



### Enoncé du besoin initial :

Aujourd’hui, les transports publics traditionnels (bus, tramway et métro) fonctionnent de manière optimale dans les centres-villes denses et les proches banlieues. Pour autant les quartiers à plus faible densité sont souvent moins bien desservis, ce qui constraint l’usager à marcher sur de longues distances ou à avoir recours à son véhicule personnel pour rejoindre la destination de son choix ou accéder à la station du **Réseau de Transports en Commun** (RTC) la plus proche.

Cette desserte insuffisante du premier ou dernier kilomètre a donc un impact direct sur le choix du mode de déplacement du citadin.



Les intentions sont de développer une mobilité multimodale et décarbonée en complément des services de transports en commun déjà existants. Les enjeux sont doubles :

- **Proposer une mobilité sans contrainte, garantissant :**
  - une continuité totale des déplacements quelque soit la multimodalité<sup>1</sup> ou l’intermodalité<sup>2</sup> ;
  - une diminution significative de l’usage de la voiture individuelle.
- **Favoriser, de fait, la réduction de l’impact environnemental et physique de la mobilité avec :**
  - la maîtrise des consommations d’énergie et des émissions de polluant ;
  - la fluidité d’écoulement de la circulation routière.

<sup>1</sup> La multimodalité désigne la présence de plusieurs modes de transport différents entre deux lieux.

<sup>2</sup> L’intermodalité est l’utilisation de plusieurs modes de transport au cours d’un même déplacement.



### Hypothèses simplificatrices :

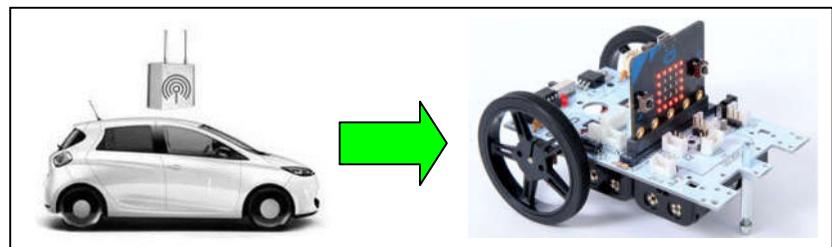
Pour se faire, vous allez progressivement concevoir, mettre en œuvre et expérimenter, à échelle réduite, un service de mobilité à la demande doté d’une flotte de **Véhicules Autonomes Electriques** (VAE) circulant sur une voie exclusivement réservée à ce type de transport.

La figure qui suit, montre un périmètre d’expérimentation - grandeur nature - qui serait située en zone périurbaine, à proximité d’une grande métropole dotée d’un réseau RTC.

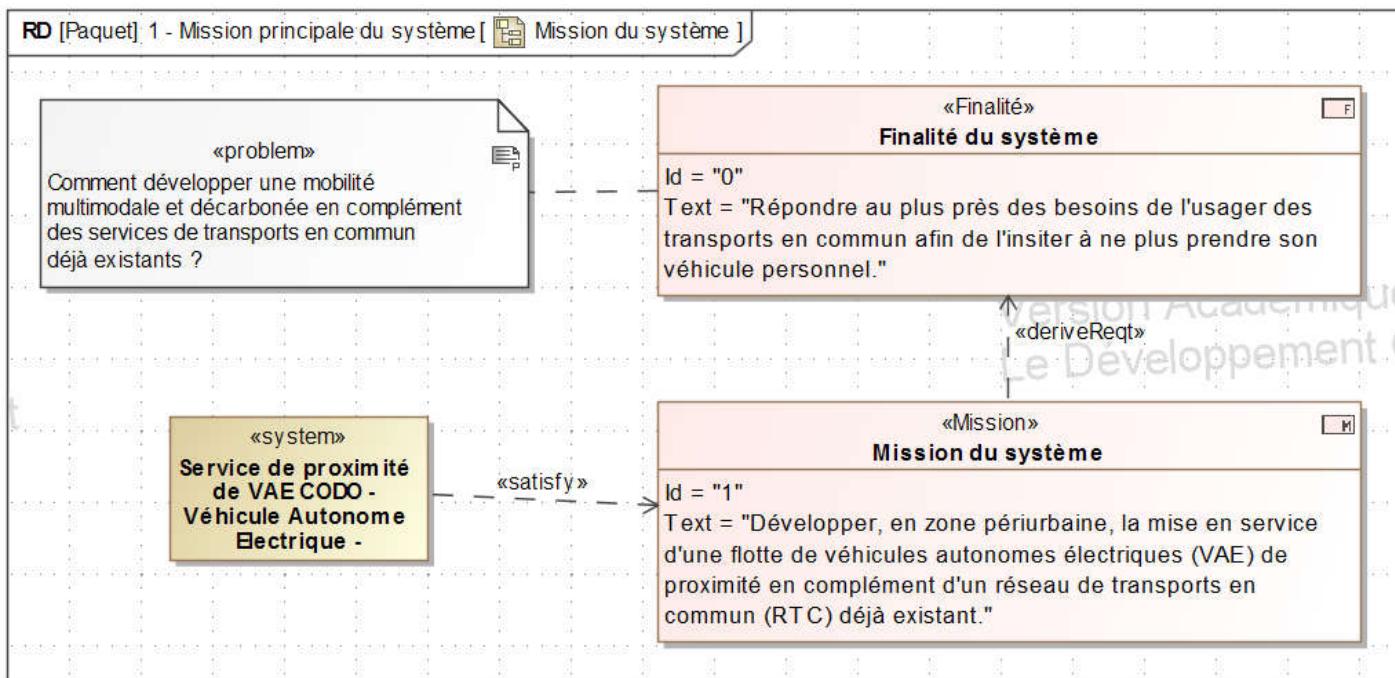


Le VAE doit permettre à tout utilisateur qui en fait la demande, de relier un point d'arrêt à proximité de son domicile à la station du RTC la plus proche dotée d'une borne TERMINUS, en empruntant un circuit bouclé à sens unique de quelques kilomètres. Ce point d'arrêt comporte une borne d'appel du VAE mise à disposition pour le client. Dans ce mini-projet, seul le trajet Aller (domicile → terminus) est abordé. Notons que le chemin du retour (Terminus → Domicile) emprunterait un parcours similaire sur une voie parallèle par l'intermédiaire d'une autre flotte de VAE circulant en sens opposé.

A l'échelle du laboratoire de SI et du mini-projet proposé, la flotte de VAE dédiée au trajet ALLER sera fixée de façon arbitraire à 4 véhicules apparentés à 4 Robots CODO ou équivalents. Dans la réalité, le nombre de VAE dépendrait de la fréquence des sollicitations afin de s'assurer que le client n'est pas à s'impatienter plus de quelques minutes.

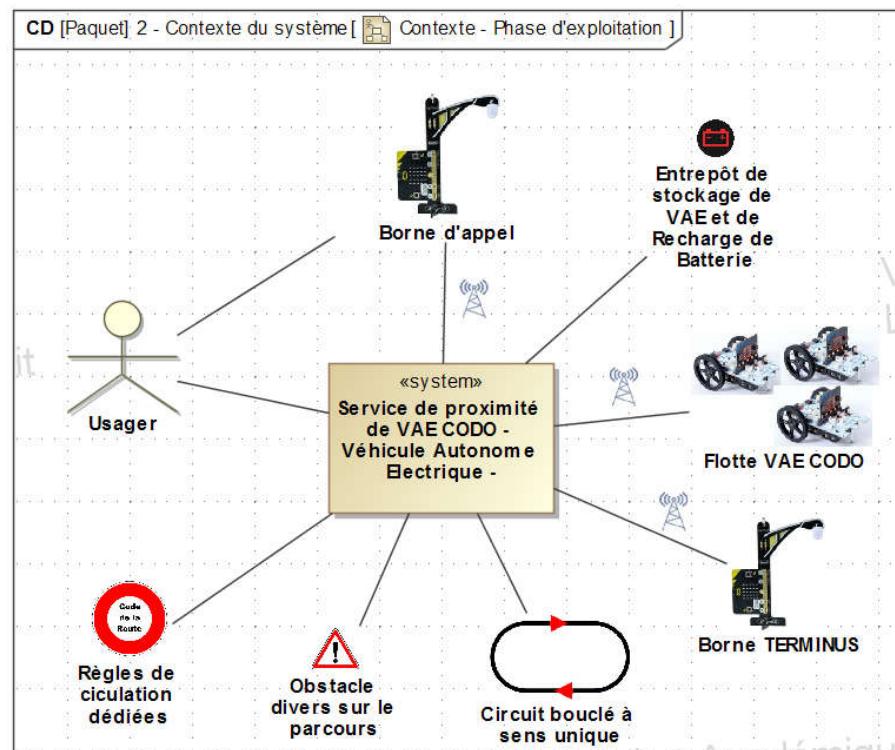


### Mission principale du SERVICE DE PROXIMITE VAE CODO

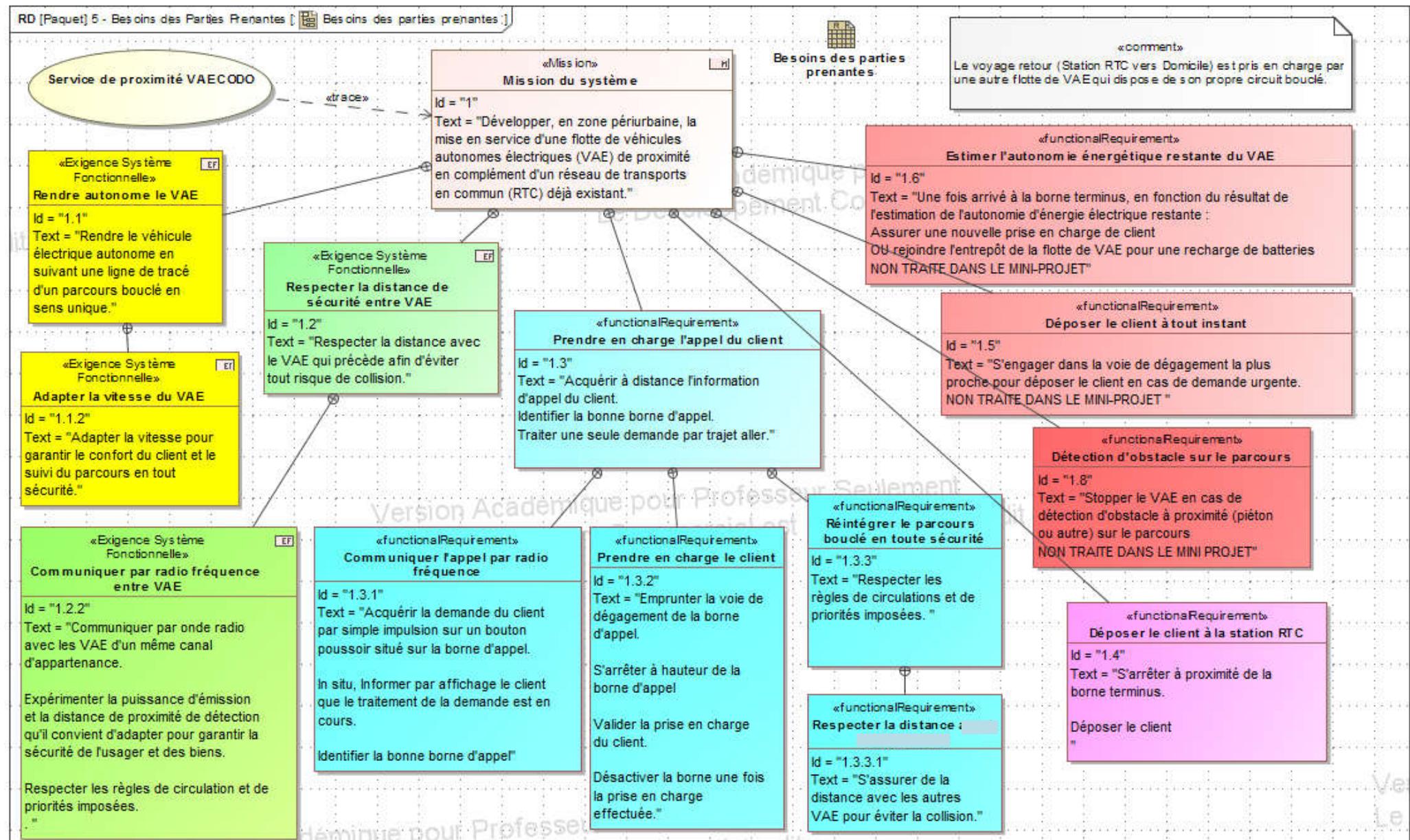


### Diagramme de contexte

Ce diagramme présente les acteurs et les éléments environnant du véhicule autonome en phase d'utilisation.



## Diagramme des exigences du mini-projet



### Règles de circulation imposées

Au départ de l'entrepôt ou de la borne Terminus, le VAE CODO 1 part en premier.

Ce même VAE CODO 1 arrive en premier à la borne terminus du trajet aller.

Les VAE CODO 1 puis 2, 3 et 4 partent au fur et à mesure de l'apparition des appels clients.

Le VAE CODO Ci+1 ne dépasse jamais le VAE Ci.

Un VAE CODO ne peut répondre qu'à un seul appel par trajet Aller.

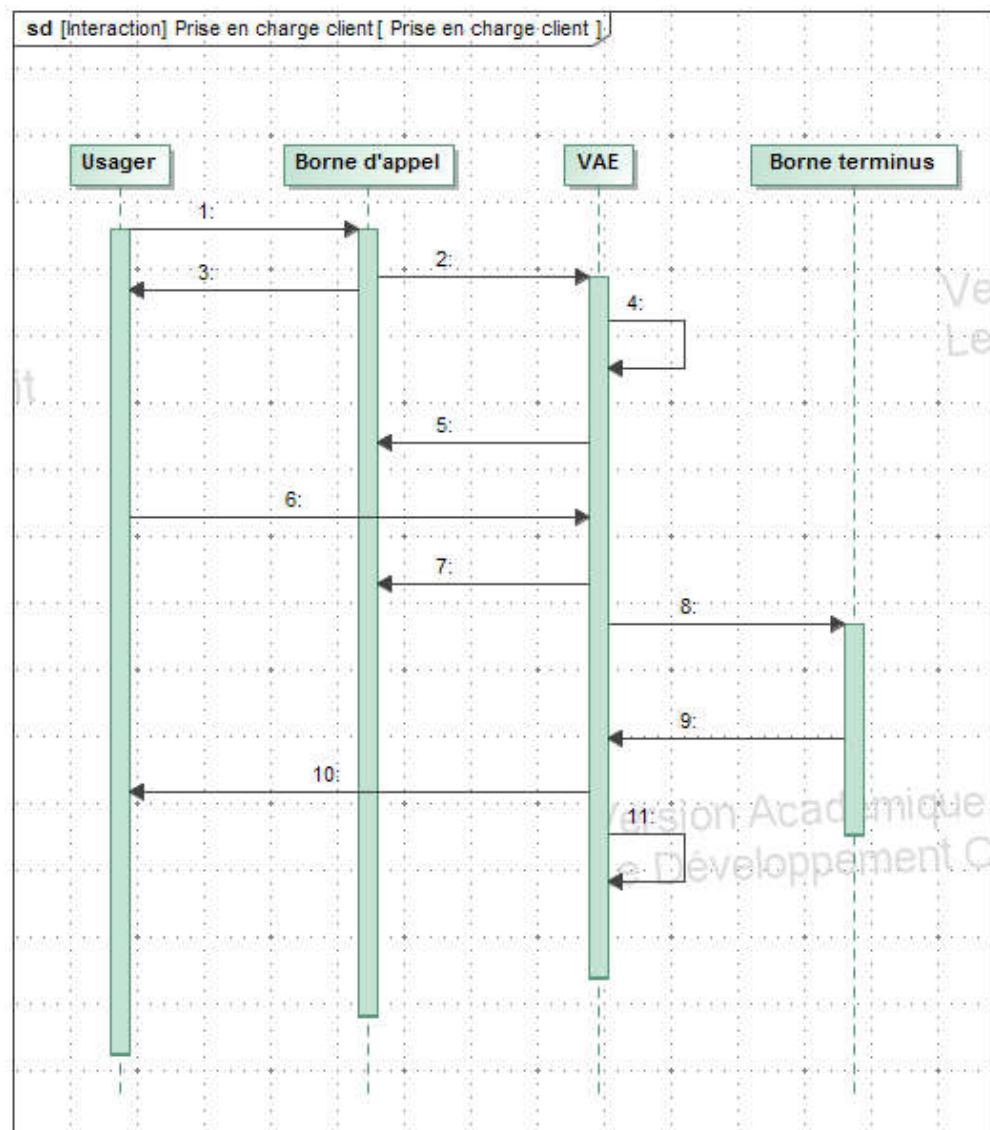
### Scénario d'utilisation simplifié lors d'un déplacement domicile vers terminus (station de RTC)

L'utilisateur appelle le VAE depuis la borne d'appel le plus proche de son domicile.

Une fois l'appel effectué, le premier VAE disponible se rend à proximité de la borne d'appel, s'engage dans la voie de dégagement pour récupérer le client. Une fois le client embarqué, le VAE achemine ce dernier jusqu'à la station de RTC la plus proche (le terminus).

### Diagramme de séquence

Ce diagramme illustre la chronologie des échanges entre les différents acteurs du système lors d'un cycle de déplacement aller domicile vers terminus.



- 1- Compléter le tableau suivant en indiquant le numéro de lien correspondant à chaque échange entre acteurs du diagramme de séquence.

N° du lien	Nature de l'échange ou de l'action
4	Enregistrer l'appel si le VAE CODO est disponible
	Appuyer sur le bouton poussoir d'appel
	Se rendre à la borne d'appel sollicitée et s'arrêter à hauteur de celle-ci
	Emettre l'appel avec l'identifiant de borne
	Monter dans le VAE le temps de l'ouverture des portes et Valider la prise en charge du client
	Informier le client de la prise en compte de la demande
	Estimer l'autonomie restante de charge des batteries
	Désactiver l'appel en cours une fois la prise en charge validée
	Déposer le client à hauteur de la station de RTC
	Ouvrir les portes du VAE durant un certain temps
	Acheminer le client jusqu'à la borne terminus

**Précisions :** En raison du faible temps imparti (# 6 heures), il ne s'agit pas de simuler l'ensemble du trafic du service de proximité mais de résoudre, par partie, les exigences qui ont un rôle majeur.

## Répartition des tâches du Mini projet – Service de proximité VAE CODO

Activité	Chronologie	Travail		Innover	Expérimenter	Répartition /éq. et /binôme	INVENTAIRE DES ACTIVITES ET PROGRESSION sur la base de 4 équipes de 4 élèves			Matériel pour 4 équipes  <b>4 CODOS</b> + 4 BORNES d'APPEL (avec carte µBit + Batterie) + 6 BALISES (carte µBIT alimentées en USB)
		Collaboratif	Coopératif				Intitulé de l'activité	Problème technique à résoudre		
<b>ACTIVATION (H1)</b>							<b>ETRE CAPABLE D'EMPRUNTER LA VOIE PRINCIPALE ET LES CONTRE-ALLEES DE FACON AUTONOME</b>			
1	H1	X			X	Eq 1 à 4	VAE CODO Suiveur de ligne	Comment suivre un parcours avec la meilleure fluidité de déplacement qui soit ? (Suivre un parcours en autonomie et Adapter la vitesse de déplacement)		1 CODO avec batterie / équipe
2A	H2		X	X	X	Bin 1 /Eq	Emprunt de la voie de dégagement	Comment s'engager dans la voie de droite qui se présente avec la meilleure fluidité de déplacement qui soit ? (Adapter la vitesse de rotation vers la droite)		1 CODO avec batterie / équipe
2B	H2				X	Bin 2 /Eq	Communication radio unidirectionnelle entre 2 cartes (simplex ou UNICAST)	Comment programmer une communication unidirectionnelle (simplex ou UNICAST) entre 2 balises ? Comment paramétrier correctement, dans le contexte d'étude, la balise émettrice (TX) et la balise réceptrice (RX) ? (Adapter au mieux la valeur de puissance d'émission et la distance de détection de l'émetteur)		2 Balises (carte µBIT alimentée en USB) / équipe
							<b>Restitution des travaux menés pour valider les programmes proposés</b>			
							<b>ETRE CAPABLE D'EVITER LES COLLISIONS ET DE RESPECTER L'ORDRE DE DEPART</b>			
3	H3		X		X	Eq1 & Eq2  Eq3 & Eq4	Respect de la distance de sécurité entre 2 VAE CODO	Comment respecter, en mode simplex ou UNICAST, la distance d'anticollision entre un CODO1 TX et un CODO2 RX ? (Hypothèse simplificatrice : Le VAE CODO1 reste toujours devant le VAE CODO2 Valider les paramètres de l'activité 2B)		1 CODO TX avec batterie pour équipe 1 (idem eq3) 1 CODO RX avec batterie pour équipe 2 (idem pour eq4)
4A	H4		X		X	Eq1  Eq3	Respect de priorité en mode simplex (UNICAST)	Comment respecter, en mode simplex ou UNICAST, l'ordre de circulation des VAE CODO malgré l'emprunt de la voie de dégagement par le VAE CODO1 ? (Hypothèse simplificatrice : Le VAE CODO1 reste toujours devant le VAE CODO2)		1 CODO TX et 1 CODO RX (avec batteries) pour équipe 1 (idem pour équipe 3)
4B	H4		X		X	Eq2  Eq4	Respect de priorité en half-duplex Puis en full-duplex (Bidirectionnel)	Comment rendre le VAE CODO 2 à la fois RX et TX (mode half-duplex) ? RX vis-à-vis du VAE CODO1 (qui reste TX) TX vis-à-vis du VAE CODO 3 (qui reste en RX) Comment affecter un mode bidirectionnel (full-duplex) TX et RX à chaque VAE CODO et les paramétrier de sorte que la détection mutuelle ne provoque pas un blocage en position arrêt ?		3 Balises (carte µBIT alimentée en USB) pour équipe 2 (idem pour eq4)
5	H4	X			X	Eq1 2,3,4	« Petit train »	Comment garantir l'ordre chronologique de sortie des 4 VAE CODO lors d'un trajet aller ?		1 CODO TX et RX avec batterie / équipe

Travail coopératif # Typologie d'activités plutôt complémentaires

Travail collaboratif # Typologie d'activités plutôt communes

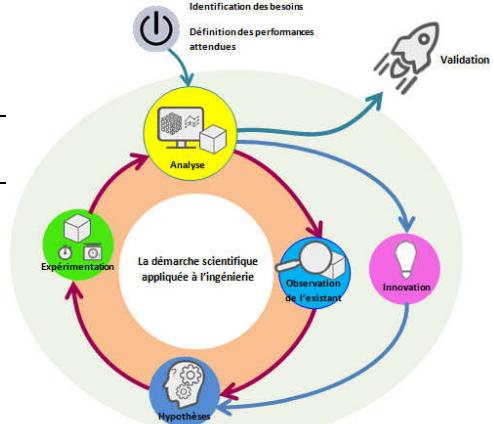
## Répartition des tâches du Mini projet – Service de proximité VAE CODO (suite)

Activité	Chronologie	Travail		Innover	Expérimenter	Répartition /éq. et /binôme	INVENTAIRE DES ACTIVITES ET PROGRESSION sur la base de 4 équipes de 4 élèves		Matériel pour 4 équipes  <b>4 CODOS</b> + 4 BORNES d'APPEL (avec carte µBit + Batterie) + 6 BALISES (carte µBIT alimentées en USB)
		Collaboratif	Coopératif				Intitulé de l'activité	Problème technique à résoudre	
<b>ETRE CAPABLE D'EMETTRE UN APPEL ET DE PRENDRE EN CHARGE LE CLIENT</b>									
6A	H5	X		X	X	Eq 1 à 4	Lancer un appel depuis une borne  Déetecter la présence du VAE	<b>Travaux préliminaires :</b>  Comment, à partir d'une simple impulsion sur un bouton poussoir, émettre un appel radio périodique depuis la borne tout en affichant sa prise en compte sur la matrice à LEDs ?  Quelle solution (aisément praticable) proposer pour provoquer l'arrêt du VAE CODO au plus près de la borne ?	1 BORNE D'APPEL avec batterie / équipe + 1 Balise (carte µBIT alimentée en USB) / équipe
6B	H5	X		X	X	Eq 1 à 4	Gestion de l'appel client <b>Identification de la borne d'appel</b> Prise en charge du client appelant	Comment prendre en compte l'appel en s'arrêtant devant la bonne borne avec le VAE CODO ?  Hypothèses simplificatrices Appropriation de la solution proposée Test et mise au point des programmes	1 BORNE D'APPEL avec batterie + 1 CODO avec batterie / équipe
6C	H6	X		X	X	Eq 1 à 4	Identification de la borne Terminus	Comment arrêter la course du VAE CODO une fois la borne terminus atteinte ?	1 BORNE D'APPEL avec batterie + 1 CODO avec batterie / équipe
7	H6	X				Eq 1 à 4	Gestion de la communication radio de l'ensemble du service de proximité VAE CODO  Peut-on généraliser la méthode proposée d'identification des acteurs radios ? Proposer d'autres solutions d'identification à partir du document annexe	XXXX	
8	<b>HORS TEMPS COLAIRE</b>				<b>Restitution des travaux par groupe de travail – Présentation type POWER POINT... Synthèse et validation des acquis</b>				

Travail coopératif # Typologie d'activités plutôt complémentaires

Travail collaboratif # Typologie d'activités plutôt communes

Cette répartition des tâches permet notamment l'analyse de situation, l'apport de solutions, la conception et mise en œuvre de programmes par parties et l'expérimentation des solutions proposées afin de valider progressivement l'essentiel des exigences retenues selon la démarche scientifique appliquée à l'ingénierie suivante :



## ETRE CAPABLE D'EMPRUNTER LA VOIE PRINCIPALE ET LES CONTRE-ALLEES DE FACON AUTONOME

### Activité 1 - Gestion du VAE CODO en suiveur de ligne



- 2- Mettre au point le programme de la carte micro :bit permettant le suivi de ligne d'un parcours bouclé à une vitesse de mobile favorisant la fluidité du déplacement.

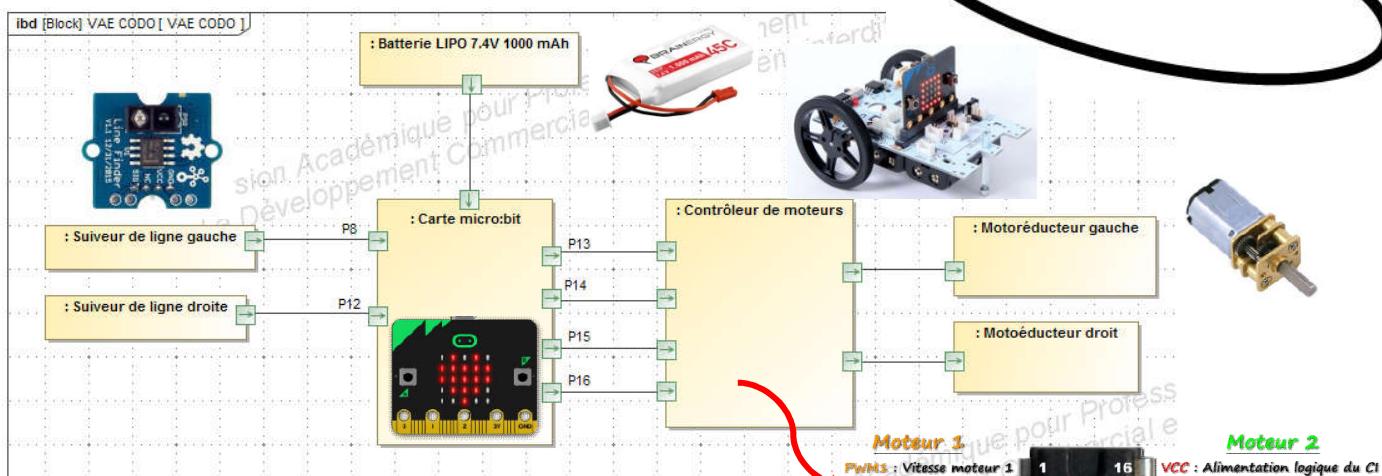
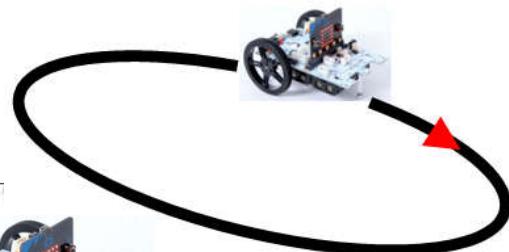
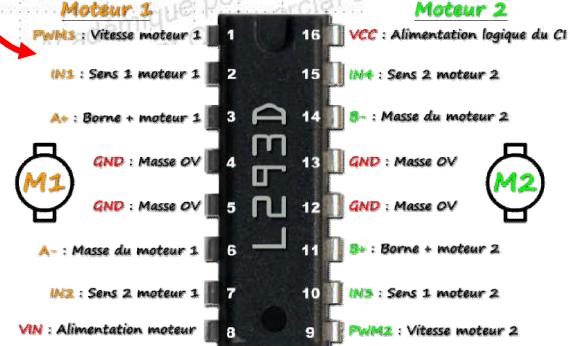


Table de vérité des actions de déplacements autorisées :

P8	P12	P13	P14	P15	P16	Action réalisée
0	0	0	0	0	0	Stopper les moteurs
0	1	1	0	0	1	Pivoter sur place à droite
1	0	0	1	1	0	Pivoter sur place à gauche
1	1	1	0	1	0	Avancer tout droit



### Exemple de programmation sous MAKECODE et son code PYTHON associé

VAE CODO en marche avant tant que la ligne est détectée.



NB : Ce programme fait appel au sous-programme « deplacementCODO »

```

def deplacementCODO():
    if CODO.detectline(DigitalPin.P8) == 1 and CODO.detectline(DigitalPin.P12) == 1:
        CODO.robot_move(CODO.RobotDirection.FORWARD, 50)
    else:
        CODO.motor_stop()

def on_forever():
    deplacementCODO()
    basic.forever(on_forever)
  
```

**Attention !!! Si vous utilisez MAKECODE, penser à télécharger l'extension « CODO »**



- 3- Préciser le nom et la valeur du paramètre de réglage retenu.

### Activité 2A - Gestion d'emprunt d'une voie de dégagement



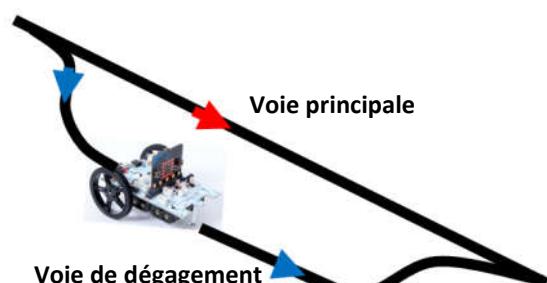
- 4- Proposer une solution constructive permettant au VAE CODO de prendre une voie de dégagement à chaque fois quelle se présente. Compléter le diagramme de blocs internes ci-dessus.



- 5- Equiper le VAE CODO de cette nouvelle solution constructive. Adapter le programme réalisé à la question 2 et tester.



- 6- Valider le bon fonctionnement.

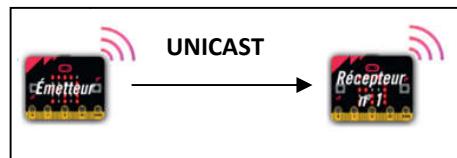


## ETRE CAPABLE D'EVITER LES COLLISIONS ET DE RESPECTER L'ORDRE DE DEPART

### Activité 2B – Communication radio entre 2 cartes micro :bit



- 7- Programmer une communication radio unidirectionnelle (simplex ou unicast) entre une carte émettrice (TX) et une carte réceptrice (RX) afin de détecter sur la carte RX la proximité de la carte TX.



**Voir la fiche d'information sur la communication radio de la carte micro:bit en annexe.**



- 8- Adapter la valeur des paramètres de puissance d'émission et de distance de détection au plus près des besoins du mini-projet. Compléter le tableau de mesure suivant et préciser le couple de valeurs à retenir.

Puissance de transmission (Variable Force de 0 à 9)	Variable Distance (de 0 à 9)	Estimation de la distance de détection en cm
1	3	
1	2	
1	1	
0	5	
0	4	
0	3	

### Activité 3 - Gestion du respect des distances de sécurité entre 2 VAE CODO



- 9- Adapter les programmes précédents sur les VAE CODO (CODO1 en TX et CODO2 en RX) afin de valider le couple de paramètres retenus. S'assurer d'un maximum de fluidité de parcours sans risque de collision mutuelle.



**Rappel de la règle de circulation :** Le CODO1 reste toujours devant le CODO2.



- 10- Préciser les noms et les valeurs des paramètres de réglages retenus.



- 11- Les exigences ayant trait à cette problématique sont-elles satisfaites ?



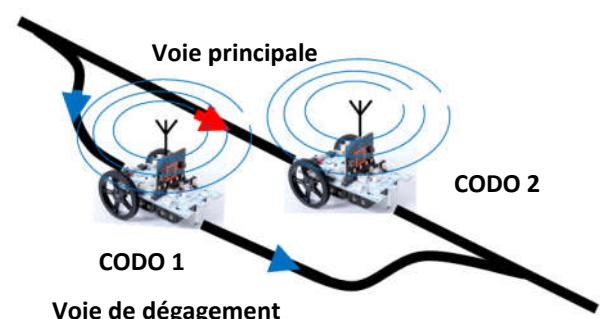
### Activité 4A - Gestion de la priorité en sortie de la voie de dégagement



- 12- Compléter le programme précédent afin de respecter l'exigence de respect de la priorité du VAE CODO 1 sur le VAE CODO 2.



- 13- Valider cette contrainte.



### Activité 4B – AMELIORATION de la Gestion de la priorité – Echanges Bidirectionnels (full-duplex)



- 14- Comment rendre le VAE CODO2 à la fois émetteur (RX) et récepteur TX ?  
RX vis-à-vis du VAE CODO1 (qui reste TX)  
TX vis-à-vis du VAE CODO 3 (qui reste en RX)



- 15- Tester et valider la proposition.

**3 cartes en E/R**



- 16- Comment affecter un échange bidirectionnel TX et RX à chaque VAE CODO et paramétrier de sorte que la détection mutuelle ne provoque pas un blocage en position arrêt ?



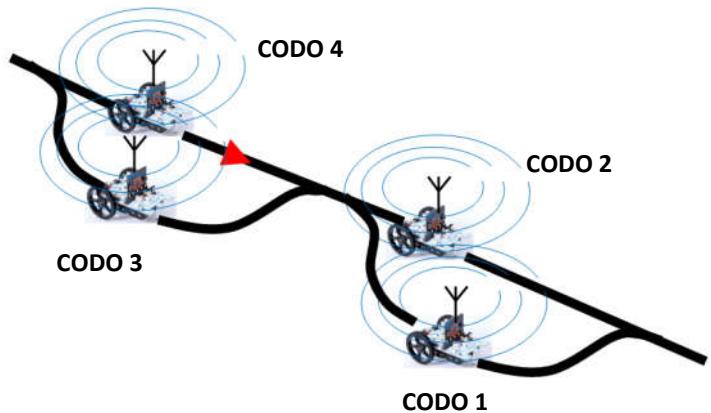
- 17- Tester et valider la proposition.

**Activité 5 – « Le petit train »**

18- Comment garantir l'ordre chronologique de sortie des 4 VAE CODO lors d'un trajet aller ?



19- Tester et valider la proposition.

**ETRE CAPABLE D'EMETTRE UN APPEL ET DE PRENDRE EN CHARGE LE CLIENT****Activité 6A - Gestion de l'appel client (pour une seule borne d'appel)**

Comment, à partir d'une simple impulsion sur un bouton poussoir, émettre un appel radio périodique depuis la borne tout en affichant sa prise en compte sur la matrice à LEDs ?

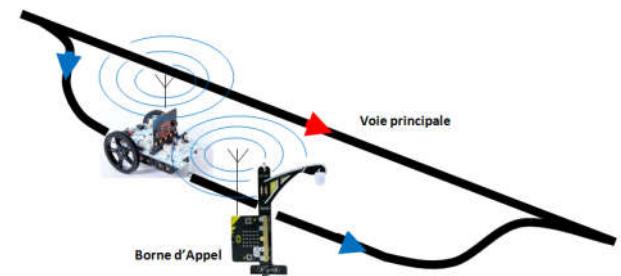
**Rappel :** Le programme est censé générer un appel radio périodique depuis la borne jusqu'à ce que le VAE CODO, parti soit de la zone d'entrepôts soit de la borne terminus au lancement de cet appel, pénètre sur la voie de dégagement de la borne concernée.

*NB : L'appel est périodique (et non unique) car il a pour but d'aider le VAE CODO à repérer la bonne borne d'appel durant son déplacement.*

**Travaux préliminaires**

20- Fort de ce rappel, préciser en quoi la proposition 1 pose problème.

21- Expliquer en quoi l'évolution de la proposition 1 est une solution permettant de résoudre ce problème.

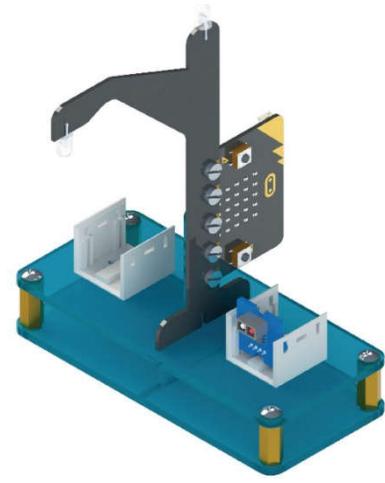
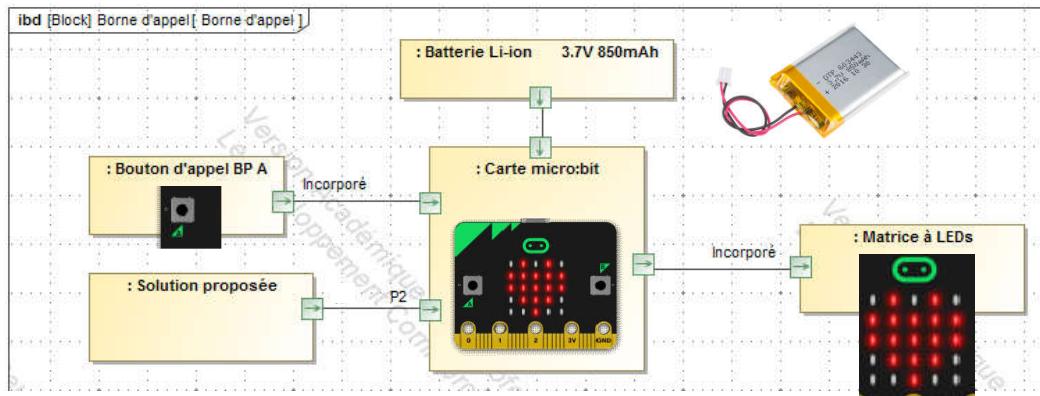


Proposition 1	Programme PYTHON généré à partir de MAKECODE
<pre> au démarrage radio définir groupe 1 radio définir puissance de transmission 2 lorsque le bouton A est pressé   envoyer la valeur "Borne1" = 1 par radio   montrer l'icône </pre>	<pre> def on_button_pressed_a():     radio.send_value("Borne1", 1)     basic.show_icon(IconNames.YES) input.on_button_pressed(Button.A, on_button_pressed_a)  radio.set_group(1) radio.set_transmit_power(2) </pre>
Evolution de la proposition 1	Programme PYTHON généré à partir de MAKECODE
<pre> au démarrage radio définir groupe 1 radio définir puissance de transmission 2 lorsque le bouton A est pressé   définir EtatMemoireA à 1 toujours   si EtatMemoireA = 1 alors     envoyer la valeur "Borne1" = 1 par radio     montrer l'icône </pre>	<pre> def on_button_pressed_a():     global EtatMemoireA     EtatMemoireA = 1 input.on_button_pressed(Button.A, on_button_pressed_a)  EtatMemoireA = 0 radio.set_group(1) radio.set_transmit_power(2)  def on_forever():     if EtatMemoireA == 1:         radio.send_value("Borne1", 1)         basic.show_icon(IconNames.YES) basic.forever(on_forever) </pre>

Quelle solution (aisément praticable) retenir pour provoquer l'arrêt du VAE CODO au plus près de la borne ?



- 22- Au regard de l'illustration ci-contre, faire une proposition de solution détection de présence du VAE CODO et compléter le diagramme de blocs internes de la borne d'appel.

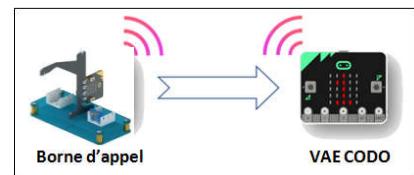


- 23- Réaliser les modifications matérielles et les branchements nécessaires sur la borne.  
Faire valider par le professeur avant la mise sous tension.



A l'aide de la borne modifiée et d'une autre carte micro:bit (en remplacement du VAE CODO), concevoir les deux programmes (émetteur (TX) et récepteur (RX)) qui permettent :

- D'émettre un signal radio périodique d'appel depuis une borne identifiable ;
- De recevoir cet appel depuis la carte micro:bit de substitution ;
- D'identifier la bonne borne d'appel à l'approche de celle-ci (la notion de gestion de distance est à nouveau une piste de solution) ;
- De s'arrêter à hauteur de la borne sollicitée grâce à détecteur de présence de la borne ;
- D'arrêter l'émission d'appel et l'affichage de prise en charge de la borne concernée une fois l'arrêt effectué à hauteur de la bonne borne ;



**CONSEIL : Il semble judicieux de s'aider de la matrice à LED de chaque carte et d'user de l'affichage de divers icônes aisément programmables pour élaborer et mettre au point ces 2 programmes.**



- 24- Tester la proposition et s'assurer de son bon fonctionnement.

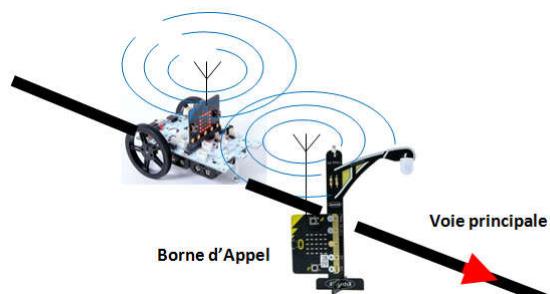
### ACTIVITE 6B - PRISE EN CHARGE DU CLIENT

Comment prendre en compte l'appel et s'arrêter devant la bonne borne avec le VAE CODO ?



Hypothèse simplificatrice : Pour cette partie, le programme du VAE CODO est volontairement réduit aux actions minimalistes de déplacement (AVANCER et STOPPER) en faisant abstraction de la voie de dégagement et des capteurs suiveurs de lignes afin de ne pas cumuler toutes les difficultés à la fois.

Conformément à l'hypothèse simplificatrice, on donne à la page suivant le programme de la borne émettrice et le programme minimaliste du VAE CODO permettant de prendre en charge le client de la borne 1



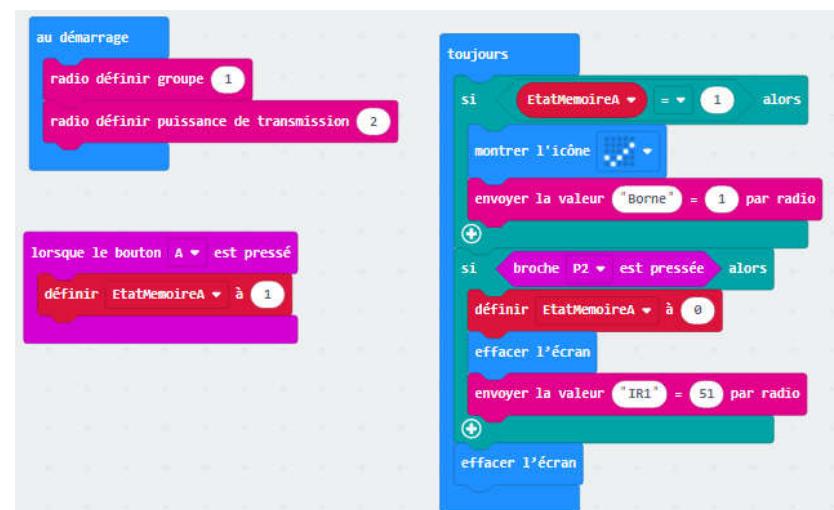
Appropriation des programmes proposés :

- 25- A partir de quand le VAE CODO se déplace ?
- 26- A quel moment le client est-il censé monter dans le véhicule ?  
Entourer directement votre réponse sur le programme.
- 27- A quel moment la matrice à leds de la BORNE n'affiche plus rien ?  
Quelle signification peut-on accorder à cette absence d'affichage ?
- 28- A quel moment la matrice à leds du VAE CODO n'affiche plus rien ?  
Quelle signification peut-on accorder à cette absence d'affichage ?

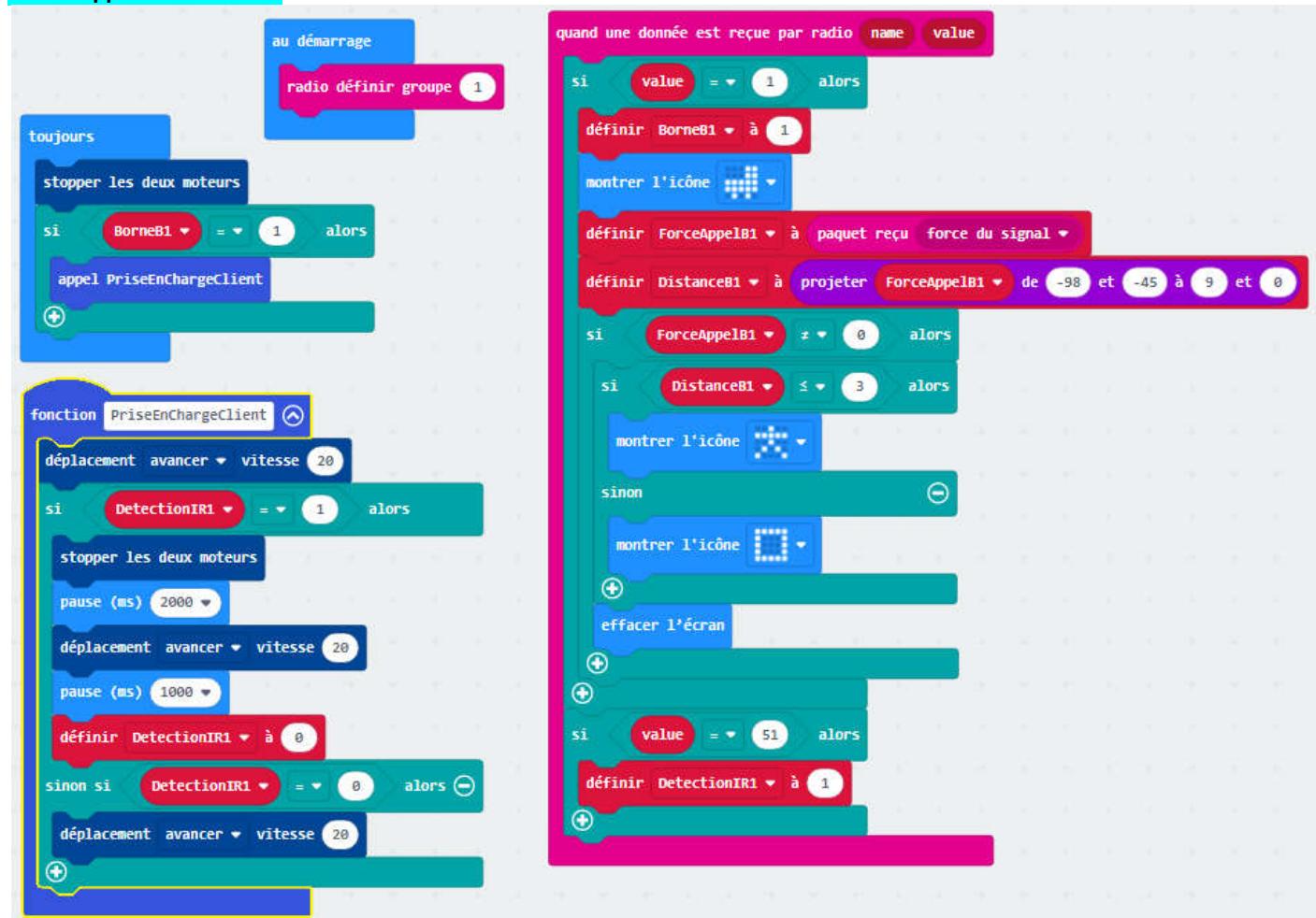


29- Télécharger les 2 programmes respectivement dans la borne et dans le VAE CODO.

[AppelBorne1-IR1-v1.hex](#)



[CODO-AppelB1-IR1-v1.hex](#)



30- Tester ces programmes (sur une courte distance et sans tracé de ligne) et confirmer ou infirmer les observations :

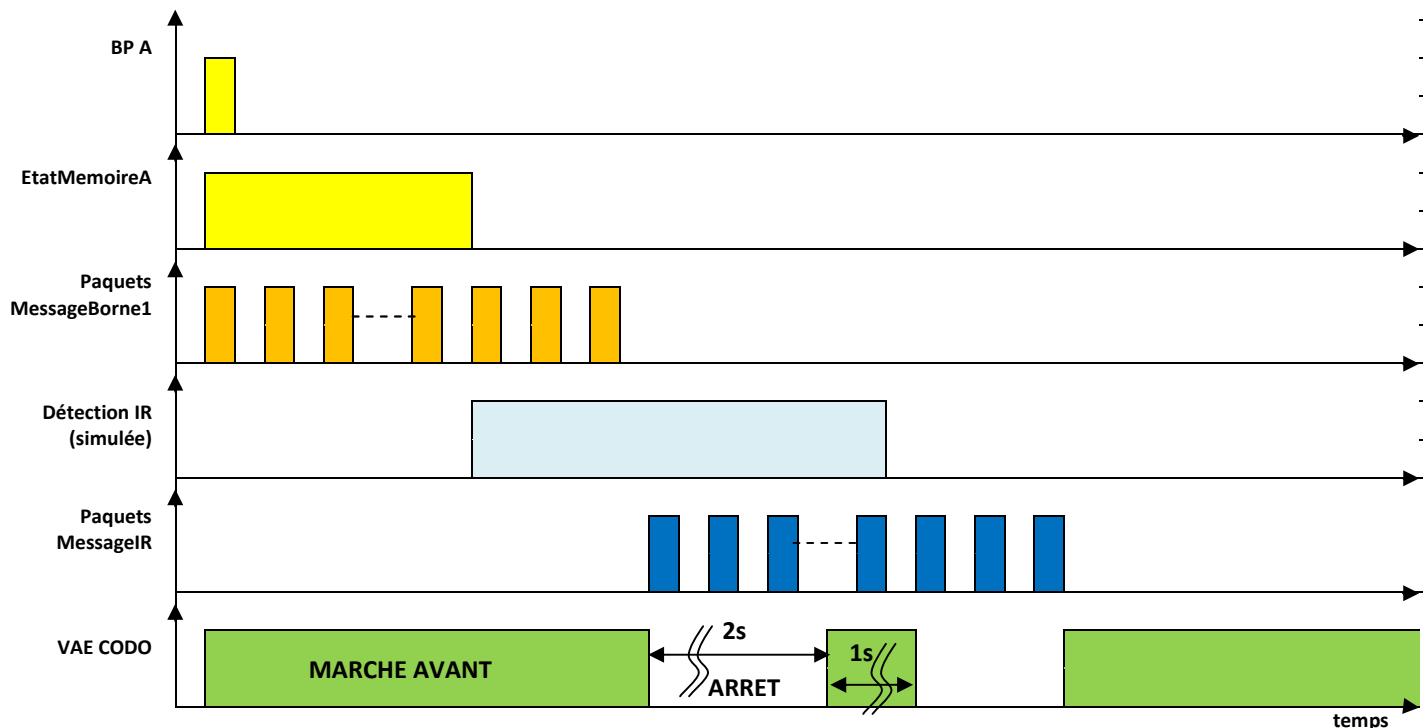
Le VAE CODO démarre dès lors que l'appel est formulé	VRAI	FAUX
Le VAE CODO détecte la proximité de la borne d'appel sollicitée	VRAI	FAUX
Le VAE CODO s'arrête à proximité du détecteur de présence	VRAI	FAUX
Le VAE CODO est correctement détecté	VRAI	FAUX
Le VAE CODO poursuit sa route et s'arrête bien après la borne si l'on simule la détection de présence à la main.	VRAI	FAUX
L'émission d'appel de la borne 1 s'interrompt une fois le VAE CODO détecté	VRAI	FAUX
L'arrêt du VAE CODO est temporisé	VRAI	FAUX
Le VAE CODO avance durant 1 s avant de poursuivre sa course afin de désactiver le détecteur de présence	VRAI	FAUX

**MISE AU POINT DES PROGRAMMES**

- 31- Sans rentrer dans la difficulté de gestion programmée de la détection sur front et d'interruption d'événement externe, proposer des solutions plus aisées qui permettent d'améliorer la détection de présence du VAE CODO.

Un des facteurs influents de la non détection du VAE CODO est lié au nombre de messages contenus dans la file d'attente (voir document annexe). Afin de mieux appréhender le problème voici les logigrammes qui font état des dysfonctionnements constatés lorsqu'on simule à la main la présence du VAE CODO.

- 32- Localiser ces problèmes en les entourant en rouge et reporter pour chaque anomalie constatée le chronogramme souhaité par dessus l'existant. (**NB : Ne pas tenir compte de l'échelle des temps**)



L'instruction `radio.config()` n'étant pas accessible, une solution intermédiaire est proposée pour résoudre le problème de file d'attente.

Elle présente également l'avantage de moins consommer d'énergie.

- 33- Déduire de l'extrait de programme de la borne émettrice suivant, la solution proposée.

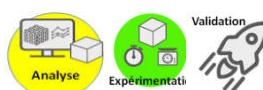


- 34- Apporter les modifications nécessaires aux deux programmes et valider le bon fonctionnement de la prise en charge du client

```
def on_forever():
    global EtatMemoireA
    if EtatMemoireA == 1:
        basic.show_icon(IconNames.YES)
        radio.send_value("Borne", 1)
        basic.pause(1000)
    if input.pin_is_pressed(TouchPin.P2):
        EtatMemoireA = 0
        basic.clear_screen()
        radio.send_value("IR1", 51)
        basic.clear_screen()
basic.forever(on_forever)
```

**Activité 6C - Identification de la borne TERMINUS.**

Comment arrêter la course du VAE CODO une fois la borne terminus atteinte ?



- 35- Compléter les programmes précédents afin de couvrir un trajet aller complet d'un VAE CODO depuis la zone de départ (entrepôt ou station de RTC) jusqu'à la borne Terminus.

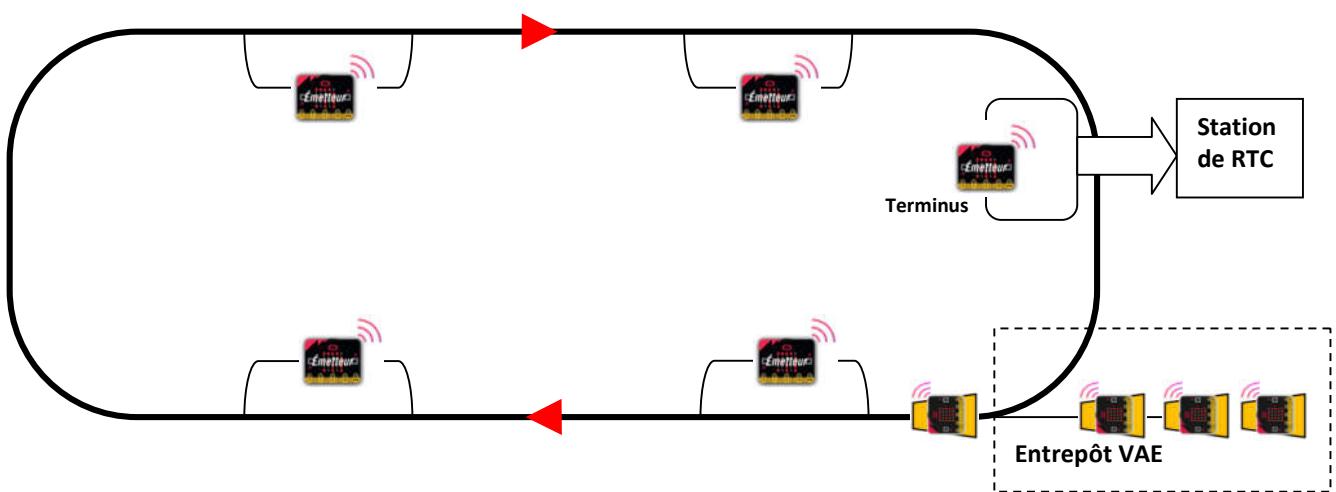
- 36- Faire valider le bon fonctionnement par le professeur.

**Activité 7 - Gestion de la communication radio de l'ensemble du service de proximité VAE CODO****Peut-on généraliser la méthode proposée d'identification des acteurs radios ?**

37- La méthode d'identification d'une borne et d'un détecteur IR proposée dans le programme de prise en charge du client permet-elle de déduire que le programme complet pourrait également identifier les autres bornes et distinguer les différentes détections IR ?

Si la réponse est NON, expliquer pourquoi ?

- 38- Si l'on souhaitait réaliser le programme complet de gestion de l'ensemble du service de proximité et conformément à la méthode d'identification utilisée dans les programmes jusqu'alors, proposer un exemple d'affectation des identifiants de tous les acteurs (bornes, détecteurs IR, VAE CODOs) faisant l'objet d'une émission de radio fréquence.  
Pour cela, reporter cette proposition sur l'illustration du circuit du trajet aller ci-dessous.



- 39- A l'aide du document annexe, proposer sommairement une ou plusieurs autres solutions pour identifier ces différents acteurs radios.

**Validation des exigences retenues et CONCLUSION GENERALE**

- 40- Compléter le diagramme des exigences en précisant les solutions constructives qui permettent de satisfaire chacune d'elles.
- 41- Parmi les exigences retenues, quelles sont celles qui n'ont pas été résolues ?  
Proposer des pistes de solutions à hauteur de chacune.
- 42- Quelle infrastructure et quels moyens matériels faudrait-il prévoir pour assurer le trajet retour (Terminus → Domicile) ?
- 43- Selon un point de vue communication radio, quelles modifications de programme seraient nécessaires pour éviter tout type d'interférence avec les acteurs du trajet aller?

**Pour le service de proximité VAE en situation réelle (à échelle 1) :**

- 44- La solution radio proposée dans ce mini-projet vous semble-t-elle viable ? Proposer le cas échéant une solution plus adaptée.

**ANNEXE - Fiche d'information sur la communication radio avec la carte micro:bit**

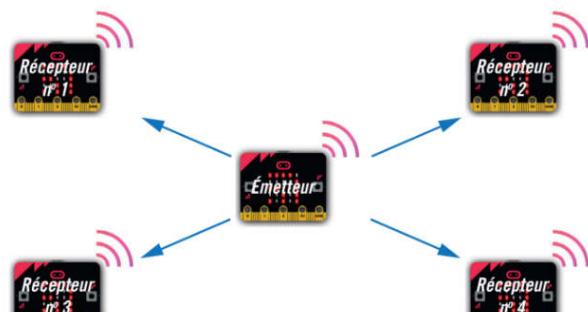
L'unité centrale de traitement (CPU) sur le micro: bit est un Nordic Semiconductor [nRF51822](#). En plus d'être un processeur informatique à usage général, cette puce contient un module émetteur radio 2,4 GHz et un module récepteur radio 2,4 GHz qui utilisent le protocole propriétaire Gazell de Nordic pour pouvoir échanger les données entre cartes micro:bit.

Lorsqu'une donnée est envoyée par un module émetteur, elle est diffusée à l'ensemble des cartes réceptrices qui sont dans le champ de portée de réception du signal radio.

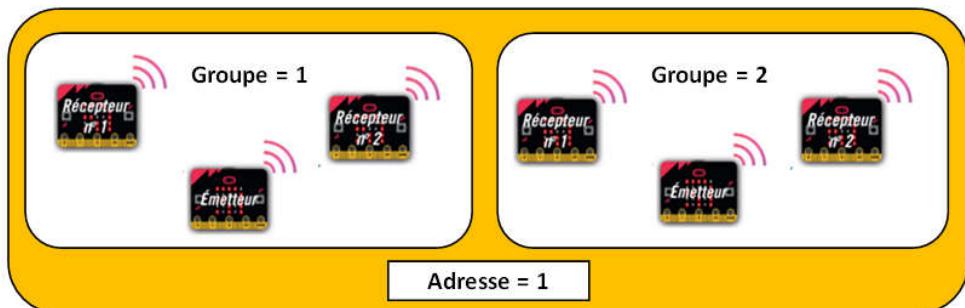
L'envoi de données n'étant pas ciblé vers un destinataire, toutes les cartes décoden la même information en réception tel une échange *broadcast*. Pour pouvoir cibler le destinataire, il est nécessaire d'identifier ce dernier sur le réseau. Le système d'identification fonctionne comme suit : l'émetteur doit spécifier une adresse et un groupe d'appartenance.

Une carte micro:bit est donc identifiée par son adresse et son groupe.

*L'exemple de la figure ci-contre propose 2 groupes à la même adresse. Les cartes du groupe 1 peuvent communiquer entre elles (idem pour les cartes du groupe 2) cependant les cartes du groupe 1 ne peuvent pas communiquer avec celle du groupe 2 même si elles ont la même adresse.*



Réseau de cartes micro:bit en mode radio



**Le paramètre adresse** (par défaut = 0x75626974) est un nom exprimé sous forme d'adresse 32 bits. Il est utilisé pour filtrer les messages entrants au niveau matériel.

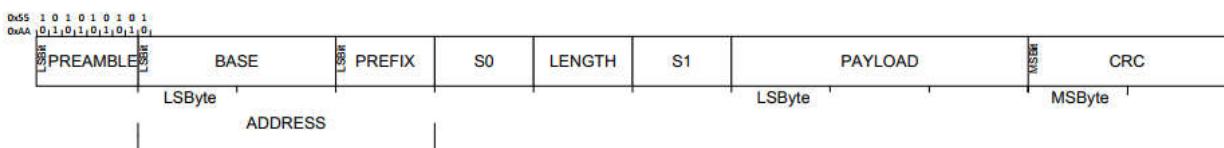
**Le paramètre groupe** (par défaut = 0) est un entier compris entre 0 et 255. Comme l'**adresse**, il est utilisé lors du filtrage des messages.

**Autres paramètres radio configurables** → Plusieurs paramètres peuvent être réglés pour ce type de communication à distance :

- **Le réglage de la puissance d'émission** (de 0 à 7). La valeur 0 correspond à une faible portée et une faible consommation tandis que la valeur 7 correspond à une portée maximale de plusieurs dizaines de mètres. Ces valeurs correspondent à une puissance émise en dBm avec respectivement -30, -20, -16, -12, -8, -4, 0 et 4 dBm. *Pour 0 dBm, la portée est d'environ 20 mètres.*
- **Le réglage du débit binaire** est également possible avec les commandes radio.RATE\_250KBIT, radio.RATE\_1MBIT et radio.RATE\_2MBIT qui permettent d'avoir respectivement des vitesses de transmission de 256Kbit/s, 1Mbit/s et 2 Mbit/s.
- **Le réglage du canal** avec la commande radio.config(chanel=7) par défaut. Un canal compris entre les valeurs 0 et 83 correspond en fait à une fréquence donnée autour de 2.4 GHz par pas de 1 MHz. Un changement de canal permet de créer un autre réseau au même titre qu'une adresse et un groupe.
- **La longueur du message** (Length = nombre de caractères du message) est définie à 32 par défaut. Mais on peut modifier ce nombre à l'aide de la commande radio.config(length=57) la valeur maximum pouvant alors atteindre 251.
- **Le nombre de messages dans la file d'attente** : radio.config(queue=3). Au delà de 3 messages en attente, ils seront supprimés.

**Configuration des paquets radios**

La radio envoie les différents champs du paquet dans l'ordre suivant : PREAMBLE, ADDRESSE, S0, LENGTH, S1, PAYLOAD et CRC comme illustré ci-dessous de gauche à droite.



**Le champ PREAMBLE** est toujours long d'un octet. Son LSB est envoyé en premier à l'antenne. Si le premier bit de l'ADRESSE est 0 alors le préambule sera mis à 0xAA sinon le PRÉAMBULE sera mis à 0x55. **Le champ ADRESSE** de l'onde radio est composé de 2 parties : la base (réglable de 0 à 7) et le préfixe. Seules les données en provenance d'une carte présentant la même adresse seront pris en compte.

**Les champs S0, LENGTH, S1, PAYLOAD** concernent les données applicatives. La longueur combinée de ses données ne peut pas dépasser 254 octets (3 octets pour S0, LENGTH, S1 et 251 octets pour le message).

**Le champ CRC** de 16 bits (Contrôle de Redondance Cyclique) qui débute par l'octet le plus significatif permet de détecter d'éventuelles erreurs de transmission.

Les messages reçus sont stockés dans une file d'attente de la mémoire vive (RAM) de la carte micro:bit type FIFO. Par défaut, le nombre maximum de messages est de 3 mais il est possible d'augmenter la taille de la file d'attente. Notons que plus elle est grande et plus la mémoire utilisée est importante. La lecture d'un message efface ce dernier de la file d'attente, mais la mémoire est pleine, les nouveaux messages sont ignorés. Les messages sont classés par adresse et par groupe.

### Programmation de la communication radio

L'outil **Radio** sur le logiciel MakeCode (par exemple) permet la communication entre plusieurs cartes micro:bit. Il est nécessaire de créer deux programmes, un pour chaque carte micro:bit. Elles doivent être programmées selon le même « groupe radio » pour que les programmes fonctionnent.

Voici l'essentiel des instructions et fonctions utilisées pour la communication radio :

Instruction ou fonction	Bloc MAKECODE	Code PYTHON
Définir l'identifiant du groupe (0 à 255)	radio définir groupe 1	def radio.set_group(id: int32): None
Modifier le niveau de puissance en sortie d'émetteur (0 à 7)	radio définir puissance de transmission 7	def radio.set_transmit_power(power: int32): None
Retourner les propriétés du dernier paquet radio reçu	paquet reçu force du signal	def radio.received_packet(type: number): number
Envoyer une information à l'aide de paquets radio (un nombre ou un nom associé à un nombre ou une chaîne de caractères)	envoyer le nombre 0 par radio envoyer la valeur "name" = 0 par radio envoyer la chaîne "" par radio	def radio.send_number(value: number): None def radio.send_value(name: str, value: number): None def radio.send_string(value: str): None
Réceptionner une information radio (un nombre ou un nom associé à un nombre ou une chaîne de caractères)	quand une donnée est reçue par radio receivedNumber quand une donnée est reçue par radio name value quand une donnée est reçue par radio receivedString	def radio.on_received_number(cb: (number) -> None): None def radio.on_received_value(cb: (str, number) -> None): None def radio.on_received_string(cb: (str) -> None): None
Changer la bande de transmission d'émission et de réception au canal donné (0 à 83)	radio régler la bande de fréquence 0	def radio.set_frequency_band(band: int32): None
Envoyer un événement par radio aux périphériques environnants	radio déclencher l'événement de la source MICROBIT_ID_BUTTON_A avec la valeur MICROBIT_EVT_ANY	def radio.raise_event(src: int32, value: int32): None

### Exemples de programme radio (sous MAKECODE et le code PYTHON associé)

Communication unidirectionnelle (simplex ou UNICAST) entre 2 cartes micro:bit avec affichage du nombre reçu sur le récepteur une fois la détection à une certaine distance de l'émetteur effectuée		
Côté émetteur (TX) :		.set_group(1) .set_transmit_power(7)  .n_forever(): lio.send_number(1) .forever(on_forever)
Côté récepteur (RX) :		def on_received_number(receivedNumber): global Force, Distance Force = radio.received_packet(RadioPacketProperty.SIGNAL_STRENGTH) Distance = Math.map(Force, -98, -45, 9, 0) if Distance <= 4: basic.show_number(receivedNumber) else: basic.clear_screen() radio.on_received_number(on_received_number)  Distance = 0 Force = 0 radio.set_group(1)

NB : Projeter... → ré-assignation d'un nombre au sein d'une plage (entre -98 et -45) à une nouvelle plage (entre 9 et 0)

La force du signal peut se situer entre -98 (plus faible valeur) et -45 (plus forte valeur).

**Communication bidirectionnelle (full-duplex) entre 2 cartes micro:bit avec affichage du nombre reçu sur l'émetteur et le récepteur une fois la détection mutuelle à une certaine distance effectuée**

**Côté émetteur (TX) :**



```

def on_received_number(receivedNumber):
    global Force, Distance
    Force = radio.received_packet(RadioPacketProperty.SIGNAL_STRENGTH)
    Distance = Math.map(Force, -98, -45, 9, 0)
    if Distance <= 4:
        basic.show_number(receivedNumber)
    else:
        basic.clear_screen()
    radio.on_received_number(on_received_number)

Distance = 0
Force = 0
radio.set_group(1)
radio.set_transmit_power(7)

def on_forever():
    radio.send_number(1)
    basic.forever(on_forever)

```

**Côté récepteur (RX) :**



```

def on_received_number(receivedNumber):
    global Force, Distance
    Force = radio.received_packet(RadioPacketProperty.SIGNAL_STRENGTH)
    Distance = Math.map(Force, -98, -45, 9, 0)
    if Distance <= 4:
        basic.show_number(receivedNumber)
    else:
        basic.clear_screen()
    radio.on_received_number(on_received_number)

Distance = 0
Force = 0
radio.set_group(1)
radio.set_transmit_power(7)

def on_forever():
    radio.send_number(2)
    basic.forever(on_forever)

```