

Licence d'Informatique  
Rapport de Stage  
Du 5 novembre 2015 au 27 juin 2016

Interface de pilotage  
du robot "Thymio" avec I-score

**Stagiaire :** Gaëtan CHAMBRES

**Responsable de stage :** Pierre COCHARD

**Co-dirigeant de stage :** Phillippe GUILLEM

Stage effectué au sein de :  
Université de Bordeaux  
351 cours de la libération  
33405 Talence cedex

# Remerciements

A RÉDIGER PLUS TARD

# Résumé

A RÉDIGER PLUS TARD

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Domaine du Stage . . . . .	1
1.1.1	Le SCRIME [?] . . . . .	1
1.1.2	Les missions du SCRIME [?] . . . . .	1
1.1.3	L'équipe du SCRIME [?] . . . . .	2
1.2	Le contexte du stage . . . . .	2
1.2.1	Pourquoi ce projet ? . . . . .	2
1.2.2	Les objectifs du stage . . . . .	3
1.3	Contenu du rapport . . . . .	3
<b>2</b>	<b>Le Robot Thymio</b>	<b>4</b>
2.1	Présentation [?] . . . . .	4
2.2	Philosophie d'un projet collaboratif[?] . . . . .	4
2.3	Les différents acteurs du projet [?] . . . . .	5
2.4	Les engagements de la société créatrice [?] . . . . .	6
<b>3</b>	<b>Thymio et le son [?]</b>	<b>7</b>
3.1	Détection de l'intensité du son . . . . .	7
3.2	Son de synthèse . . . . .	7
3.3	Modification de l'onde primaire . . . . .	7
3.4	Enregistrement . . . . .	8
3.5	Relecture . . . . .	8
3.6	Création de son depuis un ordinateur . . . . .	8
3.6.1	Son système . . . . .	9
3.6.2	Bibliothèque de son système . . . . .	9
<b>4</b>	<b>Les outils informatiques utilisés</b>	<b>10</b>
4.1	Qu'est-ce qu'Aseba ?[?] . . . . .	10
4.2	Le langage de Thymio : ASEBA [?] . . . . .	10

<b>5</b>	<b>Récupérer les événements</b>	<b>12</b>
5.1	Le transfert d'information entre Thymio et un ordinateur . . . . .	12
5.2	La communication locale . . . . .	12
5.3	Le mapping des capteurs . . . . .	13
5.3.1	Les boutons [?] . . . . .	13
5.3.2	Les capteurs de distance horizontaux . . . . .	13
<b>6</b>	<b>Extraire l'événement de Thymio</b>	<b>14</b>
6.1	Aseba exec . . . . .	14
6.2	Aseba dump . . . . .	14
6.3	script avec inscription dans un fichier . . . . .	14
<b>7</b>	<b>Bilan</b>	<b>15</b>
7.1	Apports du stage . . . . .	15
7.2	Conclusion . . . . .	15
7.3	Perspective . . . . .	15
<b>8</b>	<b>Annexe</b>	<b>16</b>

# Chapitre 1

## Introduction

### 1.1 Domaine du Stage

#### 1.1.1 Le SCRIME [?]

Ce stage a été effectué au sein du Studio de Création et de Recherche en Informatique et Musiques Expérimentales( ou Électro-acoustiques).

Ce groupement d'intérêt scientifique et artistique est rattaché à l'équipe Image et Son du Laboratoire Bordelais de Recherche en Informatique (LaBRI).

Les champs d'activités du SCRIME s'étendent de la recherche scientifique à la création artistique, en passant par la formation, la diffusion des musiques contemporaines et la pédagogie en milieu scolaire et universitaire.

Le SCRIME est soutenu par le Ministère de la Culture, le Conseil Régional d'Aquitaine, la DRAC et le CNRS.

#### 1.1.2 Les missions du SCRIME [?]

Les différentes missions du SCRIME sont :

- Favoriser la collaboration entre scientifiques et artistes à des fins de recherche et de création
- Favoriser la diffusion internationale des résultats des recherches et des oeuvres artistiques
- Promouvoir les collaborations scientifiques et artistiques avec d'autres centres de recherche et de création en France ou à l'étranger
- Favoriser les actions pédagogiques dans les institutions partenaires ainsi que dans les autres établissements publics d'enseignement et de création et ce à tous les niveaux

### 1.1.3 L'équipe du SCRIME [?]

L'équipe de ce groupement d'intérêt scientifique et artistique est essentiellement composée de :

**Myriam Desainte-Catherine** : Direction scientifique

**Jean-Louis Agobet** : Direction artistique

**Gyorgy Kurtag** : Coordination Arts et Sciences

**Annick Mersier** : Gestionnaire

**Christian Faurens** : Assistant ingénieur audiovisuel

**Pierre Cochard** : Réalisateur en Informatique Musicale

**Antoine Hubineau** : Chargé de communication

**Julia Hanadi Al Abed** : Régisseur son

**Jean-Michel Rivet** : Compositeur et Enseignant

**Raphaël Marczak** : Ingénieur de recherche

**Nicolas Vuaille** : Développeur

Pierre COCHARD aura été mon responsable de stage, mais j'ai également travaillé avec Philippe GUILLEM, professeur de musique et enseignant en classe de maternelle.

## 1.2 Le contexte du stage

### 1.2.1 Pourquoi ce projet ?

Philippe GUILLEM est maître formateur en musique et enseignant en école maternelle. Son objectif est de montrer à ses élèves ce que le monde moderne leur offre. Cela va des réseaux sociaux à la robotique, en passant par la programmation de manière plus vaste. Seulement une interrogation naissait en même temps que cette idée :

Comment chez des élèves de 5 ans, aider à construire une image du Robot et des relations que l'homme peut entretenir avec ces machines ?

Lors d'une conférence, Philippe GUILLEM a découvert le projet du robot THY-MIO. De part sa robustesse, sa simplicité d'accès, d'utilisation et de programmation, ainsi que par sa conception, il était tout à fait adapté aux jeunes enfants et donc à ce projet. Enfin, il restait la question de représenter la relation homme / robot. Pour cela, Mr GUILLEM a tenu à faire un lien avec la musique, car c'est ce qu'il y a de plus parlant pour les jeunes auditorats. C'est donc à ce moment que

le SCRIME a été sollicité, en particulier Pierre COCHARD, et ils ont travaillé sur le projet de ce stage, qui finira par m'être proposé.

### **1.2.2 Les objectifs du stage**

L'objectif principal de ce stage était de réaliser une interface permettant de piloter le robot Thymio avec le logiciel i-score afin d'en permettre l'utilisation en classe avec des élèves de maternelle par monsieur Phillipe GUILLEM.

Dans ce but, ce stage devait permettre d'implémentation d'un système de communication entre Pyo (ou tout autre logiciel adapté) et i-score.

L'objectif pédagogique était d'obtenir des compétences en robotique et dans le traitement des informations d'un capteur.

## **1.3 Contenu du rapport**



# Chapitre 2

## Le Robot Thymio

### 2.1 Présentation [?]

Thymio, conçu par l'École polytechnique fédérale de Lausanne est un petit robot dédié à la pédagogie. Ses deux roues lui permettent déplacement et manoeuvres, tandis qu'il embarque des boutons ainsi que de nombreux capteurs infrarouges pour détecter des obstacle ou des surfaces, un micro, un haut parleur, un accéléromètre et même un thermomètre. Son accessibilité est due à la mise à disposition de ses propres interfaces de programmation. Il y en a en effet plusieurs, à commencer par l'interface de programmation texte, ASEBA STUDIO qui permet d'exploiter entièrement le langage ASEBA. Une adaptation de BLOCKY est aussi disponible, qui permet d'aborder la programmation texte de manière simplifiée, et enfin il y a VPL qui permet de programmer THYMIO visuellement grâce à des vignettes. C'est grâce à ces interfaces que ce robot se trouve adapté aux plus jeunes.

Pour résumer, il s'agit d'un robot complet, qui embarque un grand nombre de capteurs, et accessible à tous, aux plus jeunes comme aux programmeurs confirmés, c'est pourquoi ce robot a été choisi pour ce projet de stage.

### 2.2 Philosophie d'un projet collaboratif[?]

Le but du projet THYMIO était de permettre à un large publique, et en particulier aux enfants, la découverte et l'apprentissage de la programmation, de la robotique, de l'ingénierie et des technologies numériques. Pour permettre cela, il a fallut regrouper plusieurs corps de métiers et des connaissances poussées dans différents domaines. C'est pourquoi THYMIO est un projet communautaire, qui implique plusieurs entités. Il s'agit donc d'un modèle basé sur le partage libre des connaissances, c'est à dire que Tout le matériel, le logiciel, et la documentation

sont développés par les partenaires sous licence libre. Ainsi, les partenaires s'engagent à fournir l'information de conception gratuitement, de manière à ce que chacun puisse comprendre, utiliser, ou modifier cette information, à condition de distribuer à son tour les résultats de son travail. Il est donc important que ce stage s'inscrive dans cette dynamique, et que les travaux ainsi que les résultats soient partagés à la communauté.

## 2.3 Les différents acteurs du projet [?]

Comme expliqué précédemment, plusieurs catégories d'acteurs se sont alliés dans ce projet :

**Les institutions de recherche :** qui permettent de diffuser le travail et de l'amener auprès du public. Plus la base d'utilisateurs grandit, meilleures sont les possibilités d'évaluation et d'amélioration. Les résultats de la recherche sont ainsi mis rapidement en application.

**Les entités de production :** qui donnent accès à des produits proches de l'état de l'art qui n'ont plus qu'à être industrialisés. Les frais liés à des licences sont réduits et les produits sont validés scientifiquement. Les entités de production distribuent toute l'information (schémas, plans, pdf des exercices, etc.) gratuitement et commercialisent les supports physiques (robots, accessoires, livrets imprimés, etc.).

**Les contributeurs** permettent de participer à un grand projet selon ses moyens et de choisir son implication, de proposer des idées, de recevoir de l'aide et de l'information et de voir son travail diffusé.

**Les utilisateurs** leur apport se perçoit dans du matériel moins cher, beaucoup de ressources gratuites, une transparence dans le projet, et une plus grande pérennité du produit.

Parmi les principaux acteurs, on pourra citer en particulier :

**EPFL - École Polytechnique Fédérale de Lausanne** Initiatrice du projet.

**Association Mobsya** Association responsable de la production et de la distribution du robot.

**ETH - Eidgenössische Technische Hochschule Zürich** Le laboratoire de Systèmes Autonomes et le centre de technologies de jeu de l'ETH Zürich qui ont contribué à ASEBA et à VPL.

**NCCR Robotics** Aide financière et aide à la validation scientifique.

**la Loterie Romande, Suisse**

**et le Fond National Suisse de la Recherche Scientifique** Respectivement pour l'aide financière pour les premières production et l'aide financière pour la mise en place de kits éducatifs

**INRIA - L'Institut National de Recherche en Informatique et en Automatique**

Développeur principal de l'adaptation de SCRATCH au THYMIO, qui donnera BLOCKY, et développement du matériel éducatif pour les enseignants, notamment avec le projet "Dessine-moi un robot".

L'INRIA nous aura grandement aidé dans notre projet, grâce notamment au contact de Didier Roy; docteur en informatique cognitive, lié de près au projet Thymio, qui nous aura aidé dans nos réflexions, mais aussi prêté un THYMIO de test pendant plusieurs mois avant que le SCRIME n'ai pu en faire l'acquisition.

## 2.4 Les engagements de la société créatrice [?]

Pour suivre cette philosophie basée sur la collaboration et les sources ouvertes (open source), la société créatrice a choisi les modes de diffusion suivants :

- pour le logiciel, le code source est en open source, sous licence LGPL.
- pour le matériel, sont distribués tous les documents de conception, y compris schémas électronique, plans des pièces, etc. en open hardware sous une licence CC-BY-SA 3.0
- pour la documentation, les exercices, les cursus, sont distribués également tous les documents en format numérique sous une licence CC-BY-SA 3.0.

Afin de permettre une production et une distribution à grande échelle, une association à but non lucratif a été créée, Mobsya. L'association assure la production du matériel. Pour le logiciel, elle contribue à la maintenance et au développement en collaboration avec les autres partenaires. L'association gère aussi l'infrastructure qui permet de créer une communauté d'utilisateurs, qui peuvent s'entraider grâce à des listes de discussion et un site de partage d'information qui contient la documentation, des exemples de réalisation, du support technique, etc. Cette infrastructure aura également aidé durant ce stage.

# Chapitre 3

## Thymio et le son [?]

### 3.1 Détection de l'intensité du son

Le Thymio peut détecter lorsque le bruit ambiant est au-dessus d'une intensité donnée et émet un événement.

La variable `mic.intensity` montre l'intensité du microphone en cours, tandis que la variable `mic.threshold` contient la limite de l'intensité pour l'événement. Si `mic.intensity` est au-dessus de `mic.threshold`, l'événement `mic` est généré. Lecture et enregistrement de sons.

Vous pouvez jouer des sons de synthèse ou du système. De plus, si vous avez installé une carte micro-SD formatée FAT dans le robot, il est possible d'enregistrer et de jouer vos propres sons. Les fichiers sont stockés sur la carte micro-SD au format wave, 8-bit non-signé, 8 kHz. Lorsque Thymio fini la lecture d'un son demandée par ASEBA, il déclenche l'événement `sound.finished`. Il ne se déclenche pas un événement si la lecture est interrompue ou si un nouveau son est joué.

### 3.2 Son de synthèse

La fonction native `sound.freq` joue une fréquence, spécifiée en Hz, pour une certaine durée, spécifié dans  $1/60$  s. Une durée égale à 0 joue le son à l'infini et une durée de -1 arrête le son.

### 3.3 Modification de l'onde primaire

La génération de sons de synthèse fonctionne par ré-échantillonnage d'une onde primaire. Par défaut, l'onde est triangulaire, mais vous pouvez définir votre propre

onde à l'aide de la fonction native `sound.wave`. Cette fonction prend en entrée un tableau de 142 échantillons, avec des valeurs de -128 à 127. Ce tableau doit représenter une onde tonique de la fréquence spécifiée dans `sound.freq`. Comme Thymio joue des sons à 7812,5 Hz, ce tableau est joué entièrement à la fréquence de  $7812.5/142 = 55$  Hz. Jouer un son de fréquences plus élevées fait sauter des échantillons dans le tableau.

### 3.4 Enregistrement

Vous pouvez enregistrer des sons en utilisant la fonction native `sound.record` qui prend en paramètre un numéro d'enregistrement de 0 à 32767. Le fichier est stocké sur la carte micro-SD sous le nom `Rx.wav` où `x` est le paramètre passé à la fonction `sound.record`. Pour arrêter un enregistrement, appelez la fonction `sound.record` avec la valeur -1.

### 3.5 Relecture

Vous pouvez rejouer un son enregistré en utilisant la fonction native `sound.replay`. Cette fonction prend en paramètre un nombre de 0 à 32767 et rejouer le fichier `Rx.wav` de la carte SD où `x` est le paramètre passé à la fonction `sound.replay`. Pour arrêter une lecture, veuillez appeler la fonction `sound.replay` avec une valeur de -1.

### 3.6 Création de son depuis un ordinateur

Vous pouvez créer des sons pour le Thymio II depuis votre ordinateur. La façon plus efficace de le faire est d'utiliser le logiciel audacity, version 1.3, qui existe pour les différents OS. Voici la démarche pour créer un son compatible avec Thymio II :

- Une fois démarré audacity, changez le project rate situé en bas à gauche de 44100 Hz (défaut) à 8000 Hz.
- Enregistrez votre son avec la touche record rouge en haut à gauche. Vous devez voir le curseur avancer et la forme d'onde se visualiser. Stoppez une fois fini avec le bouton stop.
- Votre son doit être en mono (Piste->Piste stéréo vers mono)
- Allez sur le menu File sous Export...
- Donnez un nom de fichier, par exemple `P0.wav` pour le premier fichier à jouer en utilisant la fonction native `sound.play`.
- Choisissez comme format : other uncompressed files

- Sous options choisissez un header WAV (Microsoft) et comme Encoding choisissez Unsigned 8 bit PCM.
- Assurez que le métadonnées de le fichier ne sont pas valorisé.
- Sauvez ou copiez le fichier sur une carte micro-SD.

Pour faire un test rapide vous pouvez sauver ou copier ce fichier sous le nom de S0.wav. Il sera joué au démarrage du Thymio II.

Vous pouvez jouer des sons en utilisant la fonction native `sound.play` qui prend en paramètre un numéro d'enregistrement de 0 à 32767. Le fichier doit être stocké sur la carte micro-SD sous le nom Px.wav où x est le paramètre passé à la fonction `sound.play`. Pour arrêter de jouer un son, appelez la fonction `sound.play` avec la valeur -1.

### 3.6.1 Son système

Vous pouvez jouer un son système à l'aide de la fonction native `sound.system`, qui prend un nombre de 0 à 32767 en tant que paramètre. Certains sons sont disponibles dans le firmware (voir ci-dessous), mais vous pouvez remplacer ces sons et ajouter de nouveaux en utilisant la carte micro-SD. Dans ce cas, le fichier doit être nommé Sx.wav où x est le paramètre passé à la fonction `sound.system`. Pour arrêter la lecture d'un son, appelez la fonction `sound.system` avec la valeur -1.

### 3.6.2 Bibliothèque de son système

Les sons suivant sont disponibles :

paramètre description

-1 arrête de jouer un son

0 son de démarrage

1 son d'arrêt (son non reconfiguration)

2 son des boutons fléchés

3 son du bouton central

4 son de chute libre (peureux)

5 son de choc

6 son de suivi d'objet

7 son de détection d'objet pour suivi

Pour ne plus entendre de son système, vous pouvez décompresser sur une carte micro-SD cette archive qui contient des sons muets.

# Chapitre 4

## Les outils informatiques utilisés

### 4.1 Qu'est-ce qu'Aseba ?[?]

*Aseba est un ensemble d'outils permettant à des novices de programmer des robots facilement et efficacement. Pour ces raisons, Aseba est bien adapté à la recherche [1, 5, 2] et l'enseignement [4] en robotique. Aseba est open-source (license GNU Lesser General Public License), vous pouvez donc le télécharger et jouer avec. Techniquement, Aseba est une architecture de contrôle temps réel et distribué de robot mobile, basée sur des événements. Aseba vise les robots à plusieurs micro-contrôleurs ou les groupes de robots à simple micro-contrôleurs, réels ou simulés. Le cœur d'Aseba est une machine virtuelle légère, suffisamment compacte pour fonctionner sur micro-contrôleurs. Aseba permet de programmer les robots dans un langage ergonomique et simple d'accès, à partir d'un environnement de développement intégré*

### 4.2 Le langage de Thymio : ASEBA [?]

*Le langage de programmation d'Aseba permet de programmer le comportement des nœuds Aseba. Syntaxiquement, ce langage ressemble à une classe populaire de langages, incluant Pascal et Matlab (un langage de programmation scientifique commun) par exemple ; nous nous attendons à ce que cette ressemblance permette aux développeurs ayant des connaissances préalables de ces langages de se sentir à l'aise et d'apprendre rapidement. Sémantiquement, Aseba est un langage de programmation impératif avec un seul type simple (nombres entiers signés sur 16 bits) et des tableaux. Cette simplicité permet aux développeurs sans connaissances préalables d'un système de type de programmer des comportement ; les nombres entiers étant le type de variable le plus naturel. De plus, les nombres entiers sont particulièrement adaptés aux robots à micro-contrôleurs.*

Il est à noter que ce langage n'est utilisé que pour le robot Thymio.



# Chapitre 5

## Récupérer les événements

Le coeur du stage était composé de 2 missions : récupérer les événements du robot (comme une simulation des capteurs) et les extraire afin de les utiliser comme événements dans i-score et créer ainsi de la musique.

### 5.1 Le transfert d'information entre Thymio et un ordinateur

le stage a été réalisé sur 2 Thymio : un classique et un "wireless". Le Thymio classique ne possède pas de système de communication à distance ce qui rend la récupération de données pour un ordinateur très compliqué. Une possibilité de transfert par onde radio à été envisagée mais le Thymio wireless a été commercialisé. Ce robot est spécialement étudié pour communiquer à distance avec un ordinateur ce qui rend le transfert d'information plus facile.

### 5.2 La communication locale

Thymio peut utiliser ses capteurs horizontaux de distance infrarouge pour communiquer une valeur à un autre robot à proximité dans un rayon de 15 cm. La valeur est envoyée à 10 Hz lors du traitement des capteurs de distance. Thymio envoie une valeur de 11 bits (mais un futur firmware pourrait utiliser 1 bits pour utilisation interne, il est donc préférable de se limiter à 10 bits). Pour utiliser la communication, appelez la fonction `prox.comm.enable(state)`, avec 1 comme paramètre `state` pour enclencher et 0 pour désactiver. Si la communication est activée, la valeur dans `prox.comm.tx` est transmise toutes les 100 ms. Lorsque Thymio reçoit une valeur, l'événement `prox.comm` est déclenché et la valeur peut être lue dans la variable `prox.comm.rx`.

## 5.3 Le mapping des capteurs

### 5.3.1 Les boutons [?]

Thymio possède 5 boutons capacitifs correspondant aux flèches et au bouton central. Un tableau de 5 variables, `buttons.binary`, contient l'état de ces boutons (1 = appuyé, 0 = relâché) :

- `button.backward` : flèche arrière
- `button.left` : flèche gauche
- `button.center` : bouton central
- `button.forward` : flèche avant
- `button.right` : flèche droite

Thymio met à jour ce tableau à une fréquence de 20 Hz, et génère l'événement `button` après chaque mise à jour. En outre, pour chacune de ces touches, quand elle est appuyée ou relâchée, un événement correspondant avec le même nom est émis.

### 5.3.2 Les capteurs de distance horizontaux

Thymio possède 7 capteurs de distances autour de lui. Un tableau de 7 variables, `prox.horizontal`, contient les valeurs de ces capteurs :

- `prox.horizontal[0]` : avant gauche
- `prox.horizontal[1]` : avant milieu-gauche
- `prox.horizontal[2]` : avant milieu
- `prox.horizontal[3]` : avant milieu-droite
- `prox.horizontal[4]` : avant droite
- `prox.horizontal[5]` : arrière gauche
- `prox.horizontal[6]` : arrière droite

Les valeurs dans ce tableau varient de 0 (le robot ne voit rien) à plusieurs milliers (le robot est très proche d'un obstacle). Thymio met à jour ce tableau à une fréquence de 10 Hz, et génère l'événement `prox` après chaque mise à jour.

# Chapitre 6

## Extraire l'événement de Thymio

Plusieurs pistes ont été étudiées :

### 6.1 Aseba exec

piste avec lancement d'un événement dans i-score

### 6.2 Aseba dump

piste avec lancement d'un script

### 6.3 script avec inscription dans un fichier

# Chapitre 7

## Bilan

7.1 Apports du stage

7.2 Conclusion

7.3 Perspective



# Chapitre 8

## Annexe



Objet : **Prêt de robots Thymio**

Je soussigné Gaëtan Chambres, résident Résidence Edmond Rostand, entrée 2 appartement 34, 33185 Le Haillan, étudiant à l'Université de Bordeaux en 3ème année de licence informatique, certifie avoir en prêt 1 robot Thymio numéroté 28, fourni par la médiation scientifique Inria Bordeaux.

Je les restituerai au plus tard le 18 décembre. Inria peut à tout moment demander la restitution des robots en cas de besoin.

En contrepartie, il est demandé de valoriser les activités IniRobot produites par Inria, de remplir les questionnaires disponibles sur le site inirobot.fr en prévision d'un article scientifique à venir, de transmettre des statistiques d'utilisation et des retours d'utilisation, de rédiger des retours d'expérience publiables sur le site web inirobot.fr ou dm1r.inria.fr, accompagnés si possible d'une ou plusieurs photos ou de liens vers des vidéos.

Fait à Bordeaux, le 18 Novembre 2015

Signature M<sup>r</sup> CHAMBRES  
Gaëtan