



Etat de l'art : Projet Tanit

UV 4.4 Projet SPID

Jean BAUDINAT

28/03/14

Introduction

Le présent document donne un aperçu de l'état de l'art des systèmes cyber-physiques (CPS) ainsi que des outils et technologies nécessaires à la mise en œuvre du projet TANIT (This Is Not A Train).

Aussi, il s'articulera de la sorte. Une première partie décrira ce qu'est un système cyber physique, et en quoi cette notion est en rupture avec la vision conventionnelle que nous avons des systèmes embarqués.

Enfin, la deuxième partie présentera le projet TANIT, un train électrique dernière génération muni du récent protocole DCC, muni de nombreux capteur et d'une chaîne de décision. Ce système servira de démonstrateur afin d'appréhender les CPS, et de mettre en œuvre différents moyens d'actions et de contrôle sur un tel système.

1. Les systèmes cyber-physiques

Un système cyber-physique est un système constitué de différents éléments computationnels, collaborant ensemble afin de contrôler une entité physique (Seshia 2011). De tels systèmes sont déjà existant dans l'aéronautique, l'énergie, la santé ou encore le BTP. Si, à première vue, ils peuvent être rapprochés du système embarqué, ils diffèrent cependant de ceux-ci par la variété de plate-forme pouvant être utilisées et mises en réseau afin de réussir leur mission. Quand un système embarqué est le seul à observer son environnement, un CPS sera composé de nombreuses plates-formes hétérogènes, connectées ensemble, et en étroite relation avec leur environnement. Ainsi, le CPS peut agir sur son environnement et réciproquement. Ce champ d'étude a été identifié par la National Science Foundation (NSF) comme étant un champ de recherche clé pour l'évolution des futurs systèmes informatiques (Wolf 2007).

1.1. Des systèmes de plus en plus complexes

L'arrivée massive de nouveaux objets connectés, pouvant capter des informations et les communiquer à d'autres objets crée de nouvelles problématiques quant à la gestion de tels systèmes. En effet, ce type d'objet envahit peu à peu l'ensemble des sphères de notre société, que ce soit dans des domaines comme la santé ou l'aéronautique, mais également dans des domaines plus quotidiens comme les divertissements. Or, la variété de ces objets induit une grande hétérogénéité des plateformes, une multitude d'interfaces et de moyens de communication diverses. Cette variété engendre donc une complexité exponentielle des systèmes, ce qui peut rendre leur comportement imprévisible. Si ce type de comportement peut être admis dans des applications non sensible, il devient en revanche dangereux dans le cas de systèmes critiques, comme c'est le cas pour un avion, une fusée, ou encore un automate de santé.

De plus, ces objets sont de plus en plus en interactions avec leur environnement, il est donc important de connaître exactement l'action que pourra avoir l'environnement sur leur comportement, afin de pouvoir en prévoir les conséquences. Aussi, une nouvelle approche doit être adoptée, afin de conceptualiser les CPS, pouvoir les simuler en respectant exactement le comportement réel, et pouvoir démontrer que le comportement simulé sera celui adopté dans l'environnement réel.

1.2.L'introduction de la notion de temps et d'évènements

Une approche formulée par Gérard Berry (Berry 2013), serait de prendre en compte pleinement les notions de temps et d'évènements dans la conception des systèmes. En effet, ces notions ont tendance à être négligées dans les langages actuels purement séquentielle. Or, de par leur interaction constante avec l'environnement, et la variété des éléments les constituant, les CPS ne peuvent faire l'économie d'une prise en compte de cette dimension. Ceci peut se faire par l'utilisation de nouveaux langages, faisant appel à de nouveaux paradigmes, et où la notion d'évènement est totalement prise en compte. C'est notamment le cas de langages tels que Esterel, où un approche multiforme du temps est prise en compte.

1.3.Une vérification formelle du bon fonctionnement du système

A terme, l'ensemble des outils développés pour les CPS visent à avoir une démonstration formelle du bon fonctionnement d'un tel système, en tenant compte de l'ensemble des paramètres de l'environnement. De tels outils de vérifications existent déjà pour les preuves mathématiques, comme le langage Coq (INRIA 2014), qui a déjà permis de prouver formellement le théorème des trois couleurs et le théorème de Feit et Thomson. Coq a également permis de vérifier formellement le compilateur de langage C CompCert, ce qui permet d'être certain de la bonne exécution du code compilé. Ce formalisme dans la compilation est un premier pas dans la validation du bon fonctionnement des CPS. Cela montre qu'une approche plus mathématique doit être utilisée afin de résoudre la problématique de l'analyse et de la vérification des systèmes cyber-physiques.

2. Un exemple de démonstrateur : le projet TANIT

2.1. Introduction

Comme support aux futures expérimentations concernant le domaine des CPS, il a été décidé de créer une plateforme. Cette plateforme devait être indépendante de toutes contraintes spécifiques de milieu, que ce soit maritime, médical ou aéronautique, et également assez simple à mettre en œuvre. Enfin, il devait être possible d'observer facilement ce système physique, tout en pouvant agir sur celui-ci.

Aussi, un train miniature est un système pouvant approcher les nombreuses caractéristiques des CPS. Il est en effet muni de divers actionneur, et peut être facilement observé au moyen de divers capteurs placés tout au long des voies. Enfin, la norme DCC permet d'avoir un contrôle pointu sur l'ensemble des composant de ce système, que ce soient les motrices, ou les aiguillages.

De plus, afin d'agir sur ce système, il sera fait usage de nombreux systèmes embarqués, aussi hétérogènes que possibles, connectés en réseau. Ainsi, le but est de pouvoir faire remonter des informations diverses sur le système, tout d'abord dans un but de *monitoring*, puis d'être à même de le contrôler automatiquement.

Aussi, bien plus qu'un simple jouet, le projet TANIT vise à créer un véritable CPS, doté de nombreuse plateformes dialoguant entre elles, et soumis aux contraintes de l'environnement, que ce soient des évènements inattendus, des cyber-attaques ou encore de simples défaillances.

2.2. Le train miniature

Le train choisi est à l'échelle N, de la marque Minitrix. Bien que de nombreuses autres échelles existent, il a été choisi car il permet de limiter l'espace occupé par la plateforme, ce qui rend ainsi le démonstrateur plus compact. L'alimentation des motrices se fait via les deux rails directeurs. Ce train est contrôlé à l'aide d'un protocole, le protocole DCC (Digital Command Control).



Figure 1 Une locomotive à l'échelle N

2.2.1. Le protocole DCC

Le protocole DCC est désormais un standard dans le domaine du modélisme ferroviaire. Il est défini par le National Model Railroad Association (NMRA 2004), et adopté par de nombreuses associations de modélismes ferroviaire. Aussi, l'utilisation d'un train muni de ce protocole est justifiée, car il est le plus utilisé, ce qui facilite l'acquisition de matériels spécifiques facilement adaptables au circuit. De plus, le protocole est assez simple à mettre en œuvre, ce qui facilite également le contrôle sur le train. Les ordres sont envoyés aux périphériques (motrices, aiguillages, feux, ...) sous formes de paquets. Une station de commande DCC, en combinaison avec son *booster* (amplificateur de signal), module la tension de la voie pour envoyer des messages numériques tout en fournissant l'alimentation de puissance. Chaque ordre est destiné à une adresse unique, qui est incorporée dans le paquet.

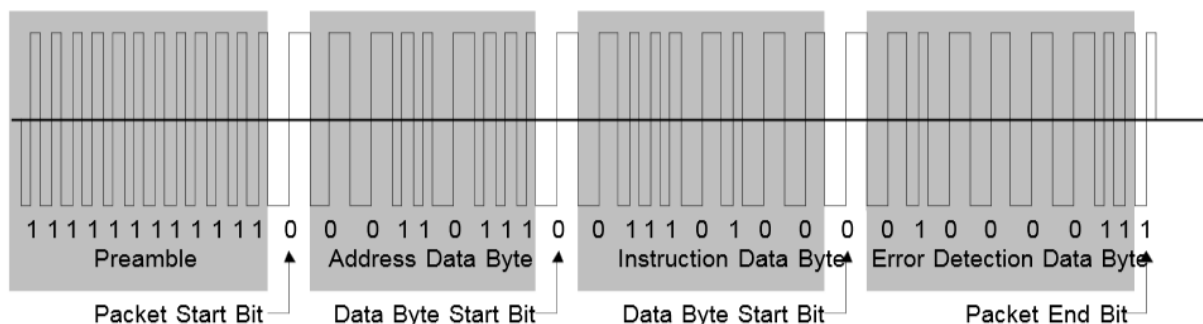


Figure 2 Exemple de trame DCC

2.2.2. Commande du train

Ce train est pour le moment contrôlé par une centrale dotée d'une interface utilisateur. Celle-ci permet de générer les trames contenant les ordres aux motrices, et son interface utilisateur permet d'avoir un à l'aide de commandes simples (boutons pour sélectionner la motrice sur laquelle agir, molette pour ajuster la vitesse et le sens de marche). Cependant, cette centrale se révèle être un obstacle dans le contrôle automatisé du train que nous voulons mettre en œuvre, puisqu'elle ne dispose pas s'interfacer avec un ordinateur, ou tout autre système embarqué permettant de générer les ordres à l'aide des différents signaux reçus.

Aussi, plusieurs solutions s'offrent à nous afin de résoudre ce problème. Une première solution serait de modifier la centrale afin de la rendre commandable par ordinateur. Cependant, cette centrale étant une boîte noire, cela nécessiterait beaucoup de travail de rétro-ingénierie, pour un résultat sans doute approximatif.

Une autre solution serait de créer une centrale à l'aide d'une carte électronique grand public type Arduino ou Raspberry-Pi, comme cela a déjà pu être fait dans le projet *Railuino* (Pleumann s.d.). Nous reviendrons dans une section suivante sur l'ensemble de ces plateformes.

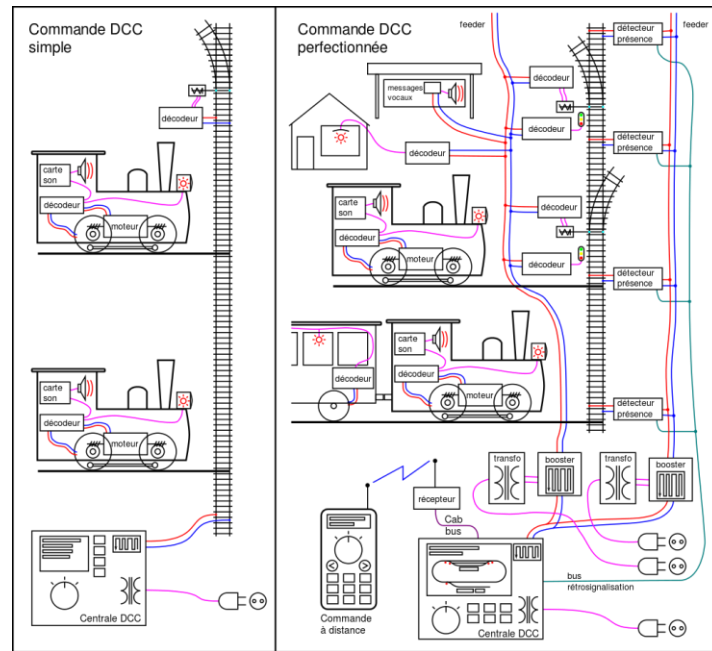


Figure 3 Exemple d'un circuit ferroviaire commandé à l'aide du protocole DCC

2.3. Les capteurs et leurs plateformes de traitement

Afin d'être en mesure de contrôler automatiquement le train, nous devons être à même d'observer celui-ci, mais également tous les autres événements de son environnement qui pourrait avoir une incidence sur celui-ci. Aussi nous devons installer de nombreux capteurs, afin de nous fournir diverses informations, telles que l'identification des motrices, leur vitesse, leur position. Puis, nous devons être en mesure de centraliser toutes ces informations pour un traitement futur. Dans cette section, nous présentons les différents modules capteurs qui pourraient appareiller notre démonstrateur.

2.3.1. Les capteurs

Barrière Infrarouge

La barrière infrarouge est un capteur facile à implémenter. Il consiste en une diode infrarouge servant d'émetteur, et d'un récepteur. Quand le faisceau infrarouge est interrompu, c'est qu'un objet est passé devant. Aussi, en appareillant le train avec un réflecteur de dimension connue, il est possible de connaître la position du train, mais également sa vitesse.

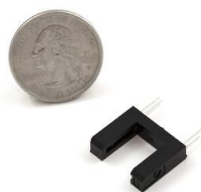


Figure 4 Exemple de barrière infrarouge pouvant être utilisée

Technologie RFID

Les puces RFID (*Radio Frequency Identification*) ont en quelques années envahis notre quotidien (Finkenzeller 2003). Elles remplacent aujourd'hui les codes-barres identifiant les produits, mais sont également présentes dans les badges d'identifications ou les cartes de transport. Cette technologie permet de mémoriser et récupérer des données à distance en utilisant des marqueurs appelés « radio-étiquettes ».

Les radio-étiquettes peuvent prendre diverses formes (capsules, circuit imprimé sur une étiquette, carte magnétique, ...), mais elles sont toutes composées au moins de deux composants. Tout d'abord, une petite mémoire, permettant de stocker au moins une adresse. Cette adresse permet d'identifier avec certitude le porteur de l'étiquette, si toutefois l'étiquette a une adresse unique. Enfin, une antenne, active, ou passive qui va permettre d'émettre l'adresse.

Pour lire les radio-étiquettes, il est nécessaire d'avoir un émetteur-récepteur radiofréquence. L'émetteur va permettre d'activer les puces par induction, et le récepteur pourra ainsi lire les données émises par l'antenne. Différents types de technologies existent, basées sur des longueurs d'onde différentes. Celles-ci ont ainsi une portée différente, pouvant aller de quelques centimètres, à plusieurs mètres. Dans le cas de notre démonstrateur, celui-ci ayant une petite taille, il est préférable de choisir un détecteur à courte portée, de l'ordre de quelques centimètres, afin de ne détecter les motrices que quand celle-ci passent devant le détecteur.

L'implémentation d'un tel capteur présente l'avantage de pouvoir identifier la motrice, contrairement à la barrière infrarouge. De plus, à l'aide de deux puces RFID placées de part et d'autre du train, il est possible d'identifier le début et la fin du train, et également de connaître sa vitesse.

La reconnaissance à l'aide d'une caméra

Les techniques de traitement d'images sont de plus en plus développées, et ont des applications toujours plus grandes. Ainsi, il est possible d'utiliser une caméra afin de détecter le passage du train, mais également de l'identifier. Cela permet ainsi de détecter tous les événements pouvant avoir lieu autre que le passage du train. Deux actions doivent ainsi être menées par ce capteur ; détecter un événement, c'est-à-dire le mouvement d'un objet. Puis, savoir identifier cet objet, pour déterminer s'il s'agit d'un train, ou si un objet extérieur est en mouvement sur le circuit. Ceci permet de contrôler l'environnement du mobile, et de pouvoir agir en conséquence pour éviter tout incident.

Si la détection du mouvement sur une vidéo est assez simple, il n'en est pas de même pour l'identification du train. De plus, cette identification se doit s'être faite de manière fine, afin d'éviter au maximum les faux positifs, accidentels, ou à visée malveillante.

Aussi, un moyen simple d'identification peut être de reconnaître un logo placé sur la locomotive. Cela limite les faux positifs, ainsi que les attaques malveillantes, puisque le logo doit être connu à l'avance. Pour reconnaître ce motif, une bonne méthode peut être d'implémenter la méthode de détection d'objet de Viola et Jones (Paul Viola 2001).

La méthode de Viola et Jones consiste à parcourir l'ensemble de l'image en calculant un certain nombre de caractéristiques dans des zones rectangulaires qui se chevauchent. Elle a la

particularité d'utiliser des caractéristiques très simples mais très nombreuses. Ainsi, une simple fenêtre de taille 24×24 génère environ 160 000 caractéristiques. Bien sûr il n'est pas possible de tester l'ensemble des caractéristiques. Aussi, afin d'accélérer les calculs, une première innovation de la méthode est l'introduction des images intégrales, qui permettent le calcul rapide de ces caractéristiques. Une deuxième innovation importante est la sélection de ces caractéristiques par *boosting*, en interprétant les caractéristiques comme des classifieurs. Enfin, la méthode propose une architecture pour combiner les classifieurs boostés en un processus en cascade, ce qui apporte un net gain en temps de détection.

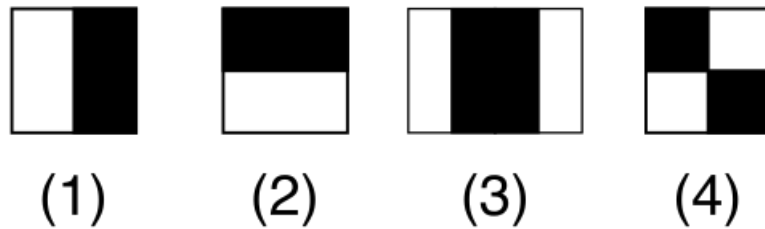


Figure 5 Exemple de caractéristiques utilisées dans la méthode de Viola et Jones

La méthode de Viola et Jones repose donc sur l'entraînement d'une cascade de classifieurs, à l'aide de l'image à reconnaître, et d'images négatives. Cet entraînement va sélectionner les caractéristiques correspondant aux seuils de détection positifs et de faux négatifs définis par l'utilisateur. Une cascade de classifieurs va donc sélectionner les images pouvant contenir le motif recherché, en utilisant des caractéristiques de plus en plus discriminantes. Quand une image réussit une étape du classifieur, elle passe à l'étape suivante, jusqu'à arriver à la dernière étape. Dans ce cas, cette image contient le motif recherché, selon la cascade.

Pour implémenter cette méthode, un moyen rapide est d'utiliser OpenCV, qui contient déjà l'ensemble des programmes pour entraîner une cascade de classifieurs (OpenCV s.d.). A l'aide de cette bibliothèque C++, il nous sera donc possible d'implémenter cet algorithme sur une plateforme de type Raspberry Pi par exemple. Mais il est également possible d'implanter la méthode sur un support de type FPGA (Daniel Hefenbrock 2010), ce qui permet de bénéficier du parallélisme inhérent à cette plateforme, et nous donne ainsi une vitesse de traitement beaucoup plus élevée.

2.3.2. Les plateformes

Comme nous l'avons déjà dit précédemment, un CPS peut comporter un grand nombre de composants hétérogènes, dialoguant en permanence. Aussi, pour reproduire cette caractéristique, il nous faut multiplier les plateformes différentes, afin d'engendrer la complexité. Dans cette partie, nous allons donc présenter les différentes plateformes électroniques à notre disposition.

Arduino

Le projet Arduino (Arduino 2014) est un projet libre de circuit imprimé doté d'un processeur ATmega et d'un ensemble de connectiques facile à mettre en œuvre. Il se programme très facilement à l'aide d'un C++ légèrement modifié. Fourni avec une bibliothèque de fonctions facilitant son usage, il permet de mettre en œuvre rapidement des capteurs simples, comme la barrière infrarouge vue plus haut. L'Arduino embarque peu de connectiques nativement, mais il est cependant possible

d'étendre ses fonctionnalités grâce à des « shields », des cartes électroniques pouvant se brancher sur l'Arduino. Ainsi, afin de pouvoir communiquer avec les autres composants du CPS, il lui sera adjoint un *shield* Ethernet, permettant une communication de type client-serveur par exemple au moyen du protocole TCP/IP.

Arduino permet également de développer des applications plus complexes comme nous l'avons vu précédemment avec le projet Railuino, mais également de nombreux autres projets comme celui du ballon sonde piloté par un Arduino (Walook 2014) .

Raspberry Pi

Contrairement à Arduino qui est un circuit imprimé embarquant peu de puissance, la Raspberry Pi (Fondation Raspberry Pi 2014) est un véritable micro-ordinateur, comportant toutes les caractéristiques d'un tel système, avec notamment un processeur ARM pouvant faire des calculs à haute fréquence, la possibilité d'installer un système d'exploitation de type Linux, mais également une connectique élargie, permettant de relier la Raspberry Pi à de nombreux périphériques. Ceci permet le développement de nombreuses applications sans être limité par la rapidité du processeur ou la mémoire embarquée comme cela peut être le cas pour Arduino. Ainsi, le module capteur de reconnaissance de logo sera plus à même de fonctionner sur une telle plateforme. Ceci est d'ailleurs facilité par l'adjonction d'une caméra spécialement conçue pour la plateforme Raspberry, ce qui accélère le traitement.

Là aussi, la petite taille de Raspberry et sa grande puissance embarquée permettent le développement d'applications variées sans être limité outre mesure. Et des systèmes existent déjà pour créer des clusters de Raspberry si les ressources venaient à manquer (Cox 2013).

FPGA

Les FPGA sont des puces reconfigurables organisées en matrices de blocs logiques. Chaque bloc logique est constitué d'une table de correspondance et d'un Flip-Flop, ce qui permet d'implémenter une équation logique ou de stocker une petite quantité de mémoire. Par une programmation en langage HDL (VHDL ou Verilog), il est possible de programmer ces blocs et de les relier entre eux afin d'implémenter matériellement des algorithmes. Cette approche matérielle de l'implémentation permet un plus grand parallélisme dans les calculs, notamment grâce à la création de pipelines, et une rapidité supérieure par rapport à une approche logicielle.

Ce type de circuit trouve de nombreuses applications dans le prototypage, permettant de connaître le fonctionnement d'une architecture de puce avant de la fondre. De plus, la possibilité de reconfigurer dynamiquement la puce trouve de nombreuses applications dans les industries de l'Aéronautique et du Spatiale ou dans la Défense. Ceci peut s'expliquer par les petites séries généralement produites par ces industries, qui ne justifient pas la création de puces ASIC, mais également le très long cycle de vie des matériels produits. Avoir une architecture reconfigurable permet donc de faire évoluer les produits sans avoir à modifier en profondeur le matériel, ce qui évite les surcoûts et limite le temps de développement.

Ces industries étant les principales intéressées dans les CPS, il paraît donc important d'intégrer des FPGA dans notre démonstrateur. Ceci pourra se faire par exemple par l'implantation

d'un capteur sonore permettant de trianguler la position du train à l'aide de plusieurs micros connectés au FPGA.

2.4. La mise en réseau

La dernière étape à la création de notre démonstrateur est la mise en réseau de l'ensemble des capteurs. Pour ce faire, tous ces capteurs sont munis d'une liaison Ethernet permettant de faire remonter les informations aux niveaux supérieurs afin de traiter les données et de prendre les décisions.

Une première étape sera dans un premier temps de centraliser toutes les informations recueillies afin de les afficher sur une interface de type Web. Puis, une seconde étape sera de créer une intelligence artificielle afin de prendre les décisions sur la circulation des motrices. Enfin, la décision prise, il faudra reboucler sur le circuit à l'aide d'une centrale DCC afin d'envoyer les ordres aux éléments composant le système.

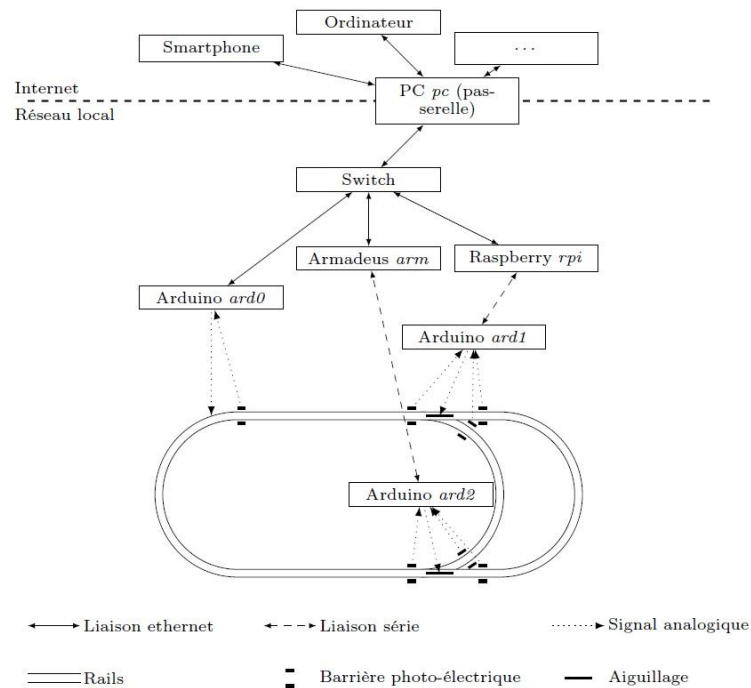


Figure 6 Schéma de la plateforme TANIT

Conclusion

Nous avons pu voir que les CPS sont un domaine en plein essor, et que répondre à la problématique de leur validation formelle a une très grande importance afin d'assurer un bon fonctionnement de ces systèmes souvent critiques. Des outils existent déjà, comme le logiciel Coq, ou les langages multi-paradigmes comme Esterel. Ils sont une première étape à la simulation formelle des CPS. Ils montrent que l'élaboration de tels systèmes doit être menée dès la première phase de conception à l'aide d'une approche mathématique permettant cette vérification formelle.

Pour tester l'ensemble des solutions afin de mieux gérer les CPS, le projet de démonstrateur TANIT tente d'approcher au mieux les caractéristiques d'un tel système. Cela se traduit par une grande diversité de plateformes (Arduino, Raspberry, FPGA, PC, ...), mettant en œuvre de nombreuses techniques d'observation de l'environnement (Méthode de Viola et Jones, barrière infrarouge, triangulation sonore,...), et interconnectées. Grâce à cela, nous sommes à même de mettre en œuvre un CPS, il ne reste plus qu'à le piloter de manière robuste.

Bibliographie

Arduino. *Arduino Official Website*. 2014. <http://www.arduino.cc/fr/#.UzTkAlcimp>.

Berry, Gérard. *Leçon Inaugurale du Collège de France / L'informatique du temps et des événements*. Collège de France / Fayard, 2013.

Cox, Simon. *Steps to make Raspberry Pi Supercomputer*. Édité par University of Southampton. 09 01 2013.
http://www.southampton.ac.uk/~sjc/raspberrypi/pi_supercomputer_southampton.htm
(accès le 03 27, 2014).

Daniel Hefenbrock, Jason Oberg, Nhat Tan Nguyen Thanh, Ryan Kastner, Scott B. Baden.
«Accelerating Viola-Jones Face Detection to FPGA-Level using GPUs.» *18th IEEE Annual International Symposium on Field-Programmable Custom Computing Machines*. Charlotte (USA), 2010.

Finkenzeller, Klaus. *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*. John Wiley & Sons Ltd, 2003.

Fondation Raspberry Pi. *Site Officiel Raspberry Pi*. 2014. <http://www.raspberrypi.org/>.

INRIA. *The Coq proof assistant*. 2014. <http://coq.inria.fr/>.

NMRA. «Communications Standards For Digital Command Control.» 2004.

NMRA. *DCC Standards & Recommended Practices Index*
http://www.nmra.org/standards/DCC/standards_rps/DCCStd.html. 14 Avril 2013.

NMRA. «Electrical Standards For Digital Command Control.» 2004.

OpenCV. «Face Detection using Haar Cascades.» *OpenCV Documentation*. s.d.
http://docs.opencv.org/trunk/doc/py_tutorials/py_objdetect/py_face_detection/py_face_detection.html (accès le 03 27, 2014).

Paul Viola, Michael Jones. «Rapid Object Detection using a Boosted Cascade of Simple Features.» *IEEE CVPR*, 2001.

Pleumann, Joerg. *Railuino Project*. s.d. <https://code.google.com/p/railuino/> (accès le 03 27, 2014).

Seshia, E. A. Lee and S. A. *Introduction to Embedded Systems - A Cyber-Physical Systems Approach*. LeeSeshia.org, 2011.

Walook. *Walook - Projet de Ballon Sonde Arduino*. 2014. <http://walook.fr/index.html>.

Wolf, Wayne. «The Good News and the Bad News.» *Embedded Computing Column IEEE Computer*, 2007.