

Rapport Projet MAKI

Robot-Visage pour enfants autistes



KOENIG Pierre-Olivier
CAILLOT Adrien



Licence Professionnelle SESAM
Groupe 1

Sommaire

Table des matières

Contexte	3
Introduction.....	3
Cahier des charges.....	3
1. Partie LABVIEW.....	4
1.1) Le fonctionnement du logiciel :	4
1.2) Les améliorations :	4
1.3) Le Media Player :	5
2. Partie microprocesseur	6
2.1) Choix du microprocesseur	6
2.2) Configuration de la Raspberry	6
2.3) Communication avec la Raspberry	6
2.4) Partage de connexion Internet Raspberry / PC.....	8
2.5) Implémentation du code LABVIEW dans la Raspberry.....	10

Table des figures

Figure 1: Nouvelle interface utilisateur LABVIEW	4
Figure 2: Interface du Média Player	5
Figure 3: Contrôle réussi de la Raspberry par SSH	7
Figure 4: Changement de mot de passe.....	7
Figure 5: Configuration câble Ethernet IPv4	8
Figure 6: Partage de connexions Internet	8
Figure 7: Configuration correcte de la passerelle	9
Figure 8: Connexion Internet de la Raspberry réussie	9

Contexte

Aujourd'hui, en France on estime à 650 000 le nombre de personnes touchées les maladies du Troubles du Spectre Autistique (TSA). Parmi ces 650 000 personnes, se situent de nombreux enfants qui ne sont très peu scolarisée (20%). Ces enfants ont notamment des soucis à reconnaître les émotions humaines. Pour cela, l'IUT de Cachan développe un robot d'apprentissage appelée MAKI.

Notre équipe s'occupe uniquement de la partie software dans ce projet, une autre équipe s'occupe de la partie hardware.

Introduction

Le robot MAKI est une tête d'enfant robotisée capable de reproduire les émotions humaines. C'est un projet s'étendant sur plusieurs années à l'IUT de Cachan et donc bien connus de tous. Nous avons donc eu accès à la version précédente du logiciel. Notre équipe travaillant sous environnement LabVIEW en entreprise, ce projet est donc idéal pour appliquer nos connaissances apprises à nos bureaux respectifs en cours. Notre travail principal consiste à développer un code LabVIEW modulaire, bien écrit et documenté afin que la prise en main pour les étudiants futurs soit simplifiée. Le deuxième objectif de notre projet consiste lui, à venir assurer la portabilité du logiciel sur une Raspberry pi afin de simplifier son déploiement et/ou coût de mise en place chez les institutions éducatives.

Cahier des charges

- Commander sous LabVIEW des servomoteurs pour réaliser des émotions du visage
- Mettre en œuvre un système embarqué de type MyRio ou Raspberry PI 3
- Ajout de fichiers de configuration des émotions permettant des améliorations futures
- Enregistrement des profils des utilisateurs pour permettre un suivi individualisé
- Bilan visuel des compétences acquises par l'enfant

1. Partie LABVIEW

1.1) Le fonctionnement du logiciel :

Le logiciel vient initialiser la lire des fichiers .ini afin de récupérer les informations des émotions à exécuter connexion avec des moteurs XL -320 de la marque Dynamixel en liaison série protocole RS-485, et ce dès son lancement. Il vient également

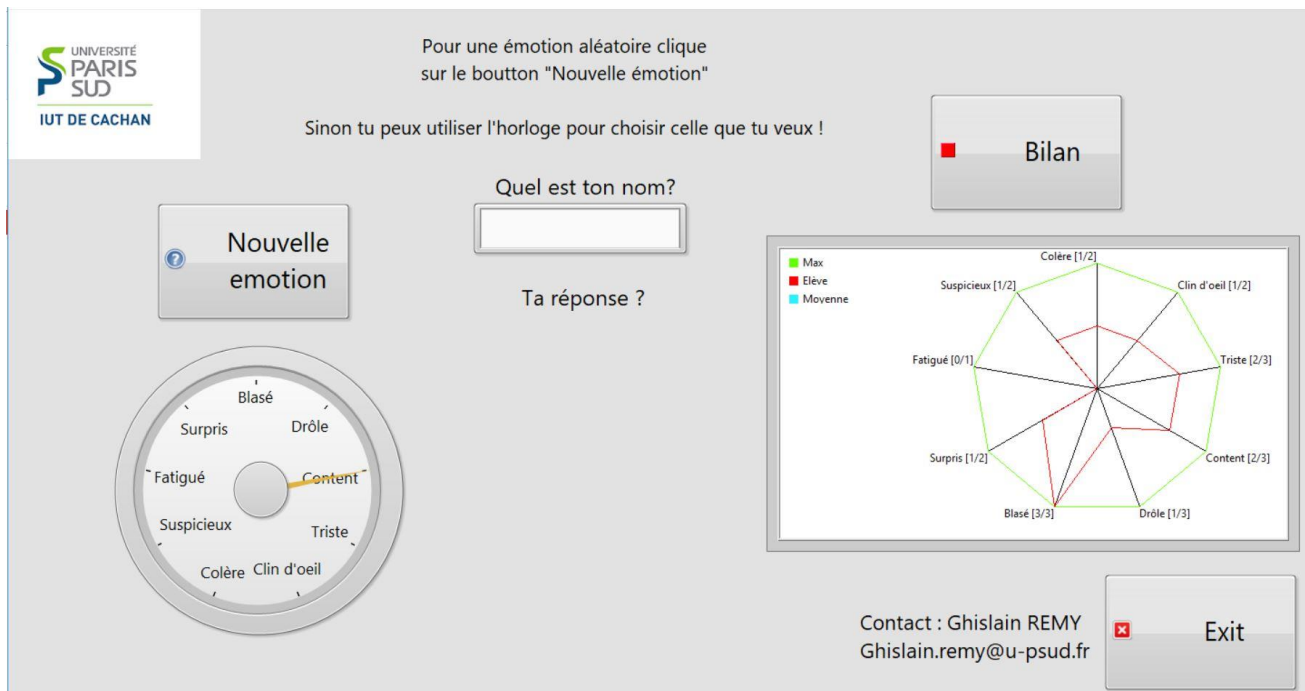


Figure 1: Nouvelle interface utilisateur LABVIEW

On peut voir sur l'interface ci-dessus, une « horloge » permettant à l'utilisateur de choisir une émotion précise, il y a également la possibilité de générer une émotion aléatoire en cliquant sur le bouton « Nouvelle émotion ». Ensuite le logiciel ordonne aux moteurs de se déplacer selon un angle précis afin de reproduire les émotions. Une fois que le test est terminée, l'utilisateur peut cliquer sur le bouton « Bilan » afin d'avoir un résumé des résultats sous la forme d'un diagramme de pieuvre (cf interface MAKI). L'utilisateur a la possibilité de venir rentrer son nom afin de bénéficier d'une troisième courbe dans le diagramme retraçant ses résultats.

1.2) Les améliorations :

Les améliorations de notre logiciel sont :

- Code plus modulaire, mise en place d'une machine à états gestionnaire de message via une FIFO.
- Utilisation d'un template National Instrument comme architecture de base.
- Utilisation plus adaptée de Sous-VI afin de rendre le code plus lisible.
- Documentation plus approfondie du code.
- Rajout du suivi de l'élève.

- Nouvelle interface, plus ludique

Cependant le point le plus important est le fait que, l'utilisateur peut modifier les données et rajouter /supprimer des émotions sans ouvrir LabVIEW. En effet le logiciel vient lire un fichier.ini dès son lancement. L'utilisateur peut donc modifier le fichier .ini comme il souhaite (à condition de respecter la syntaxe National Instrument) et le code s'adaptera. Un utilisateur ne connaissant pas LabVIEW peut donc rajouter des émotions et piloter les moteurs.

1.3) Le Media Player :

Nous avons également développé une deuxième application qui utilise des vidéos montrant un acteur qui simule les émotions.

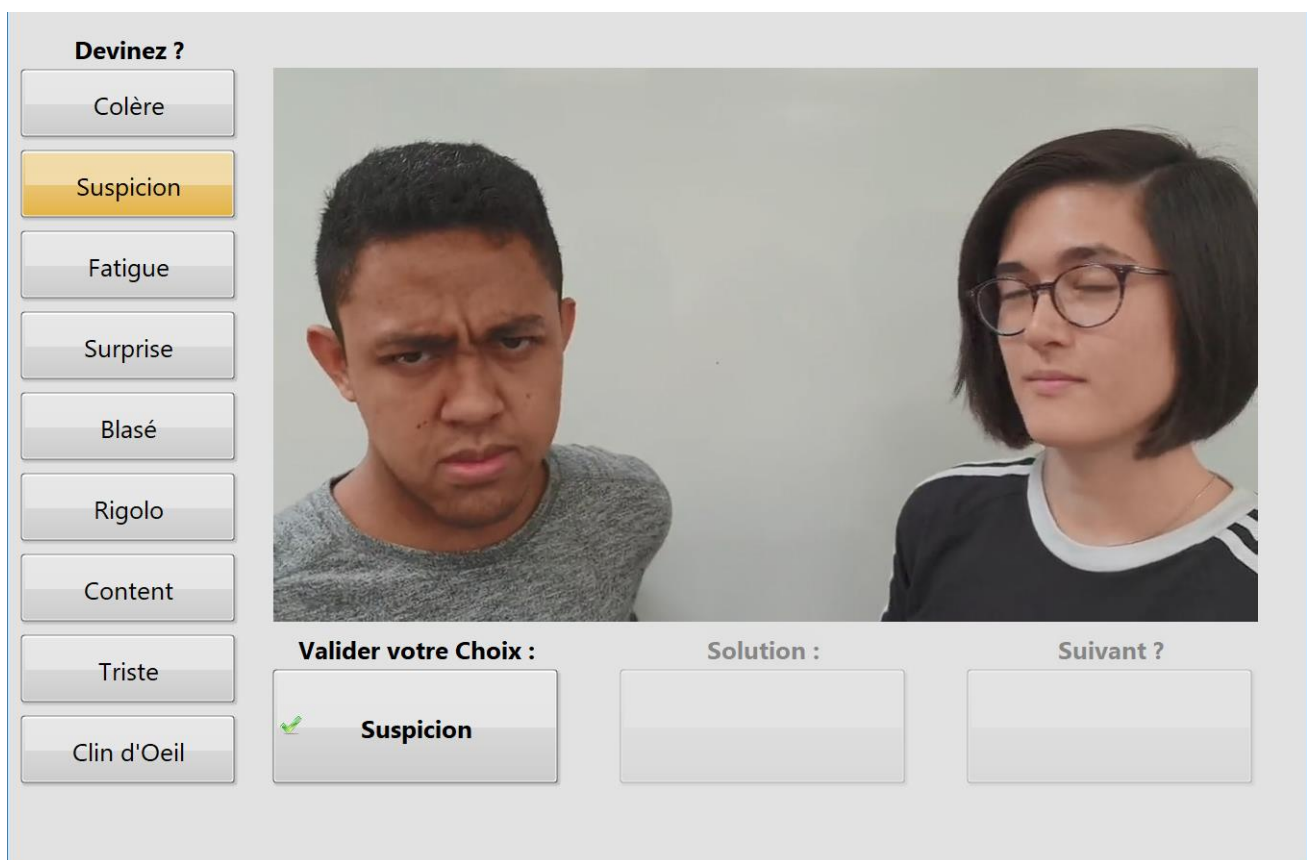


Figure 2: Interface du Média Player

On peut voir sur la partie gauche de l'interface, la liste des émotions possible. L'utilisateur est invité à choisir l'émotion qu'il croit reconnaître. Ensuite il doit valider son choix avec le bouton « Valider votre choix ». La réponse est ensuite affichée dans la case « Solution ». L'utilisateur peut ensuite cliquer sur suivant pour jouer la prochaine vidéo.

2. Partie microprocesseur

2.1) Choix du microprocesseur

La deuxième partie de notre projet a pour but d'implémenter notre code LABVIEW dans un système embarqué afin d'éviter l'utilisation d'un ordinateur qui peut être encombrant.

Initialement, nous devons utiliser un **NI MyRio** pour récupérer le code LABVIEW mais nous avons remarqué que le boîtier ne possédait pas de connectiques adaptées pour afficher une interface graphique sur un écran TV ou un moniteur d'ordinateur.

Nous nous sommes donc focalisés sur une **Raspberry Pi 3 model B** qui possède une prise HDMI.

2.2) Configuration de la Raspberry

Dans un premier temps, il faut installer le système d'exploitation dans une carte Micro SD pour utiliser la Raspberry.

Nous avons suivi le tutoriel vidéo sur YouTube suivant : [\[lien de la vidéo\]](#)

Pour notre carte, nous avons installé la version 2.8.1 de NOOBS mise à jour le 24-04-2018.

Attention ! L'installation de NOOBS peut durer très longtemps !

Dans notre cas, il a fallu 2h d'installation et environ 7.5 Go de mémoire dans la carte Micro SD.

2.3) Communication avec la Raspberry

Durant notre projet, nous avons communiqué avec la carte avec deux méthodes :

- Avec l'interface graphique, en utilisant un écran, une souris et un clavier;
- Sans interface, en la contrôlant directement par notre PC en protocole SSH.

Nous avons utilisé l'interface graphique afin d'activer le SSH qui est désactivé par défaut et pour fixer une adresse IP fixe pour la Raspberry.

Activer le SSH :

- Ouvrir la console de commande Raspberry et entrer la commande :
sudo raspi-config
- Sélectionner *Interfacing Options > SSH*.
- Choisir "yes" dans la fenêtre de confirmation.
- Sélectionner "OK" puis "Finish".

Déterminer une adresse IP fixe :

- Entrer dans le fichier *cmdline.txt* en tapant la commande suivante :
sudo nano /boot/cmdline.txt
- Insérer à la fin de la ligne et sans saut votre adresse IP :
[...] rootwait ip=192.168.137.3
- Quitter le fichier en utilisant Ctrl + X.
- Un message de confirmation de sauvegarde apparaît, appuyer sur la touche "o" puis sur "Entrée".
- Vérifier le changement d'adresse en utilisant la commande :
ifconfig

```
pi@raspberrypi:~ $ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.137.3 netmask 255.255.255.0 broadcast 192.168.137.255
    inet6 fe80::ba27:ebff:fe03:fc1f prefixlen 64 scopeid 0x20<link>
    ether b8:27:eb:03:fc:1f txqueuelen 1000 (Ethernet)
    RX packets 2630 bytes 215939 (210.8 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1602 bytes 200925 (196.2 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Ensuite nous utilisons la Raspberry sans interface à cause d'une contrainte de matériel.

Méthode pour contrôler la Raspberry en SSH (sans deuxième écran) :

- Sous Windows, nous avons besoin d'un logiciel nommé [Putty](#).
- Lancer Putty et insérer l'IP fixe de la Raspberry.
- Aller dans *Connection > SSH > X11* et cocher *Enable X11 forwarding*.
- Cliquer sur "Open" et taper le login et le mot de passe (par défaut : login = *pi* et password = *raspberrypi*).

Remarque : Sous LINUX, le mot de passe ne s'affiche pas lorsque l'on appuie sur le clavier.



```
pi@raspberrypi: ~
login as: pi
pi@192.168.0.2's password:
Linux raspberrypi 4.14.34-v7+ #1110 SMP Mon Apr 16 15:18:51 BST 2018 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu May 24 12:45:11 2018
pi@raspberrypi:~ $
```

Figure 3: Contrôle réussi de la Raspberry par SSH

Notre nouveau mot de passe : *pi*

Pour changer le mot de passe, taper la commande suivante :

sudo passwd

```
pi@raspberrypi:~ $ sudo passwd
Entrez le nouveau mot de passe UNIX :
Retapez le nouveau mot de passe UNIX :
passwd: password updated successfully
pi@raspberrypi:~ $
```

Figure 4: Changement de mot de passe

2.4) Partage de connexion Internet Raspberry / PC

Notre PC est connecté par Internet grâce au Wifi, nous effectuons un partage de connexion entre la carte Wifi du PC et le câble Ethernet relié à la Raspberry puis nous configurons une passerelle pour que la Raspberry accède à Internet.

Configuration Internet du PC

- Ouvrir l'invite de commandes Windows et taper la commande suivante pour repérer le port de votre câble Ethernet :

Ipconfig

Pour nous : Ethernet

- Ouvrir le Centre Réseau et partage : clic droit sur l'icône de connexion Internet.
- Sélectionner "Connexions : Ethernet" (Double clic).
- Dans "Etat de Ethernet", choisir *Propriétés > Protocole Internet version 4 (TCP/IPv4)*.
- Cocher *Utiliser l'adresse IP suivante* et écrire l'adresse IP fixe du PC, **différente de l'IP fixe de la Raspberry !** (Cf figure *Configuration câble Ethernet IPv4*)

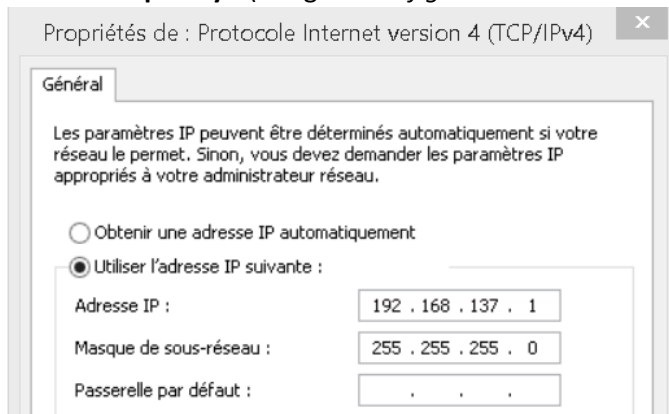


Figure 5: Configuration câble Ethernet IPv4

- Valider puis sélectionner votre connexion Wifi.
- Dans "Propriétés de Wi-Fi", aller dans l'onglet *Partage*.
- Cocher *Autoriser d'autres utilisateurs du réseau à se connecter via la connexion Internet de cet ordinateur*.
- Choisir le port du câble Ethernet (Cf figure *Partage de connexions Internet*).

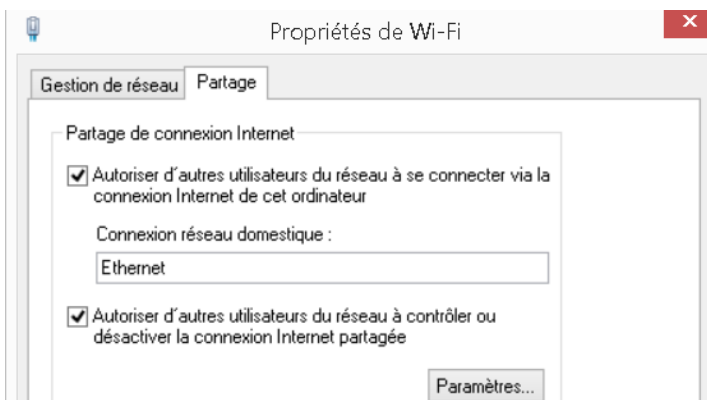


Figure 6: Partage de connexions Internet

Configuration Internet de la Raspberry

- Entrer la commande suivante :
`sudo nano /etc/network/interfaces`
- Insérer à la suite :
[...]
`auto eth0`
`allow-hotplug eth0`
`iface eth0 inet manual`
 `adress 192.168.137.1`
 `netmask 255.255.255.0`
 `broadcast 192.168.137.255`
`up ip route add 0.0.0.0/0 via 192.168.137.1 dev eth0`

Explications :

Les six premières lignes de code autorise la Raspberry à communiquer avec l'IP fixe de notre PC.

La dernière ligne crée une passerelle Internet entre l'IP fixe de la Raspberry et l'IP fixe du PC.

- Quitter le fichier en utilisant Ctrl + X.
- Un message de confirmation de sauvegarde apparait, appuyer sur la touche "o" puis sur "Entrée".
Vérifier que la passerelle est bien établie en entrant la commande :

`route`

(Cf figure *Configuration correcte de la passerelle*)

```
pi@raspberrypi:~ $ route
Table de routage IP du noyau
Destination      Passerelle      Genmask          Indic Metric Ref       Use Iface
default          192.168.137.1   0.0.0.0          UG    0      0          0 eth0
192.168.137.0    0.0.0.0         255.255.255.0    U     0      0          0 eth0
pi@raspberrypi:~ $
```

Figure 7: Configuration correcte de la passerelle

- Vérifier que la Raspberry a accès à Internet en tapant la commande :
`ping www.google.com`
(Cf figure *Connexion Internet de la Raspberry réussie*)

```
pi@raspberrypi:~ $ ping www.google.com
PING www.google.com (216.58.209.228) 56(84) bytes of data.
64 bytes from parl0s29-in-f4.1e100.net (216.58.209.228): icmp_seq=1 ttl=51 time=37.1
ms
64 bytes from parl0s29-in-f4.1e100.net (216.58.209.228): icmp_seq=2 ttl=51 time=36.9
ms
64 bytes from parl0s29-in-f4.1e100.net (216.58.209.228): icmp_seq=3 ttl=51 time=76.5
ms
64 bytes from parl0s29-in-f4.1e100.net (216.58.209.228): icmp_seq=4 ttl=51 time=30.5
ms
^C
--- www.google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 30.542/45.320/76.593/18.251 ms
```

Figure 8: Connexion Internet de la Raspberry réussie

Astuce : Pour arrêter une commande en cours d'exécution, utiliser Ctrl + C.

2.5) Implémentation du code LABVIEW dans la Raspberry

Pour implémenter un code LABVIEW dans une Raspberry, nous avons besoin de LINUX, projet en open source permettant de développer des applications embarquées avec LabVIEW.

Problème de compatibilité non résolu :

Le projet LINUX ne fonctionne que pour la version **LABVIEW 2014 SP (32-bit)** et notre code du robot MAKI fonctionne sous **LABVIEW 2017**.

Nous essayons d'exécuter le tutoriel [Raspberry Pi with LabVIEW LINUX](#) permettant de résoudre ce problème.

Par manque de documentation sur LINUX, nous n'avons pas réussi à faire fonctionner l'exemple de la LED présenté dans le tutoriel.

Le code est bien présent dans la Raspberry mais ne s'exécute pas correctement.

Nous recommandons de changer de microprocesseur car l'implémentation dans une Raspberry présente peu de documentation et il y a peu d'aide sur Internet pour résoudre les problèmes.