# Fondamentaux et architecture des systèmes

Projet NAO

Florent BERLAND et Hugo LECLER

# Introduction:

Notre projet consiste en l'utilisation de capteurs sur un ordinateur Raspberry portatif pour pouvoir guider un utilisateur vers une destination voulue.
L'objectif lors de ce projet était tout d'abord de prendre en main divers capteurs pour combiner leurs utilisations et parvenir à un résultat ayant une utilisation restreinte, mais fonctionnel.

# Mode d'emploi:

Montage du boîtier utilisateur :
Brancher une batterie sur un ordinateur Raspberry. Monter un shield adapté sur les connecteurs. Brancher un récepteur infrarouge sur le port analogique 0 du shield, 3 boutons sur les ports digitaux 6, 7 et 8 et un écran LCD.
Pour initialiser le programme : installez la bibliothèque i2c-dev.h pour utiliser le shield. Téléchargez le code source du programme, compilez-le et lancez-le. Le Raspberry peut alors fonctionner de manière autonome.

Montage des émetteurs :
Branchez un microcontrôleur Arduino à une source d'alimentation (piles ou secteur). Connectez un shield adapté, et ajoutez dessus une LED infrarouge sur le port digital 4. Connectez l'Arduino à un ordinateur. Dans le code source, vous pouvez changer les numéros de salle dans les variables globales. Téléversez le code source dans la carte. Débranchez la carte de l'ordinateur, elle est prête à fonctionner. Fixez-la sur le montant d'une porte à un 1,3 m de hauteur, la LED doit être orientée perpendiculairement au mur, en direction de l'intérieur de la salle. Assurez-vous que le numéro de la salle entré dans le code corresponde à la salle se trouvant de l'autre côté de la porte.

# Moyens matériels:

Notre projet utilise un Raspberry mais nous n'avons jamais obtenu de shield malgré les multiples demandes.
Nous avons tout de même écrit le code nécessaire au bon fonctionnement sur le Raspberry. Les capteurs que nous avions demandé n'ayant pas non plus été commandés, nous avons dû les commander nous même ce qui nous a légèrement retardé sur le projet.

Le système se compose de deux parties : le boîtier utilisateur et les émetteurs. Le boîtier est portatif, il a besoin de piles. Il contient le Raspberry ainsi que l'écran LCD, un capteur infrarouge et trois boutons. Les émetteurs sont fixes, ils peuvent être branchés sur une prise électrique ou avoir un boîtier de piles. Ils ont aussi besoin d'une carte Arduino avec un shield adapté et une LED infrarouge.

Florent BERLAND et Hugo LECLER

## Moyens humains:

La répartition du travail s'est faite ainsi :
Florent : codage des émetteurs sur Arduino
Hugo : codage du récepteur en C pour le Raspberry

A deux : fonctions du code principal du Raspberry.

## Architecture du projet:

Code des émetteurs:

void setup(): Initialise les paramètres nécessaires pour l'émission du signal.

void loop(): Envoie le signal suivi d'un délai (la fonction s'exécute en boucle donc l'émission est périodique). Le signal est codé de cette façon : il comporte 6 bits, le premier et le dernier sont à 1, les bits 1 à 4 contiennent le numéro de la salle suivante (de 0 à 15) codé en binaire, les bits de poids faible en premier.

Code du boîtier utilisateur:

int main(): Programme bouclant exécutant les différentes étapes du projet.
Il commence par initialiser l'écran LCD, appelle demander() qui permet de choisir un numéro de salle grâce aux boutons,
la salle de départ de l'utilisateur et celle d'arrivée, puis exécute la fonction salleX(int) une ou plusieurs fois, en fonction du trajet. Par exemple, si on se trouve en salle 1, avec pour destination salle 3 et qu'il est nécessaire de passer par la salle 2, le programme lancera salleX(2) puis salleX(3). Une fois cela fait, on redemande à l'utilisateur une salle de départ et une destination.

int salleX(int): lance la fonction décoder, puis analyse le code obtenu. Si le code obtenu est égal à la variable en paramètre, c'est la bonne salle et la fonction finie.
Sinon, si le code est 0 c'est que l'utilisateur arrête le programme et le programme s'achève. Sinon, on indique à l'utilisateur via l'écran LCD que ce n'est pas la bonne salle, et on attend un nouveau code.

int decoder(): boucle en attente d'un code valide avec la fonction recevoir(), c'est à dire ayant un tableau avec la première et dernière valeur égale à 1 (le cas échéant signifie que le signal a été capté en cours, il ne peut pas être décodé). Si le tableau est valide, retourne decode() qui réécrit le nombre envoyé par l'Arduino en base 10.

void recevoir(): Demande en boucle une lecture du récepteur IR jusqu'à trouver une valeur supérieur au seuil de précision (Cela correspond à un bit à 1 du signal). Une fois trouvé, il enregistre les 6 prochaines valeurs captées dans un tableau de 6 booléens.

Nous avons également écrit des fonctions pour permettre à l'écran LCD d'afficher divers messages grâce à un tableau de caractères fournit en paramètre.

## Perspectives:

Le projet n'est pas terminé, n'ayant pas pu tester notre programme sur un Raspberry pi (nous n'avons jamais pu obtenir le matériel nécessaire, lequel était censé être disponible), nous avons dû convertir le code pour un Arduino et nous avons parfois été confrontés à un problème de mémoire.
De plus, pour que ce projet puisse être commercialisable, il faudrait créer une véritable base de donnée des différents chemins vers les salles et possiblement obtenir des capteurs plus performants.