

Gabriel Le Guay  
Jonas Daverio

Microinformatique  
Rapport du microprojet



16/05/2022

# Projet de microinformatique : Mapping

# Table des matières

## **1. Introduction**

## **2. Principe de fonctionnement**

### 2.1 Déplacement du robot

#### 2.1.1 itinéraire

#### 2.1.2 Détection des objets

#### 2.1.3 Changement de direction

### 2.2 cartographie

#### 2.2.1 Principe

#### 2.2.2 IMU et Calculs

#### 2.2.3 Bluetooth

## **3. Organisation du code**

### 3.1 Fonctionnement général

### 3.2 Fonction remarquables

### 3.3 Organisation entre les modules

## **4. Résultats**

## **5. Conclusion**

## **6. Références**

# 1.Introduction

Dans le cadre du mini projet de semestre du cours de systèmes embarqués et robotique dispensé par le professeur Francesco Mondada, nous avons cherché à simuler un robot qui cartographie la pièce dans laquelle il se déplace. Ce projet utilise un robot E-Puck ainsi que ses capteurs de distance, moteurs pas à pas, accéléromètre (IMU) et sa transmission bluetooth. Le robot E-Puck va donc se déplacer dans la pièce de manière à recouvrir entièrement la surface de celle-ci. En simultan  nous rel verons sa position en temps r el afin de cartographier son chemin et la pi ce dans laquelle il s'est d plac .

## 2.Principe de fonctionnement

### 2.1 D placement du robot

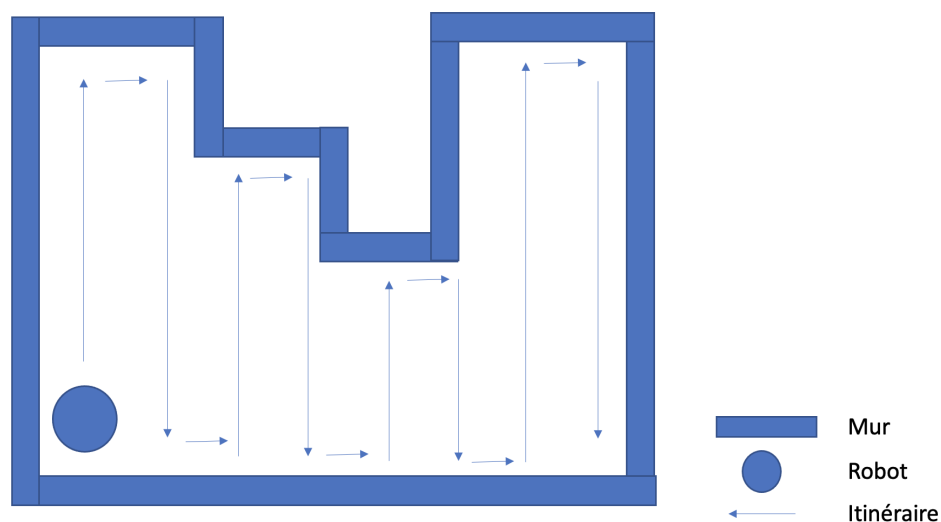


figure 1 : sch ma du chemin du robot dans une certaine pi ce

#### 2.1.1 itin raire

Le robot sera initialement d pos  dans le coin de la pi ce. Il commencera par avancer gr ce   ses moteurs   une vitesse de 3 cm/s. Lorsqu'il arrive sur un mur, il tourne   90 degr s sur la droite, si il n'y a pas de mur devant lui, a

proximité, il avance pendant une seconde et tourne encore de 90 degrés. S'il y a un mur, il tourne directement. Ensuite, il continue d'avancer dans l'autre direction et répète l'opération en tournant à sa gauche.

### 2.1.2 Détection des objets

Lorsque le robot se trouve à une distance de 10 cm d'un objet sur son chemin (en face de lui), il effectue alors une rotation de 90 degrés. Cette détection se fait à l'aide du capteur de distance situé à l'avant du robot. Nous avons choisi d'utiliser le capteur de distance et non pas les InfraRouges, car ceux-ci sont trop peu précis et détectent une distance trop petite pour laisser le robot s'arrêter et tourner.

### 2.1.3 Changement de direction

Une fois le mur détecté, le robot commence par une rotation de 90 degrés vers la droite. Il test la distance à laquelle il est du mur en face de lui, si elle est inférieure à 10 cm, il tourne directement de 90 degrés encore à droite (demi tour au final). Si la distance est supérieure, le robot va avancer pendant 1 seconde et ensuite tourner. Il répètera l'opération en tournant cette fois à gauche. Et ainsi de suite, en alternant à droite du robot deux virages puis deux à gauche. On peut bien le voir sur la figure 1.

## 2.2 cartographie jonas

2.2.1 Principe

2.2.2 Calculs

2.2.3 Bluetooth

## 3. Organisation du code

### 3.1 Fonctionnement général

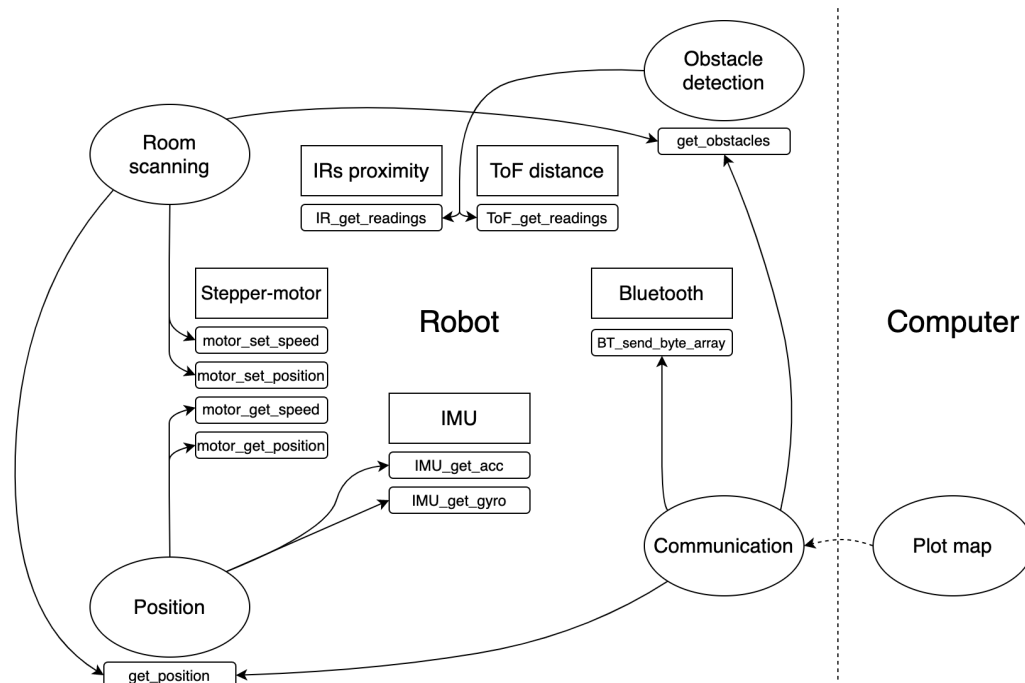


figure 2 : schéma des modules et de leur dépendance mutuelles  
 $A \rightarrow B = A$  dépend de  $B$

### 3.2 Fonction remarquables

Fonction 'turn' :

Elle prend comme input 'angle' pour déterminer de quel angle on veut que le robot tourne et le booléen 'i', pour savoir si on veut que le robot tourne à gauche ou à droite.

Nous utilisons la librairie 'motor.h' et notamment 'motor\_get\_pos' et 'motor\_set\_pos'. Nous commençons par mettre sa vitesse et sa position à 0, et tant que la roue (droite ou gauche selon 'i') n'a pas effectué, dans notre cas, un quart de tour (=90 degrés), une roue avance et l'autre recule (ce qui fait tourner le robot). Une fois la rotation terminée, le robot continue d'avancer.

### 3.3 Organisation entre les modules

4. Résultats

5. Conclusion

6. Référence