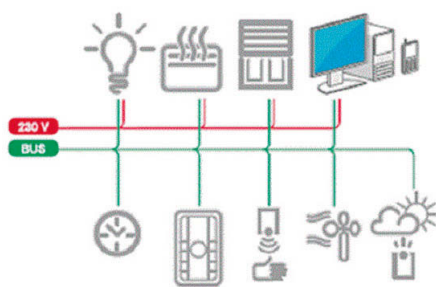
Le principe est simple : câblez en étoile et pilotez l'ensemble des dispositifs du tableau électrique.

En effet l'idée est que chaque dispositif, que ce soit un volet, un point lumineux, une prise soit individuellement ramené au tableau électrique sans aucune dérivation. Bien sûr cela fait beaucoup de câble dans le tableau, c'est à prévoir !

On installera ensuite dans le tableau des actionneurs qui ne sont ni plus ni moins que des relais permettant d'ouvrir ou fermer le circuit à l'image d'un interrupteur. Ces relais seront bien sûr pilotables depuis une interface informatique. Ainsi on sera capable de contrôler chaque point lumineux, prise ou volet individuellement.

Les interrupteurs physiques, c'est-à-dire les points de contrôle seront eux aussi câblés jusqu'au tableau mais sans être des « coupeurs de phases », ils ne transportent plus de courant 220V.

Sur ce point le principe est différent entre le KNX et les « Web Relais » type IPX800. En KNX l'interrupteur est à un élément actif (c'est à dire équipé d'électronique et alimenté pour fonctionner). De ce fait les points de contrôle comme un simple interrupteur coûtent plus cher qu'un simple va-et-vient mécanique qu'on trouve dans n'importe quel magasin d'électricité générale. 2



2 Bus KNX

Chaque point de contrôle en KNX est un dispositif qui fait partie du réseau KNX et dispose d'une adresse. Physiquement tous les points de contrôle sont connectés sur un bus. On doit le programmer pour lui définir l'action à réaliser comme par exemple allumer ou éteindre tel circuit.

Le mapping entre l'interrupteur et le(s) dispositif(s) contrôlé(s) est donc configurable au niveau du logiciel. Rien n'est figé dans vos murs ! Si vous voulez changer le comportement de tel ou tel interrupteur il suffira de le reprogrammer ! **3**

C'est ainsi pour résumer que fonctionne le KNX : un réseau de puissance (220V) en étoile piloté par des actionneurs et de l'autre un réseau de commande (24V) sous forme d'un bus où l'on connecte les points de commande comme des interrupteurs. Et bien entendu on pourra sur ce bus de commande connecter une passerelle IP.

Sur les cartes « Web relais » comme l'IPX800 de GCE Electronics par exemple on retrouve le même principe pour le circuit de puissance : l'IPX800 est un module au format DIN qui s'installe dans un tableau et dispose de 8 sorties, c'est-à-dire 8 relais sur lesquels on peut connecter nos différents dispositifs à piloter. On pourra également étendre l'IPX800 avec des extensions pour ajouter des entrées ou sorties supplémentaires, une passerelle sans fil (EnOcean), GSM, des variateurs, etc. **4**

On ramène ainsi individuellement chaque point lumineux à notre tableau au niveau des sorties de l'IPX que l'on pourra piloter informatiquement (l'IPX embarque un serveur Web avec une interface RJ45 connectée sur votre réseau LAN).

Pour les points de contrôle, l'IPX800 (et les équivalents) proposent des entrées digitales. Il suffit donc d'envoyer une impulsion pour activer une entrée.

De ce fait l'idée est de câbler individuellement jusqu'au tableau chaque point de contrôle. Je recommande ici d'utiliser du câble Ethernet (4 paires de fils par câble). Tous vos interrupteurs (type va-et-vient ou bouton-poussoir) seront donc tous connectés sur les entrées de l'IPX. Si le poussoir n'est pas pressé, l'entrée n'est pas activée et lorsque vous appuyez sur le poussoir vous activez l'entrée.

Le « mapping » entre les entrées (par exemple un bouton poussoir) et les sorties (points lumineux par exemple) sera défini dans la configuration de l'IPX. Vous aurez ainsi la souplesse de configurer les mappings via l'interface d'administration de l'IPX selon vos besoins.

La différence entre le KNX et l'IPX est donc que dans le premier cas les interrupteurs (éléments actifs) portent la configuration, alors que dans le cas de l'IPX, les interrupteurs (éléments passifs) ne font qu'activer une entrée sur l'IPX central qui, lui, porte la configuration. Autrement dit à l'inverse de l'IPX, le KNX est une architecture décentralisée.

On comprend bien que ce genre de technologies ne peut être déployée qu'en cas de rénovation (refonte complète du réseau électrique) ou d'une nouvelle construction.

Les technologies sans nouveau-fil

Quand il n'est pas possible de passer de nouveaux câbles, une so-

lution consiste à utiliser ceux déjà existants, c'est-à-dire le câblage du réseau de tension. On appelle cela « technologie par courant porteur » où l'on utilise le câblage électrique pour communiquer des informations à une fréquence et tension différentes. Aussi connu sous l'acronyme CPL (courant porteur en ligne), on l'utilise généralement dans les foyers pour mettre en réseau des équipements informatiques facilement sans configuration Wifi ni passage de nouveaux câbles RJ45 : d'où le surnom « sans nouveaux fils » ! Ce genre de technologie demande une bonne qualité de l'installation électrique existante, ce qui dans des vieux bâtiments est rarement le cas. La prolifération de boîtes de dérivation dans le foyer tend à affaiblir le signal. De plus, de nombreux appareils électroniques peuvent créer d'importantes perturbations sur le réseau électrique ce qui « brouille » la communication entre les équipements. En tête : la machine à laver, le sèche-cheveux, le micro-onde ou encore certaines alimentations des téléphones et ordinateurs. Il faut avant tout auditer son réseau électrique pour déterminer sa qualité, ajouter des filtres sur les équipements qui pourraient perturber le signal et supprimer les sources parasites bien que certaines technologies par courant porteur embarquent directement des filtres anti-bruit. Parmi les technologies domotiques par courant porteur, on y trouve la solution « In One » de Legrand, le X2D de Delta Dore, le PCLBUS, des passerelles pour le protocole KNX et Lonworks ainsi que le protocole X10.

Le X10 est un excellent moyen de découvrir la domotique très facilement et à moindre frais. Ce protocole a été développé à l'origine par Pico Electronics en 1975. On trouve actuellement différents fabricants de produits X10 comme XDOM, Marmitek, Powerhouse ou Home System. Le protocole X10 supporte aussi le sans fils par transmission radio (433Mhz) grâce à un convertisseur radio/porteuse.

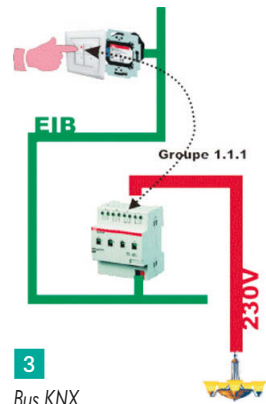
En X10 vous retrouvez trois types d'équipement : les émetteurs, les récepteurs et les contrôleurs.

Les émetteurs permettent d'envoyer des messages X10 sur le réseau comme les télécommandes, certains interrupteurs ou les capteurs (de météo ou de sécurité). En général, les émetteurs fonctionnent par radio. **5**

Les récepteurs correspondent à des actionneurs répondant aux ordres X10 dans lesquels on retrouve les modules d'Appliance (contact On/Off) et les modules Lampe (qui gèrent en plus la variation de la tension). Chaque module (émetteur ou récepteur), dispose d'une adresse comportant le code « maison », lettre de A à P, et un code « unité », chiffre de 0 à 16 permettant d'adresser jusqu'à 256 adresses différentes. Chaque adresse est définie physiquement sur chacun des modules, et plusieurs modules peuvent avoir la même adresse (et donc répondre aux mêmes ordres). **6**

Les récepteurs peuvent prendre la forme de module « gigogne », c'est-à-dire qu'ils s'intercalent entre la prise électrique et l'équipement à brancher (ou entre la douille et l'ampoule à viser) ou bien sous forme de module Rail DIN à installer dans le tableau électrique (ce qui rend l'installation plus discrète).

Les contrôleurs permettent d'émettre et de recevoir des messages



4 IPX800-V4



5 Détecteur X10 (MS13)



6 Récepteurs X10



7 Module X10 sur Rail DIN

X10 sur le réseau électrique ou radio. Ils disposent généralement d'une interface USB ou TCP/IP pour être exploités par un PC, une tablette, etc.

Bien que le X10 soit relativement peu coûteux et très simple à mettre en œuvre, il souffre de sérieux inconvénients comme sa fiabilité et sa sécurité !

En effet, tout le monde peut émettre et écouter sur le réseau X10 ! Soit par les ondes soit par le réseau électrique, en branchant un simple contrôleur sur une prise électrique, vous pourrez donner n'importe quel ordre X10 sur le réseau ! Dans un immeuble on recommande d'ailleurs de mettre un filtre juste après l'arrivée EDF pour être sûr qu'aucun ordre émis dans l'appartement à côté ne puisse arriver chez vous !

Hormis la sécurité, c'est son degré de fiabilité qui nous pousse à réfléchir quant à son installation à travers toute la maison ! Tout d'abord certaines perturbations générées par les appareils électriques (TV, Hifi, PC, électroménager) brouillent les signaux X10 et l'ordre n'arrive pas à destination ! Ces perturbations électriques peuvent également déclencher des ordres X10 non souhaités ! Il m'est déjà arrivé d'avoir une lampe qui s'éteint ou s'allume toute seule ! Les ordres sont relativement longs à être transmis sur le réseau (50 bits/sec), et comme le

protocole ne gère pas le retour d'état (accusé de réception), lorsque l'on donne un ordre (éteindre une lumière par exemple), il faut « croiser les doigts » pourvu que l'ordre soit bien transmis, bien reçu et bien exécuté par le récepteur. Aucune garantie ne sera donnée si ce n'est la vérification visuelle par soi-même (il existe aussi la possibilité d'envoyer au récepteur X10 un message de type « Query » lui demandant de renvoyer un message indiquant son état). 7

Pour obtenir une bonne qualité de service avec le X10, la « best practice » est de privilégier l'usage des modules Rail DIN de manière à centraliser l'ensemble des récepteurs dans le tableau électrique afin de minimiser la distance à parcourir entre le contrôleur et les récepteurs (et éviter l'exposition aux perturbations électriques sur réseau). Mais cela demande comme pour une installation filaire (KNX, IPX ou autre) de câbler en étoile son réseau électrique pour ramener individuellement chaque prise/lampe au tableau. Dans le cas d'un logement existant il est donc impossible de mettre en œuvre sans entamer un lourd chantier de rénovation !

Les technologies « sans fil »

Quand il est difficile d'équiper son habitat avec une technologie filaire (demandant de revoir complètement le câblage) la dernière possibilité reste l'utilisation du « sans fil » !

Comme nous l'avons vu, il existe des extensions aux technologies domotiques filaires ou par courant porteur pour fonctionner sans fil. C'est le cas du KNX qui propose une passerelle « KNX radio » permettant d'ajouter des interrupteurs (contrôleurs) sans fil lorsqu'il n'est pas possible de les connecter sur le bus de commande. Il en va de même avec l'IPX800 et sa passerelle EnOcean. Aussi, les gammes de produits déjà citées, comme celles

de Legrand (InOne), Insteon, Lonworks ou encore X2D de Deltadore, proposent des extensions sans fils. Il faut cependant les considérer comme des « extensions » venant étendre un réseau domotique filaire ou CPL, et non comme la possibilité de créer un réseau domotique 100% « sans fil ». 8

En entrée de gamme, il existe un ensemble de produits domotiques « grand public » que l'on trouve dans les grandes surfaces ou magasins spécialisés. Des packs généralement composés de prises gigognes et télécommandes permettent (pour moins de 30 euros) de télécommander quelques prises, très pratiques pour contrôler les lampes du salon. On y trouve la technologie HomeEasy utilisée notamment par les produits Chacon que l'on retrouve un peu partout en France. Cette technologie monodirectionnelle peut être pilotée « informatiquement parlant » grâce à des émetteurs/récepteurs de trame RF 433Mhz comme avec le RFXCOM (en version USB à 95 euros ou en version LAN pour 200 euros). Le RFXCOM permet de piloter des produits X10, Insteon, sondes Oregon Scientific, etc. 9

Chez Velux, Assa Abloy (serrure électronique) ou encore Honeywell, on utilise la technologie IO Homecontrol, plus haut de gamme, avec l'avantage d'être bidirectionnelle. Chez Somfy on utilise le protocole RTS.

Depuis 2010, on a pu voir apparaître une technologie prometteuse : EnOcean. Cette société allemande a inventé une technologie de communication sans fil et sans pile ! Celle-ci est généralement utilisée pour les capteurs et émetteurs. Elle capte l'énergie de l'environnement, soit de la lumière (cellule photovoltaïque), soit d'une différence de température ou encore de la pression mécanique sur un interrupteur. Il n'y a donc plus besoin de câble ni de pile ! 10

La fiabilité reste le principal défi du « sans fil ». En effet comment pouvons-nous être certains que l'ordre arrive bien à sa destination ? Comment fait-on dans un grand logement, avec différentes contraintes (murs porteurs, canalisations, ...) qui affaiblissent considérablement le signal ?

Lorsqu'il s'agit d'équiper une seule pièce pour piloter quelques lampes grâce à une télécommande, cela ne pose pas de problème (la proximité entre la télécommande et les lampes limite les problèmes de transmission et si la lampe ne reçoit pas l'ordre, elle ne s'allumera pas. J'aurai donc une confirmation visuelle et dans ce cas, j'appuierai à nouveau sur le bouton de la télécommande). Mais lorsqu'il s'agit d'équiper entièrement son habitat pour contrôler l'ensemble des équipements, il semble primordial d'avoir confiance sur le fait que l'action émise localement ou à distance sera quoiqu'il en soit bien exécutée ! Pour cela il est nécessaire d'utiliser une technologie bidirectionnelle à retour d'état, afin de confirmer au contrôleur et/ou à l'utilisateur que l'ordre a bien été réceptionné et exécuté par l'actionneur cible grâce à l'envoi d'un accusé de réception.

La sécurité peut aussi être une problématique. Par exemple, une télécommande de la marque IDK permettant de piloter jusqu'à 16 prises est sécurisée par la programmation d'un « code maison/unité » entre la télécommande et les prises. Mais sur ce



8 Produits DIO de Chacon



9 RFXCOM en version USB



10 Interrupteur EnOcean

produit, le code n'a que 16 combinaisons possibles ! Autant vous dire qu'il paraît très facile de piloter le réseau de votre voisin en essayant toutes les combinaisons de code un à par un (comme avec le X10 d'ailleurs, où il n'y a que 16 codes « maison » possibles, les lettres de A à P !). **11**

Enfin, pour résoudre les problèmes liés à la distance de transmission et contrer les différents obstacles, il est nécessaire d'utiliser des répéteurs/relais ou routeurs afin de couvrir l'ensemble de l'habitat. Dans cette lignée, il existe deux technologies très intéressantes : le ZigBee et le Z-Wave qui permettent de créer des réseaux maillés. Dans cette topologie certains éléments du réseau (émetteurs ou récepteurs) peuvent se comporter comme des relais et ainsi créer un réseau de maillage. Dans le cas où le contrôleur ne parvient pas à joindre directement le récepteur cible, il peut transmettre l'ordre à d'autres équipements, physiquement plus proches, qui se chargeront de router le message jusqu'à « bon port » en utilisant un ou plusieurs nœuds. À l'image des protocoles de routage sur internet, chaque équipement échange avec ses voisins la liste des équipements qu'ils peuvent joindre directement et ainsi définir une table de routage pour joindre l'ensemble des nœuds du réseau. Lors de l'installation du réseau Z-Wave, il faudra garder à l'esprit cet aspect pour placer judicieusement ces équipements afin de couvrir au mieux son habitat et éviter les « zones d'ombres ». **12**

Pour la partie théorique, le Z-Wave a été développé par une société danoise Zensys, rachetée en 2008 par Sigma Design qui utilise la bande de fréquence de 868,42Mhz pour l'Europe. Bien que propriétaire, une alliance a été créée en 2008 et réunit aujourd'hui plusieurs centaines de fabricants commercialisant des produits Z-Wave. Cela permet de garantir une certaine pérennité du protocole mais aussi un large choix de produits (on compte aujourd'hui plus de 2 400 produits Z-Wave sur le marché).

Le Z-Wave se définit comme une technologie bidirectionnelle à retour d'état permettant de garantir la bonne exécution des ordres. Très rapide (environ 50ms pour émettre un ordre et obtenir l'accusé de réception) et beaucoup plus sûre par l'utilisation d'un code sur 32 bits (soit plus de 4 milliards de combinaisons possibles) qui bloque tout équipement ne disposant pas du bon code bien qu'une faille de sécurité exploitant le processus d'appairage ait été découverte en 2017. Enfin, comme décrit précédemment, sa topologie en réseau de maillage permet de couvrir aisément l'ensemble de l'habitat en prenant compte des différentes contraintes physiques. Un réseau Z-Wave se compose d'un ou plusieurs contrôleurs (clé USB, box Wifi/Ethernet, interrupteurs ou télécommande) « maîtres » gérant la sécurité ainsi que le routage. Il est possible de gérer jusqu'à 232 périphériques Z-Wave (au-delà, il faudra créer un autre réseau Z-Wave et établir un pont entre les réseaux). Les périphériques peuvent être des actionneurs (prises gigognes, relais encastrables, sirènes, modules pour volets, ou actionneur de chaudière ...) ou des émetteurs (de consommation, de sécurité, météo, interrupteurs, ou télécommandes). Pour lier les émetteurs/récepteurs au réseau Z-Wave (géré par un contrôleur), il faut les appairer en entrant dans le mode « association » sur le contrôleur et en activant le périphérique à ajouter au réseau (appuie sur un

bouton ou une séquence de boutons en fonction du produit) dans un rayon de moins de 2 mètres afin de garantir une certaine sécurité quant à l'inclusion de périphériques étrangers ! C'est à ce moment-là que seront recalculées les tables de routage en tenant compte des nouveaux périphériques. C'est pourquoi il est préférable d'associer les périphériques à leurs emplacements définitifs.

**11**

Prises télécommandables IDK

Mixité technologique : ajoutez des objets connectés

Une fois sa technologie domotique choisie, par exemple basée sur l'IPX800 ou sur un réseau Z-Wave, ne tombez pas dans le piège de tout vouloir faire avec celle-ci.

Non pas que je recommande d'installer tout et n'importe quoi, mais pour une fonction précise vous n'aurez pas forcément le meilleur choix en vous retréignant à une seule technologie.

D'ailleurs vous pouvez aussi avoir connecté votre maison avec des IPX800 couplés à du Z-Wave, EnOcean ou autre pour certaines pièces ou usage.

Prenez l'exemple du thermostat : il y a très peu de thermostats Z-Wave et les seuls actuellement disponibles ressemblent aux thermostats des années 90. À côté de ça on trouve pour un prix équivalent des thermostats à la fois performants, fonctionnels et très design tel le Nest ou le NetAtmo.

Il ne faut donc pas se limiter à une technologie mais choisir la meilleure dans chaque domaine. Attention toutefois à ne pas tomber dans un « patchwork » d'objets connectés où vous retrouvez avec des dizaines d'applications / fonctionnements différents !

Tous ces objets connectés doivent/devront communiquer ensemble et être connectés dans un « système global », que ce soit une plateforme d'interconnexion telle que Constellation, ou bien une box/système domotique suffisamment ouvert avec une notion de plugin et/ou d'API pour pouvoir ajouter de nouveaux équipements au gré des années.

Le thermostat connecté

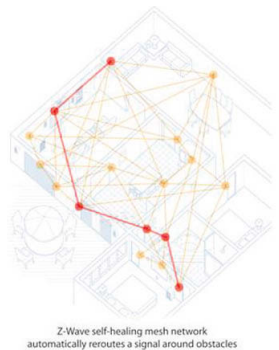
Le thermostat Nest est un très bon choix, design il s'intégrera parfaitement dans votre intérieur. Il communique avec une liaison filaire ou sans fil avec le Heat Link, un petit boîtier à connecter sur votre système de chauffe. **13**

Il embarque des fonctions d'autoapprentissage et de détection de présence même si (selon votre installation) je vous recommande de les désactiver. En effet, la programmation et le contrôle du thermostat seront réalisés par notre propre « IA » et la détection de mouvement confiée au système d'alarme bien plus précis (réseau de capteur).

Côté développeur, Nest met à disposition un service basé sur les « Server-sent events » permettant de piloter et recevoir les informations en temps réel des équipements de la marque.

Les lampes connectées

Si vous n'êtes pas passé par la case « rénovation » ou « construction » il y a peu de chance que vos points lumineux soient pilotables



Z-Wave self-healing mesh network automatically reroutes a signal around obstacles

12

Réseau maillé

**13**

Thermostat Nest



14 Ampoules connectées Hue



individuellement. Ainsi dans mon salon pas moins de 8 appliques murales illuminent la pièce à vivre.

Sans technologie filaire, j'ai donc un seul circuit qui alimente ces appliques en passant par deux in-

terrupteurs va-et-vient pour contrôler l'éclairage. Bien entendu derrière un des interrupteurs on peut placer un micromodule Z-Wave comme le FGD221 fonctionnant sans le neutre (généralement absent dans la boîte d'encastrement d'un interrupteur). De cette façon on peut piloter via le réseau Z-Wave l'éclairage du salon ! Le seul problème est qu'on a ici un « tout ou rien » : soit toutes les lampes sont allumées soit tout est éteint ! Pour pallier ce problème quand il n'est pas possible de recâbler, on peut utiliser des ampoules connectées.

Il en existe plusieurs mais mon choix en 2014 s'est porté sur les ampoules Hue. Elles communiquent en ZigBee via une passerelle Ethernet nommée le « Bridge » qui expose un service http permettant à une application ou un système domotique de piloter chaque lampe individuellement. 14



15 PoolCop



16 Amplificateur home-cinéma connecté

On peut donc adapter l'intensité et même la couleur de chaque ampoule pour créer l'ambiance désirée. Chaque ambiance pouvant être enregistrée sous forme d'une « Scène » qu'on adaptera en fonction de l'activité : « réception », « lecteur », « feu de cheminée », « cinéma », etc.

Attention toutefois aux ampoules : il faudra « supprimer » le fait de couper la phase avec l'interrupteur d'origine. En effet, si vous coupez la phase les ampoules ne seront plus alimentées donc plus connectées. Impossible alors de les piloter ou de les allumer ! Quand vous éteignez les lumières, l'ampoule doit rester alimentée.

Dans mon cas, l'interrupteur a été by-passé avec un domino de manière à ce que le courant 220V arrive toujours aux appliques murales. Les deux interrupteurs

ont été connectés à un micro-module Z-Wave qui ne pilote aucune charge mais le fait d'appuyer sur un des deux boutons poussoirs lève un événement Z-Wave, ce qui me permet de le traiter dans le système domotique.

On peut par exemple changer l'ambiance lumineuse du salon de manière cyclique à chaque appui ou encore éteindre toutes les lampes lors d'un appui long : tout cela n'est que du code qu'on programmera à souhait !

Pour les plus chanceux qui ont une piscine, PoolCop offre une solution de gestion de la piscine connectée : analyse de l'eau, gestion de la filtration, du chauffage, du niveau de l'eau, du pH, désinfection, de l'éclairage, etc. 15

Bien entendu la solution met à disposition une API http/REST pour l'intégration dans un système domo-

tique. On pourrait aussi parler des serrures connectées, sonnette/interphone connectés, capteurs en tous genres, etc. La seule contrainte est d'avoir sur chaque objet une interface de communication pour permettre une intégration dans un système global sans quoi on tombe dans le « patchwork » des objets connectés où chaque système reste dans son coin sans aucune coopération tuant dans l'œuf « l'intelligence ambiante ».

Au-delà de la domotique classique

Dans l'énumération des objets connectés prenons également le cas des équipements multimédia : TV, ampli, système de diffusion sonore, etc.

Là encore assurez-vous d'avoir des services plus ou moins ouverts pour permettre une intégration dans votre système domotique.

A cet égard, les amplificateurs home-cinéma Pioneer exposent un service TCP pour le pilotage intégral de l'ampli. Il devient alors possible d'automatiser certaines tâches. 16

Même principe pour les téléviseurs. On peut ainsi automatiser le fait de lancer un film depuis la voix : allumer la TV, l'ampli, et le NUC (via un WoL), sélectionner le NUC comme entrée de l'ampli, démarrer votre film ou série via l'API Kodi, tamiser l'ambiance du salon en changeant la scène sur les ampoules Hue, fermer le volet avec le micromodule Z-Wave, etc.

On voit bien dans l'exemple ci-dessous la coopération entre plusieurs technologies : TV et ampli (objets connectés en IP), NUC (système Windows ou Linux), Kodi (application), lumières (ampoules ZigBee), volet (module Z-Wave).

On peut aussi parler de la diffusion audio multi-room. J'ai par exemple opté pour des enceintes encastrées dans les faux-plafonds de différentes pièces. 17

Comme toujours, tout cela est câblé en étoile, c'est-à-dire que les câbles de chaque enceinte de la maison descendent jusque dans ma baie technique.

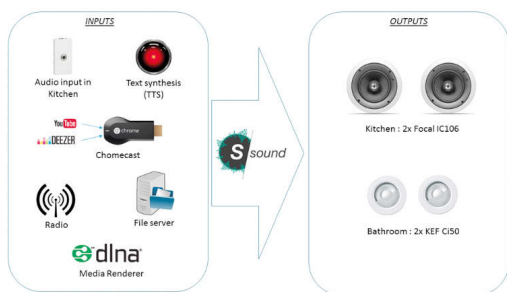
Toutes les enceintes de chaque pièce sont connectées sur leur propre amplificateur type SMSL50 (amplificateur de petite taille) lui-même alimenté par une carte son USB dédiée type Behringer UCA222. Côté OS on a donc autant de sorties audio que de pièces.

Avec un système comme S-Sound (disponible sur le store Constellation) on peut donc, pour chaque pièce, diffuser un fichier audio (MP3, WMA, etc.), un stream audio (comme une webradio), un message vocal (TTS) ou encore diffuser une source en entrée (prise jack mural dans certaine pièce ou encore un chromecast permettant d'envoyer la musique depuis son smartphone). 18

Grâce à cela vous dotez votre maison d'une interface de communication vocale à moindre frais. Pour finir cette partie, on pourrait également ajouter les capteurs météo ou autres. A ce sujet NetAtmo propose une solution intéressante intégrant la température, humidité, sonomètre, pression atmosphérique ou encore taux de CO₂ très utile dans les chambres. Comme toujours avec NetAtmo, les données sont disponibles à travers une API REST. On peut aussi déployer des capteurs RF type Oregon qu'on captera avec une passerelle comme le RFXCOM. 19



17 Enceinte encastrable Kef



18 S-Sound

L'alarme comme système de sécurité et source de donnée

Quoi de plus rassurant que de dormir sur deux oreilles tout en sachant que la maison veille sur nous. Au-delà du simple système de sécurité qui détecte les intrusions, l'alarme est également un réseau de capteurs très intéressant pour alimenter les règles d'automatisation de la maison pour ne pas dire de « l'intelligence de la maison ».

En effet une alarme est composée d'une multitude de capteurs (volumétrique et périmétrique essentiellement) avec des panneaux de contrôle et des sirènes.

Si vous souhaitez avoir de la fiabilité et sérénité oubliez les systèmes d'alarme intégrés avec des fonctions domotique. Une alarme doit être autonome et ne requiert aucun système de plugin ou de mise à jour telle une box domotique ! N'hésitez pas à consulter mon blog, j'ai publié un billet complet en 2014 sur ce sujet.

Dans cette lignée je vous recommande les systèmes Paradox et notamment la gamme EVO, système d'alarme de très grande qualité dans lequel on retrouvera l'extension IP150 ou l'APRT3 mettant à disposition une interface IP ou série respectivement. **20**

Le connecteur Constellation d'ailleurs disponible sur le store vous permettra de contrôler l'alarme (armement et désarmement) et surtout de pouvoir récupérer en temps réel l'état de chaque zone.

Autrement dit, vous pouvez réagir aux événements produits par les détecteurs de l'alarme : une porte ou une fenêtre s'ouvre, un mouvement est détecté dans telle pièce, etc. dans vos applications Python, Arduino, .NET, Javascript, etc.

Cela vous donne une source non négligeable d'informations sur l'environnement de la maison pour alimenter l'intelligence de la maison : une fenêtre est restée ouverte alors qu'il pleut, la porte du garage est ouverte lorsqu'il n'y a plus de mouvement depuis un certain temps, une présence est détectée dans le salon alors que la nuit est tombée, l'alarme vient de passer en mode nuit, etc.

La vidéo-surveillance

Pour compléter le système de sécurité vous pouvez également déployer des caméras IP ou analogiques.

Les versions analogiques sont bien moins chères et se connectent sur un câble coaxial sans oublier l'alimentation (il existe des câbles deux-en-un). Toutes les caméras analogiques sont à connecter sur un DVR. Oubliez les fonctionnalités natives d'une DVR souvent de mauvaise qualité, voyez plutôt le DVR comme une interface d'acquisition des flux vidéo. Dans cette optique je vous recommande les DVR Floureon qui propose 4, 8 ou 16 entrées analogiques et qui pour chaque entrée expose le flux vidéo sur une URI RTSP (*Real Time Streaming Protocol*). **21**

Autrement pour des caméras IP, je vous recommande de partir du filaire RJ45 alimenté en PoE (l'alimentation passe par le câble RJ45). Il faudra soit utiliser un injecteur PoE, ou mieux, utiliser un switch PoE.

L'important est de s'assurer que votre camera expose son flux vidéo en RTSP (flux vidéo temps réel) plutôt qu'un simple M-JPEG (série de photo continue) ou pire d'avoir un système fermé ce qui est parfois le cas sur les caméras connectées grand-public obligeant alors

l'utilisation de l'application du constructeur.

Pour fédérer l'ensemble de vos caméras (et là aussi éviter l'effet patchwork) je vous recommande vivement ZoneMinder ou Shinobi. L'un comme l'autre vous permettront de connecter tous types de camera dans un système unique sans avoir à jongler entre les applications pour voir telle ou telle camera. De plus le système de détection de mouvement est très avancé et vous permettra de définir pour chaque camera les zones à prendre en compte ou non avec un paramétrage très fin. **22**

L'interface bien pensée vous permettra aussi de retrouver facilement les traces des mouvements enregistrés. De plus ZoneMinder comme Shinobi exposent un service REST pour, là encore, pouvoir intégrer la solution dans votre « système global ».

Une fois n'est pas coutume un connecteur Constellation/Zone Minder est disponible. Cela vous permet d'avoir accès à chaque camera sans se soucier du protocole ou sécurité de chacune d'entre elles, de s'abonner aux événements levés par la détection de mouvements dans vos algorithmes ou encore de pouvoir piloter l'enregistrement de ces dernières (par exemple si alarme en cours sur le système d'alarme on passera toutes les caméras en enregistrement forcé).

Le réseau informatique

Bien qu'il ne s'agisse pas à proprement parler de domotique, vous devez anticiper le déploiement de votre réseau informatique qui pourra d'ailleurs être le réseau de communication de vos dispositifs domotique ou autres objets connectés (beaucoup d'objets connectés fonctionnent par Wifi sur la couche IP).

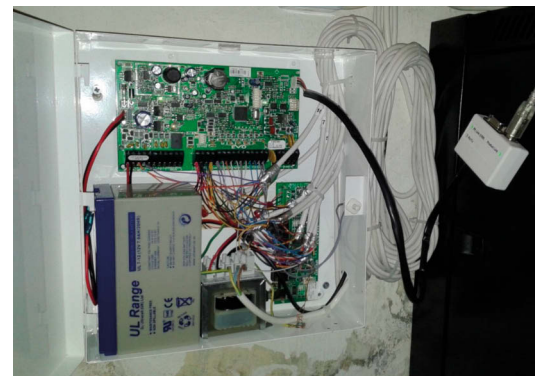
Si vous êtes dans un chantier de rénovation ou de construction, privilégiez une installation en étoile, en ajoutant des prises RJ45 Cat6 (avec blindage) dans chaque pièce et ramenez-les tous en un point central pour les connecter sur un panneau de brassage. Placez également des prises à l'extérieur et en hauteur pour y placer éventuellement des caméras IP ou bornes Wifi. Au prix du mètre de câble n'hésitez pas aussi à tirer des câbles en attente, une fois les cloisons montées il est sera trop tard.

Pour les câbles partez sur du Cat7 blindé sans oublier de relier le blindage de chaque câble à la masse. Pour les références, la marque Ligawo offre un très bon rapport qualité / prix que ce soit pour le panneau de brassage (comptez 30€ pour un panneau 24 ports Cat6a) comme pour le câble (environ 40€ la bobine de 100 mètres en Cat7 S/FTP).

Une fois l'ensemble des prises de la maison disponibles sur le panneau de brassage, optez pour un simple switch Giga Ethernet ou un switch administrable si vous avez quelques notions en réseau. Cela vous permettra par exemple



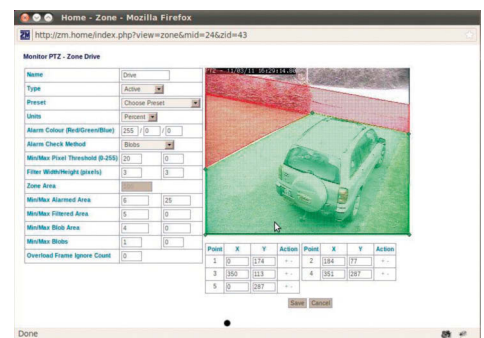
19 Sonde Oregon



20 Alarme Paradox EVO192



21 DVR Floureon



22 ZoneMinder



23 Ma baie réseau

de séparer votre réseau domestique des équipements domotique ou encore des serveurs et équipements réseau en déclarant plusieurs VLAN (réseau LAN virtuel), de gérer la CoS ou encore l'IGMP Snooping si vous diffusez la TV en multicast IP.

Les switches TP-Link ou Netgear sont très abordables pour une installation domestique (un GS724T dans mon cas). Pensez également aux versions

avec des ports PoE si vous installez des bornes Wifi ou cameras IP alimentées en PoE.

Côté routeur, je ne peux que vous recommander de remplacer votre box internet par un véritable routeur comme les Mikrotik (RouterOS) ou encore Ubiquiti. Vous pourrez alors déployer des bornes Wifi dans l'ensemble de votre logement pour diffuser proprement le signal à tous les étages sans devoir avoir recours à des solutions onéreuses telles que les Orbi de Netgear ou Velop chez Linksys. 23

Côté serveur, optez de préférence pour un véritable serveur x64 plutôt qu'un NAS tout en un. Vous aurez plus de liberté quant à son utilisation. Dans mon cas un serveur 1U sur un châssis LGX qui renferme un Core i3 et 16Go de RAM sur une carte Mini ITX où sont déployées différentes machines virtuelles pour mes tests. Juste en dessous, un serveur dans un rack 2U avec un Core i5 8400 et 32Go de DDR4 et 20To de stockage au total agrégé sous DrivePool : une superbe solution qui se rapproche dans l'idée de ce qu'on a pu connaître avec le Windows Home Server.

Le serveur héberge là encore plusieurs machines virtuelles dont le contrôleur de domaine, le serveur Constellation, le serveur de média avec DrivePool (le NAS en résumé), le serveur ZoneMinder pour les caméras, le serveur Kibana/ElasticSearch pour la boîte noire domotique, le serveur de supervision Cacti, ou de téléphonie Asterisk et j'en passe.

Côté alimentation n'oubliez pas d'ajouter un onduleur, dans mon cas des APC connectés en USB pour la supervision. Le package « BatteryChecker » sur le store Constellation vous permettra d'ailleurs d'intégrer l'état de l'onduleur dans votre système global permettant d'automatiser certaines tâches. Par exemple en cas de coupure de courant, au passage en mode batterie, l'intelligence de la maison coupera graduellement différents dispositifs hardware

(pont Hue, système sonore, caméras notamment la nuit où l'éclairage IR consomment beaucoup, bornes Wifi, etc.) ou software (arrêt des VM non critiques) pour prolonger l'autonomie de la batterie au maximum.

Mes installations domotique

Le cas d'une rénovation

Démarré fin 2017, nous avons rénové la maison familiale dans le Périgord. Seuls les murs en pierre sont conservés autrement tout est refait : toiture, dalle béton, intérieur, extérieur en passant par

l'électricité, le réseau, la plomberie, la géothermie, la climatisation, la piscine, etc.

Pour ce projet j'ai opté pour une installation IPX800. Comme expliqué précédemment, l'ensemble des câbles alimentant les points lumineux et prises contrôlées ont été ramenés au tableau pour être connectés sur les sorties d'un IPX800-V4. Pour faciliter le câblage on a un tableau dédié à la protection et un autre pour la partie IPX800. Chaque câble est connecté sur un bloc de jonction de type Phoenix Contact à la manière d'un panneau de brassage pour le réseau informatique.

N'oubliez pas que les sorties de l'IPX supportent jusqu'à 10 ampères, il faut donc, en amont de l'IPX, protéger chaque circuit avec des disjoncteurs 10A.

Pour compléter les 8 sorties et 8 entrées digitales (et 4 analogiques) de base de l'IPX800 nous avons ajouté deux modules X-8R (soit 24 circuits pilotés au total) et deux X-24D soit un total de 56 entrées pour les interrupteurs, capteurs d'énergie à impulsion, détecteurs périmétriques et volumétriques. Un X-ENO est également prévu pour des prises d'appoint sans fil et télécommandes EnOcean, un X-PWM pour la variation des spots 12v et des X-4VR pour la gestion des volets. 24

Tous les points de contrôle (interrupteur de type bouton-poussoir) sont câblés avec du câble RJ45. On peut donc remplacer certain bouton-poussoir par les X-DISPLAY offrant un écran OLED encasté dans un interrupteur avec affichage dynamique. 25

Côté informatique, l'ensemble du logement est câblé en RJ45 Cat6 S/FPT vers un panneau de brassage dans une baie réseau au niveau de l'armoire technique. Un Netgear GS724 central avec 24 ports et un deuxième switch TP-Link TL-SG108PE de 8 ports dont 4 ports PoE pour l'alimentation des bornes Wifi. Routeur Mikrotik RB3011 avec 5 bornes cAP AC en intérieur et 2 bornes wAP AC Weatherproof pour la diffusion du réseau Wifi à l'extérieur. Connectivité Internet en LTE/4G avec antenne Yagi sur le toit et connexion de backup en ADSL. Côté piscine, les deux bassins sont régulés par PoolCop et la géothermie régulée et supervisée directement sur l'IPX (contacts sec).

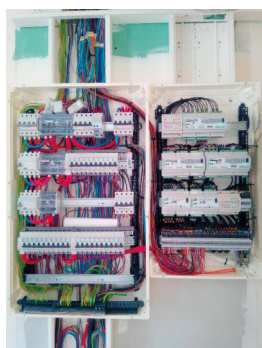
Pour finir une sentinelle Constellation a été installée sur un Raspberry Pi 3 physiquement installé dans la baie réseau. Cette sentinelle est connectée sur ma Constellation centrale hébergée dans mon logement principal que nous détaillerons dans la section suivante.

C'est donc depuis ma Constellation centrale, que je déploie sur cette sentinelle locale le package « IPX » permettant de connecter la domotique IPX800 à Constellation, le package « PoolCop » pour la piscine et le package « Mikrotik » pour le routeur.

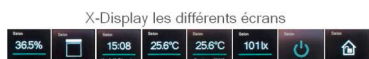
Ainsi dans ma Constellation est synchronisé en temps réel l'état de tous les capteurs, détecteurs, piscines et équipement réseau de la maison en Dordogne. Un simple code C# ou une page Javascript peut donc, en se connectant sur le bus, récupérer l'ensemble de ces variables en temps réel et invoquer des actions sur ces systèmes.

Le cas d'une maison 1930 sans rénovation

Lorsque nous avons acheté notre maison il y a environ quatre ans



24 Domotique IPX



25 X-Display

dans le Nord de la France, l'électricité venait d'être refaite quelques années auparavant tout comme l'intégralité de l'intérieur (les murs, les sols, la plomberie, etc...). De ce fait je n'ai réalisé que le tirage des câbles alarme et informatique ce qui a tout de même engendré pas mal de saignées mais rien de comparable s'il avait fallu refaire l'électricité.

On est donc ici dans le cas d'un réseau électrique très classique où les câbles en sortie du tableau sont (très souvent) mutualisés pour plusieurs dispositifs (par exemple le disjoncteur « Eclairages salon » alimente pas moins de 12 points lumineux dans le salon). **26**

L'usage des technologies filaires tel l'IPX800 est impossible. J'ai donc opté pour le Z-Wave qui consiste essentiellement à installer des micromodules derrière les interrupteurs dans les boîtes d'encastrement. Au total environ 40 devices Z-Wave essentiellement des micromodules Fibrao et prise Everspring.

Le contrôleur Z-Wave est une Vera Lite équipée avec une antenne externe bien plus performante que l'antenne intégrée. **27**

Avec son API HTTP supportant le long-pooling la Vera est la meilleure interface Z-Wave du marché pour une intégration dans un système global comme la plateforme Constellation. Par contre si vous souhaitez l'utiliser comme système central ce choix n'est pas le plus pertinent (interface et catalogue de plugin assez pauvre).

Lumières, volets et quelques prises commandées sont donc gérés par Z-Wave. Pour le reste le thermostat est un Nest, capteurs météo NetAtmo et Oregon (via un RFXCOM directement connecté à Constellation), ampoules/lampes Hue, détecteur de fumée Nest Protect, des prises Wifi, interrupteurs et télécommandes d'appoint EnOcean, etc... Sans oublier une multitude d'objets connectés home-made que nous découvrons dans la suite de ce dossier.

Comme pour la Dordogne, le cœur de réseau est un Netgear GS724T où plusieurs réseaux LAN se croisent (VLAN) avec routeur Mikrotik sur une ligne FTTH/GPON (en lieu et place de la Livebox restée dans son carton), réseau de bornes Wifi cAP AC, wAP AC et mAP Lite via CAPSMAN, serveurs de virtualisation pour héberger pas loin de 20 serveurs virtuels, cameras IP et analogiques sur un DVR Floureon, système de diffusion audio multi-room, etc.

Le serveur Constellation supervise plus de 80 packages (applicatifs) réparties sur une vingtaine de sentinelles. L'ensemble produisant plus de 650 StateObjects (objets de donnée) et exposant plus de 180 MessageCallbacks (fonctions).

Interconnecter les objets et services de la maison

Comme détaillé précédemment on prend le meilleur des technologies selon la fonction dont on a besoin : Z-Wave ou EnOcean pour la domotique sans-fil, IPX/KNX en filaire, Nest ou NetAtmo pour un thermostat ou un détecteur de fumée, etc.

Seulement pour ne pas se retrouver avec un patchwork d'objet connecté et autant d'application que de fonction, il faut centraliser et fédérer l'ensemble et c'est dans cette idée qu'a été conçue la plateforme Constellation. **28**

Constellation permet de créer une architecture décentralisée où

chaque programme, service ou objet peut communiquer avec les autres sur un bus.

S'il fallait résumer la plateforme Constellation, on pourrait dire qu'elle repose sur trois grands principes :

- **Orchestration** : permettant de déployer des programmes (services ou applications) que l'on nomme des packages sur un ensemble de devices en gérant la configuration, le versioning et les logs de manière centralisée.
- **Messaging & MessageCallback** : tout système connecté à Constellation peut envoyer et recevoir des messages en temps réel dans un langage universel. Les MessageCallbacks font l'association entre un message et une méthode (fonction) d'un programme afin de permettre à tous les intervenants d'exposer et d'invoquer des méthodes entre eux très facilement et rapidement.
- **StateObjects** : l'état d'un CPU, d'une variable, d'une lampe, d'une batterie, d'une porte, le volume de la radio, le nombre de visiteurs d'un site web, le temps de réponse d'un équipement réseau ou la température d'une chambre ou d'un disque dur, ... sont tous des StateObjects. Chaque intervenant peut interroger ou suivre en temps réel les StateObjects produits par les packages d'une Constellation.

Ainsi chaque « package » (applicatif) peut produire des données (StateObject) et exposer des fonctions (MessageCallback), et inversement invoquer des MessageCallbacks ou consommer des StateObjects.

L'API Constellation supporte le .NET, Arduino/C++, Python, Javascript ainsi qu'une interface HTTP pour les autres technologies. De ce fait tout devient possible et va au-delà des limites des solutions domotique existante : vous codez dans votre langage favori avec votre IDE favori sans être bloqué dans un modèle d'extensibilité qui n'est pas forcément le fort des plateformes domotique.

Prenons un exemple très simple : écrivons un petit algorithme dans notre langage favori, ici en C#, pour allumer la lumière lorsqu'un mouvement est détecté dans l'entrée et que la luminosité est faible. Pour cela nous pouvons utiliser le « StateObject Explorer » pour explorer l'ensemble des StateObjects produits par les packages de votre Constellation.

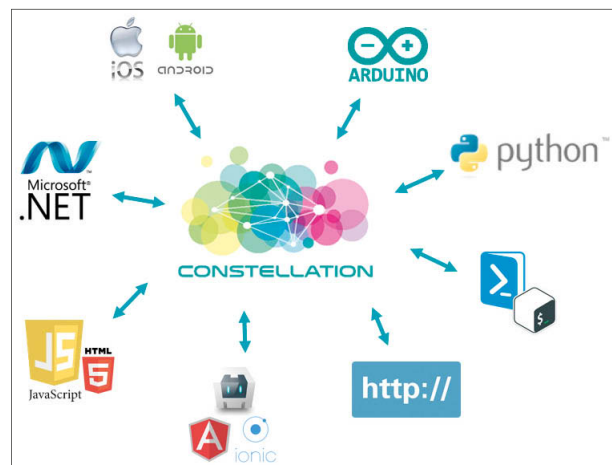
Le mouvement dans l'entrée est connu avec le StateObject nommé « ZoneInfo12 » du package « Paradox » (le package qui commu-



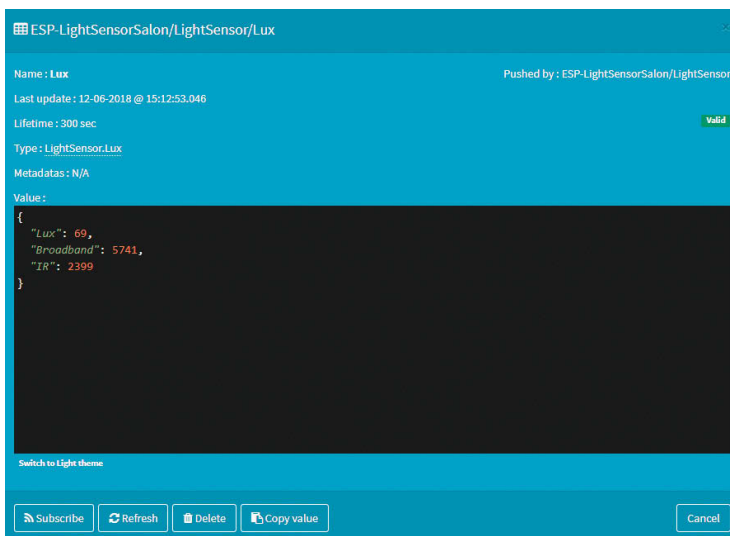
26
Micromodule Z-Wave FGD211



27 Vera Lite



28
Constellation



29 StateObject Explorer

nique avec l'interface PRT3 de l'alarme Paradox). La zone n°12 étant le détecteur de mouvements installé dans cette pièce. La valeur de ce StateObject est un objet complexe dont le type est automatiquement décrit dans Constellation. On retrouvera une propriété « IsOpen » indiquant si la zone est ouverte. La luminosité est connue avec le StateObject nommé « Lux » produit par le package « LightSensor » de la sentinelle « ESP-LightSensorSalon ». **29**

Il s'agit là encore d'un StateObject de type complexe composé de 3 propriétés du « Lux », un champ de type entier contenant la luminosité. Ce package est en réalité un ESP8266 décrit dans la prochaine section. Pour allumer la lumière de l'entrée qui est pilotée par un micromodule FGD221, on a sur le package « Vera » (connecteur avec le contrôleur Z-Wave) un MessageCallback nommé « SetSwitchState » permettant d'envoyer un ordre Z-Wave. **30**

Celui-ci prend en paramètre un objet contenant l'ID du device (information qu'on trouvera dans le StateObject représentant le micromodule Z-Wave de l'entrée) et le nouvel état (allumé ou éteint). En .NET on vient simplement déclarer des propriétés et les décorer avec l'attribut « StateObjectLink » pour injecter en temps réel, à la manière d'un framework IoC, les valeurs StateObjects produits par d'autres systèmes d'une Constellation dans votre code .NET :

```
[StateObjectLink("Vera", "Eclairage Entrée")]
public StateObjectNotifier EclairageEntree { get; set; }
```

```
[StateObjectLink("Paradox", "ZoneInfo12")]
public StateObjectNotifier MouvementEntree { get; set; }
```

```
[StateObjectLink("LightSensor", "Lux")]
public StateObjectNotifier Lux { get; set; }
```

Ainsi mes trois propriétés .NET contiendront respectivement l'état du module Z-Wave pilotant l'éclairage de l'entrée, l'état du capteur de mouvements de l'entrée, la luminosité mesurée par l'ESP/Arduino. Pour allumer ou éteindre la lumière il suffira d'invoquer le MessageCallback « SetSwitchState » en créant un « proxy dynamique » vers le package Vera.

On exécutera notre règle seulement lorsqu'il y a un mouvement dans l'entrée, c'est-à-dire quand le StateObject, ici lié à la propriété « MouvementEntree », change :

```
// Si le StateObject du detecteur Paradox change
this.MouvementEntree.ValueChanged += (s, e) =>
{
    if (this.MouvementEntree.DynamicValue.IsOpen) // Si mouvement (= zone ouverte)
    {
        // Envoyer un message au package Vera pour invoquer le MC "SetSwitchState"
        PackageHost.CreateMessageProxy("Vera").SetSwitchState(new
        {
            // ID du module de l'entrée connu avec son StateObject
            DeviceID = (int)this.EclairageEntree.DynamicValue.Id,
            // l'état de la lampe est conditionné par rapport au StateObject des lux
            State = this.Lux.DynamicValue.Lux > 20
        });
    }
};
```

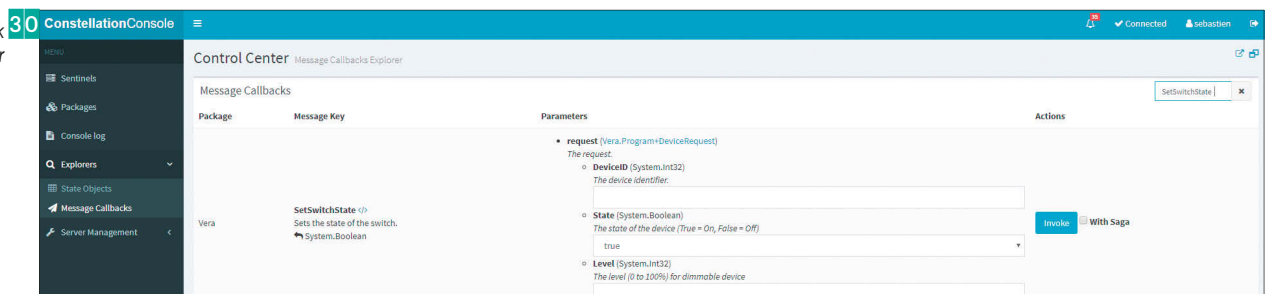
En quelques lignes de code, ici du C#, on peut facilement créer des algorithmes faisant intervenir différents systèmes ou objets, home-made ou non. L'exemple est en C# mais on aurait aussi pu le faire avec l'API Python de Constellation.

Pour plus d'informations sur la plateforme Constellation rendez-vous sur le portail développeur : <https://developer.myconstellation.io>

Dans la suite de ce dossier créons nos propres objets connectés : capteurs en tout genre, pilotage intelligent de la porte de garage, miroir connecté, automatisation de l'abri de jardin, intégration de la sonnette, etc... Nous découvrons également comment ajouter de l'intelligence dans notre maison, une introduction vers « la maison (réellement) intelligente ».

A suivre ...

MessageCallback
Explorer





Principes avancés de conception objet

Partie 1

La POO, ou programmation orienté objet, permet de représenter un concept, une idée ou une entité du monde physique par des entités appelées objets ayant des propriétés intrinsèques et exposant des opérations publiques pour les manipuler. Ainsi les objets peuvent être vus comme des briques rendant des services aux autres objets et donc réutilisables. L'interaction entre les objets via leurs relations permet de concevoir et réaliser les fonctions attendues. La conception est donc une étape importante pour modéliser les éléments du monde réel et les transcrire en code.

Cependant, la conception reste difficile dans le développement logiciel, car :

- Les principes de base de la POO que sont l'encapsulation, l'héritage et le polymorphisme ne suffisent pas à guider dans la conception
- Les design patterns qui sont des abstractions de solutions à des problèmes récurrents ne suffisent pas à former pas un tout cohérent pour la construction de design complets

Le présent document présentera quelques principes utiles en matière de conception et illustrera par des exemples dans le langage Java.

Les risques de dégénérescence de l'application

Lorsqu'une application est en PRODUCTION les phénomènes suivants sont observés pendant les activités de développement.

La rigidité

Chaque évolution risque d'impacter d'autres parties de l'application. Le coût de développement augmente et avec l'approche de la date de livraison, la qualité de code est négligée. Il fonctionne, mais le développeur ne prend pas le temps de refactorer, par conséquent le code devient difficile à modifier. Un cercle vicieux s'installe puisque le coût de modification devient élevé et le logiciel a peu de chances d'évoluer au risque d'entraîner des régressions.

La fragilité

Modifier une partie du code entraîne des erreurs dans une autre partie du logiciel qui devient peu robuste au changement avec un coût de maintenance élevé.

L'immobilité

Il est difficile d'extraire une partie de l'application pour la réutiliser. Le développeur a très vite tendance à copier/coller en modifiant les parties qui le concernent.

Les problèmes énoncés sont d'autant plus importants avec la volumétrie de l'application.

La source du problème : la gestion des dépendances

Les dégradations tirent leurs origines dans la multiplication des dépendances et de leur architecture : les modules, packages et classes finissent par dépendre les uns des autres aboutissant au code spaghetti.

Les objectifs de la conception

Les modifications de code sont inévitables avec l'évolution des besoins. La conception d'un logiciel a un pour objectif d'amortir l'impact des dépendances et à aboutir aux qualités recherchées de :

- robustesse : les changements n'introduisent pas de régression
- extensibilité : l'ajout de fonctionnalité doit être facile
- réutilisabilité : il est possible de réutiliser certaines parties de l'application pour en construire d'autres.

Les principes présentés ci-dessous doivent aider à éviter les phénomènes de rigidité, fragilisée et d'immobilité énoncés plus haut. Ainsi l'application sera capable de s'adapter au changement

Principes SOLID pour l'organisation des classes

Single Responsibility Principle : principe de responsabilité unique (S) :

Ce principe stipule qu'il ne doit avoir qu'une raison et une seule raison de modifier une classe/ un module. Il doit cadrer pour en donner une définition de la responsabilité et les indications sur la taille d'une classe.

Un nom simple donné à une classe est un indicateur du principe de responsabilité. Si le développeur éprouve des difficultés à nommer alors il est fort probable que la classe endosse trop de responsabilités. Le respect de ce principe augmente par ailleurs la cohésion de la classe qui est un concept indiquant le degré d'interdépendance entre variables d'instance et méthodes.

Ce principe simple d'énoncé est pourtant celui qui est le plus transgressé lors du développement logiciel. En effet, lorsqu'une évolution est demandée l'impact logiciel a été analysé et la modification se fait rapidement pour faire fonctionner le logiciel sans considérer un refactoring pour mieux organiser le code et le maintenir propre. A force le module continue de grossir et augmente en responsabilité.

Quand appliquer : systématiquement

Le maintien d'une haute cohésion donne un logiciel avec de nombreuses petites classes. Il n'y a pas plus de parties entre un logiciel fait de nombreuses petites classes et un logiciel avec peu de grandes classes puisqu'il y a autant de parties. Le développeur devra chercher s'il existe une fonction dont il a besoin et il est indiscutable qu'il est préférable de chercher dans une boîte à outils avec plusieurs compartiments étiquetés plutôt qu'un sac avec tous les outils en vrac dedans.

Comment appliquer :

- est-ce que le nom donné à la classe est facile ?
- si cette fonctionnalité est ajoutée dans cette classe existante, est-ce que cela donne de nouvelles responsabilités à la classe si elle doit être modifiée dans le futur ? **1**

Nous pouvons voir dans la classe Person qu'elle possède 2 responsabilités : des données et opérations propres à une personne, mais aussi pour un emploi.

Nous pourrions imaginer que la classe Person contienne aussi des informations/méthodes sur la maison où elle vit (avec une adresse, rue, ville, pays), une voiture qu'elle possède (marque, modèle, numéro de plaque d'immatriculation). Tout ceci alourdirait la classe Person et ferait supporter trop de responsabilités.

Il est plus efficace de refactorer de sorte qu'il y ait 2 classes distinctes. **2**

**Open/close principe :
principe d'ouverture/fermeture :**

La rigidité et la fragilité vues précédemment viennent de l'impact d'un changement d'une partie de l'application sur d'autres parties avec des effets indésirables. Bertrand Meyer, créateur du langage orienté objet Eiffel, a stipulé que tout module (package, classe, méthode) doit être :

- ouvert aux extensions : on peut ajouter des fonctionnalités non prévues à la création
- fermée aux modifications : les changements introduits ne modifient pas le code existant

Autrement dit, l'extensibilité se traduit par de l'ajout de code uniquement. Une fois le code produit, testé unitairement, qualifié par des procédures et enfin livré en production, le seul moyen de modifier est d'étendre le code pour s'assurer que le code existant ne sera pas altéré au risque d'entraîner des régressions.

L'abstraction et polymorphisme sont les moyens pour y parvenir en faisant reposer le code stable sur une abstraction d'une entité variable pouvant être amenée à évoluer.

Utilisation de la délégation abstraite

Soit une classe Geometry qui permet de calculer l'aire et le périmètre d'une figure géométrique comme dans le schéma suivant et pseudo-code ci-dessous : **3**

```
class Geometry {
    void area(Shape shape) {
        if (shape instanceof Circle) {
            // calculate for circle
        }
        else if (shape instanceof Square) {
            // calculate for square
        }
    }

    void perimeter(Shape shape) {
        if (shape instanceof Circle) {
            // calculate for circle
        }
        else if (shape instanceof Square()) {
            // calculate for square
        }
    }
}
```

La classe Geometry gère les 2 cas des formes Circle et Square. Si une nouvelle forme doit être ajoutée, il faut modifier Geometry en ajoutant un 3e bloc de test if sur la nouvelle forme, ce qui violerait le principe d'OCP. Or il est préconisé d'ajouter du code.

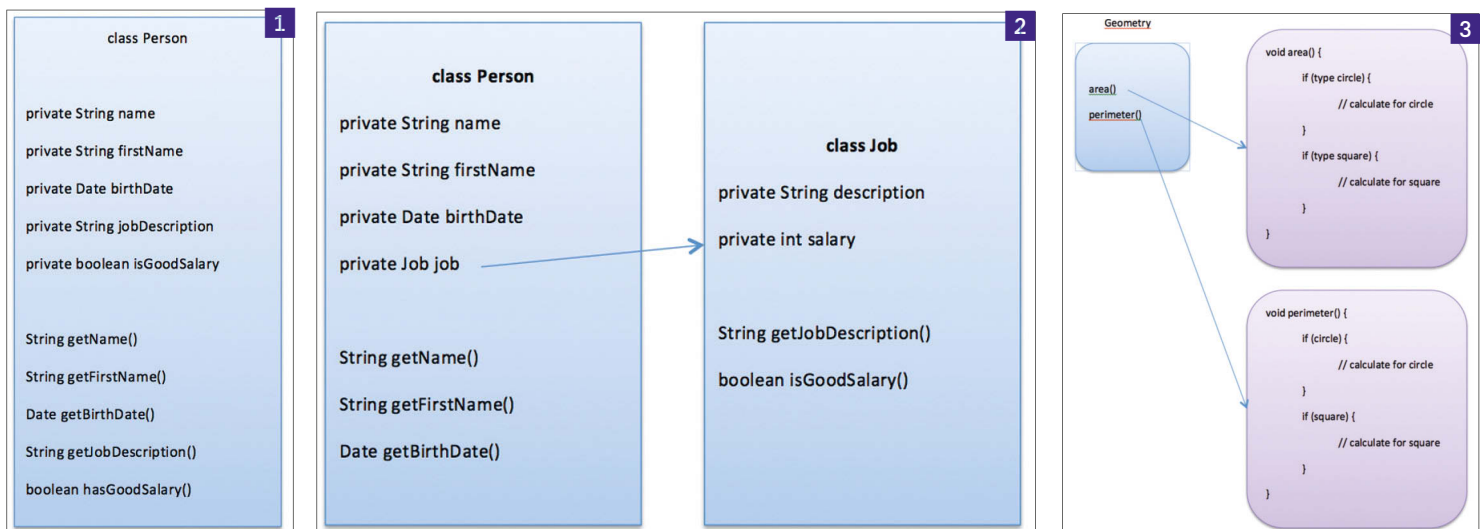
Pour respecter l'ouverture/fermeture, Geometry va s'appuyer sur une interface Shape avec une implémentation correspondant à chaque forme. Geometry délègue à l'interface Shape le traitement.

4 5

Puisque Geometry dépend de Shape, il devient alors possible d'ajouter une nouvelle forme Rectangular sans modifier Geometry en créant une classe implémentant Shape.

Le principe se retrouve dans quelques design patterns :

- Strategy : le code qui doit être ouvert/fermé travaille avec plusieurs algorithmes possibles sans impacter le code client.
- Abstract factory : une classe qui sert de fabrique d'instances d'un certain type
- Template method : la structure générale d'une méthode est fermée, mais certaines sous parties peuvent être ouvertes pour des implémentations spécifiques.



Quand appliquer :

L'OCP est incontournable pour rendre le code flexible. L'erreur classique consisterait à ouvrir/fermer systématiquement toutes les classes de l'application. En effet, tout n'est pas sujet à la flexibilité et cela complexifierait le code. Ce qui rend néfaste d'autant plus que la flexibilité recherchée n'est pas entièrement exploitée. Il convient d'étudier les points d'ouverture/fermeture :

- quand des algorithmes divers interviennent et sont exprimés par le client
- en fonction des besoins de flexibilité pressentis par le développeur
- à mesure des changements répétés constatés au cours du développement

Comment appliquer :

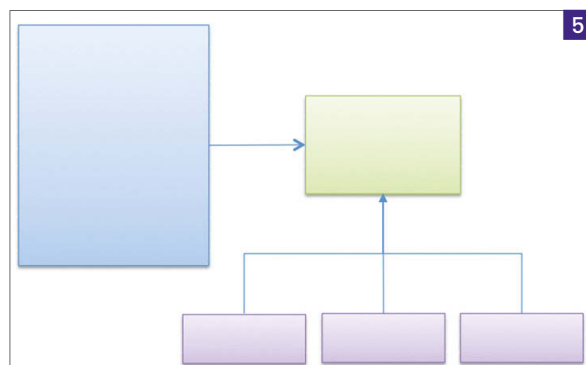
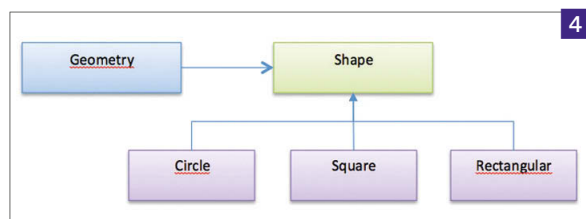
Utiliser les design patterns cités plus haut pour faire une délégation abstraite.

Exemple de cas concret :

On peut imaginer un traitement de génération de rapport après extraction des données. Concrètement une application est amenée à produire de tels rapports pour aider les décideurs dans la prise de décision : rapport Excel, PDF, fichiers plats, XML.

L'algorithme peut être schématisé comme suit :

```
class BankAccountProcessor {
    public void extractAndGenerateReport(Form form) {
        Result res = search(form);
        if (form.isExcel()) {
            generateExcel(res);
        } else if (form.isPdf()) {
            generateExcel(res);
        } else if (form.isXml()) {
            generateXml(res);
        }
    }
}
```



Dans l'exemple ci-dessus, 3 types de rapports sont implémentés. Ajouter un autre type impliquerait donc de modifier la méthode **extractAndGenerateReport** en ajoutant un autre bloc de test. Il est judicieux d'identifier que le type de rapport doit être une abstraction par exemple ReportSupport avec une méthode **generate(Result res)** auquel on déléguera la suite avec 3 implémentations possibles Excel, PDF et XML. 6

Code de la classe de service de génération de rapport

```
class BankAccountProcessor {
    private ReportSupport delegate;

    public void extractAndGenerateReport(Form form) {
        Result res = search(form);
        delegate = getExtractor(form);
        delegate.generate(res);
    }

    private ReportSupport getExtractor(Form form) {
        // find the right implementation according to support choice
    }
}
```

Code de la classe abstraite et des 3 implémentations

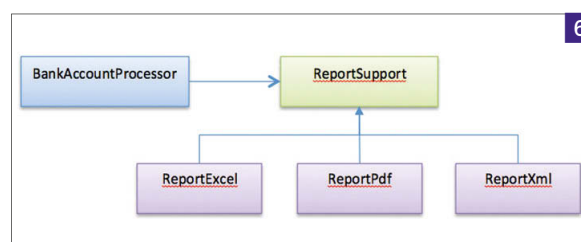
```
abstract class ReportSupport {
    public abstract void generate(Result res);
}
```

```
class ReportExcel extends ReportSupport {
    public void generate(Result res) {
        // generate in Excel
    }
}
```

```
class ReportPdf extends ReportSupport {
    public void generate(Result res) {
        // generate in PDF
    }
}
```

```
class ReportXml extends ReportSupport {
    public void generate(Result res) {
        // generate in XML
    }
}
```

La suite dans *Programmez ! 221*.





Guillaume ANDRÉ
CTO Wexperience & UX.Care
UX.Care est une solution en ligne dédiée à la mesure et à la compréhension de l'expérience utilisateur sur votre site web ou mobile.



Progressive Web App (PWA)

Le fichier Web App Manifest

Partie 1

Sans aucun doute la première porte d'entrée pour tout développeur web qui s'intéresse au domaine des Progressive Web Apps (PWAs), la fonctionnalité Web App Manifest permet à tout un chacun de transformer son site internet, son blog, son site e-commerce, ou sa single-page application (SPA) en Progressive Web App. C'est le fichier Web App Manifest qui va guider le navigateur en lui spécifiant les points d'entrées et les points clés de la PWA. L'une des fonctionnalités phares et par défaut liée à l'usage des Web App Manifest est la capacité de rendre un site "installable" sur l'écran d'accueil du smartphone de l'utilisateur et ce, au même niveau qu'une application native. On parle ici de la fonctionnalité A2HS pour Add 2 Home Screen aussi appelée Web App Install Banners côté Android.

Web App Manifest, les présentations Un simple fichier

Le Web App Manifest est un fichier JSON ayant pour extension `.webmanifest` qui permet aux développeurs de centraliser les informations relatives à la PWA. Pour le navigateur, c'est uniquement ce fichier qui permet de faire la distinction entre un site web classique et une PWA, et ce, à condition qu'il soit référencé. Ce fichier permet de préciser des méta-données comme par exemple, l'icône, le nom, la catégorie et la description de la PWA mais aussi certains comportements spécifiques comme le paramétrage de l'affichage de la PWA au sein de l'environnement hôte, la gestion d'un mode offline et des coupures de réseaux, la possibilité de traiter ou non des notifications push, etc. Ce fichier doit-être de type MIME `application/manifest+json` et respecter un certain format (schéma) pour être interprété correctement par le navigateur.

Après avoir déposé le fichier Manifest à la racine du dépôt ou dans un dossier (au même titre qu'un fichier lambda faisant partie de la solution de votre site), il sera nécessaire de renseigner une balise `link manifest` dans votre page web (`index.html` par exemple) juste avant la fermeture du `</head>` afin de référencer le fichier Manifest au sein du document. Cela dans le but de prévenir le navigateur qu'il est en présence d'une PWA et qu'il va devoir traiter ce fichier Manifest. A noter que l'utilisation de HTTPS sur votre site est un prérequis.

```
<link rel="manifest" href="/manifest.webmanifest">
```

Nous verrons par la suite que certaines fonctionnalités avancées ne se limitent pas à ce fichier Manifest et nécessitent l'exécution d'un proxy appelé Service Worker pour fonctionner.

Exemple de Manifest.webmanifest

```
{
  "short_name": "WAM",
  "name": "Web App Manifest",
  "theme_color": "#eb5252",
  "background_color": "#000000",
  "display": "fullscreen",
  "Scope": "/",
  "orientation": "portrait",
}
```

```
"icons": [
  {
    "src": "images/android/android-launchericon-48-48.png",
    "type": "image/png",
    "sizes": "48x48"
  },
  {
    "src": "images/android/android-launchericon-96-96.png",
    "type": "image/png",
    "sizes": "96x96"
  },
  {
    "src": "images/android/android-launchericon-192-192.png",
    "type": "image/png",
    "sizes": "192x192"
  }
],
"start_url": "index.html?utm_source=homescreen"
}
```

Un référentiel ?

Pour bien comprendre la construction et le formatage d'un fichier Web App Manifest un schéma référentiel (non officiel) est disponible à cette adresse : <http://json.schemastore.org/web-manifest>. Cela permet de prendre connaissance du nom des propriétés, des types attendus et aussi d'accéder à une liste précise des *enums* utilisables. Mieux vaut connaître tous les cas possibles avant de se lancer. Aussi vous retrouverez toutes les spécifications concernant les Web App Manifest (encore au stade de Draft) sur le site du W3C (<https://www.w3.org/TR/appmanifest/>). Nous verrons par la suite qu'il existe aussi un validateur.

Liste non-exhaustive des propriétés et fonctionnalités

- **name** : le nom de la PWA retrouvable au sein de l'environnement ayant installé la PWA ;
- **short_name** : le nom utilisé par l'environnement juste en dessous de l'icône de la PWA ;
- **start_url** : il faut renseigner l'url de démarrage de la PWA. Une

des bonnes pratiques consiste à tracker cette url en spécifiant une query string notamment avec les paramètres UTM et l'outil Campaign URL Builder de Google (<https://ga-dev-tools.appspot.com/campaign-url-builder/>) ;

- **display** : permet de spécifier comment la PWA va être présentée. *fullscreen* permet d'ouvrir l'application en plein écran, *stand-alone* de garder la statusbar et le bouton back, *minimal-ui* avec aucun bouton et *browser* pour une vue par défaut (navigateur classique). Cette fonctionnalité permet aussi de filtrer certains périphériques qui ne sauraient pas gérer le mode d'affichage renseigné. *fullscreen* et *minimal-ui* ne sont pas encore supportés par Safari ;
- **orientation** : permet de préciser le type d'affichage pour la PWA, à savoir portrait, paysage ou les deux ;
- **background_color** : force la couleur du background avant le chargement des CSS. On voit le `background_color` généralement au lancement de la PWA sur une très courte durée ;
- **scope** : par défaut c'est la `start_url`, il est possible de renseigner par exemple un `scope` spécifique pour chaque type de Service Workers ;
- **icons** : peut contenir une collection d'icônes avec différentes tailles ;
- **serviceworker** : permet de renseigner un Service Worker, mais il est coutume d'enregistrer et d'instancier directement un Service Worker via JavaScript au sein de l'application.

La liste exhaustive est accessible sur le Draft W3C officiel : <https://www.w3.org/TR/appmanifest/#webappmanifest-dictionary>

Les différents acteurs n'hésitent pas à apporter des fonctionnalités propriétaires dans ce fichier Manifest, il convient donc de bien faire attention aux propriétés utilisées afin de connaître leur impact sur chaque plateforme.

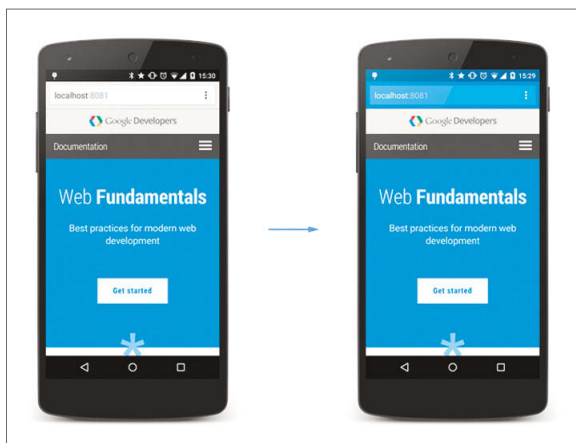
Plus de personnalisation

Pour aller plus loin dans la personnalisation et l'identité visuelle de votre PWA, les navigateurs proposent la gestion de balises link et meta spécifiques. Leur principal but est d'améliorer l'expérience utilisateur en modifiant les couleurs de l'environnement, la qualité des icônes, voire même de modifier le rendu de la status bar ou l'écran de chargement sur iOS. Voici un exemple non exhaustif de ces balises :

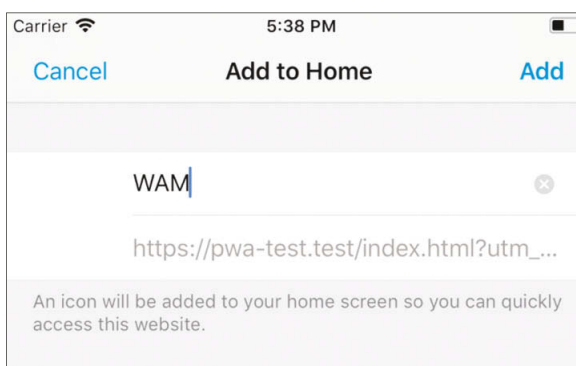
```
<!-- Chrome -->
<link rel="icon" sizes="192x192" href="icon.png"><link rel="apple-touch-icon"
href="ios-icon.png">
<!-- Safari -->
<link rel="apple-touch-startup-image" href="icon.png">
<!-- IE -->
<meta name="msapplication-square310x310logo" content="icon_targetile.png">
<!-- Chrome -->
<meta name="theme-color" content="#4285f4">
<!-- Safari -->
<meta name="apple-mobile-web-app-status-bar-style" content="black">
```

C'est Safari qui est le plus concerné par cette fonctionnalité. Safari se sert principalement des balises pour palier sa "discutable" gestion du fichier Web App Manifest. Consulter la documentation officielle pour une liste exhaustive des possibilités sur Safari : <https://developer.apple.com/library/content/documentation/AppleApplications/Reference/SafariWebContent/ConfiguringWebApplications/ConfiguringWebApplications.html>

Dans une optique d'automatisation, PWACOMPAT (<https://github.com/GoogleChromeLabs/pwacompat>) est une librairie qui permet de générer ces balises link et meta. Aussi, elle permet de pro-



1 Exemple de personnalisation sur Chrome avec la balise meta theme-color



2 Icône blanche avec Safari sur écran A2HS

poser une version dégradée du Manifest pour les anciens devices qui auraient des versions de Webkit ou Chrome non à jour ou qui ne supporteraient pas les PWAs. Attention, depuis la version 11.3 d'iOS, il est possible que l'icône de l'application ne se charge pas en local. 2. Cela est sûrement lié à un problème de certificat SSL erroné lié à votre environnement de développement. En publiant avec un bon certificat ou en spécifiant une url absolue, il est possible de régler le problème.

Les Service Workers

Dans le cadre d'une PWA, notifications push, navigation hors connexion, synchronisation en arrière plan, capture d'écrans sont les principales fonctionnalités dites "service workers dépendantes". Un Service Worker est un composant technique qui se positionne entre l'application et le navigateur (ou le réseau—semblable à un proxy). Il fonctionne de manière événementielle, et peut fortement modifier la manière dont une application se comporte (cache, rendu, offline, etc.). C'est le Web App Manifest qui permet d'enregistrer les différents Service Workers au sein d'une PWA et donc d'adresser les différentes fonctionnalités exposées ci-dessus.

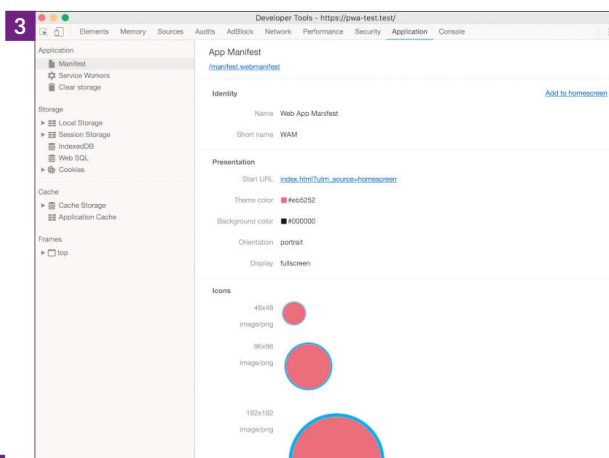
Des outils pour accélérer l'intégration d'un Web App Manifest

Les navigateurs :

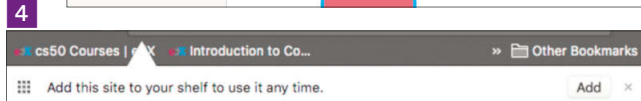
Les navigateurs et leurs outils de développement n'ont pas tous un traitement spécifique concernant le fichier .webmanifest.

- **Google Chrome** : les outils de développement Google (Developer

Developer Tools >
Application > Manifest sur
Google Chrome



Exemple du comportement
d'ajout d'une application
Google Chrome



Tools) permettent de vérifier la bonne prise en charge d'un fichier Manifest au sein d'une page web. *Clic droit > Inspecter > Application > Manifest* **3**

On remarquera le bouton *Add to homescreen* qui permet aussi de créer une Application Google Chrome (*Home > Applications > Applications Chrome*) sur base du fichier Manifest. En cliquant sur ce bouton A2HS, le processus d'installation commence : **4**

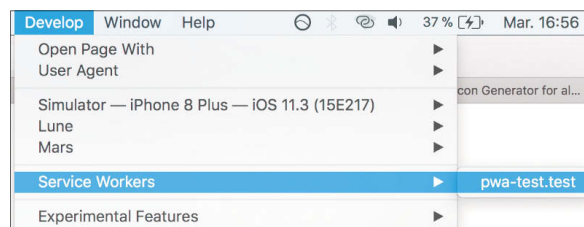
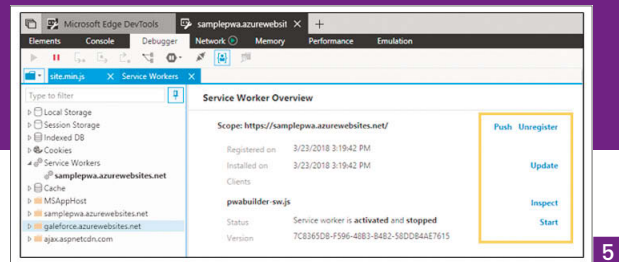
L'onglet Service Workers sera aussi utile. Pour accéder aux Service Workers déjà installés taper *chrome://serviceworker-internals/* dans la barre de recherche.

- **Microsoft Edge** : pour la plateforme Windows, il faudra utiliser Microsoft Edge DevTools Preview, ou, par défaut, la console développeur via *Inspecter* au sein de Edge. L'avantage de cet exécutable non lié au navigateur est qu'il offre la possibilité de debugger en remote : <https://www.microsoft.com/fr-fr/store/p/microsoft-edge-devtools-preview/9mzbfrnz0mnj?activetab=pivot%3aoverviewtab> **5**
- **Firefox** : lui aussi propose via les outils développeurs la possibilité de debug. Ça se trouve dans le menu *Développement Web > Outils de développement*. Il faudra taper *about:debugging* dans la barre de recherche pour aller plus loin et accéder aux Service Workers installés.
- **Safari** : Safari Technology Preview (<https://developer.apple.com/safari/technology-preview/>) vous permettra d'accéder à la totalité des fonctionnalités implémentées par Apple. A aujourd'hui, la version de Safari (11.0.3) ne permet pas de travailler sur les PWAs. La version Tech Preview permet en effet d'accéder aux Service Workers de votre PWA et aux fonctionnalités dites expérimentales. **6**

Un validateur et une extension :

Il est possible de vérifier qu'un fichier Web App Manifest est bien formaté. La spécification étant encore au stade de Draft, le validateur officiel W3C n'est pas disponible. Cependant, il existe un validateur disponible à cette adresse : <https://manifest-validator.appspot.com/>. Il conviendra de valider votre fichier pour vérifier sa conformité.

Google propose "Lighthouse" (extension Chrome) qui permet de dresser un audit de votre site et de votre Progressive Web App. Le but est d'atteindre le score de 100. Le Manifest sera en partie traité par cet outil (<https://developers.google.com/web/tools/lighthouse/>). Plus



6 Menu Develop dans Safari Tech Preview

généralement Lighthouse dresse la checklist concernant les PWAs (<https://developers.google.com/web/progressive-web-apps/checklist>).

Générer son Web App Manifest, c'est possible !

La génération d'un Web App Manifest pouvant être assez chronophage, il existe des générateurs sur la toile. Se tourner vers ce type de solution vous permet de gagner du temps, il suffit de se laisser guider.

- <https://www.pwabuilder.com/> et <http://appimagegenerator-prod.azurewebsites.net/> pour les images (resizing)
- <https://app-manifest.firebaseapp.com/>
- <https://tomitm.github.io/appmanifest/>

Dans cette même optique de génération, il existe aussi des packages NPM et des plugin Webpack (<https://github.com/arthurbergmz/webpack-pwa-manifest>) qui, dans un objectif d'automatisation et d'intégration continue notamment sur la partie Images & Icons sont compétitifs. En effet, ce type de plugin vous permet par exemple de générer sur base d'un tableau paramétrable toutes les tailles d'icônes nécessaires, ce qui vous évitera bien des manipulations au sein de votre éditeur d'images préféré...

```
icons: [
  {
    src: path.resolve('src/assets/icons/ios-icon.png'),
    sizes: [120, 152, 167, 180, 1024],
    destination: path.join('icons', 'ios'),
    ios: true
  },
  {
    src: path.resolve('src/assets/icons/ios-icon.png'),
    size: 1024,
    destination: path.join('icons', 'ios'),
    ios: 'startup'
  },
  {
    src: path.resolve('src/assets/icons/android-icon.png'),
    sizes: [36, 48, 72, 96, 144, 192, 512],
    destination: path.join('icons', 'android')
  }
]
```

Pour les utilisateurs de VS Code il existe une extension PWA Tools qui permet d'ajouter des snippets (push, notifications, etc) et des modèles de Manifest :

<https://marketplace.visualstudio.com/items?itemName=johnpapa.pwa-tools> •

La suite dans le N°221



Adrien Clerbois
Microsoft MVP VSDT,
Lead Designer @ Ingenico ePayments
@AClerbois



À la conquête du global tool dans .NET Core

Vous le savez peut-être déjà, le .NET Core nous permet de créer et d'exécuter des applications à partir d'une multitude de plateformes. Il peut se révéler difficile de les distribuer, surtout, dans le domaine des applications console.

Si l'on observe ce qui existe dans le monde open source et notamment sous NodeJS, on peut télécharger et installer assez aisément une application. Ensuite, elle est accessible automatiquement depuis n'importe quel dossier du PC.

Voici un exemple avec node :

```
$ npm install -g myLibrary
$ myLibrary
```

À partir de la version du NET Core SDK 2.1, disponible depuis le 30 mai 2018, Microsoft nous offre une expérience similaire :

```
$ dotnet install tool -g myLibrary
$ myLibrary
```

Avec la première ligne de commande, l'application est téléchargée et installée dans un espace commun. Une fois l'application installée, grâce à la seconde commande, vous exécutez l'application console depuis n'importe quel endroit, dans mon exemple myLibrary.

Hey Jamy, mais comment ça fonctionne ?

Cette fonctionnalité permet d'exposer des applications .Net Core et plus spécifiquement des netcoreapp en version 2.1 de type exécutable. Le résultat est encapsulé dans un paquet NuGet et peut être distribué sur un serveur public (tel que nuget.org), ou un espace privé (Visual Studio Online avec l'extension Packages, myget,...) ou même dans un dossier local ou un répertoire distant.

Pour pouvoir utiliser cet exemple, je vous recommande de créer un répertoire pour contenir vos paquets générés. Choisissez ensuite si vous souhaitez contenir les paquets dans l'espace de travail (dossier de votre solution/projet) ou plus globalement dans votre ordinateur.

Voici la manière de procéder pour rester dans l'espace de travail :

- Créez un dossier à la racine de votre espace de travail, dans mon exemple nupkgs ;
- Créez un fichier de configuration : dotnet new nugetconfig ;
- Dans ce fichier de configuration, ajoutez la ligne qui définit votre répertoire :

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <packageSources>
    <add key="local-packages" value="./nupkgs" />
  </packageSources>
</configuration>
```



```
</packageSources>
</configuration>
```

Don't panic : ce n'est pas compliqué !

Maintenant que notre environnement est mis en place, partons en quête de résultats : créons notre application console :

```
$ dotnet new console -o aclerbois.sayhello.programmez
```

Cette commande crée un nouveau dossier nommé aclerbois.sayhello.programmez et y ajoute un projet de type console applicative .NET Core. Un fichier program.cs et aclerbois.sayhello.programmez.csproj a été ajouté.

Commençons par modifier le fichier program.cs :

```
static void Main(string[] args)
{
    Console.WriteLine("Salut les lecteurs du magazine Programmez!");
}
```

Marquez l'application comme un outils

Au niveau de la configuration de notre projet, nous allons définir l'application comme un outil. Rendez-vous dans le fichier aclerbois.sayhello.programmez.csproj et configurez le projet comme suit :

```
<Project Sdk="Microsoft.NET.Sdk">

  <PropertyGroup>
    <OutputType>Exe</OutputType>
    <TargetFramework>netcoreapp2.1</TargetFramework>
    <PackAsTool>true</PackAsTool>
    <ToolCommandName>helloprogrammez</ToolCommandName>
    <PackageOutputPath>../nupkgs</PackageOutputPath>
  </PropertyGroup>
</Project>
```

```
<Version>1.0.0</Version>
</PropertyGroup>

</Project>
```

Vous êtes certainement familier des éléments `OutputType`, `Version` et `TargetFramework`, mais vous vous demandez quels sont ces nouveaux éléments :

- `PackAsTool` : ceci indique à la commande `dotnet pack` d'empaqueter correctement notre application pour qu'elle soit installable en tant qu'outil.
- `ToolCommand` : ceci permet de choisir le nom de l'exécutable en ligne de commande pour l'application. Grâce à ce paramètre vous pourrez écrire `helloprogrammez` au lieu de `aclerbois.sayhello.programmez`.
- `PackageOutputPath` : ceci indique à la commande `dotnet pack` où placer le résultat `.nupkg`. Cette propriété n'est pas nouvelle et existe déjà dans l'outillage publié. Pour notre exemple, il doit pointer vers le chemin que nous avons configuré dans le fichier `nuget.config`.

Note

Si vous souhaitez utiliser le mot `dotnet` devant l'application, on peut utiliser la valeur `dotnet-helloprogrammez`.

La commande pour appeler la console sera :

```
$ dotnet helloprogrammez
```

Et maintenant... ?

Nous sommes prêts à compiler et empaqueter notre application. La commande `dotnet pack` va compiler les binaires si ce n'est déjà fait et l'empaqueter :

```
$ dotnet pack -c Release
...
Successfully created package
'C:\repos\clitools\nupkgs\aclerbois.sayhello.programmez.1.0.0.nupkg'.
```

Il ne nous reste plus qu'à installer le paquet généré :

```
$ dotnet install tool -g aclerbois.sayhello.programmez
```

If there were no additional instructions, you can type the following command to invoke the tool: `helloprogrammez`
 Tool '`aclerbois.sayhello.programmez`' (version '`1.0.0`') was successfully installed.

Pour jouer le jeu, ouvrez une nouvelle invite de commande et tapez le mot magique :

```
c:\temp>helloprogrammez
Salut les lecteurs du magazine Programmez!
```

Désinstaller une application

C'est très simple de désinstaller une application CLI. Vous devez utiliser la commande `dotnet uninstall tool [nom du paquet]`.

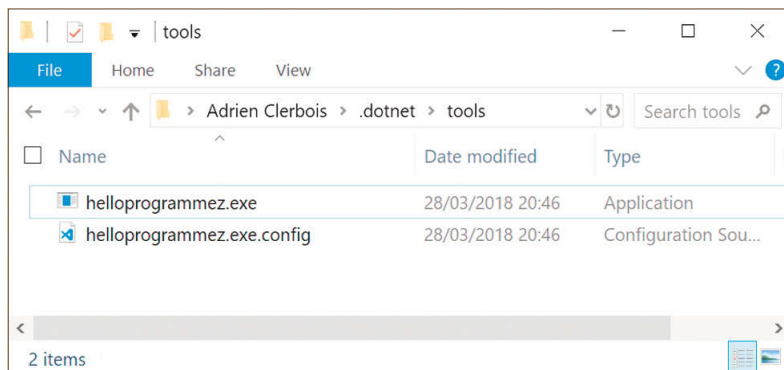
Dans notre cas :

```
$ dotnet uninstall tool -g aclerbois.sayhello.programmez
```

Les détails font la perfection, et la perfection n'est pas un détail, Léonard de Vinci

Intéressons-nous à la commande `dotnet install tool`.

Il est permis de spécifier la version du paquet que l'on souhaite installer à l'aide de l'argument CLI `--version`.



1

```
$ dotnet install tool -g aclerbois.sayhello.programmez --version 1.1.0
```

Vous pouvez également installer l'application uniquement dans le contexte de votre répertoire de travail. Pour cela, n'indiquez pas le paramètre `-g` (ou `--global`)

```
$ dotnet install tool aclerbois.sayhello.programmez
```

Si vos paquets applicatifs se trouvent sur un répertoire distant, utilisez l'argument `--source` accompagné de l'endroit où se situent vos paquets.

Welcome home

Notre outil `helloprogrammez` est installé dans le dossier `.dotnet\tools` du répertoire de l'utilisateur : 1

La commande `install` génère un fichier.exe comme un wrapper sur Windows et des scripts shell sur MacOS / Linux. Sous Windows, il s'agit actuellement d'un exécutable .NET Framework, mais il est prévu de le remplacer par un exécutable natif dans une version à venir.

Les sources de mon exemple se trouvent sur :

<https://github.com/aclerbois/sayhello.programmez>

Sky's the limit

(Enfin sauf pour Elon Musk)

Aujourd'hui, le nombre de ressources disponibles sur le répertoire public de npm est d'environ 600 000 paquets. La diversité de fonctionnalités proposées par la communauté est impressionnante. L'arrivée de ce nouvel apport du SDK et les possibilités offertes par le .NET Core sont sans limites. J'ai l'espoir que le nombre de package NuGet.org va augmenter de façon exponentielle avec plein de nouveaux outils qui nous permettront d'accélérer notre façon de travailler.

PS. Merci à mes collègues (Denis, Noël, Philippe, Charly), mon ami Denis et surtout ma petite femme Anaïs pour leurs relectures et recommandations.



Fouad JADOUANI
Développeur
@Société Générale



Jordan NOURRY
Développeur
@La combe du lion vert

Comment migrer de AngularJS vers Angular sans douleur



Le monde du développement web évolue très vite et il est parfois compliqué de s'y retrouver dans la pléthore de Frameworks JavaScript disponibles. Quand on effectue une recherche, on tombe rapidement sur un des frameworks phares de ces dernières années, propulsé par Google : Angular.

Des milliers de projets ont passé le cap et ont démarré sur la première version d'AngularJS. Quelques années plus tard Angular n'a pas dérogé à la règle du monde du web et a beaucoup évolué et gagné en maturité. Nous en sommes à la 6e version et autant dire qu'elle n'a pas grand-chose à voir avec sa première forme. Pour autant, des milliers de projets sont en production avec la première version d'AngularJS. Si vous travaillez encore avec, vous devez déjà savoir que le Framework ne sera plus maintenu dans quelques temps.

De ce fait, quels sont les bénéfices des dernières versions et pourquoi sont-elles apparues ? Quelles problématiques viennent-elles régler ? Ces problèmes sont-ils les vôtres ? Quels seront l'impact et l'effort à fournir pour migrer vers la dernière version ? Dans cet article, nous allons répondre à toutes ces questions et expliquer le processus de migration le plus pertinent, ainsi que les différentes stratégies à suivre, pour le succès de celui-ci.

Petit retour sur les apports d'AngularJS

AngularJS a été l'un des Frameworks les plus tendance durant ces dernières années et tout le monde a sauté sur ce Framework hybride entre HTML et JavaScript pour simplifier la construction de ses applications web. La tendance AngularJS et Angular est toujours en cours. À l'époque AngularJS n'était pas le seul Framework « front-end » disponible. Alors, pourquoi cette ruée vers l'or ? Visiblement, AngularJS a su trouver la formule secrète pour répondre aux besoins des entreprises et des développeurs.

Voici quelques éléments qui ont favorisé son succès :

SÉCURITÉ

Grâce à ses patches de sécurité, AngularJS permet aux applications d'être toujours à jour face aux failles de sécurité du web. AngularJS vient avec des stratégies prédéfinies afin de se prémunir contre certaines attaques (XSRF/CSRF, Json Hijacking, XSS...) lors des appels HTTPS vers le serveur.

UI DECLARATIVES

AngularJS vient survitaminé le code HTML grâce à de nouveaux attributs de balise. En effet, se baser sur le HTML est l'assurance d'être vite pris en main par tous les développeurs « front ». De plus, l'intégration de bibliothèques tierces dans une application AngularJS est une chose plutôt facile à faire, ce qui est un point essentiel pour les entreprises qui souhaitent intégrer leurs nouvelles applications dans le cadre et les outils existants.

MOINS DE CODE QU'EN NATIF

Par ailleurs, il permet d'écrire moins de code qu'avec les autres Frameworks. Ainsi, les développeurs ont besoin de moins de temps pour développer une fonctionnalité, ce qui représente un gain non négligeable.

Les principaux défauts d'AngularJS

AngularJS a beaucoup de points positifs et c'est pour cette raison que Google continue d'investir sur cette technologie. AngularJS a su répondre à de nombreuses problématiques par le passé, mais de nouveaux défis se présentent constamment. C'est pour cette raison que nous en sommes à la version 5 d'Angular et que la 6e est en bêta. Dès que nous développons autre chose qu'une « to do » liste, cela devient très frustrant et nous laisse à penser que l'architecture a été mal conçue derrière ce Framework performant.

DATA BINDING BIDIRECTIONNEL

Le Data binding n'est rien d'autre que la **synchronisation des données entre le modèle et la vue**. Avantage de ce mécanisme : si la valeur change depuis UI, le modèle sera mis à jour et vice-versa. En revanche, le développeur lutte considérablement contre le problème de performance lié à ce mécanisme dans AngularJS. Ce problème influe notamment sur l'affichage, et le Framework souffre de défauts de scalabilité, laissant donc entrevoir que quelque chose ne va pas au niveau fondamental.

Dans Angular, ce concept est plus performant, mais il serait bien trop long et complexe à explorer dans cet article, vous trouverez plus d'informations, si cela vous intéresse, sous ces liens(1).

INJECTION DE DÉPENDANCE

L'injection de dépendance d'AngularJS est une injection par nom passé en argument. Le problème dans cette syntaxe est que cela ne fonctionne plus après une minimisation du code parce que les variables sont injectées par nom. Depuis, de nouvelles méthodes d'injection existent, mais le fait d'avoir un Framework qui offre plusieurs possibilités pour réaliser une même action complique le choix. Sur ce point, contrairement à AngularJS qui ne propose qu'un seul injecteur de dépendance dans toute l'application, Angular laisse chaque composant utiliser son propre injecteur.

(1) <https://habiletechnologies.com/blog/data-binding-angularjs-1-angularjs-2/>

DEBUGGING

« Débugger » une application est compliqué en soit, mais c'est encore plus compliqué avec une application AngularJS. Les erreurs de 'binding' ne se déclenchent pas du tout.

De ce côté-là, Angular, grâce à Typescript, permet de trouver des incohérences plus tôt lors du cycle de développement, ce qui n'est pas négligeable.

SCOPE

C'est sans doute l'erreur la plus commune à laquelle doit faire face un développeur AngularJS. Si vous basez votre logique sur les héritages de scope, cela devient très difficile à tester. Votre logique devient plus compliquée et implicite, car vous utilisez des variables qui ne sont pas déclarées dans le module. Les variables héritées sont identiques à des variables globales et tout le monde sait que les variables globales sont le mal !

DIRECTIVES

Leurs syntaxes sont compliquées et les comprendre prend du temps. Il faut non seulement savoir quoi faire, mais également comment le faire. Or cette complexité n'est pas forcément utile. Le fait que tous ces problèmes peuvent être résolus ne signifie pas pour autant qu'il n'y ait pas de problème. Un des points les plus importants d'AngularJS reste qu'il force les développeurs « front » à découper la logique en modules et donc le code devient plus modulaire.

Pourquoi Angular ?

Angular ne corrige pas uniquement les problèmes d'architecture de la première version, il apporte également de nouvelles fonctionnalités et d'importantes améliorations que cela soit en termes de performance ou de processus de développement :

RETRO COMPATIBILITE

Dès lors que vous quittez AngularJS pour une version supérieure (dès Angular2), votre application sera sûrement **compatible avec Angular 4, 6 et toutes les versions suivantes.**

PASSAGE À TYPESCRIPT

C'est un sur-ensemble de JavaScript qui comprend un excellent outillage et permet une transition facile depuis le JavaScript. Donc TypeScript n'est pas un choix dangereux pour l'équipe de développeurs qui envisage de l'adopter, car « It's just JavaScript ». Le compilateur TypeScript n'ajoute aucune dépendance à une bibliothèque JavaScript. TypeScript s'intègre alors naturellement dans l'écosystème JavaScript. Les bibliothèques de programmation JavaScript comme

jQuery ou Modernizr sont appelées normalement depuis du code TypeScript. De plus, avec un bon IDE vous pouvez bénéficier de l'auto complétion et de la gestion des erreurs lors de la compilation. Une fois que vous commencez à l'utiliser vous allez constater que votre productivité augmente en comparaison de JavaScript.

LES COMPOSANTS

Alors que AngularJS utilise « directive », « contrôleur » et « \$scope », ces concepts sont maintenant réunis en un seul « component ».

Angular est orienté « component », où même l'application est un « component ».

AOT (ahead-of-time)

Face aux concurrents qui prétendent avoir les mêmes performances qu'Angular avec un code plus léger, l'équipe Angular a priorisé l'AOT, afin d'avoir un code généralement réduit à 60%.

LAZY LOADING

Technique qui permet de charger uniquement les ressources nécessaires à l'affichage d'une page.

AMELIORATION DU *NGIF ET *NGFOR

L'une des principales fonctionnalités ajoutées dans Angular est l'amélioration des bases structurales *ngIf et *ngFor.

La directive ngIf a été beaucoup critiquée dans AngularJS. C'était dû au manque de la clause « else » pour simuler un bloc « if-then-else » dans le « template ». Voici un exemple :

```
<div ng-if="condition">...</div>
<div ng-if="!condition">...</div>
```

Dans la nouvelle version on peut maintenant utiliser le « else » qui est partie intégrante de la directive ngIf.

```
<div *ngIf="condition; then thenBlock else elseBlock"></div>
<ng-template #thenBlock>...</ng-template>
<ng-template #elseBlock>...</ng-template>
```

La directive ngFor est une implémentation Angular du ng-repeat qui apporte des améliorations telle que des variables locales (ex : index, first, last, even et odd). Celles-ci vous permettent de ne plus utiliser TrackBy comme dans AngularJS. ngFor permet aussi d'itérer sur un observable.

```
<div *ngFor="let radio of radios$ | async; #i=index;">
  {{i}} - {{radio.name}}
</div>
```

Parmi les autres apports, Angular supporte les « web crawlers » pour un meilleur référencement.

Comment vendre au métier cette migration vers Angular ?

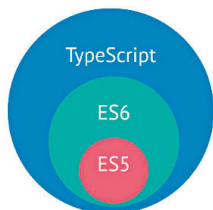
Angular est un framework très tendance, il ne passe pas 3 mois sans en entendre parler. Ce qui nous amène à nous poser les questions suivantes :

- N'est-ce pas justement qu'une tendance ?
- Est-ce que le projet/métier a vraiment besoin de cette technologie ?
- Pourra-t-il compter dessus pour les besoins à venir et dans quelles mesures ?

Le métier, quant à lui, peut se poser la question du coût d'entrée sur la nouvelle version. **En effet, qu'importe le point de départ, le passage à Angular (et plus) n'est pas gratuit !** Mais cette nouvelle version arrive avec plusieurs innovations versus AngularJS. La migration comporte donc une vraie valeur ajoutée.

Bien que la courbe d'apprentissage ne soit pas trop élevée, il y a quelques changements radicaux par rapport à la première version qui pourraient être difficiles à appréhender. Ces différentes mises à jour ont suscité beaucoup de craintes et de questions au sein de la

	JavaScript	ECMA Script 6	TypeScript
Better Structure	✗	✓	✓
Editor Friendly	✗	✗	✓
Better Speed	✗	✗	✗
Large Applications	✗	✓	✓
Brevity of Code	✗	✗	✗
Ease of Learning	✓	✓	✓
Easy Debugging	✓	✓	✓



communauté. Nous en sommes arrivés au point où beaucoup ne savaient pas s'ils devaient vraiment mettre à jour leurs applications ou tout recommencer à zéro. **Ce qui est du point de vue métier difficile à comprendre quand on a déjà une application fonctionnelle en production.** Allons, faisons une passe sur les points forts de ces nouvelles versions, qui ne manqueront pas de faire réagir (positivement) votre métier.

PERFORMANCE

On en parlait précédemment et c'est évidemment une chose que **le métier peut comprendre facilement, Angular est 5 fois plus rapide que AngularJS.** Ce qui fait du boost de performance, la raison principale de migration. Le rendu de l'application peut commencer à être fait efficacement côté serveur, grâce au serveur-rendering, donc cela augmente les performances d'exécution. *Pour l'explication technique :* l'équipe d'Angular a fait des changements dans l'AOT (Ahead-Of-Time Compilation). De ce fait l'application est compilée avant d'être envoyée au navigateur. Par conséquent, il est plus facile de télécharger le code et le navigateur peut l'exécuter plus rapidement en raison de ce temps de compilation réduit. Ce qui favorise par ailleurs la sécurité de l'application. **Donc Angular est plus léger, rapide et plus sécurisé vis à vis des injections de code.**

ORIENTE MOBILE

Si votre métier cherche à déployer l'application sur des appareils mobile, cela peut être un plus pour l'adoption d'Angular qui est une approche axée sur les mobiles. Contrairement à la version AngularJS, les dernières versions ont certaines stratégies inspirées des Frameworks orientés mobile, comme Nativescript, Ionic Framework ou encore React Native. Par exemple avec le Lazy Loading, qui permet de compiler différents modules uniquement quand on en a besoin. Dans tous les cas, **attaquer le segment du mobile sera plus simple avec Angular qu'avec AngularJS.**

MEILLEURE EXÉCUTION GRÂCE AUX COMPOSANTS

La structure des applications Angular, basée sur les composants, rend l'application plus facile à comprendre qu'une application en AngularJS. Cette nouvelle approche **permet aux développeurs de se concentrer sur ce qu'ils sont en train de développer plutôt que de chercher à comprendre comment faire.** La testabilité et la maintenabilité de l'ensemble est aussi accrue.

MIGRER EN DOUCEUR

Finalement, le bénéfice majeur qui permettra de vendre Angular est le fait que les deux versions du framework peuvent vivre ensemble au sein même d'une application.

De plus, être à la pointe des dernières tendances technologiques incitera les développeurs à devenir plus motivés et compétents, ce qui est un avantage non négligeable !

Après vous avoir présenté les ajouts principaux d'Angular versus AngularJS, ainsi que leurs avantages, nous allons vous expliquer les stratégies de migration disponibles.

Planification de la migration

Afin de mettre toutes les chances de votre côté, il est impératif d'avoir un plan d'action. Ce plan de migration est fonction de l'état actuel de votre projet et de la manière dont il a été réalisé.

Vous devez avoir conscience des différentes stratégies qui existent en vue d'en choisir une :

NE PAS FAIRE DE MIGRATION

Dans le cas d'une application qui est **proche de la fin de son cycle de vie** (exemple : une application qui gère tout ce qui est en relation avec une loi qui n'est plus d'actualité...) ou encore d'une application qui tourne très bien sous AngularJS et pour laquelle **aucune amélioration ne sera demandée dans le futur** ; en effet, migrer serait un effort qui n'aurait **pas vraiment de bénéfice significatif**. Toutefois il est important de **garder un œil sur les différentes vulnérabilités** qui potentiellement peuvent apparaître, ou sur l'utilisation de certaines fonctions qui sont obsolètes.

MIGRATION PROGRESSIVE EN UTILISANT NG-UPGRADE

Comme mentionné précédemment, la migration peut être progressive en faisant cohabiter les deux versions d'Angular ensemble. Ce point sera le sujet majeur de cet article et sera plus détaillé par la suite.

RÉÉCRIRE L'APPLICATION FROM SCRATCH

Cette approche est optimale, mais elle demande une préparation spécifique, qui sera abordée dans le dernier paragraphe de cet article. Dans tous les cas, que vous choisissiez l'une ou l'autre des stratégies qui s'offrent à vous, vous devez avoir une approche précise par rapport à votre code base existant :

- **Migrer fonctionnalité par fonctionnalité** : consiste à identifier le groupe de composants et services qu'utilise cette fonctionnalité puis à l'implémenter en Angular.
- **Migrer route par route** : consiste également à identifier les composants et services, mais cette fois-ci par rapport à une route spécifique, puis les implémenter en Angular. Cette stratégie fonctionne bien si la route ne comporte pas trop de composants et qu'ils ne sont pas trop complexes.

Maintenant que vous savez pourquoi migrer et que vous connaissez les stratégies qui s'offrent à vous, **il convient de préparer le terrain afin de s'assurer d'arriver rapidement au bout de cette migration.**

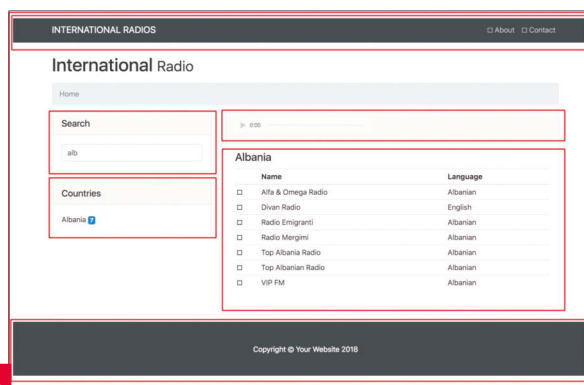
Préparation du terrain

Afin que la migration soit plus facile, votre projet doit répondre et s'aligner à certains concepts et conventions :

LES TESTS

Les tests représentent l'un des principaux leviers de la réussite d'un projet, ils vous permettent de développer une application avec un minimum de défauts. Sur une application conséquente, il y a inévitablement plusieurs développeurs qui travaillent ensemble. L'intégration des différentes fonctionnalités par la suite peut engendrer de nombreux bugs. C'est pourquoi il est nécessaire d'**avoir une bonne couverture du code qui vous permettra d'éviter d'éventuelles régressions.** Cet article n'a pas pour vocation de faire l'apologie des tests, mais **ceux-ci constituent la base d'un code professionnel et de qualité**, critères principaux du mouvement Software Craftmanship. L'écosystème de test pour Angular est très riche et des outils comme Karma, Angular Mocks, Mocha ou encore Protractor sont des incontournables !

Il est important de configurer une étape de **contrôle de la couverture de test** dans votre pipeline de déploiement, afin qu'à chaque



1

« build » une règle de 50% minimum de couverture s'applique, sinon le « build » ne passe pas. Puis, monter progressivement le seuil minimum de couverture. Ce qui encouragera continuellement l'équipe à s'améliorer et à fiabiliser l'application dans le temps.

Pour étayer nos propos au sujet de la préparation du terrain en vue d'une migration, nous vous avons mis à disposition un projet **AngularJS(2)**, afin que vous puissiez visualiser des exemples de :

- Tests de services, composants et « controllers » ;
- Génération d'un rapport de la couverture de code ;
- Mise en place d'une règle de 70% minimum de couverture de code ;
- L'invocation de ces tâches dans un pipeline de build TravisCI.

LOGIQUE MÉTIER

Aucune, ou presque, logique métier ne doit être présente dans le « front », car à la base les « front-ends » sont censés être « jetables ». Pour preuve : votre migration actuelle !

STYLE GUIDE

Un Style Guide est un ensemble de standards mis à disposition par la communauté, qui nous permet d'adopter des bonnes pratiques et qui couvre l'ensemble des parties d'un projet. Comme vous le savez, le monde du JavaScript est très permissif et AngularJS n'échappe pas à cette règle. Il se peut que vous commettiez facilement l'une des erreurs suivantes :

- Les mêmes tags AngularJS avec plusieurs des syntaxes disponibles : data-ng-app, x-ng-app ou encore ng-app ;
 - L'architecture est un mélange MV*, MVW ... ;
 - L'implémentation des directives avec Directive et Component ;
 - L'utilisation de fonctions récentes non supportées par tous les navigateurs ;
 - L'utilisation de fonctions dépréciées qui peuvent causer un dysfonctionnement du projet.
 - L'utilisation d'un style guide évite de tomber dans ce genre de travers.
- Dans le monde d'AngularJS plusieurs styles guides existent, Google(3) en fournit d'ailleurs un, mais les deux guides sortant du lot sont ceux de Todd Motto(4) et John Papa(5).

Il est difficile de choisir car les deux sont très bien. Plusieurs articles sur internet essaient de les comparer. Mais le meilleur moyen de vous faire une idée sur ces guides reste de les tester par vous-même. La sélection reste avant tout un choix d'équipe !

LE PASSAGE EN COMPOSANTS

Si votre application tourne déjà avec une version d'AngularJS supérieure à 1.5, cela permet d'utiliser la directive « component ».

Celle-ci est apparue depuis cette version et elle facilitera grandement votre migration car l'architecture Angular est orientée composant. Procéder de cette façon vous permettra d'avoir, pour chaque composant, son propre « template », « controller » et ses entrées/sorties ; ce qui ressemble grandement au fonctionnement d'un composant Angular. D'autre part pour être Angular compatible, il ne faut pas que vos composants utilisent les attributs « compile », « transclude », « terminal », « replace » et « priority ». Voici une proposition d'un découpage par composant : 1 Cette application exemple est mise à disposition sur GitHub pour appuyer cet article.

CONTROLLERS

Une autre amélioration apportée par AngularJS 1.5 est l'ajout de « hook » méthodes (\$onInit, \$onDestroy, \$onChanges), qui permettent d'interagir avec le composant durant les différentes phases de son cycle de vie.

L'UTILISATION DU MODULE LOADER

Utiliser les modules loader tel que Webpack ou Browserify vous permettra d'utiliser le même module patterns fournis par TypeScript ou ES2015.

MIGRER VERS TYPESCRIPT AVANT DE MIGRER VERS ANGULAR

Vous pouvez commencer à utiliser TypeScript dans votre application en installant un compilateur et en convertissant tous vos fichiers *.js en *.ts, cela vous permettra d'avoir les avantages suivants :

- Module Loader natif qui vous permettra d'organiser votre code en module ;
- Les annotations qui peuvent être ajoutées graduellement ;
- Les fonctionnalités ES2015 tels que les « arrow function », « let », « const » ou l'affectation destructurée ;

Migration progressive via ng-upgrade

Maintenant que les bases sont posées, nous allons voir ensemble plus en détails la technique de migration via NgUpgrade. La particularité de cette méthode est que la migration passe d'abord par une phase de cohabitation des deux versions d'Angular. C'est un bon moyen de passer le cap et de commencer directement à utiliser la nouvelle version, sans faire un « big bang ». Ce qui n'est pas négligeable si votre application est trop complexe ou que vous n'avez pas la capacité de faire la bascule d'un coup. Cela a pour bénéfice de réduire le risque autour du projet de migration et de continuer à livrer de la valeur, tout en mettant en place les fondations de la prochaine version de l'application. C'est un peu l'esprit que nous retrouvons dans le pattern d'architecture « strangler », qui consiste à créer progressivement un nouveau système autour des bords de l'ancien, en le laissant croître lentement sur plusieurs années jusqu'à ce que l'ancien système soit étranglé. L'outil NgUpgrade, mis à disposition par l'équipe Angular, a pour objectif de faciliter la migration, mais il y a aussi un outil créé par la communauté et « approuvé » par la « core team » d'Angular, qui s'appelle ngForward.

NgForward est un projet open source, qui vous permet d'écrire une

(2) <https://github.com/RinOne/ngUpgrade>

(3) <https://google.github.io/styleguide/angularjs-google-style.html>

(4) <https://github.com/toddmotto/angularjs-styleguide>

(5) <https://github.com/johnpapa/angular-styleguide>

application en AngularJS, tout en ayant un code qui sera très similaire à de l'Angular2. Je m'explique ! Vous ne commencerez pas vraiment à utiliser ng2. Au lieu de cela, vous vous familiarisez avec la syntaxe et une grande partie de votre code ressemblera presque exactement à ce qu'il aurait été dans ng2. N'étant pas une solution de migration officielle nous n'en parlerons pas plus, mais si ce chemin vous intéresse, il y a pas mal de documentation sur internet. Concernant ngUpgrade, le meilleur endroit pour commencer à se familiariser avec cette méthode est la documentation officielle fournie par Google. Par ailleurs, différentes stratégies d'installation sont disponibles sur internet.

Dans la plupart des cas, cela ressemblera à la roadmap suivante :

- Appliquer les recommandations données dans la section "Préparer le terrain" ;
- Faire tourner l'application dans un environnement Angular CLI ;
- Faire la configuration de la gestion du routage entre Angular et AngularJS ;

Sacrée roadmap, mais ce n'est pas aussi compliqué que ça en a l'air. Nous allons détailler ces étapes par la suite.

APPLIQUER LES RECOMMANDATIONS

Dans tous les cas, même si vous ne poursuivez pas votre migration vers une nouvelle version d'Angular, l'application des différents points exposés précédemment rendra votre code beaucoup plus propre, lisible et pérenne. Ce qui est déjà une très bonne chose en soi.

INITIALISER UN NOUVEAU PROJET AVEC ANGULAR-CLI

Après avoir préparé votre projet pour la migration, vous pouvez poser les premières fondations de la nouvelle version. Cela commence par créer un nouveau projet, en utilisant Angular CLI.

Angular CLI est d'une interface de commande en ligne qui permet de générer une application, composant ou service ou tous les autres concepts Angular. Au cours de l'année dernière, il est devenu l'un des outils les plus essentiels dans la boîte à outils d'un développeur Angular. Le point d'entrée officiel pour la documentation est ici <https://cli.angular.io/>.

Voici les commandes permettant de réaliser cette étape :

```
npm install -g @angular/cli
ng new my-angular-project
```

Puis lancez le nouveau projet, pour vérifier que tout est bien installé :

```
cd my-angular-project
ng serve
```

On continue avec l'installation du module ngUpgrade :

```
npm install @angular/upgrade --save
```

CONFIGURATION NÉCESSAIRE POUR LA COHABITATION DES DEUX FRAMEWORKS

Maintenant, il faut définir une structure pour votre projet, on vous propose de faire au plus simple en mettant **chaque application dans un dossier**.

Maintenant, pour pouvoir **importer votre application AngularJS dans votre nouveau projet Angular**, il faut builder l'application AngularJS via des hooks. Voici un exemple du package.json du projet Angular : **2**

```
{
  ...
  "scripts": {
    "prestart": "gulp --gulpfile ../../angular1_5/gulpfile.js",
    "start": "ng serve",
    "prebuild": "gulp --gulpfile ../../angular1_5/gulpfile.js",
    "build": "ng build --prod",
    ...
  },
  ...
}
```

Jusque-là rien de compliqué, pour la suite il faut **importer votre application AngularJS avec ses dépendances dans angular-cli.json** du projet Angular :

```
{
  ...
  "apps": [
    {
      ...
      "styles": [
        "../../angular1_5/build/css/vendors.min.css",
        "../../angular1_5/build/css/styles.min.css",
        "styles.css"
      ],
    },
    "scripts": [
      "../../angular1_5/build/js/vendors.min.js",
      "../../angular1_5/build/js/app.min.js"
    ],
    ...
  ],
  ...
}
```

La dernière étape avant de pouvoir lancer les deux applications ensemble est de « bootstrapper » l'application AngularJS avec le ngUpgrade(6). Cela se fait via deux directives HTML :

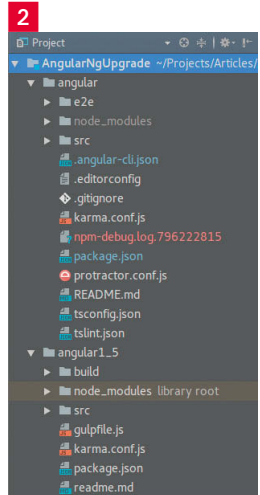
```
<!DOCTYPE HTML>
<html>
  <head>...</head>

  <body ng-app="app" ng-strict-di>
    <!-- home is AngularJS component -->
    <home></home>
  </body>
</html> Ou via du JavaScript dans votre AppModules.
angular.bootstrap(document.body, ['app'], { strictDi: true });
```

Cette partie est la plus compliquée parmi les différentes étapes de mise en place de l'environnement de développement. Un exemple figure dans le projet ngUpgrade fourni avec cet article (<https://github.com/RinOne/ngUpgrade>).

Après avoir vu comment faire tourner les deux frameworks en même temps afin d'avoir une application hybride, passons à la migration des différents concepts AngularJS en Angular.

(6) <https://angular.io/api/upgrade/static/UpgradeModule>



MIGRER LES CONCEPTS ANGULARJS VERS ANGULAR

Migrer un composant, un service ou tous les autres concepts introduits par AngularJS est plutôt facile en soit, la documentation **Angular Upgrade** est très bien renseignée. Elle vous permettra de comprendre rapidement les stratégies de migration à travers des exemples concrets. Voici un exemple tiré du projet lié à cet article, pour migrer un composant :

```
app
.component('searchMenu', {
  bindings: {
    keyword: '='
  },
  templateUrl: 'directives/search/search.html',
  controller: 'searchController as searchCtrl'
})
.controller('searchController', function() {});
<div class="card mb-4">
...
<input type="text"
  class="form-control"
  placeholder="Search for..."
  ng-model="searchCtrl.keyword"
  autocomplete="off"
  name="search">
...
</div>
```

Pour migrer ce composant, il faut le réécrire en Angular, puis le rétrograder pour qu'il fonctionne dans l'application AngularJS via la fonction « **downgradeComponent** ».

```
import {Component, Input} from '@angular/core';
import {downgradeComponent} from '@angular/upgrade/static';

declare let angular: any;

@Component({
  selector: 'app-search',
  templateUrl: './search.component.html',
  styleUrls: ['./search.component.css']
})
export class SearchComponent {
  @Input() keyword: string;
}

angular.module('app')
.directive(
  'searchMenu',
  downgradeComponent({component: SearchComponent})
);
<div class="card mb-4">
...
<input type="text"
  class="form-control"
  placeholder="Search for..."
  [(ngModel)]="keyword"
```

```
autocomplete="off"
name="search" />
...
</div>
```

Une dernière étape consiste à ajouter « **SearchComponent** » dans les **déclarations** de l'application en tant que composant, puis l'ajouter aussi à « **entryComponents** » en tant que point d'entrée à l'application Angular.

```
@NgModule({
  declarations: [AppComponent, SearchComponent],
  imports: [BrowserModule, FormsModule],
  providers: [],
  bootstrap: [AppComponent],
  entryComponents: [SearchComponent]
})
export class AppModule { }
```

Le dernier point concerne nos recommandations pour une réécriture « **From Scratch** » de votre application AngularJS en Angular.

From Scratch

Cette approche est recommandée dans la mesure où les circonstances le permettent. Une réécriture de toute l'application peut être très coûteuse, car comme évoqué précédemment, une montée en compétence de toute l'équipe peut se révéler plus ou moins compliquée en fonction de l'état d'esprit des développeurs. Idéalement, pour commencer ce type de projet, il vous faudra l'aide de quelqu'un d'expérimenté en Angular, afin de vous guider durant toutes les phases du développement (de la création de l'application via Angular-CLI, jusqu'à la mise en prod). Une réécriture complète de l'application peut ne pas être la bonne option si votre application est très grande. Une solution est la migration progressive via la cohabitation d'AngularJS et Angular, comme expliqué dans la partie **Migration progressive via ngUpgrade**. Cela vous permettra de faire évoluer votre application, tout en continuant à apporter de la valeur pour votre métier.

CONCLUSION

La procédure de monter en version d'une application AngularJS vers Angular peut être intimidante, parce qu'il faut composer et être à l'aise avec 3 domaines de connaissances : **AngularJS**, **Angular** et **TypeScript**. Evidemment, cet article ne peut pas couvrir tous les aspects d'Angular, donc d'autres sujets restent à creuser de votre côté, tels que l'utilisation de **NgRx** et **RxJS**.

Avant d'engager la migration, nous suggérons aux développeurs qui ont de bonnes connaissances en AngularJS, de **prendre le temps de se former en vue d'être confortables avec les concepts introduits par Angular et Typescript**. Cela pourrait passer par le développement de nouvelles fonctionnalités en Angular directement afin de comprendre les mécanismes de celui-ci, puis par la suite essayer de traduire les paradigmes d'AngularJS dans la nouvelle stack technique. Néanmoins, l'équipe d'Angular a fait un bon travail en fournissant des outils pour rendre ce processus moins intimidant, donc lancez-vous ! **N'oubliez pas notre repository GitHub, mis à votre disposition pour voir des exemples concrets.** •



Aurélie Vache
Cloud Dev(Ops) chez Continental
Duchess France & DevFest Toulouse Leader
Toulouse Data Science core-team & Mairaine Elles bougent
@aurelievache



niveau
400

Terraform Advanced : Gérer plusieurs environnements avec les workspaces de Terraform

Comme nous l'avons vu dans un précédent article dans ce magazine, lorsque vous voulez gérer (créer, modifier et supprimer) votre infrastructure, il est facile de commencer avec Terraform. Mais gérer plusieurs environnements et tfstate avec Terraform est une autre paire de manches. Heureusement que les workspaces sont là et vont nous prêter main forte.

Step by step

Nous allons commencer ce tutoriel en créant simplement des fichiers se terminant par .tf contenant la description des ressources que vous voulez avoir.

Par exemple, si nous voulons créer une petite infrastructure chez le fournisseur de cloud AWS :

- un bucket S3 (pour terraform) ;
- un bucket S3 (pour notre site web) ;
- un bucket S3 (pour les logs du site Web) ;
- une distribution CloudFront (sans SSL pour cet exemple).

Nous avons juste besoin de créer des fichiers .tf comme ceci :

```
aws_s3.tf
aws_cloudfront.tf
root.tf
variables.tf
output.tf
```

aws_s3.tf :

```
# s3 bucket for terraform state files
resource "aws_s3_bucket" "com_programmez_terraform" {
  bucket = "${var.aws_s3_bucket_terraform}"
  acl    = "private"

  versioning {
    enabled = true
  }

  tags {
    Tool   = "${var.tags-tool}"
    Contact = "${var.tags-contact}"
  }
}

# s3 bucket for front logs
```

```
resource "aws_s3_bucket" "front_logs" {
  bucket = "${terraform.workspace == "preprod" ? var.bucket_demo_logs_preprod : var.bucket_demo_logs}"
  acl    = "log-delivery-write"

  tags {
    Tool   = "${var.tags-tool}"
    Contact = "${var.tags-contact}"
  }
}

# s3 bucket reached by cloudfront
resource "aws_s3_bucket" "front_bucket" {
  bucket = "${terraform.workspace == "preprod" ? var.bucket_demo_preprod : var.bucket_demo}"
  acl    = "private"

  force_destroy = false

  depends_on = ["aws_s3_bucket.front_bucket-logs"]

  versioning {
    enabled = true
  }

  logging {
    target_bucket = "${aws_s3_bucket.front_bucket-logs.bucket}"
    target_prefix = "root/"
  }

  website {
    index_document = "index.html"
  }

  tags {
```

```
Tool = "${var.tags-tool}"
Contact = "${var.tags-contact}"
}

cors_rule {
  allowed_headers = ["*"]
  allowed_methods = ["GET", "PUT", "POST", "DELETE"]
  allowed_origins = ["*"]
}

policy = <<POLICY
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::${terraform.workspace} == "preprod" ? var.bucket_demo_preprod : var.bucket_demo/*"
    }
  ]
}
POLICY
}
```

aws_cloudfront.tf :

```
resource "aws_cloudfront_distribution" "front_cdn" {

  # Use All Edge Locations (Best Performance)
  price_class = "PriceClass_All"
  http_version = "http2"

  "origin" {
    origin_id = "origin-bucket-${aws_s3_bucket.front_bucket.id}"
    domain_name = "${aws_s3_bucket.front_bucket.website_endpoint}"
    origin_path = "/root"

    custom_origin_config {
      origin_protocol_policy = "http-only"
      http_port = "80"
      https_port = "443"
      origin_ssl_protocols = ["TLSv1"]
    }
  }

  enabled = true
  is_ipv6_enabled = true
  default_root_object = "index.html"

  logging_config {
    include_cookies = true
```

```
bucket = "${aws_s3_bucket.front_bucket-logs.bucket}.s3.amazonaws.com"
prefix = "cloudfront/"
}

"default_cache_behavior" {
  allowed_methods = ["GET", "HEAD", "DELETE", "OPTIONS", "PATCH", "POST", "PUT"]
  cached_methods = ["GET", "HEAD"]

  "forwarded_values" {
    query_string = false

    cookies {
      forward = "none"
    }
  }

  min_ttl = "21600"
  default_ttl = "86400"
  max_ttl = "31536000"
  target_origin_id = "origin-bucket-${aws_s3_bucket.front_bucket.id}"
  // This redirects any HTTP request to HTTPS. Security first!
  viewer_protocol_policy = "redirect-to-https"
  compress = true
}

"restrictions" {
  "geo_restriction" {
    restriction_type = "none"
  }
}

# Pre-requisites: Put a SSL cert on AWS store in us-east-1 region
# Generate a csr in localhost, make request to IT, get the returned cert
# put the cert + intermediate + private key in AWS (click in import button)
"viewer_certificate" {
  # default certificate if you don't already added one in AWS certificate manager
  cloudfront_default_certificate = true
  minimum_protocol_version = "TLSv1.1_2016"
}

aliases = ["${var.demo_domain_name}"]

depends_on = ["aws_s3_bucket.front_bucket"]

tags {
  Tool = "${var.tags-tool}"
  Contact = "${var.tags-contact}"
}
}
```

root.tf :

```
provider "aws" {
  region = "eu-central-1"
}
```

variables.tf :


```
variable "default-aws-region" {
  default = "eu-central-1"
}

variable "tags-tool" {
  default = "Terraform"
}

variable "tags-contact" {
  default = "Aurelie Vache"
}

variable "aws_s3_bucket_terraform" {
  default = "com.programmez.terraform"
}

variable "bucket_demo" {
  default = "com.programmez.demo"
}

variable "bucket_demo_logs" {
  default = "com.programmez.demo.logs"
}

variable "bucket_demo_preprod" {
  default = "com.programmez.demo.preprod.demo"
}

variable "bucket_demo_logs_preprod" {
  default = "com.programmez.demo.preprod.demo.logs"
}

variable "demo_domain_name" {
  default = "demo.programmez.com"
}
```

output.tf :

```
output "cloudfront_id" {
  value = "${aws_cloudfront_distribution.front_cdn.id}"
}
```

Maintenant, nous devons initialiser Terraform (seulement la première fois), générer un plan et l'appliquer.

```
$ terraform init
Initializing the backend...

Initializing provider plugins...

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.
```

If you ever set or change modules or backend configuration for Terraform, rerun this command to reinitialize your working directory. If you forget, other commands will detect it and remind you to do so if necessary.

La commande `init` va initialiser votre répertoire de travail qui contient les fichiers de configuration `.tf`.

C'est la première commande à exécuter pour une nouvelle configuration, ou après avoir fait une extraction d'une configuration existante, par exemple, dans un dépôt Git donné.

La commande `init` va :

- télécharger et installer les providers / plugins Terraform ;
- initialiser le backend (si défini) ;
- télécharger et installer des modules (si définis).

Depuis Terraform v0.11+, au lieu de générer un plan et ensuite de l'appliquer, si vous êtes en utilisation interactive, vous avez juste besoin d'exécuter *terraform apply*. La commande va créer le plan, inviter l'utilisateur à répondre "yes" pour l'appliquer, si vous saisissez "yes", le plan sera appliqué et les changements seront effectués.

```
$ terraform apply
...
Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value:
yes
...

Apply complete! Resources: 4 added, 0 changed, 0 destroyed.

Outputs:

cloudfront_id = 123456789
```

Notre infra est créée... Génial! Mais nous travaillons seuls... dans un seul environnement.

Stockage des tfstate en remote

Dans une entreprise ou dans un projet Open Source, vous ne travaillez pas seul, vous devez donc arrêter de stocker le tfstate localement et commencer à le stocker à distance, dans le "cloud", pour le partager (non on ne stocke pas de tfstate dans un repo Github svp ^^).

Pour information ou rappel, un tf state est un snapshot de votre infrastructure depuis la dernière fois que vous avez fait un *terraform apply*. Par défaut, le tfstate est stocké localement dans le fichier *terraform.tfstate*. Mais quand nous travaillons en équipe, nous devons stocker le tfstate à distance :

Nous créons maintenant une ressource *backend* pour stocker le tfstate dans un bucket S3 et le chiffrer.

backend.tf:

```
# Backend configuration is loaded early so we can't use variables
terraform {
```

```
backend "s3" {
  region = "eu-central-1"
  bucket = "com.programmez.terraform"
  key = "state.tfstate"
  encrypt = true #AES-256 encryption
}
```

Lorsque nous avons créé les ressources afin de fabriquer des buckets S3, dans lesquels nous mettons notre tfstate, nous avons activé le versioning. Ce n'est pas une erreur ou un copier/coller. Il est recommandé d'activer le versioning pour les fichiers d'état. Les buckets AWS S3 ont cette capacité, que vous pouvez exploiter puisque Terraform a un backend pour cela. Imaginez, soudainement, votre tfstate ait été corrompu. Merci au versioning des fichiers states, vous pouvez utiliser un state plus ancien et respirer ;-).

Notre fichier tfstate existe déjà, suite à l'exécution du terraform apply fait dans cet article, donc nous devons convertir l'état local en état distant (et stocker notre state dans le bucket s3 que nous avons créé) :

```
$ terraform state push
```

Workspaces

Avant la fonctionnalité des workspaces de Terraform, afin de gérer plusieurs environnements, la solution consistait à créer un dossier par compte d'environnement / provider de cloud et à placer les fichiers .tf. La solution n'était pas pratique, facile à maintenir avec des fichiers .tf en double.

Depuis Terraform v0.10+, pour gérer plusieurs ensembles distincts de ressources / environnements d'infrastructure, nous pouvons utiliser les workspaces de terraform.

Créer des workspaces :

```
$ terraform workspace new dev
Created and switched to workspace 'dev'
$ terraform workspace new preprod
Created and switched to workspace 'preprod'
```

```
$ terraform workspace new prod
Created and switched to workspace 'prod'
```

Sélectionnez le workspace dev :

```
$ terraform workspace select dev
```

Lister les workspaces :

```
$ terraform workspace list
default
* dev
preprod
prod
```

Après une exécution avec succès d'un *terraform apply*, votre tfstate sera stocké dans le bon dossier d'environnement dans le bucket s3 :

```
com.programmez.terraform
env:/
dev/
state.tfstate
preprod/
state.tfstate
prod/
state.tfstate
```

Parfait, car c'est une bonne pratique pour séparer les tfstates par environnement.

Comme vous pouvez le voir dans les fichiers .tf précédents, nous pouvons appeler une variable en fonction du workspace courant :

```
bucket = "${terraform.workspace} == "preprod" ? var.bucket_demo_preprod : var.bucket_demo"
```

Conclusion

Comme vous l'avez vu, avec les workspaces Terraform, nous gérons facilement plusieurs environnements sans maux de tête. Suite à la découverte des workspaces, nous verrons dans un prochain article comment déployer dans plusieurs environnements, sans toucher aux fichiers tf contenant la définition des ressources, grâce aux workspaces et aux variables de type map.



Une publication Nefer-IT, 57 rue de Gisors, 95300 Pontoise - redaction@programmez.com
Tél. : 09 86 73 61 08 - Directeur de la publication & Rédacteur en chef : François Tonic

Secrétaire de rédaction : Olivier Pavie

Ont collaboré à ce numéro : S. Saurel

Nos experts techniques : A. Brel, L. Grangeau, C. Fantoli, D. Nicolet, A. Vannieuwenhuyze, N. Decoster,

L. Ellerbach, M. Ellerbach, N. Lethiec, A. Vache, S. Warin, C. Pichaud, Duy Anh Pham, G. André, A. Clerbois, F. Jadouani, J. Nourry.

Couverture : © Y. Serra - Maquette : Pierre Sandré.

Publicité : François Tonic / Nefer-IT - Tél. : 09 86 73 61 08 - pub@programmez.com.

Imprimeur : S.A. Corelio Nevada Printing, 30 allée de la recherche, 1070 Bruxelles, Belgique.

Marketing et promotion des ventes : Agence BOCONSEIL - Analyse Media Etude - Directeur : Otto BORSCHA oborscha@boconseilame.fr

Responsable titre : Terry MATTARD Téléphone : 09 67 32 09 34

Contacts : Rédacteur en chef : ftonic@programmez.com - Rédaction : redaction@programmez.com - Webmaster :

webmaster@programmez.com

Evénements / agenda : redaction@programmez.com

Dépôt légal : à parution - Commission paritaire : 1220K78366 - ISSN : 1627-0908 - © NEFER-IT / Programmez, juin 2018

Toute reproduction intégrale ou partielle est interdite sans accord des auteurs et du directeur de la publication.

Abonnement :

Service Abonnements PROGRAMMEZ, 4 Rue de Mouchy, 60438 Noailles
Cedex - Tél. : 01 55 56 70 55 - abonnements.programmez@groupe-gli.com
Fax : 01 55 56 70 91 - du lundi au jeudi de 9h30 à 12h30 et de 13h30 à 17h00, le vendredi de 9h00 à 12h00 et de 14h00 à 16h30.

Tarifs

Abonnement (magazine seul) : 1 an - 11 numéros France métropolitaine :
49 € - Etudiant : 39 € CEE et Suisse : 55,82 € - Algérie, Maroc,
Tunisie : 59,89 € Canada : 68,36 € - Tom : 83,65 € - Dom : 66,82 €
- Autres pays : nous consulter.

PDF

35 € (monde entier) souscription sur www.programmez.com

LEROY MERLIN PRÉSENTE

Maker Faire® Paris

DU 9 AU 11 NOVEMBRE 2018 - CITÉ DES SCIENCES ET DE L'INDUSTRIE

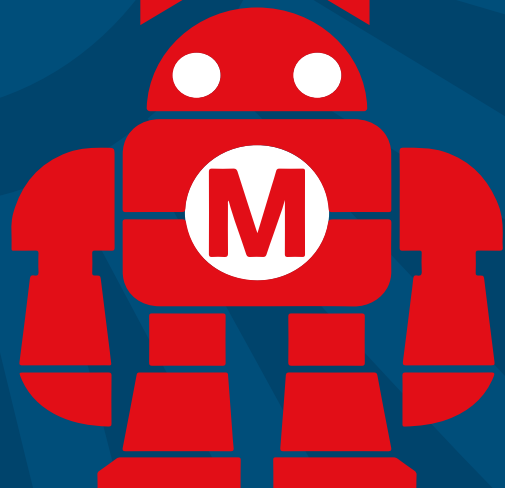
APPEL AUX MAKERS

NOUS RECHERCHONS

BRICOLEURS ★ MAKERS
DESIGNERS ★ CODEURS
GAMERS ★ ARTISTES
HACKERS ★ CREATEURS
INVENTEURS ★ YOUTUBERS
INGENIEURS ★ REVEURS

EN PARTENARIAT AVEC

DESIGNSPARK PRÉSENTÉ PAR 



LES ATELIERS LEROY MERLIN RCS 811 687 961 LILLE

INSCRIPTIONS JUSQU'AU 14 JUILLET 2018 SUR LE SITE
PARIS.MAKERFAIRE.COM

cité
sciences
et industrie

OPÉRATION POUR 1 EURO DE PLUS

Pour bénéficier de cette offre exceptionnelle, il suffit de commander WINDEV Mobile 23 (ou WINDEV 23, ou WEBDEV 23) chez PC SOFT au tarif catalogue avant le 13 Juillet 2018. Pour 1 Euro de plus, vous recevrez alors le ou les magnifiques matériels que vous aurez choisis. Offre réservée aux sociétés, administrations, mairies, GIE et professions libérales, en France métropolitaine. L'offre s'applique sur le tarif catalogue uniquement. **Voir tous les détails sur : WWW.PCSOFT.FR** ou appelez-nous au 04.67.032.032

Le Logiciel et le matériel peuvent être acquis séparément. Tarif du Logiciel au prix catalogue de 1.650 Euros HT (1.980,00 TTC). Merci de vous connecter au site www.pcsoft.fr pour consulter la liste des prix des matériels. Tarifs modifiables sans préavis.

**Aucun
abonnement
à souscrire.**
Compatible
tous opérateurs

CHOISISSEZ :

- SAMSUNG Galaxy Note8
- ou
- SAMSUNG Galaxy S9 +carte SD 128Go
- ou
- SAMSUNG Galaxy S9+
- ou
- Lot de 2 tablettes Samsung Galaxy Tab S2 9,7"
- ou
- TV Wifi 140cm 4K SAMSUNG

(Détails et autres matériels sur pcsoft.fr)

COMMANDEZ WINDEV MOBILE 23 OU WEBDEV 23 OU WINDEV 23 ET RECEVEZ LE NOUVEAU SAMSUNG Galaxy S9+



DAS S9+ : 0.362 W/kg. Le DAS (débit d'absorption spécifique) quantifie le niveau d'exposition maximal de l'utilisateur aux ondes électromagnétiques. La réglementation française impose que le DAS ne dépasse pas 2 W/kg pour une utilisation à l'oreille. Visuels non contractuels. Ecrans simulés.

Atelier de
Génie Logiciel
Professionnel
cross-plateformes