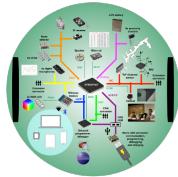


# Systèmes embarqués et robotique - Projet

8 Mai, 2023

**Benjamin Bahurel (326888)**  
benjamin.bahurel@epfl.ch

**Victor Cartier-Negadi (315587)**  
victor.cartier-negadi@epfl.ch



## 1 Introduction

Dans le cadre de ce projet, nous avons utilisé un robot appelé E-Puck2 pour créer un système de sécurité qui permet de détecter des intrus et de les signaler en produisant des sons et des lumières.

L'E-Puck2 est un petit robot mobile, équipé de plusieurs capteurs et possédant ainsi une grande variété de fonctionnalités. Il peut se déplacer facilement sur différents types de surfaces et peut être programmé pour effectuer différentes tâches. De manière à répondre aux contraintes imposées dans le cadre de ce projet, notre choix de modules s'est orienté vers les moteurs, le micro et le capteur Time of Flight. Le capteur Time of Flight (ToF) intégré au robot permet de détecter les obstacles et les objets à une distance allant jusqu'à 2 mètres. Le robot est également équipé d'un microphone et d'un haut-parleur pour produire des sons.

Nous avons créé un système de sécurité qui utilise les fonctionnalités de détection d'objets et de sons du robot E-Puck2 pour alerter les personnes présentes dans la zone en cas de détection d'un phénomène inhabituel. Le robot suit un parcours pré-défini et utilise ses capteurs pour détecter les objets ou les sons dans son environnement. En cas de détection, il passe en mode alarme et émet des signaux sonores et lumineux. Il envoie également un message sur un terminal pour notifier l'utilisateur d'une intrusion.

## 2 Méthode de travail

Pour la réalisation de ce projet, nous avons suivi les étapes suivantes :

### 2.1 Définition des objectifs

Nous avons commencé par définir les fonctionnalités et les performances attendues pour notre système de sécurité en prenant en compte les contraintes liées aux capteurs du robot.

### 2.2 Conception et planification

Une fois nos objectifs clairement définis à l'écrit(modules, tâches, interruptions), nous avons conçu le code de notre système de sécurité. Nous avons commencé par isoler et indexer les différents morceaux de code C nécessaires à notre projet, en utilisant notamment le TP5 comme base. Ensuite, nous avons évalué l'état du code disponible, déterminé les modifications nécessaires et identifié les tâches à coder en détail.

### 2.3 Implémentation du logiciel

Une fois la conception terminée, nous avons commencé à implémenter le logiciel. Nous avons utilisé le langage de programmation C pour développer le logiciel qui contrôle le robot et gère les capteurs. Nous

avons utilisé différentes fonctions pour accéder aux fonctionnalités du robot et développé des threads pour utiliser les capteurs/actuateurs.

## 2.4 Tests et validation

Les capteurs ont été testés au cours des travaux pratiques pour comprendre leur fonctionnement, choisir le sujet de projet et vérifier leur fiabilité. Le code du projet a été testé sur le robot à chaque étape et validé avant d'être poussé sur Github.

Nous avons effectué des tests approfondis pour vérifier le bon fonctionnement de notre système de sécurité. De plus, nous avons testé le robot dans différentes situations pour valider les performances de notre logiciel et pour détecter d'éventuels problèmes. Enfin, nous avons également mesuré les performances du système en termes de déplacement et de précision de détection.

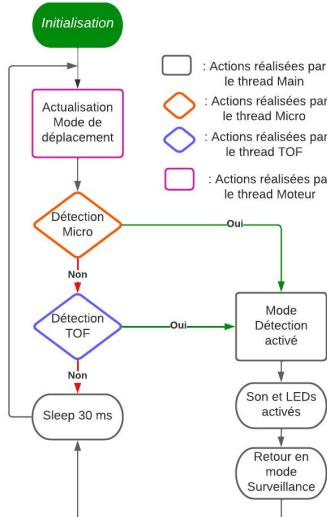
## 2.5 Documentation et rapports

Nous avons documenté notre travail à travers les descriptions de nos commits sur Github et les commentaires dans notre code pour faciliter la compréhension de notre projet.

En résumé, notre méthode de travail a consisté à définir les objectifs de notre projet, à concevoir notre système, à coder le logiciel, à effectuer des tests et à documenter notre travail. Cette méthode nous a permis de développer un système de sécurité efficace et performant en utilisant les fonctionnalités avancées du robot E-Puck2.

# 3 Conception du logiciel

Notre logiciel de robot de surveillance pour le robot e-puck2 est conçu pour détecter les objets ou êtres vivants, et les sons dans une zone définie, et pour activer un système d'alarme lorsqu'une détection est établie. Notre code source est structuré en plusieurs fichiers, se focalisant chacun sur l'implémentation ou le traitement d'une fonctionnalité.

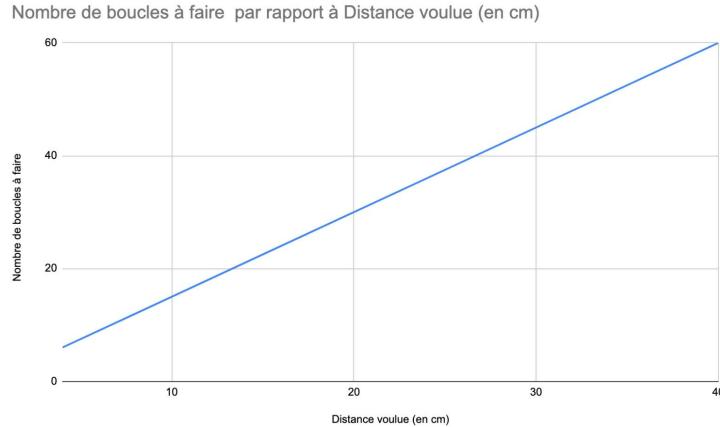


### 3.1 Gestion des moteurs

Le logiciel utilise les deux moteurs du robot pour lui permettre de se déplacer dans la zone à sécuriser selon des parcours définis. Cinq modes de parcours différents sont définis: rotation sur lui même (position

unique), allers-retours en ligne droite, déplacement en carré, déplacement en carré avec balayage supplémentaire, déplacement en triangle. Un mode arrêt (aucun mouvement) est aussi disponible.

Le paramétrage de ces modes est effectué dans le fichier *motor\_process.c* et il suffit de changer la valeur du sélecteur pour changer de mode. De plus, la distance que parcourt le robot en ligne droite est facilement modifiable en changeant une valeur dans le fichier *motor\_process.c*, distance suivante :



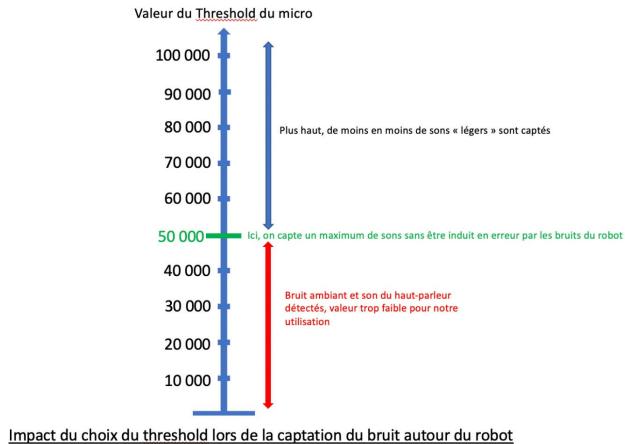
### 3.2 Gestion des capteurs

Le logiciel utilise le capteur Time of Flight (ToF) du robot pour détecter les objets et les êtres vivants dans la zone à sécuriser. Le capteur effectue des mesures régulières et envoie une notification au module de traitement dès qu'il y a détection.

Le paramétrage de la distance de détection spatiale est effectué dans le fichier *TOF\_process.c*

Notre code utilise également le microphone gauche de l'E-Puck2 pour détecter les sons dont l'intensité sonore est trop importante. Lorsqu'un son trop fort est détecté, le logiciel envoie une notification au module de traitement ainsi qu'une alerte sur le terminal.

Le paramétrage de la valeur seuil pour l'intensité sonore est effectué dans le fichier *audio\_processing.c*. Suite à des tests menés avec le micro, nous avons pu déterminé une valeur permettant une captation optimale des bruits sans prendre en compte ceux émis par le robot lui-même.



### 3.3 Module de traitement

Le fichier main.c gère le mode de fonctionnement du robot en fonction des événements extérieurs. Le mode « SURVEILLANCE » actualise le mode de déplacement et vérifie régulièrement les micros et le capteur ToF. Si l'un des deux capteurs détecte quelque chose, le robot passe en mode « DETECTION » et déclenche le système d'alarme.

### 3.4 Module d'alarme

Le module d'alarme est chargé de déclencher un système d'alarme lorsque le robot est en mode « DETECTION ». Le système d'alarme se compose de deux parties : une partie sonore avec un son strident et une partie lumineuse avec le clignotement des LEDs.

## 4 Analyse et résultats obtenus

Dans cette partie, nous allons présenter les résultats obtenus lors des tests effectués sur notre programme, ainsi que les analyses que nous avons pu en tirer.

### 4.1 Description des tests effectués

Nous avons effectué plusieurs tests pour vérifier le bon fonctionnement de notre programme. Tout d'abord, nous avons testé la détection des obstacles avec les capteurs de proximité de l'e-puck2. Nous avons placé des obstacles à différentes distances de l'e-puck2 et nous avons vérifié que le robot réagissait correctement en notifiant les obstacles. Nous avons notamment remarqué que le capteur était légèrement incliné vers le haut pouvant gêner la détection d'objet se trouvant trop proche du sol.

Pour tester la captation des sons avec le microphone, nous avons émis différents sons à des intensités spécifiques vérifiant ainsi que l'e-puck2 les percevait. Nous avons également testé les LEDs et le haut parleur du e-puck2 pour être certain que le système d'alarme soit efficace et se déclenche correctement.

### 4.2 Analyse des résultats obtenus

Nous avons pu observer que notre programme fonctionnait très correctement lors des tests que nous avons effectués pour notre usage précis. Le robot a été capable de se déplacer et de détecter les obstacles dans une plage de précision qui nous satisfaisait pour notre projet.

Cependant nous avons eu quelques résultats décevants lors des tests initiaux des capacités de ses capteurs. L'utilisation des moteurs du robot ne fonctionnait plus à partir d'une certaine vitesse demandée et nous avons pu observer que le robot avait parfois du mal à détecter les obstacles lorsqu'ils étaient très proches.

### 4.3 Discussion

Malgré ses limitations, notre programme a réussi à remplir les objectifs que nous nous étions fixés, notamment la détection des obstacles, la détection de son et la communication avec l'ordinateur. Cependant, il est important de souligner que notre projet pourrait être amélioré en tenant compte des limites que nous avons rencontrées. Par exemple, l'utilisation de capteurs plus performants pourrait améliorer la précision de la détection des obstacles, même dans des conditions difficiles. De plus, un interface utilisateur permettant de contrôler directement le robot à distance pourrait être une piste de développement intéressante.

En conclusion, notre projet nous a permis de développer un programme efficace pour contrôler l'e-puck2 dans un environnement précis, en utilisant différents capteurs et modules de ce robot. Nous sommes satisfaits des résultats obtenus lors de nos tests, et nous envisageons des améliorations possibles pour l'avenir.