

PJT

Plateforme Rover Robotique

RobArts

Luc Beltran
Mathieu Bouvet
Benoit Boyau
Thomas Merlet

8 JANVIER 2018

ARTS
ET MÉTIERS

CONCEVOIR
DEMAIN



ARTS
ET MÉTIERS
ParisTech

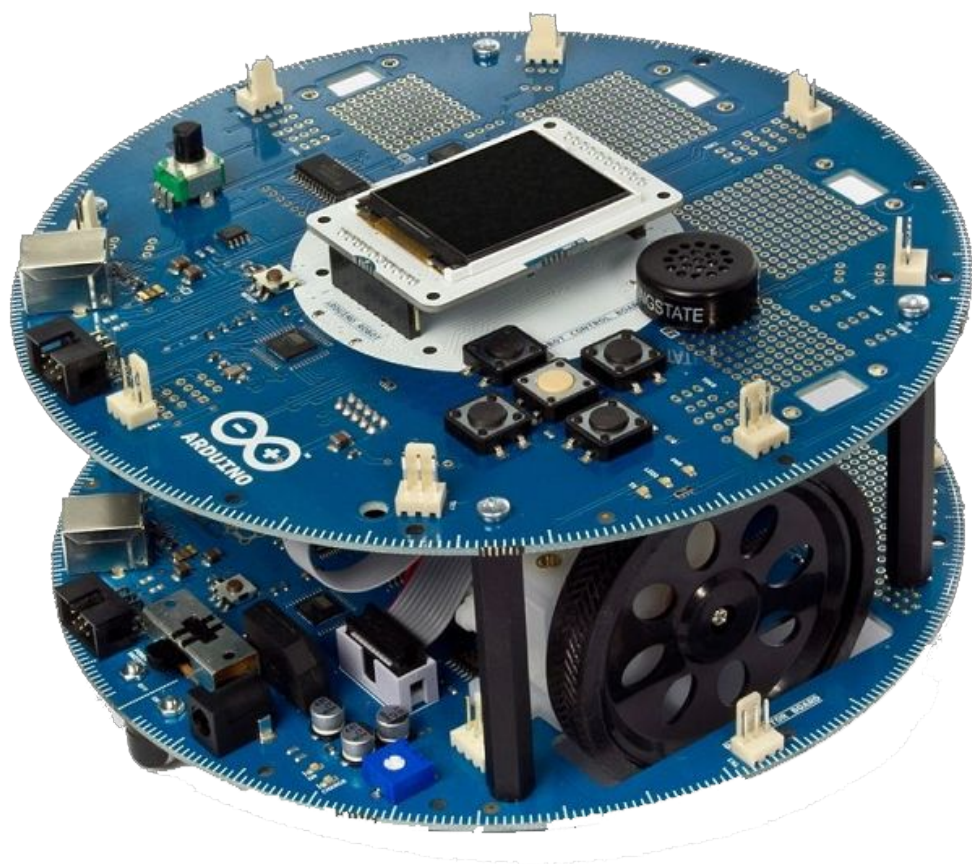
Introduction

Présentation du projet

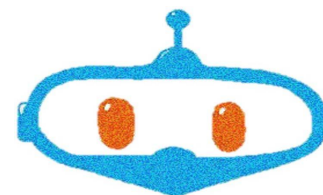
Introduction

PJT
Plateforme Rover
Robotique

8 janvier 2018

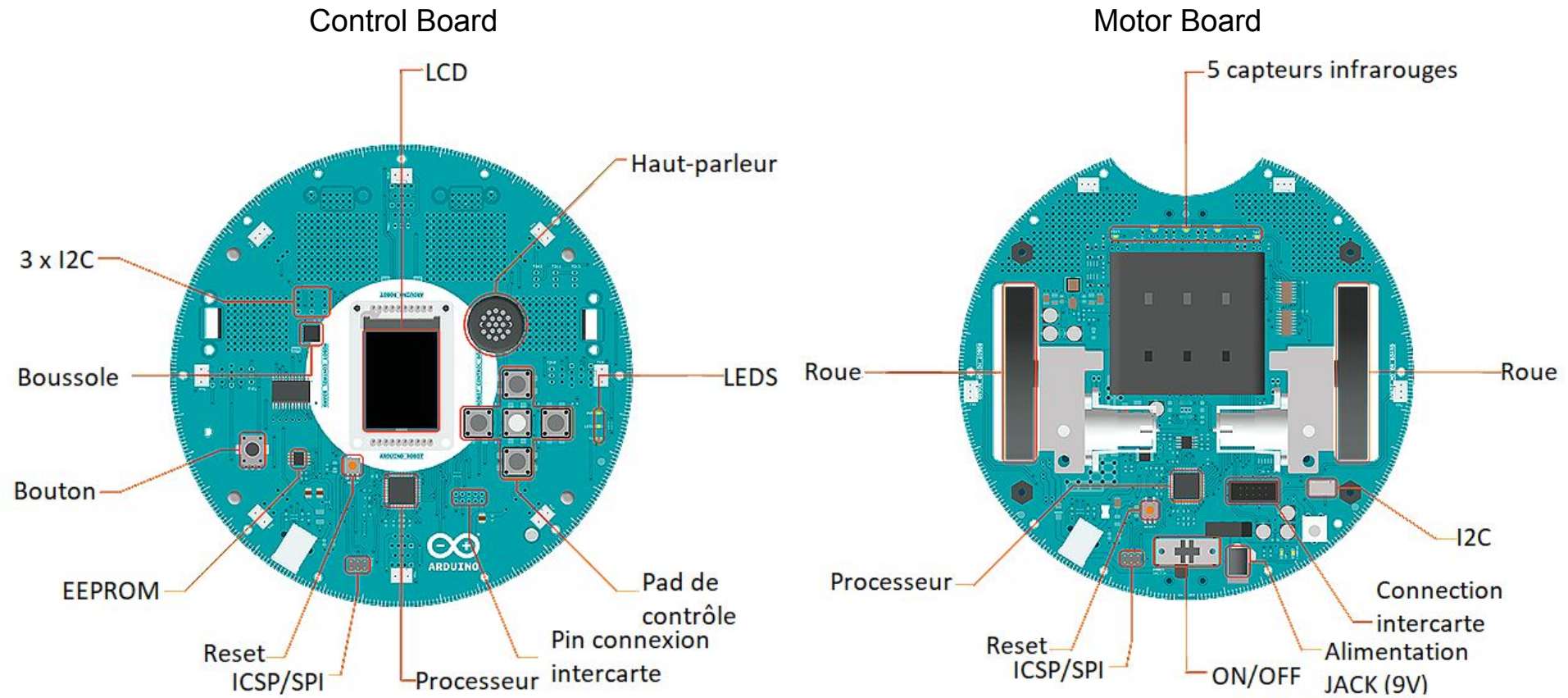



Coupe de France
DE ROBOTIQUE



Présentation du robot

Introduction



PJT
Plateforme Rover
Robotique

8 janvier 2018

SOMMAIRE

- I. *Gestion de projet*
- II. *Aire de jeu*
- III. *Les capteurs*
- IV. *Liaison sans-fil*
- V. *Problématique de localisation*

I. Gestion de projet

Objectifs du projet

- I. Gestion de projet
- II. Aire de jeu
- III. Les capteurs
- IV. Liaison sans-fil
- V. Problématique de localisation

Fonctions demandées	Performance attendu et contrôlé
Définir une localisation	Précision de 1cm sur 10m parcouru
Programmer et accès à distance (liaison sans fil)	Portée de 10 m
Traiter localement puis à distance l'information	Stockage local et transmission sans fil sur une portée de 10m
Trajectographie et correction de trajectoire	Précision de 1cm sur 10m parcouru
Recueillir des informations sur l'environnement	Consultation d'une banque de données horodatée sur une profondeur d'acquisition de 24h

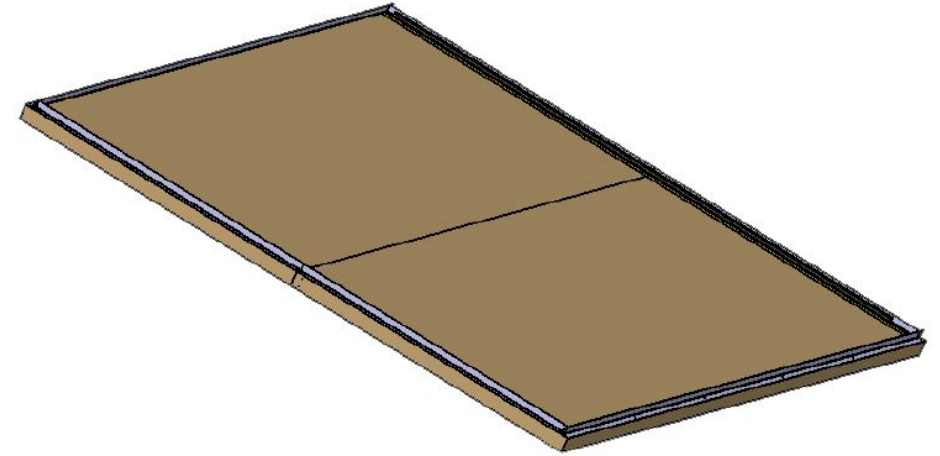
Cahier des charges

- I. Gestion de projet
- II. Aire de jeu
- III. Les capteurs
- IV. Liaison sans-fil
- V. Problématique de localisation

Fonctions	Critère	Niveau	Solution technologique proposée
Résister à l'aire de jeu	Choc avec les barrières	Respecter l'aire de jeu	Capteur à ultrasons
Respecter l'aire de jeu	Ne doit pas endommager les barrières, capteurs, balises		Capteur à ultrasons
Résister à l'air	Température	0-40°C	
	Humidité	0-87%	
	Vent	Sans objet	
	Pression	Atmosphérique	
	Poussière	Sans objet	
Respecter l'air	Ne pas polluer	Aucune émission	Moteurs électriques

II. Aire de jeu

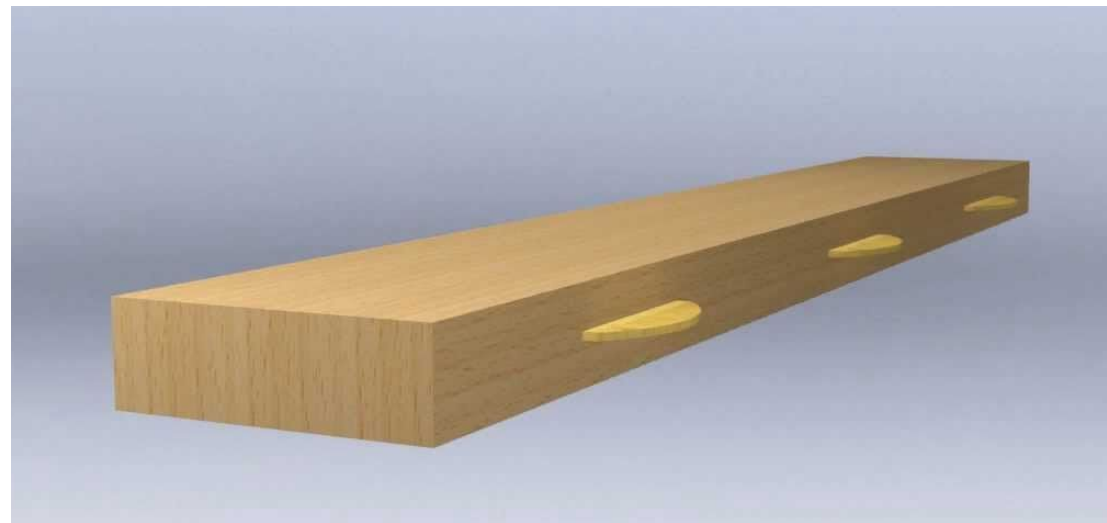
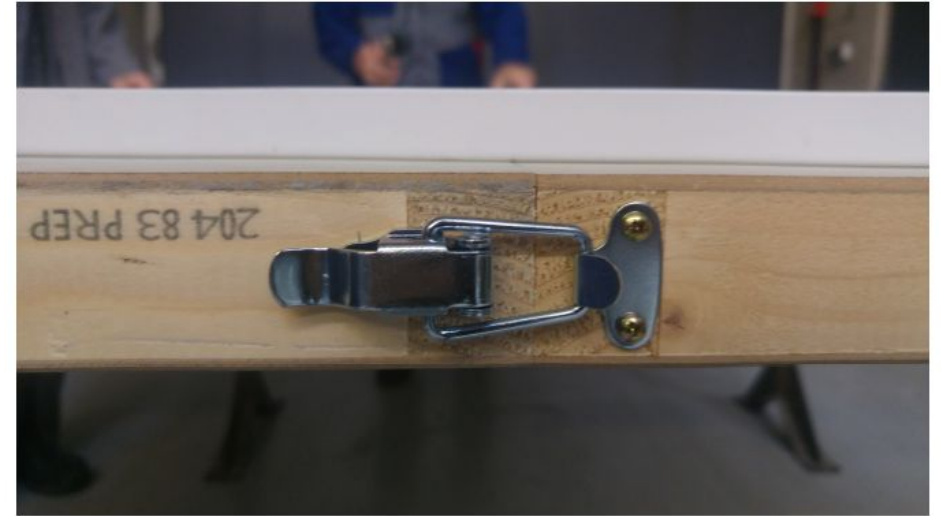
Besoins



- I. Gestion de projet
- II. **Aire de jeu**
- III. Les capteurs
- IV. Liaison sans-fil
- V. Problématique de localisation

Spécifications	Performance attendu et contrôlé	Solution choisie	Solution technologique
Dimensions	Taille minimum : 2m * 80cm		
Planéité		Planche solide	Portes
Transportabilité	L'aire de jeu doit pouvoir être rangée dans une voiture	Diviser l'aire de jeu en deux parties	2 portes 5 lamellos 2 grenouillères
Résistant au transport	L'aire de jeu ne doit pas être endommagée lors du transport		
Modulable	L'aire de jeu doit permettre la mise en place et le déplacement de capteurs et de balises	Rail	Rail en métal sur le bord de l'aire de jeu
Rugosité	Le robot doit pouvoir rouler sur la surface		
Sécurité du robot	Le robot ne doit pas tomber de l'aire de jeu	Structure autour de l'aire de jeu	

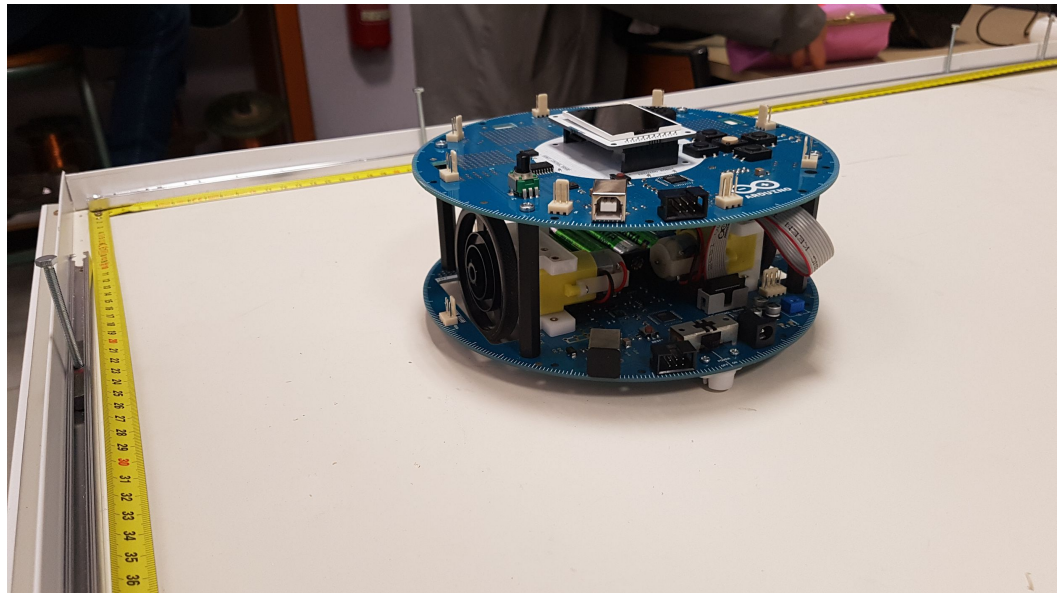
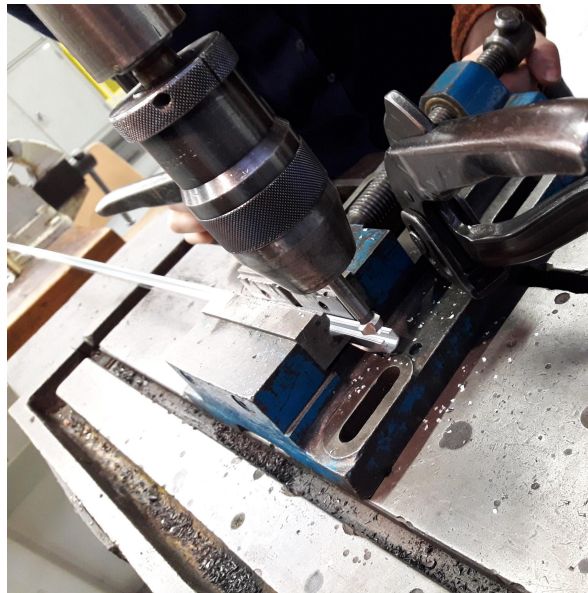
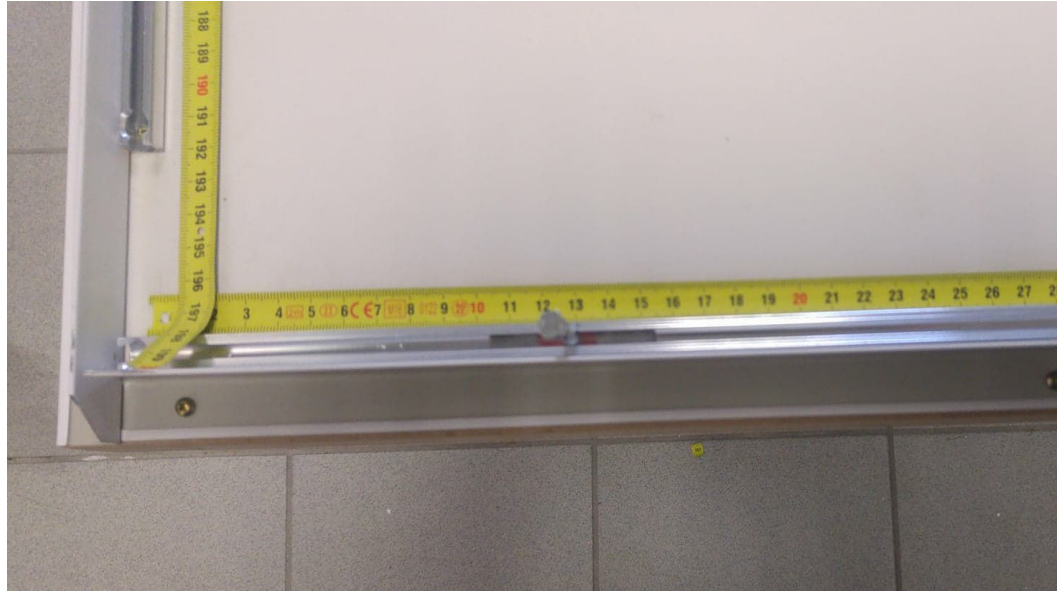
Fabrication



- I. Gestion de projet
- II. **Aire de jeu**
- III. Les capteurs
- IV. Liaison sans-fil
- V. Problématique de localisation

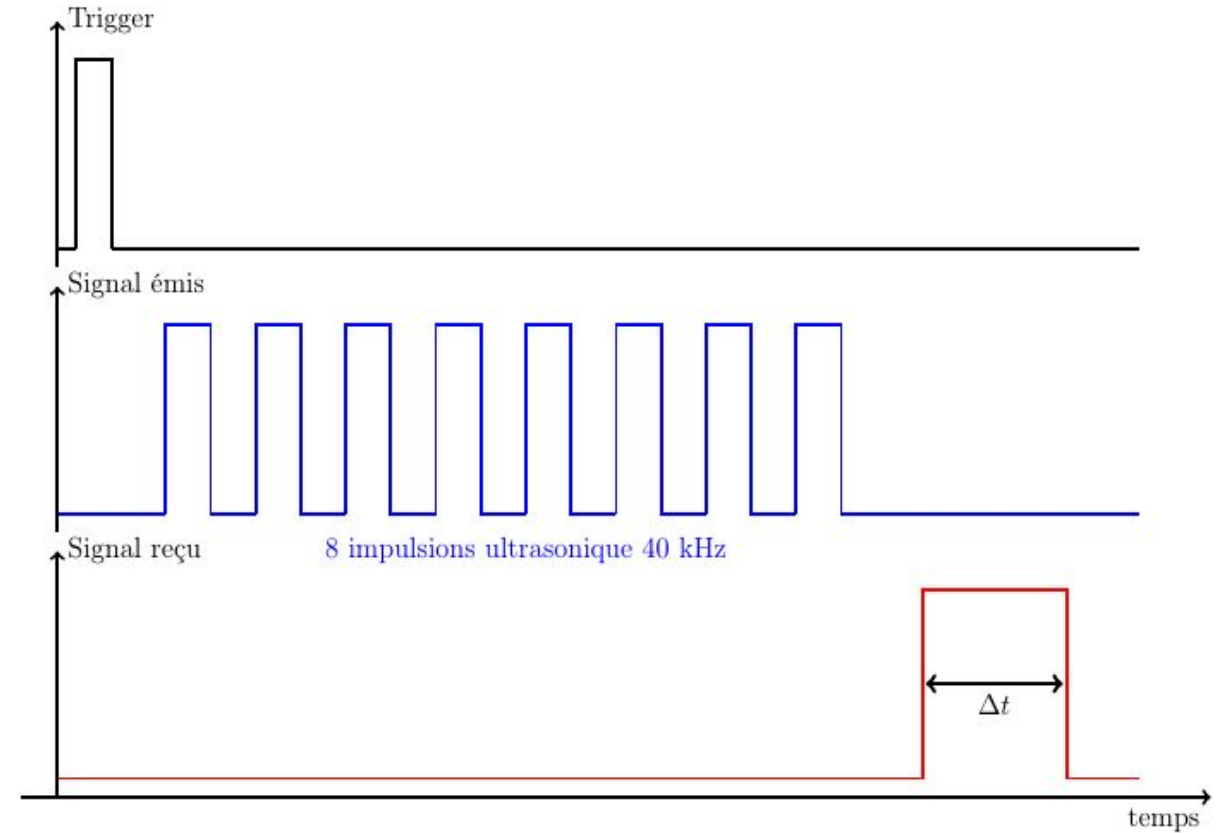
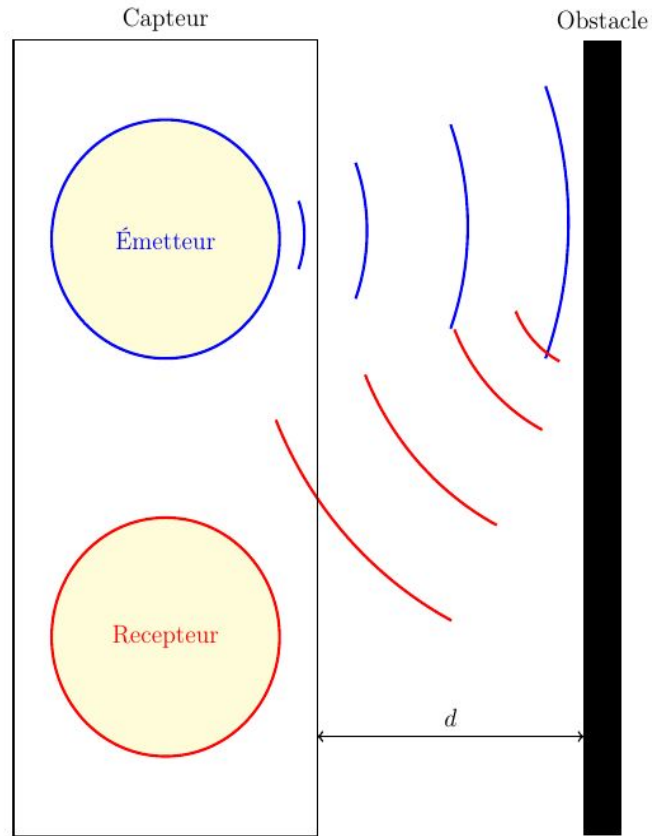
Fabrication

- Rail amovible
- Règle graduée
- Balises mobiles



III. Les capteurs

Capteur ultrason

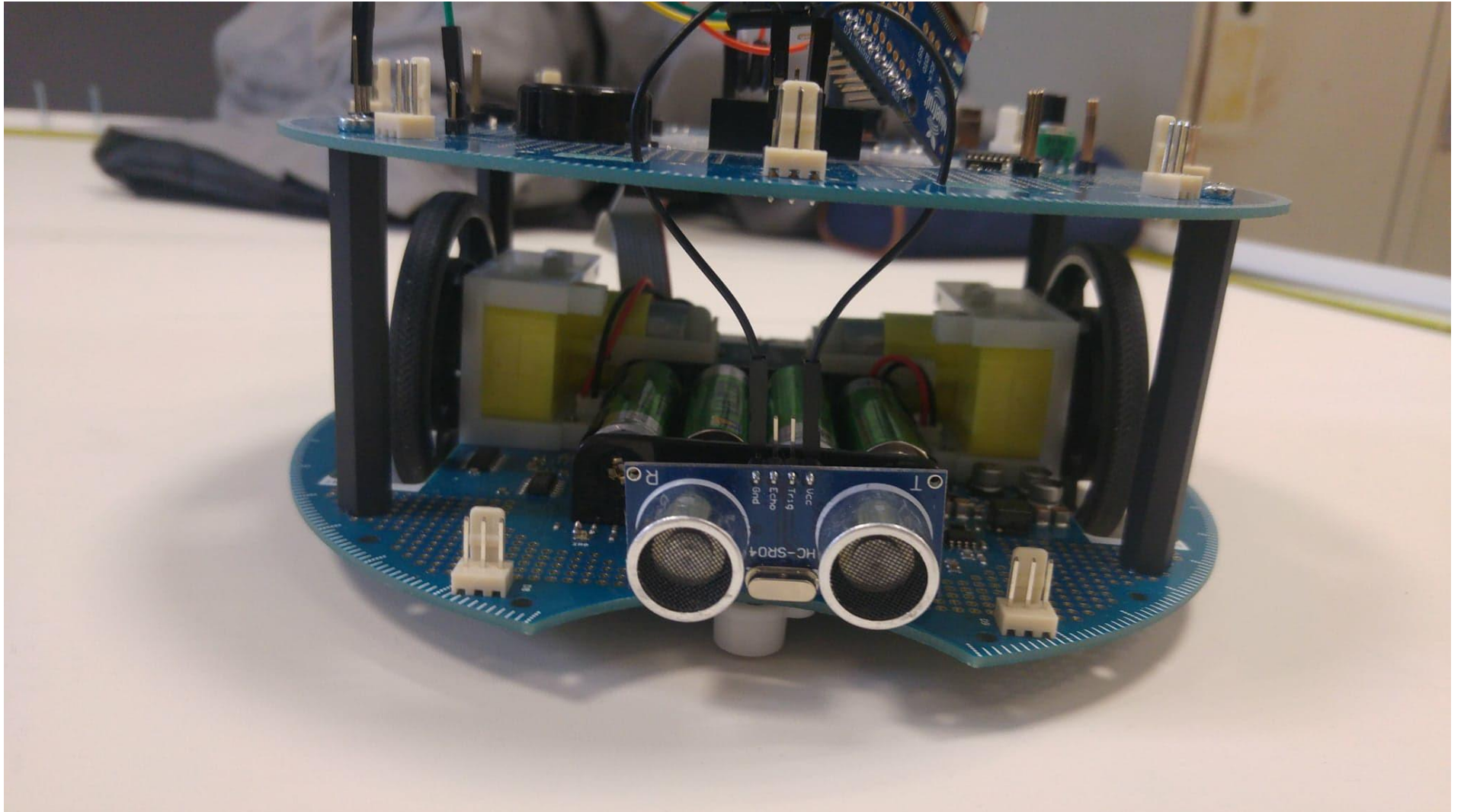


$$d = v \cdot \frac{\Delta t}{2}$$

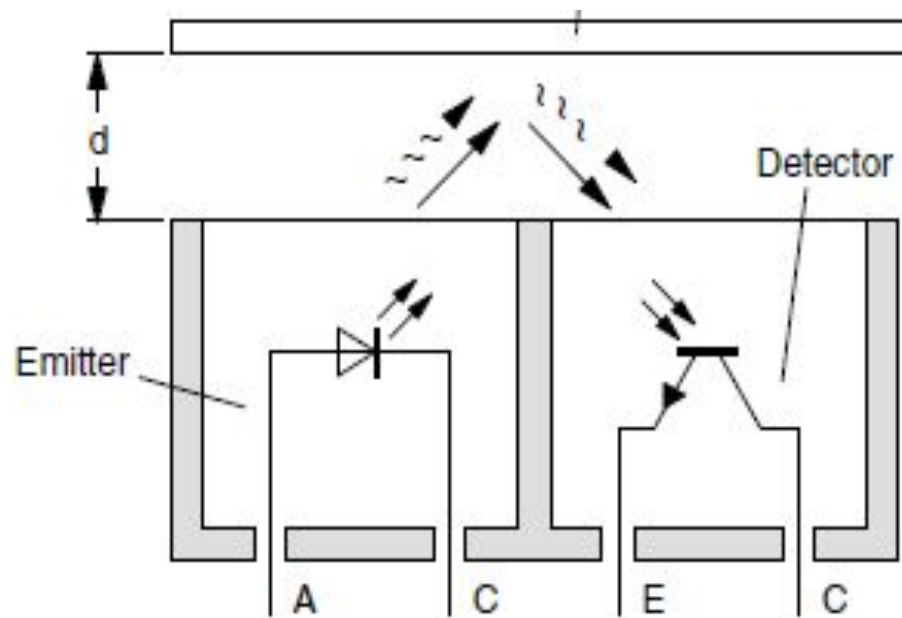
- I. Gestion de projet
- II. Aire de jeu
- III. **Les capteurs**
- IV. Liaison sans-fil
- V. Problématique de localisation

Capteur ultrason

- I. Gestion de projet
- II. Aire de jeu
- III. **Les capteurs**
- IV. Liaison sans-fil
- V. Problématique de localisation



Capteur infrarouge



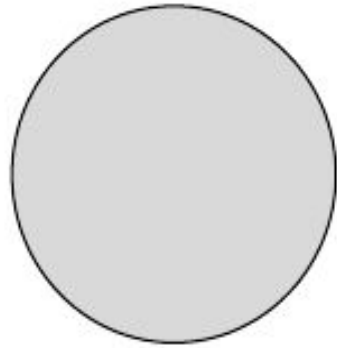
Surface noire / non réfléchissante

⇒ Pas de réflexion de la lumière

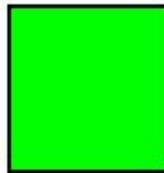
⇒ Pas de lumière détectée

- I. Gestion de projet
- II. Aire de jeu
- III. **Les capteurs**
- IV. Liaison sans-fil
- V. Problématique de localisation

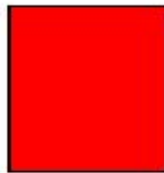
Capteur infrarouge



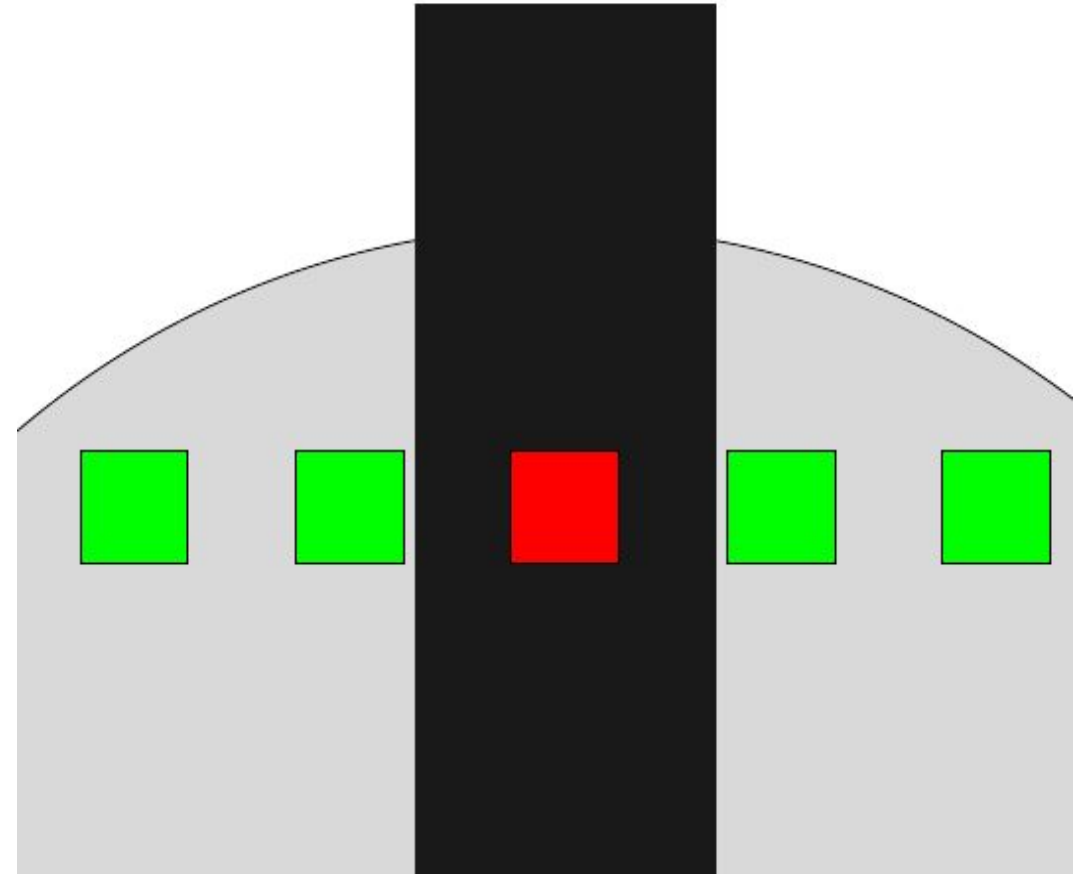
Robot



État haut

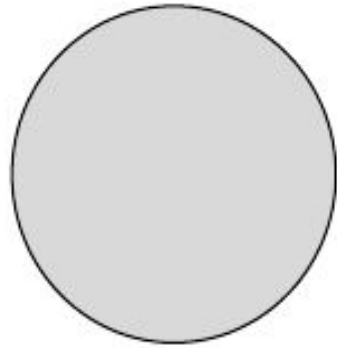


État bas

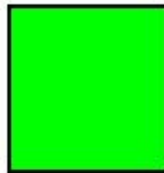


- I. Gestion de projet
- II. Aire de jeu
- III. **Les capteurs**
- IV. Liaison sans-fil
- V. Problématique de localisation

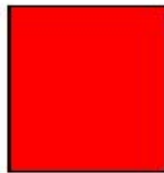
Capteur infrarouge



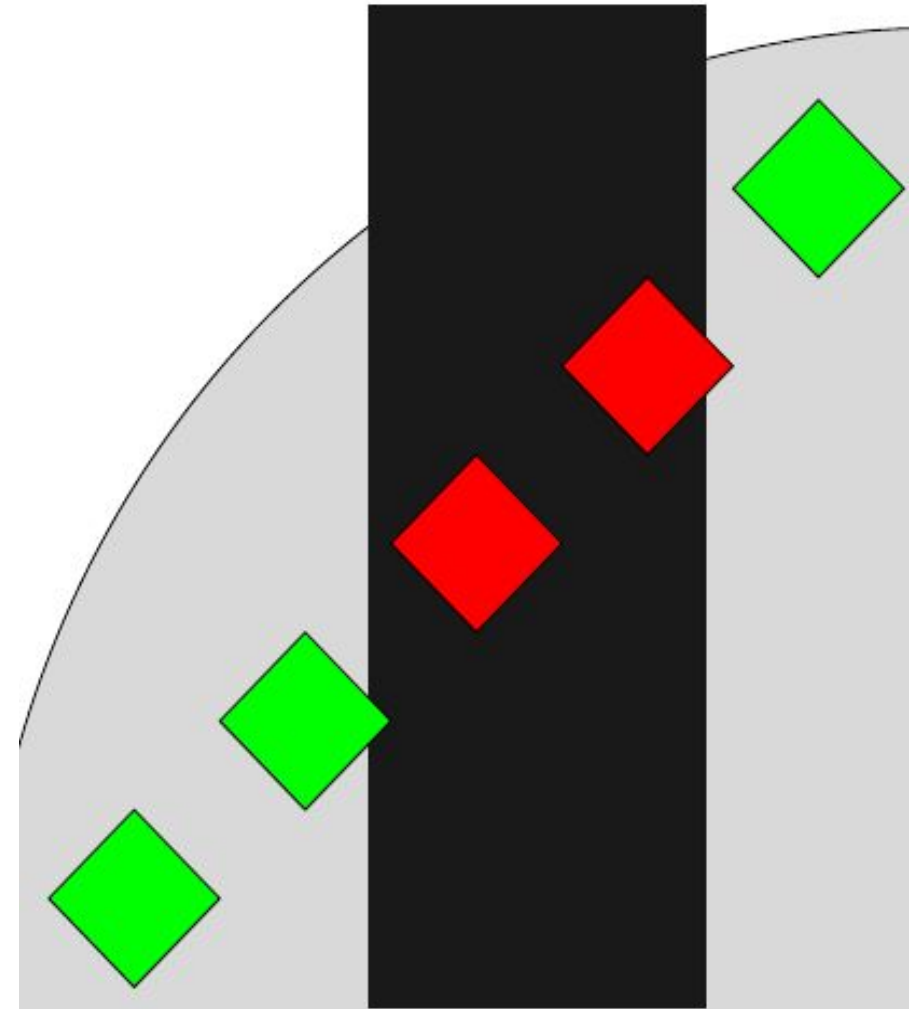
Robot



État haut



État bas



- I. Gestion de projet
- II. Aire de jeu
- III. **Les capteurs**
- IV. Liaison sans-fil
- V. Problématique de localisation

IV. Liaison sans fil

Choix de la technologie



	Bluetooth Low Energy (BLE)	Wi-Fi (norme IEEE 802.11)
Portée	≈10m	≈30m en intérieur ≈100m en extérieur
Fréquence d'émission (porteuse)	2,4 GHz	2,4 - 3,6 - 5 GHz
Débits de données	10 Mbit/s mesuré (24 Mbit/s théorique)	30 Mbit/s (54 Mbit/s théorique)

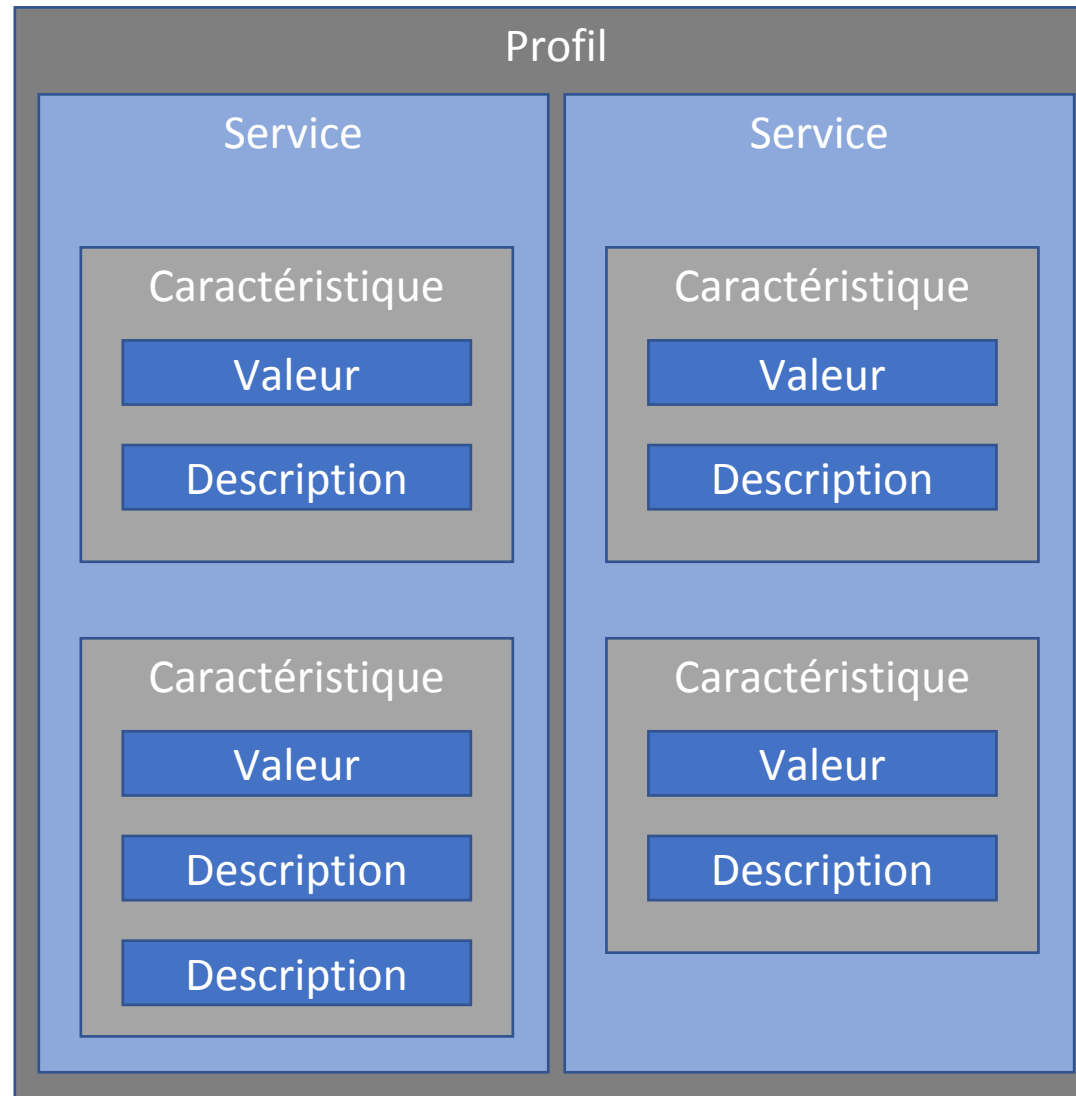
Bluetooth Low Energy (BLE) ou Bluetooth à basse consommation



- I. Gestion de projet
- II. Aire de jeu
- III. Les capteurs
- IV. Liaison sans-fil**
- V. Problématique de localisation

Protocole Bluetooth

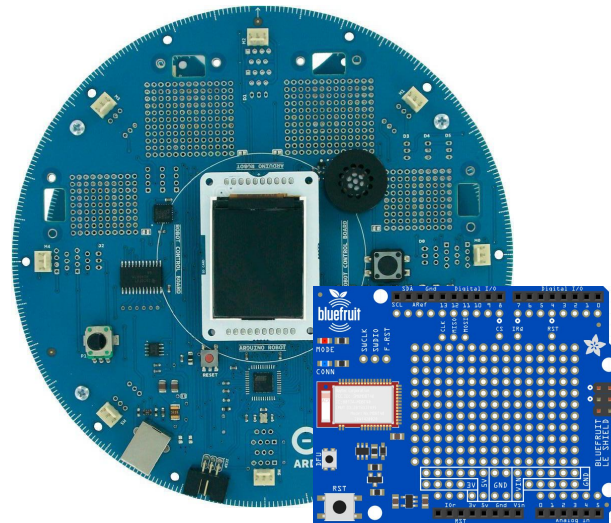
Profils Generic Attribute ou GATT



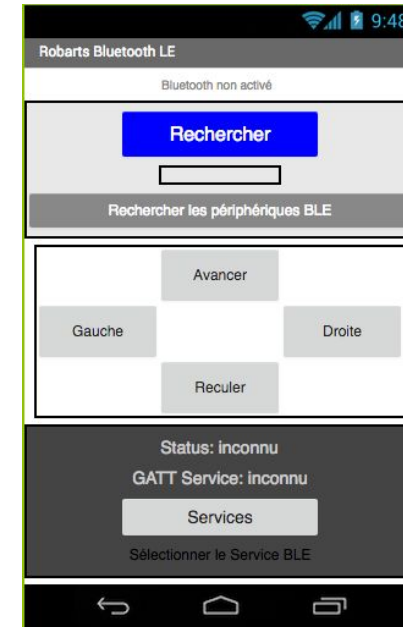
- I. Gestion de projet
- II. Aire de jeu
- III. Les capteurs
- IV. Liaison sans-fil**
- V. Problématique de localisation

Connexion

Réseau de type "Ad-hoc" (sans infrastructure)



Esclave

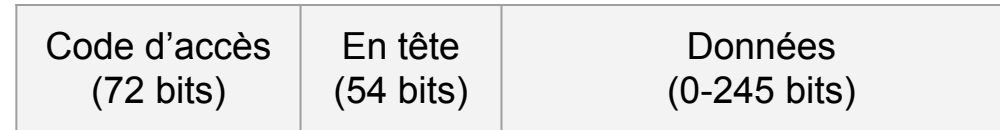


Maître

- I. Gestion de projet
- II. Aire de jeu
- III. Les capteurs
- IV. **Liaison sans-fil**
- V. Problématique de localisation

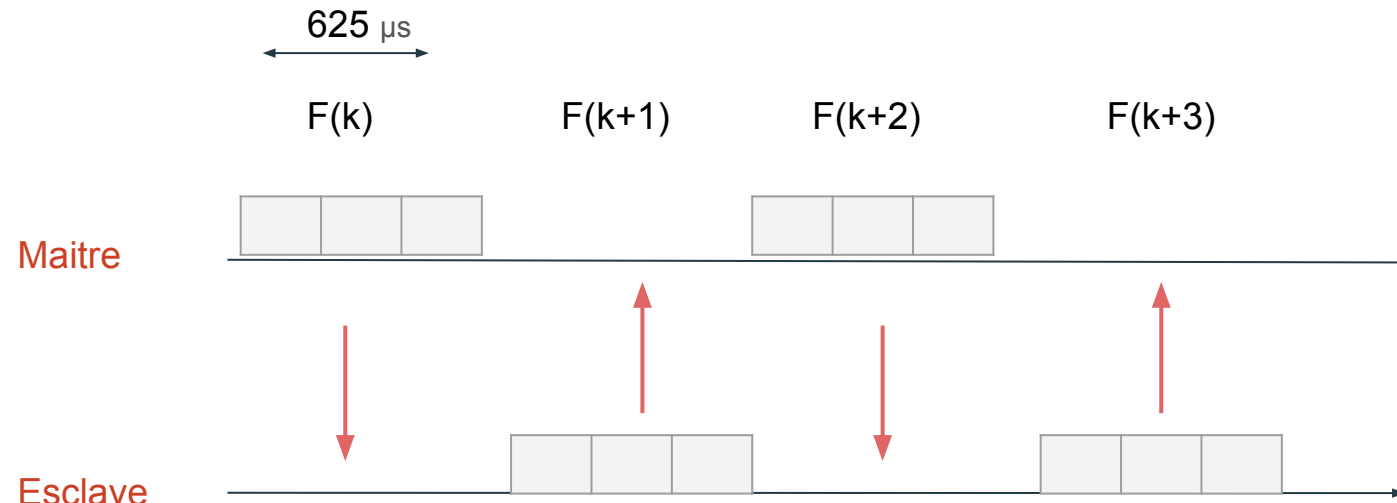
La technologie Bluetooth

Transmission par paquet :



L'Universal Unique Identifier (UUID)
Identifier de manière unique le périphérique

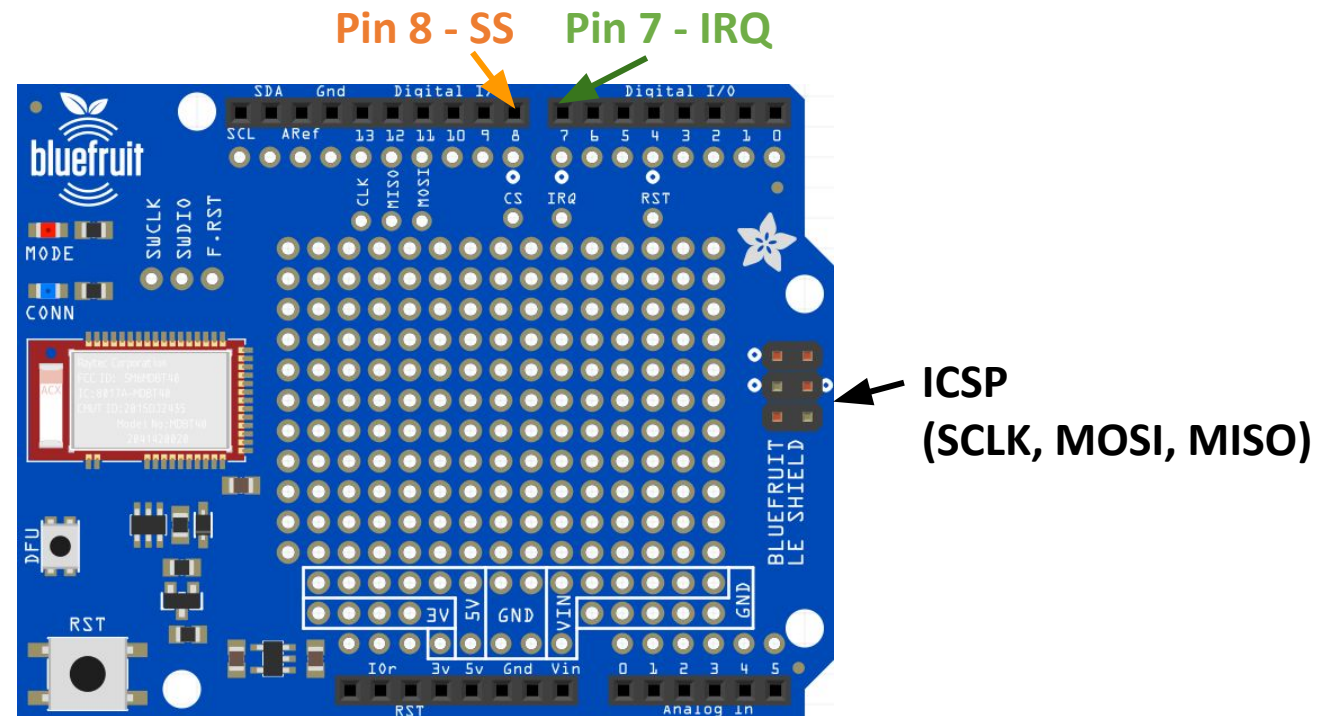
Communication single - slot :



F1:18:F8:A2:5A:61 Adafruit Bluefruit LE

- I. Gestion de projet
- II. Aire de jeu
- III. Les capteurs
- IV. Liaison sans-fil
- V. Problématique de localisation

Carte Bluetooth : Adafruit Bluefruit LE Shield

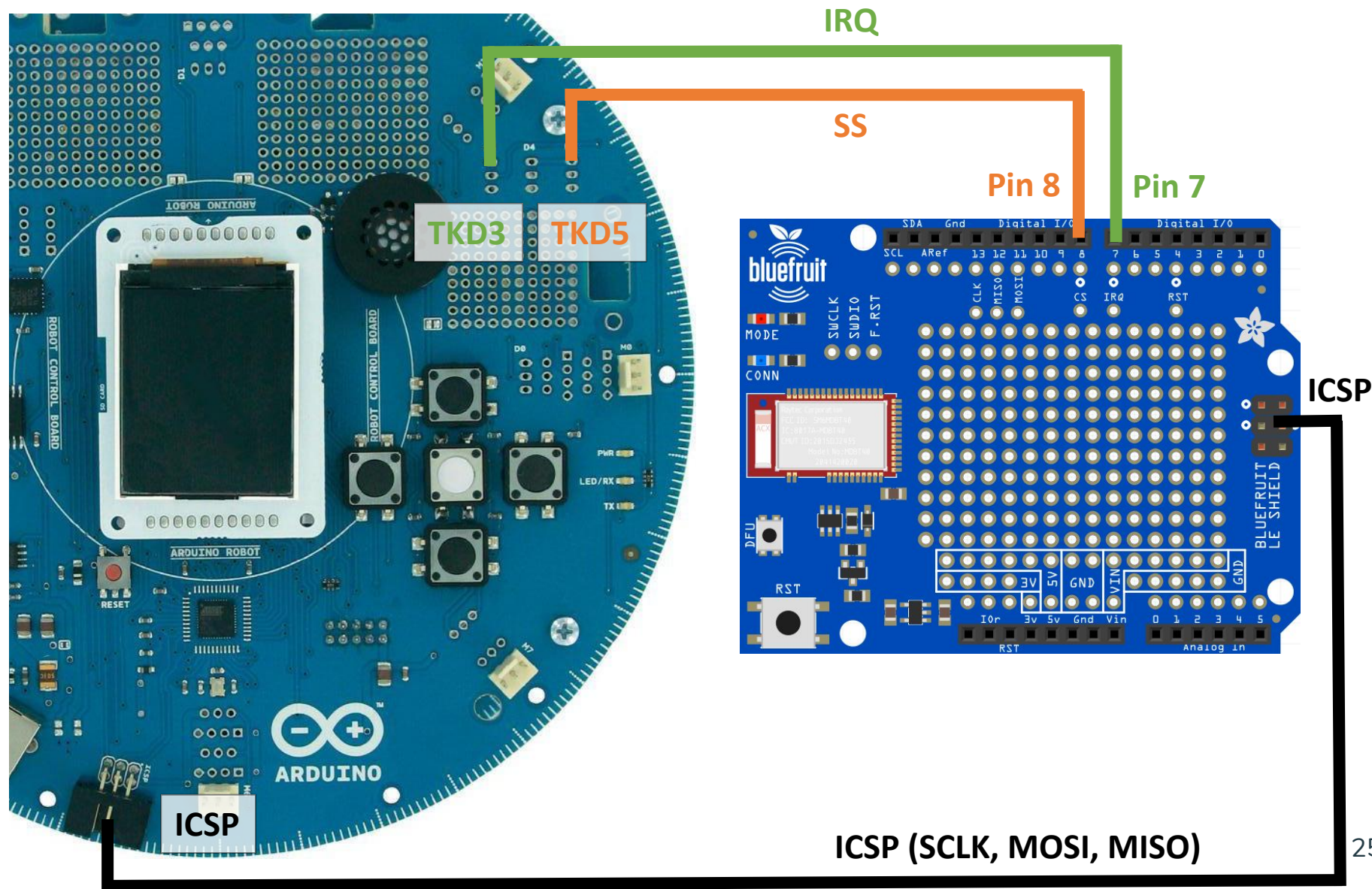


Protocole de connexion SPI (Serial Peripheral Interface)

- **SCLK** — Serial Clock, Horloge (généré par le maître)
- **MOSI** — Master Output, Slave Input (généré par le maître)
- **MISO** — Master Input, Slave Output (généré par l'esclave)
- **SS** — Slave Select, indique quel carte est utilisé sur le bus SPI
- **IRQ** — Pin d'interruption qui indique quand une donnée est reçue par la carte Bluetooth

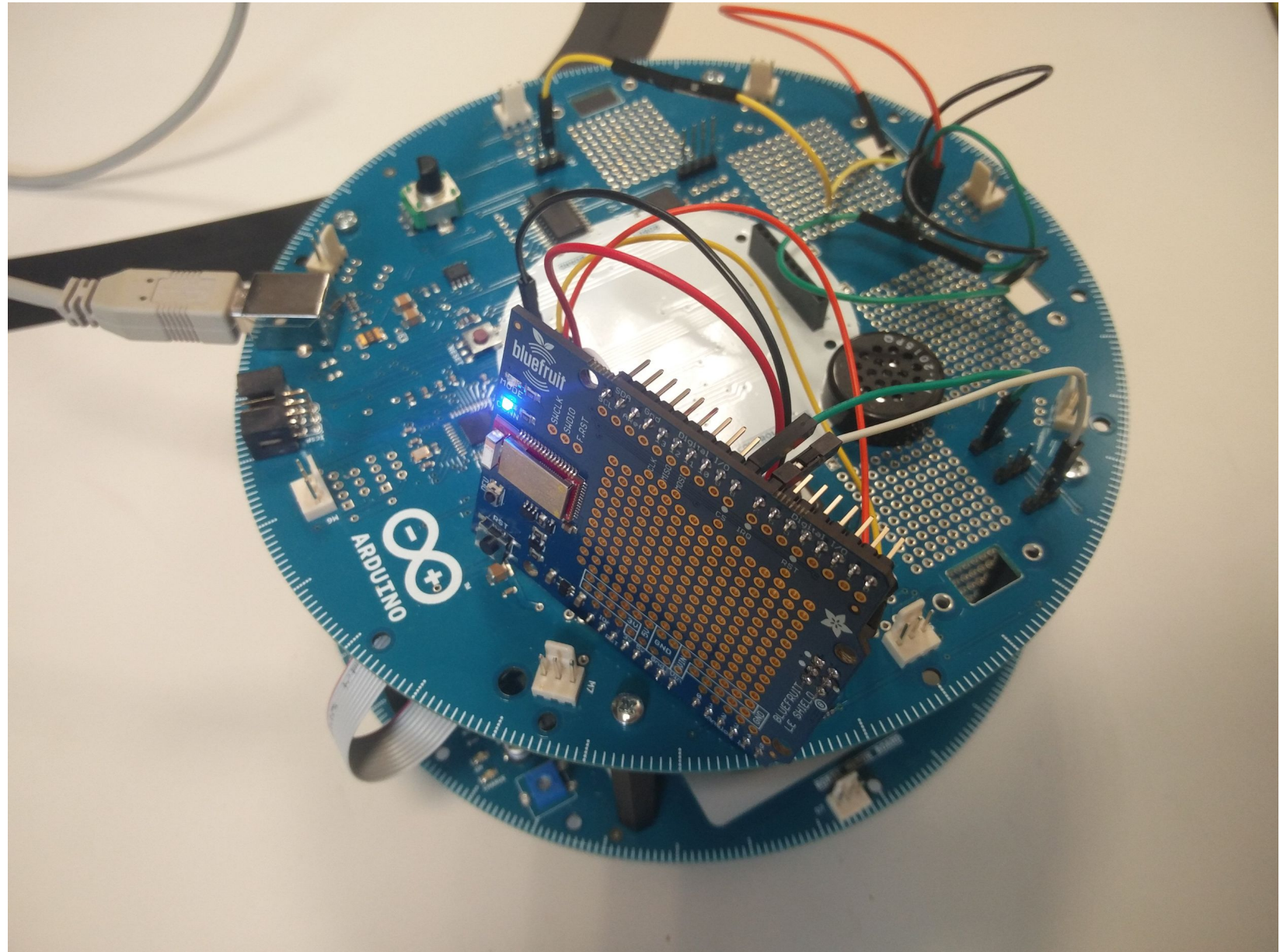
Connexion : Schéma électrique pour établir le plan de connexion de l'interface SPI entre les deux

- I. Gestion de projet
- II. Aire de jeu
- III. Les capteurs
- IV. **Liaison sans-fil**
- V. Problématique de localisation



Connexion

- I. Gestion de projet
- II. Aire de jeu
- III. Les capteurs
- IV. **Liaison sans-fil**
- V. Problématique de localisation



Développement d'une application mobile



- I. Gestion de projet
- II. Aire de jeu
- III. Les capteurs
- IV. **Liaison sans-fil**
- V. Problématique de localisation

```
when BluetoothLE1 .DeviceFound
do
  call BluetoothLE1 .StopScanning
  set ListePeripheriquesBLE . Elements to list from csv row text BluetoothLE1 . DeviceList
  call ListePeripheriquesBLE .Open

when ListePeripheriquesBLE .AfterPicking
do
  set global BLEdeviceName to call BluetoothLE1 .FoundDeviceName index ListePeripheriquesBLE . SelectionIndex
  call BluetoothLE1 .Connect index ListePeripheriquesBLE . SelectionIndex
```

Quand un périphérique Bluetooth a été trouvé

On arrête la recherche de périphérique

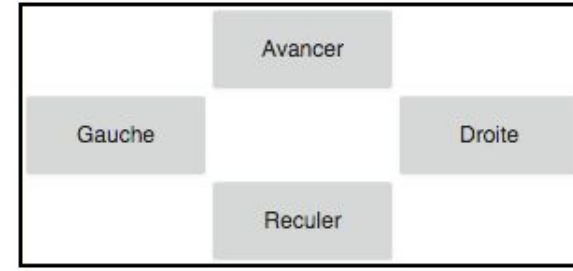
Affichage de la liste des périphériques trouvés

Après avoir sélectionné le périphérique

Enregistrement du nom du périphérique

Connexion à la carte Bluetooth

Envoi des données par bluetooth



```
when Avancer .TouchDown  
do  
  call envoiDirection  
    direction "avance"
```

Lorsqu'on appui sur le bouton Avancer
Appel de la fonction envoiDirection avec le paramètre « avance »

Fonction envoiDirection :

```
to envoiDirection direction  
do  
  call BluetoothLE1 .WriteStrings  
    serviceUuid call BluetoothLE1 .ServiceByIndex  
    index listBLEServices . SelectionIndex  
    values get direction
```

- I. Gestion de projet
- II. Aire de jeu
- III. Les capteurs
- IV. Liaison sans-fil**
- V. Problématique de localisation

Programmation de la carte du robot

```
void loop(void)
{
    String BLEbuffer = ble.buffer; //Création d'une variable BLEBuffer contenant la
    chaîne de caractère reçu

    //Si la commande reçue est « avancer »
    if (BLEbuffer.indexOf("avancer") >= 0) {

        //Robot.motorsWrite(vitesseGauche,vitesseDroit) contrôle la vitesse des moteurs
        gauche et droit du robot.

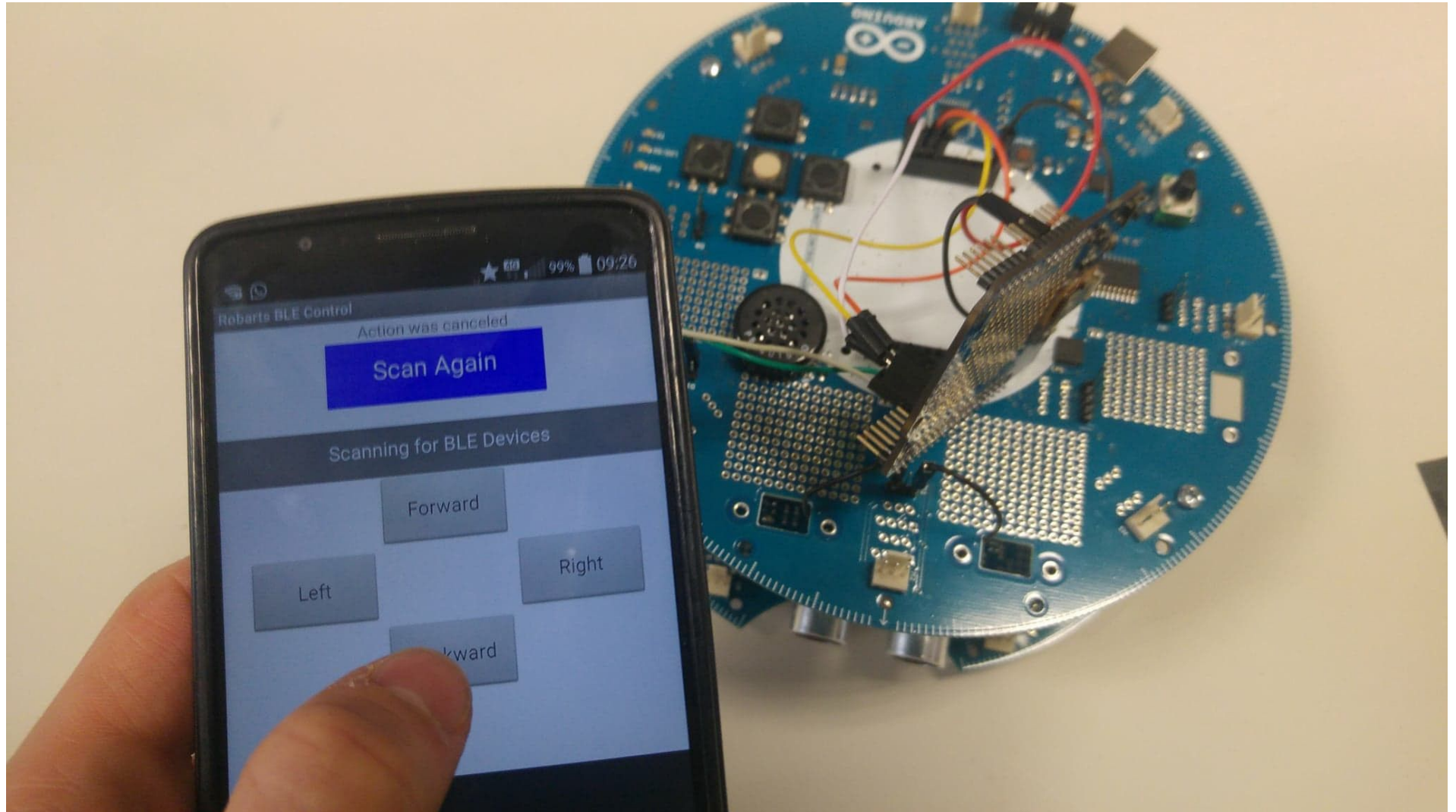
        Robot.motorsWrite(130, 130); //Le robot avance vers l'avant (même vitesse des 2
        moteurs)

    }
}
```

Fonction Loop du programme Arduino

- I. Gestion de projet
- II. Aire de jeu
- III. Les capteurs
- IV. Liaison sans-fil**
- V. Problématique de localisation

- I. Gestion de projet
- II. Aire de jeu
- III. Les capteurs
- IV. Liaison sans-fil**
- V. Problématique de localisation



V. Problématique de localisation

Localisation par ultrasons

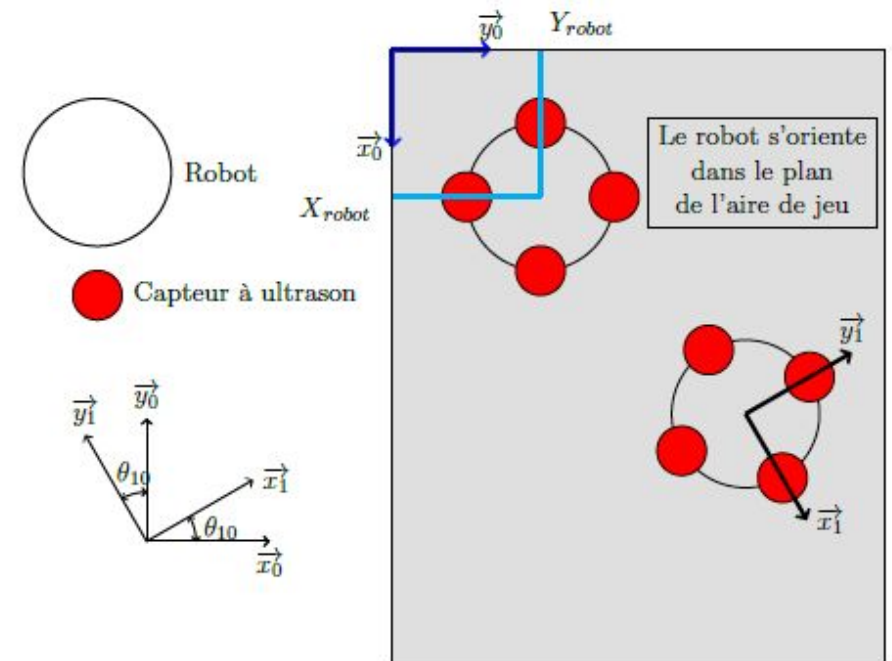
- I. Gestion de projet
- II. Aire de jeu
- III. Les capteurs
- IV. Liaison sans-fil
- V. **Problématique de localisation**

Prises de mesures:

- Définir repère boussole par rapport à la table
- Repère robot \rightarrow Distances aux bords

Problématique:

- Précision capteurs



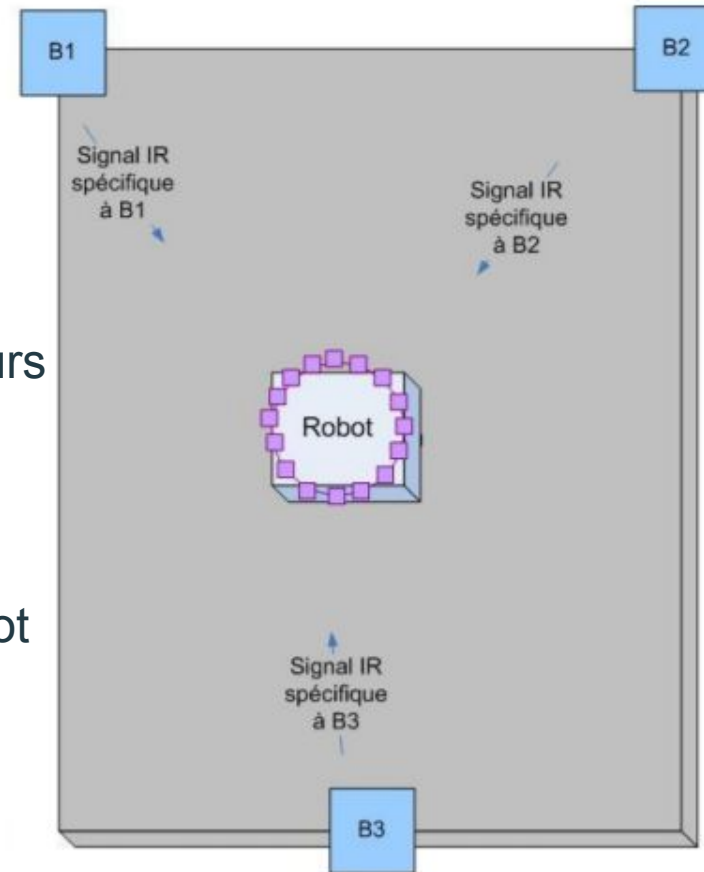
Localisation par infrarouge

- I. Gestion de projet
- II. Aire de jeu
- III. Les capteurs
- IV. Liaison sans-fil
- V. **Problématique de localisation**

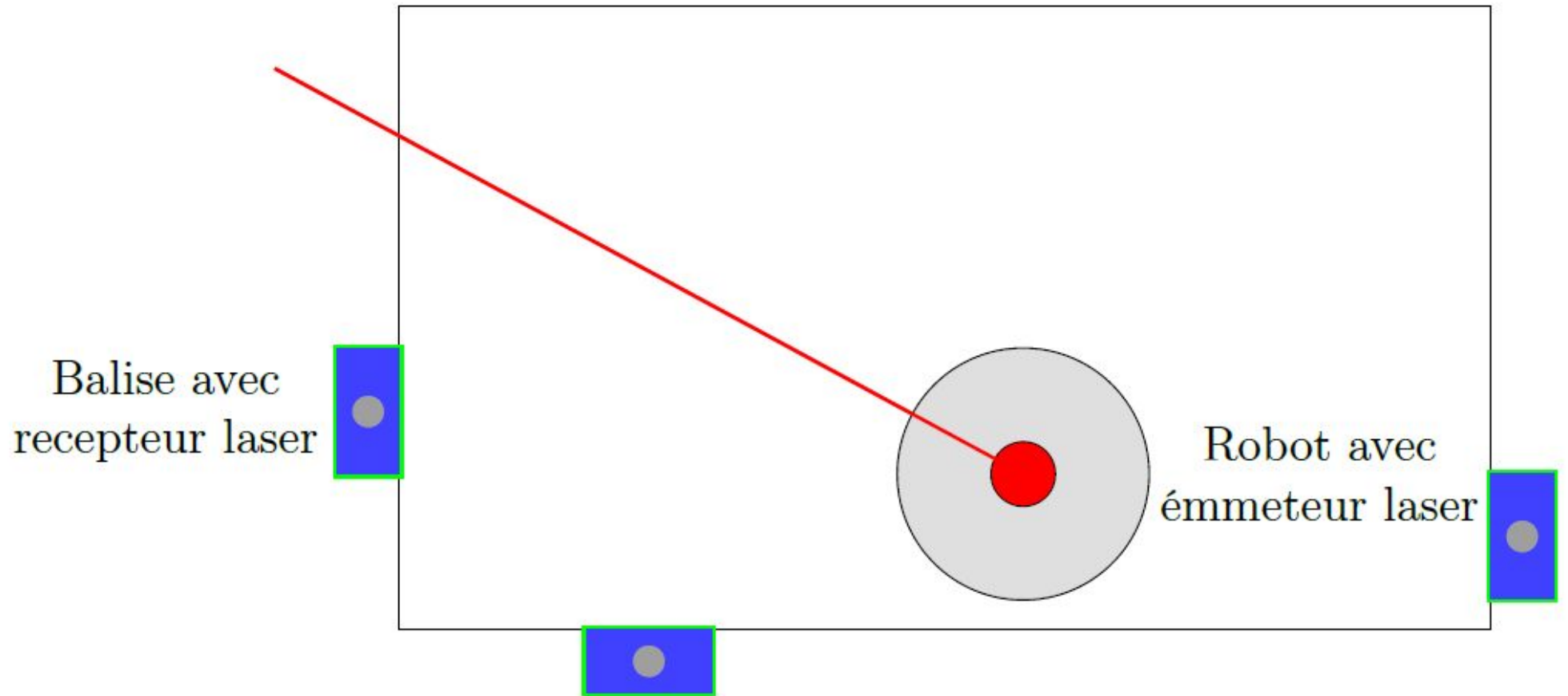
- Le robot est équipé de capteurs infrarouges
- Les balises sont équipés d'émetteurs

Problématique:

- Nombre de pin utilisable par le robot limité



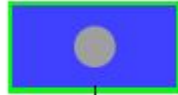
Localisation laser



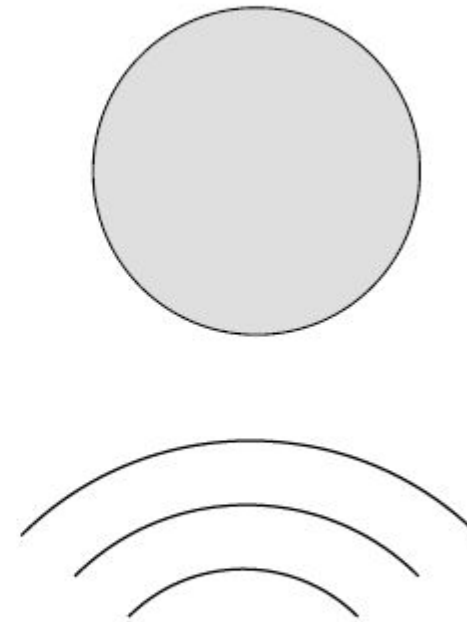
Localisation laser

Reception et analyse
du signal Bluetooth
au niveau du robot

Reception d'un
signal laser



Horloge

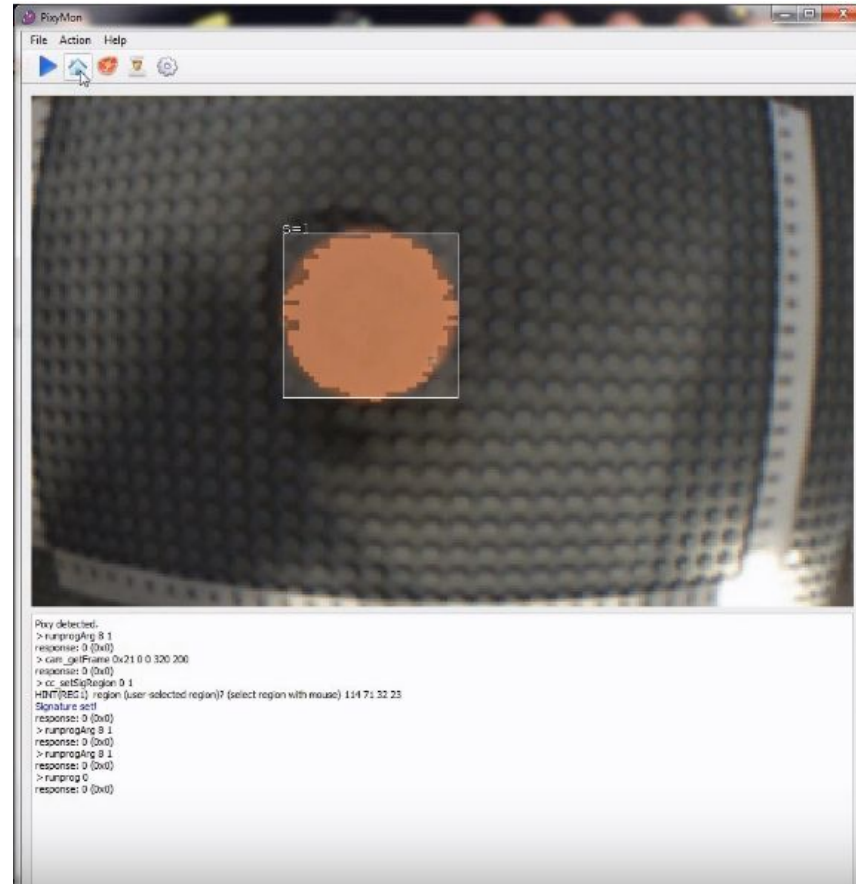


Emission d'un
signal Bluetooth

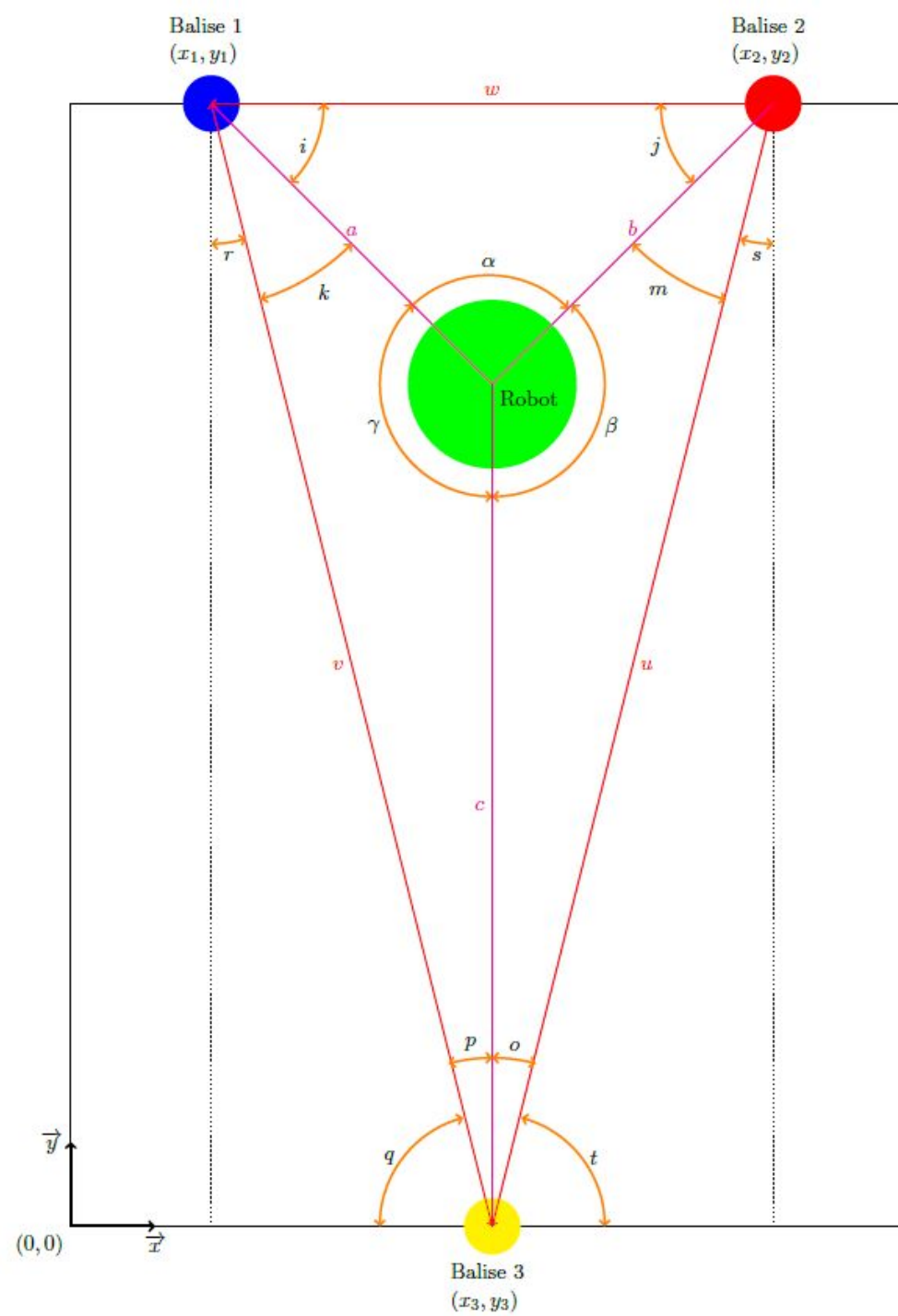
- I. Gestion de projet
- II. Aire de jeu
- III. Les capteurs
- IV. Liaison sans-fil
- V. **Problématique de localisation**

Localisation par triangulation à l'aide d'une caméra

- I. Gestion de projet
- II. Aire de jeu
- III. Les capteurs
- IV. Liaison sans-fil
- V. **Problématique de localisation**



- I. Gestion de projet
- II. Aire de jeu
- III. Les capteurs
- IV. Liaison sans-fil
- V. **Problématique de localisation**



Démonstration de la formule de triangulation

$$j = \arccos\left(\sqrt{\frac{[A \cdot \cos(s - \beta + 90) + B \cdot \cos(\alpha - 180)]^2}{[A \cdot \cos(s - \beta + 90) + B \cdot \cos(\alpha - 180)]^2 + [A \cdot \sin(s - \beta + 90) + B \cdot \sin(\alpha - 180)]^2}}\right)$$

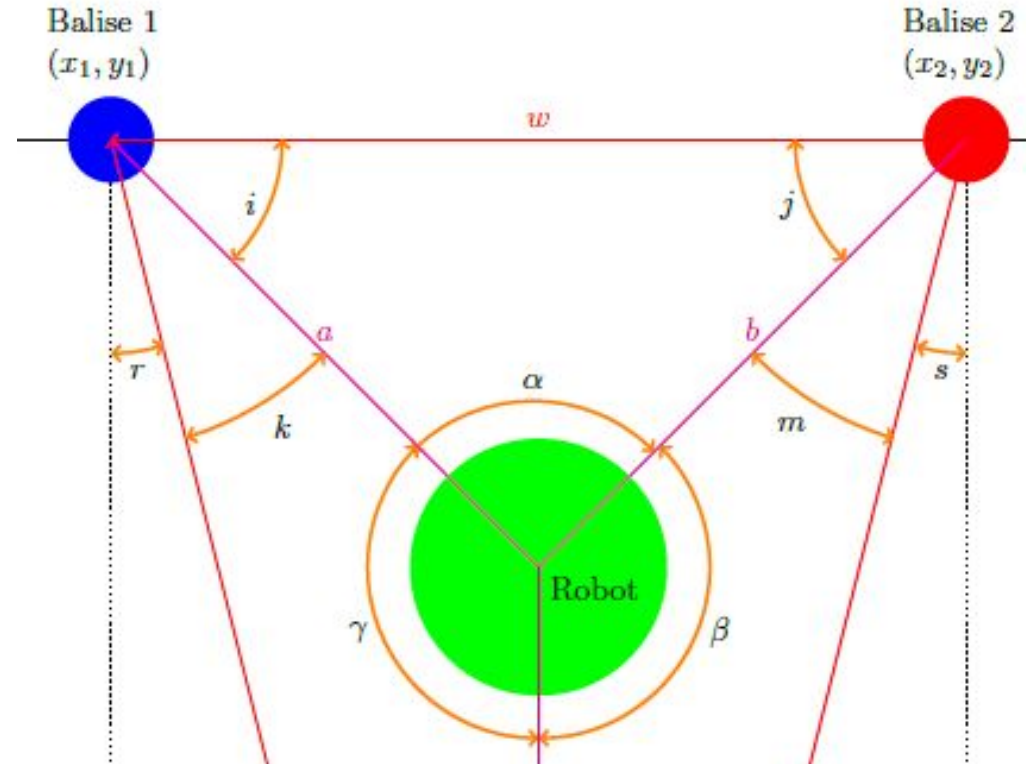
- I. Gestion de projet
- II. Aire de jeu
- III. Les capteurs
- IV. Liaison sans-fil
- V. **Problématique de localisation**

Etapes:

- Formule du sinus dans les 3 triangles
- Somme des angles d'un triangle
- Couplage des équations

$$\begin{cases} i = 180 - \alpha - j \\ a = \frac{w}{\sin \alpha} \cdot \sin j \\ b = \frac{w}{\sin \alpha} \cdot \sin i \end{cases} \begin{cases} A = \frac{u}{\sin \beta} \\ B = \frac{w}{\sin \alpha} \\ \Phi_1 = s - \beta + 90 \\ \Phi_2 = \alpha - 180 \end{cases}$$

$$\begin{cases} x = x_1 + \cos(i) * a \\ y = y_1 - \sin(i) * a \end{cases}$$



Programme : formule de triangulation

```
void position(float alpha,float beta,float gama) //Fonction de calcul de la position en
fonction des angles inter balise
{
    //Dimensions et angles connus
    float L = 2040; // longueur du terrain
    float w = 1660; // longueur entre B1 et B2 = 2 x 830 mm
    float v = sqrt(pow(830,2)+ pow(2040,2)); // longueur entre B1 et B3
    float u = sqrt(pow(830,2)+ pow(2040,2)); // longueur entre B2 et B3
    float r = atan(830/2040); // angle(0,B1,B3)
    float q = (M_PI/2)-r; // angle(B1,B3,0)
    float s = atan(830/2040); //angle
    float t = (M_PI/2)-s; //agle

    // Calcul de l'angle j
    float j =
    acos(sqrt(pow(((u/sin(beta))*cos(s-beta+M_PI/2)+(w/sin(alpha))*cos(alpha-M_PI)),2)/(pow(((u/si
    n(beta))*cos(s-beta+M_PI/2)+(w/sin(alpha))*cos(alpha-M_PI)),2)+pow(((u/sin(beta))*sin(s-beta+M
    _PI/2)+(w/sin(alpha))*sin(alpha-M_PI)),2)))));
```

- I. Gestion de projet
- II. Aire de jeu
- III. Les capteurs
- IV. Liaison sans-fil
- V. **Problématique de localisation**

Programme : formule de triangulation

- I. Gestion de projet
- II. Aire de jeu
- III. Les capteurs
- IV. Liaison sans-fil
- V. **Problématique de localisation**

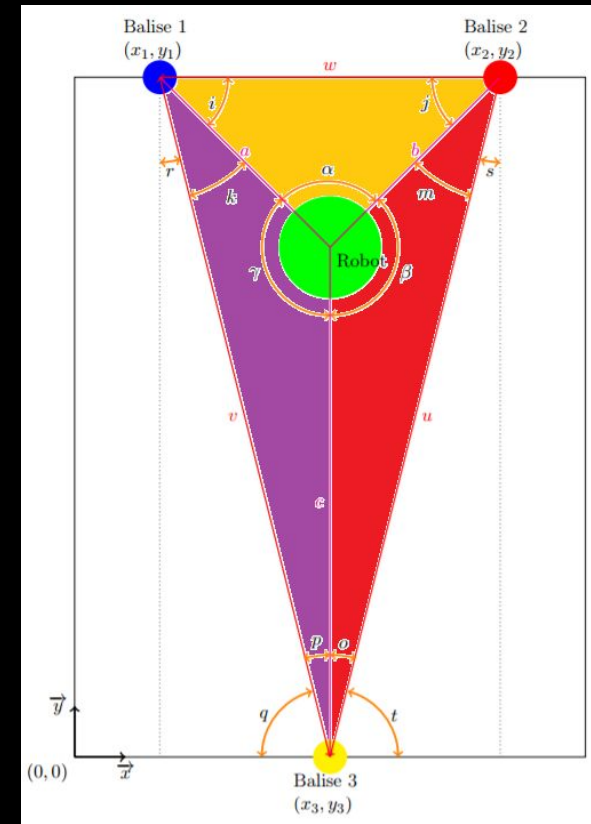
```
// Calcul de l'angle i
float i = M_PI-alpha-j;
printf("%.6f", j);
printf("\n");
printf("%.6f", i);
printf("\n");
```

```
//Calcul des longueurs inconnues
float a = w*sin(j)/sin(alpha);
float b = w*sin(i)/sin(alpha);
printf("%.6f", a);
printf("\n");
printf("%.6f", b);
printf("\n");
```

```
//Calcul des coordonnées
```

```
x = 0 + cos(i)*a;
y = L - sin(i)*a;
```

```
}
```



$$\begin{cases} x = x_1 + \cos(i) * a \\ y = y_1 - \sin(i) * a \end{cases}$$

Conclusion

Conclusion

Quelle suite pour le projet ?

- Levée de fonds
- Intégration de la Pixy Cam sur le robot
- Déplacement avec précision à une position donnée
- Réalisation du cache du robot

Démonstration de la formule de triangulation

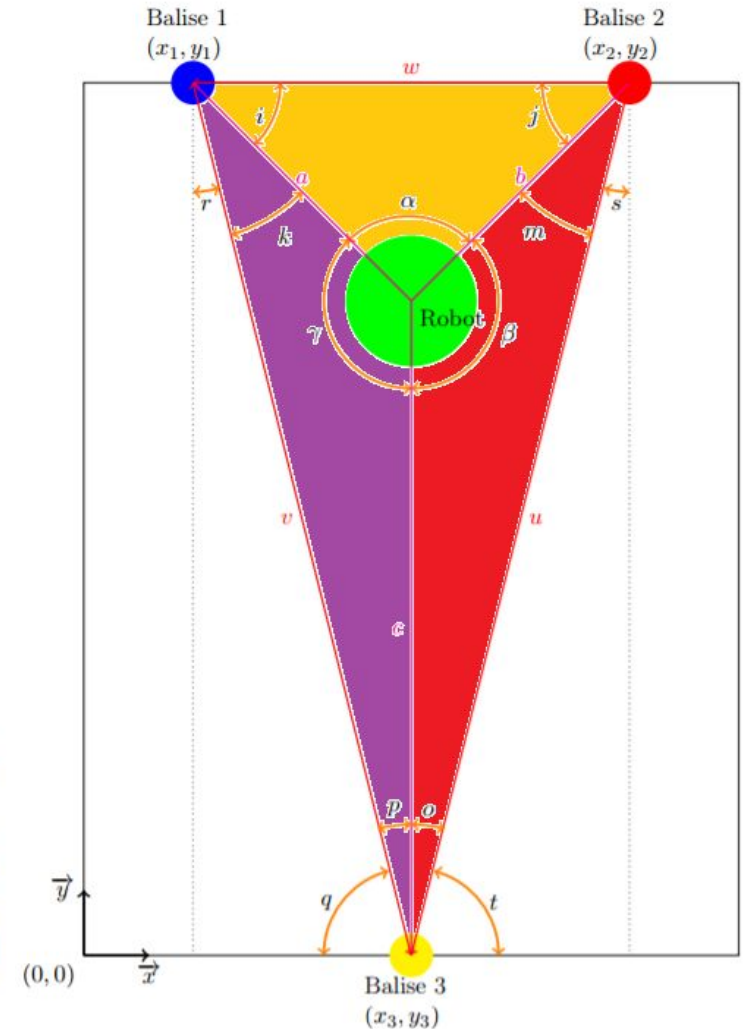
Etapes:

- Formule du sinus dans les 3 triangles

$$\left\{ \begin{array}{l} \frac{v}{\sin \gamma} = \frac{a}{\sin p} = \frac{c}{\sin k} \\ \frac{w}{\sin \alpha} = \frac{a}{\sin j} = \frac{b}{\sin i} \\ \frac{u}{\sin \beta} = \frac{b}{\sin o} = \frac{c}{\sin m} \end{array} \right.$$

- Expression de a, b et c en fonction des autres paramètres

$$\left\{ \begin{array}{l} a = \frac{v}{\sin \gamma} \cdot \sin p = \frac{w}{\sin \alpha} \cdot \sin j \quad (1) \\ b = \frac{w}{\sin \alpha} \cdot \sin i = \frac{u}{\sin \beta} \cdot \sin o \quad (2) \\ c = \frac{u}{\sin \beta} \cdot \sin m = \frac{v}{\sin \gamma} \cdot \sin k \quad (3) \end{array} \right.$$



- I. Gestion de projet
- II. Aire de jeu
- III. Les capteurs
- IV. Liaison sans-fil
- V. **Problématique de localisation**

Démonstration de la formule de triangulation

Etapes:

- Par définition:

$$\begin{cases} j + m = 90 - s \\ m = 180 - o - \beta \end{cases}$$

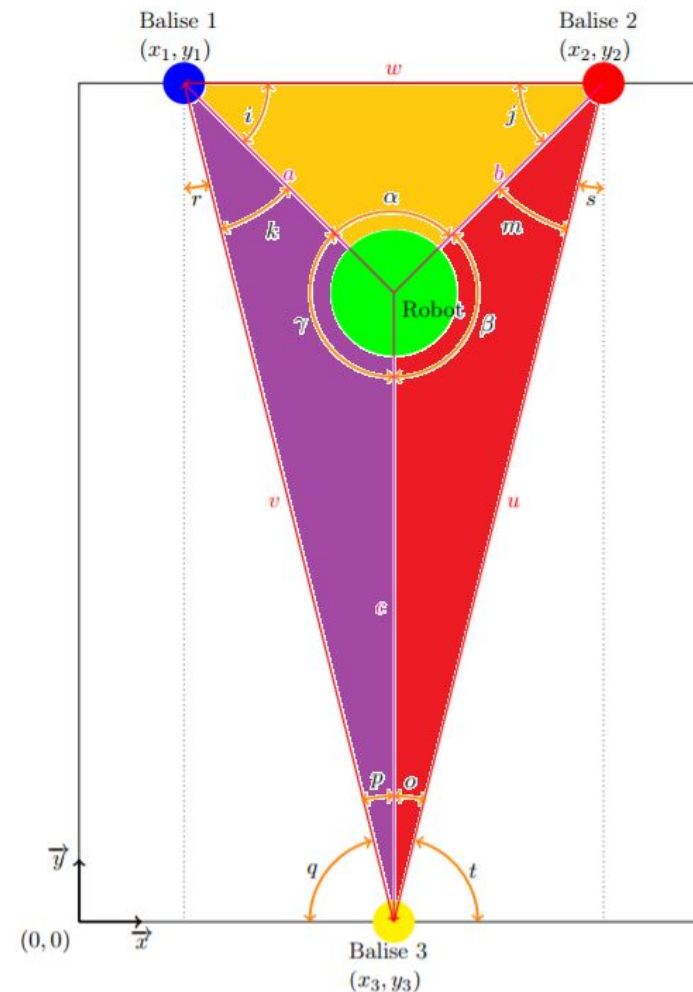
- On conclut que

$$o = j + s - \beta + 90$$

$$i = 180 - j - \alpha$$

- On injecte dans (2)

$$\frac{w}{\sin \alpha} \cdot \sin(180 - j - \alpha) = \frac{u}{\sin \beta} \cdot \sin(j + s - \beta + 90) \quad (4)$$



- I. Gestion de projet
- II. Aire de jeu
- III. Les capteurs
- IV. Liaison sans-fil
- V. **Problématique de localisation**

Démonstration de la formule de triangulation

- On injecte dans (2):

$$\frac{w}{\sin \alpha} \cdot \sin(180 - j - \alpha) = \frac{u}{\sin \beta} \cdot \sin(j + s - \beta + 90)$$

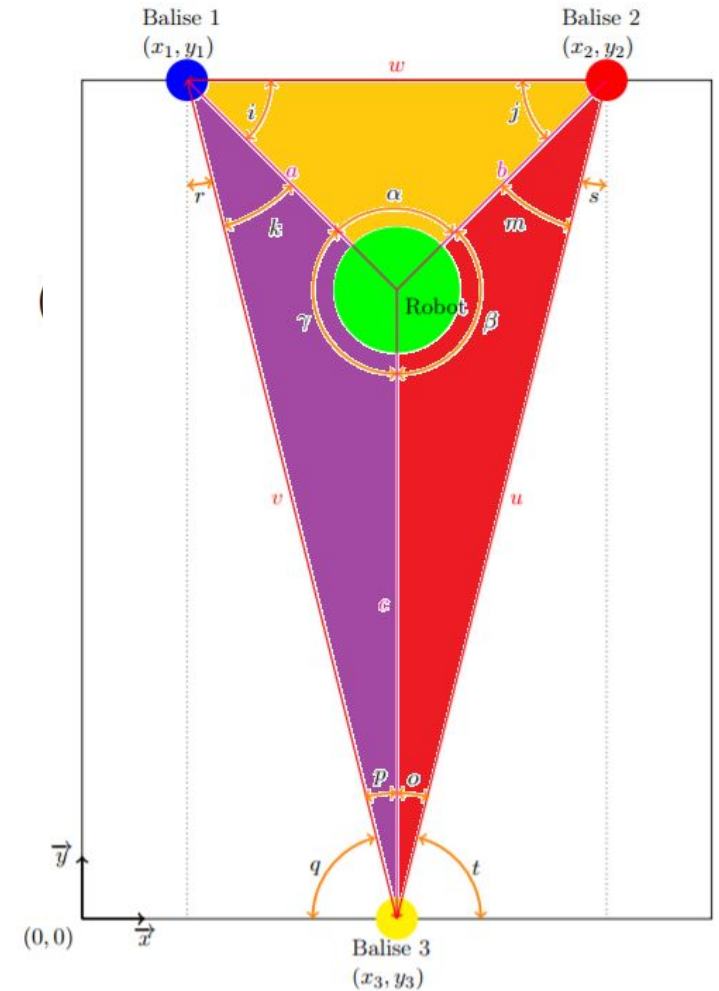
- On pose:

$$\begin{cases} A = \frac{u}{\sin \beta} \\ B = \frac{w}{\sin \alpha} \\ \Phi_1 = s - \beta + 90 \\ \Phi_2 = \alpha - 180 \end{cases}$$

- On a donc:

$$A \cdot \sin(j + \Phi_1) = -B \cdot \sin(j + \Phi_2)$$

- On développe le sinus



- I. Gestion de projet
- II. Aire de jeu
- III. Les capteurs
- IV. Liaison sans-fil
- V. **Problématique de localisation**

Démonstration de la formule de triangulation

- I. Gestion de projet
- II. Aire de jeu
- III. Les capteurs
- IV. Liaison sans-fil
- V. **Problématique de localisation**

- On pose

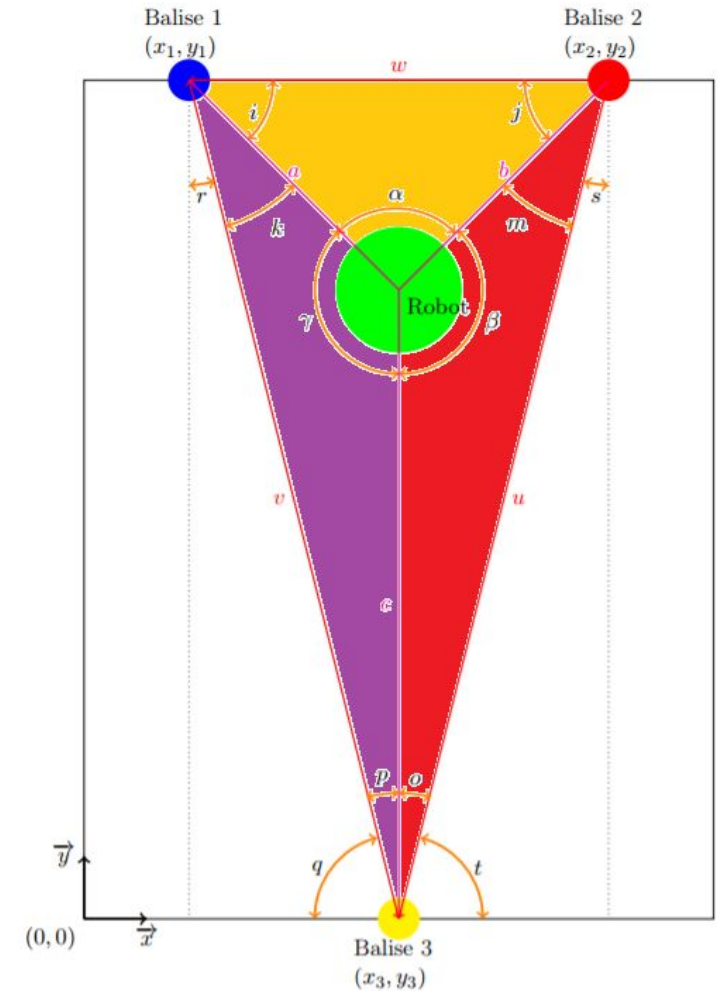
$$\begin{cases} U = A \cdot \cos \Phi_1 + B \cdot \cos \Phi_2 \\ V = A \cdot \sin \Phi_1 + B \cdot \sin \Phi_2 \end{cases}$$

- On a donc

$$U \cdot \sin j = -V \cdot \cos j$$

$$U \cdot \sqrt{1 - \cos^2 j} = -V \cos j$$

$$\cos j = \sqrt{\frac{U^2}{U^2 + V^2}}$$



Démonstration de la formule de triangulation

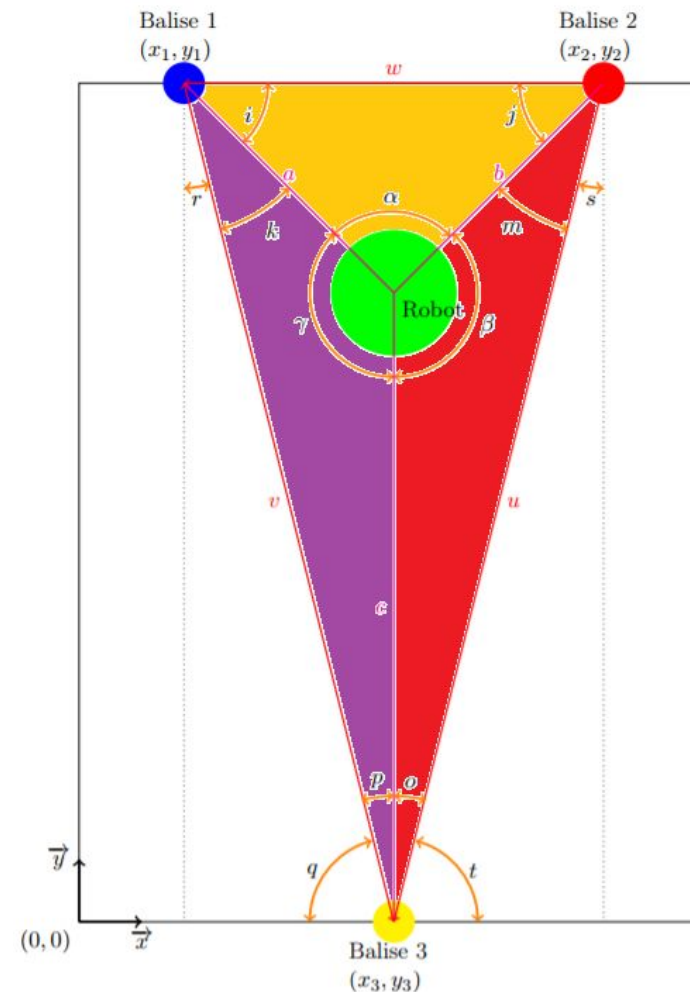
- On a donc sachant que $-1 \leq \sqrt{\frac{U^2}{U^2+V^2}} \leq 1$

$$j = \arccos\left(\sqrt{\frac{U^2}{U^2 + V^2}}\right)$$

- Enfin on a:

$$\text{Avec: } \begin{cases} i = 180 - \alpha - j \\ a = \frac{w}{\sin \alpha} \cdot \sin j \\ b = \frac{w}{\sin \alpha} \cdot \sin i \end{cases}$$

$$j = \arccos\left(\sqrt{\frac{[A \cdot \cos(s - \beta + 90) + B \cdot \cos(\alpha - 180)]^2}{[A \cdot \cos(s - \beta + 90) + B \cdot \cos(\alpha - 180)]^2 + [A \cdot \sin(s - \beta + 90) + B \cdot \sin(\alpha - 180)]^2}}\right)$$



- I. Gestion de projet
- II. Aire de jeu
- III. Les capteurs
- IV. Liaison sans-fil
- V. **Problématique de localisation**