

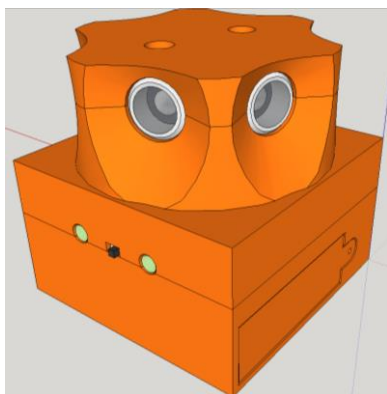
Balise à ultrasons : notice d'utilisation

Cette notice décrit les étapes à suivre pour installer le dispositif de positionnement, et effectuer des mesures. Ce document fait référence à la version « en cours de développement » de la balise à ultrasons, qui ne correspondra probablement pas à la version finale.

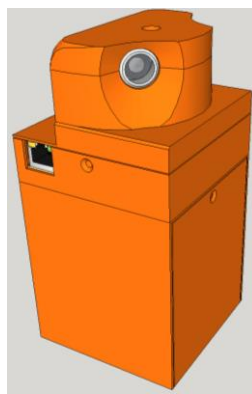
1. Positionnement des balises

Il y a au total 4 balises différentes à positionner. 3 balises réceptrices, fixes, placées sur les supports du bord de table. Une balise émettrice, mobile, placée sur l'objet à localiser, typiquement un robot.

Balise émettrice

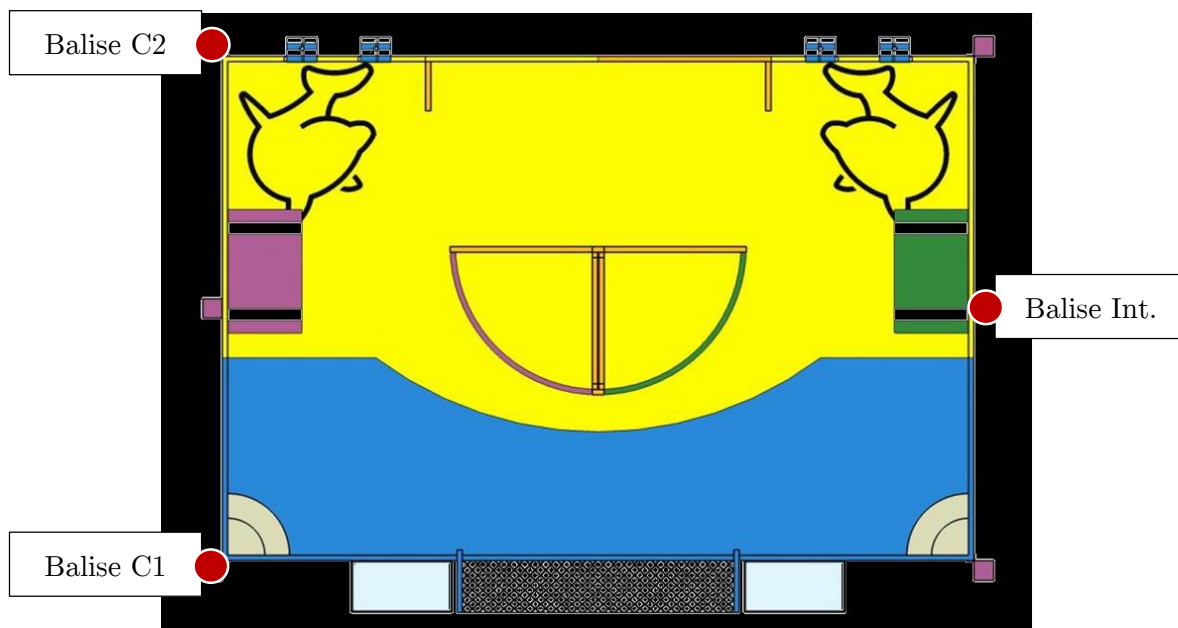


Balise réceptrice



La balise émettrice doit être placée sur l'objet à localiser, plus ou moins à la même hauteur que les balises réceptrices.

Les trois balises réceptrices sont différentes. On les différencie par leur partie supérieure qui intègre un ou plusieurs transducteur(s) à ultrason(s). Voici le positionnement :



Balise Int. : il s'agit de la balise possédant deux transducteurs à ultrasons, ainsi qu'un socle contenant le circuit d'alimentation et le microcontrôleur Teensy.

Balises C1 et C2 : il s'agit des balises n'ayant qu'un seul transducteur à ultrason chacune. Les deux sont identiques à une symétrie près.

2. Réglages

Chaque balise réceptrice possède un interrupteur à glissière à trois positions. Chaque position correspondant à une balise, il y a donc : Int. ; Ext1 ; Ext2

La balise principale, Int., embarquant l'alimentation et le microcontrôleur, doit être nécessairement réglée sur « Int. ».

Les deux autres balises peuvent être indifféremment positionnées sur Ext1 ou Ext2, mais ne peuvent pas avoir toutes les deux le même réglage.

Jusque-là, le canal Ext1 a été utilisé par la balise C1 positionnée en (-1500 ; 0), tandis que le canal Ext2 était utilisé par la balise C2 positionnée en (-1500 ; 2000). Positions données en mm, dans le repère standard d'INTech. Les programmes écrits jusqu'ici ne permettent pas forcément de choisir la disposition des balises, il est donc préférable d'utiliser ce positionnement.

3. Câblage

Balise émettrice

La balise émettrice est bien entendu « sans fil ». Elle est alimentée à l'aide d'une pile 9V se trouvant derrière le clapet. Il convient de vérifier qu'elle est correctement chargée avant d'effectuer des tests. Cette pile peut être remplacée par n'importe quelle source d'alimentation fournissant une tension entre 6,5 et 36V et délivrant au moins 2,5W.

La balise s'allume en actionnant le petit interrupteur en façade, une DEL verte fait office de témoin d'alimentation (mais n'atteste pas de la qualité de l'alimentation en question).

Balises réceptrices

Les balises réceptrices doivent être connectées entre elles à l'aide de câbles Ethernet droits. Si vous utilisez des câbles croisés, ça ne fonctionnera tout simplement pas (rien ne sera allumé) mais cela ne devrait pas tout détruire (en principe du moins, car cela n'a jamais été testé). Les trois balises peuvent être reliées dans n'importe quel ordre, vous pouvez même faire un anneau mais cette pratique sera qualifiée du terme technique « overkill design ».

Remarque importante concernant les câbles à utiliser : outre le fait qu'il est nécessaire d'utiliser des câbles droits, il convient de sélectionner des câbles Ethernet possédant un anti-retour intact sur chaque embout. Ce n'est pas futile, loin de là.

L'alimentation des balises réceptrices se fait au niveau de la balise principale (Int.), à l'aide de deux fils de type Kebab (IEEE 302.42). Il y a deux fils pour la masse (ils doivent être reliés, sinon ça ne marche pas dans tous les cas d'utilisation), et un fil rouge. Il faut fournir une tension entre 6,5 et 36V, avec une puissance d'environ 6W.

L'interrupteur d'allumage se trouve sur la balise principale (Int.). Chaque balise possède un témoin d'alimentation sur chacune de ses prises Ethernet. Les autres DEL présentes sur le socle de la balise principale témoignent de l'allumage de la Teensy ainsi que de son état (pour l'état de la Teensy, cela reste à implémenter).

4. Récupération des données

La récupération des données dépend fortement du code qui a été télé-versé sur le microcontrôleur, ce point est détaillé par la suite.

Connectez le microcontrôleur Teensy à un ordinateur à l'aide d'un câble micro-USB. Le microcontrôleur sera vu comme un port série. Vous pouvez régler le débit binaire de la liaison à 115200baud si vous aimez faire les choses bien (mais en pratique, la Teensy est signe-proof, elle règle toute seule le bon débit, probablement à l'aide de rites sataniques). Il vous suffit alors de lire ce que vous recevez sur le port série, le format des données dépendant bien entendu de ce que vous avez programmé sur le microcontrôleur.

5. Programmation, télé-versement et format des données

Le code du microcontrôleur peut être trouvé ici :

`balise-t3/software/BaliseT3/BaliseT3/BaliseT3.ino`

Pour programmer le microcontrôleur, il vous suffit de télécharger l'IDE Arduino (<https://www.arduino.cc/en/Main/Software>), ainsi que l'extension pour Teensy : Teensyduino (https://www.pjrc.com/teensy/td_download.html).

Vous pouvez alors éditer et télé-verser le code via l'IDE Arduino (réglez correctement le port série et le type de microcontrôleur, Teensy 3.2 en l'occurrence). Si vous souhaitez utiliser un véritable IDE, et pas cet espèce de sous-éditeur de texte moins performant que Word 2000, vous pouvez au choix, prendre un IDE proposant un plugin Arduino (Visual Micro pour Visual Studio, si Windows est compatible avec votre système neuronal), ou bien reconstruire une chaîne de compilation à la main...

Dans le code actuellement présent, vous pouvez faire référence aux trois balises à l'aide des constantes préprocesseur suivantes :

CANAL_1	0	<i>balise Ext1</i>
CANAL_2	1	<i>balise Ext2</i>
INT	2	<i>balise Int.</i>

Ces constantes doivent être utilisées comme indices dans un tableau, typiquement un tableau contenant les timbres temporels (timestamp).

Le code actuel envoie ses données dans le format suivant :

```
[Indice balise] ; timestamp
```

Il n'y a donc aucune association entre les timestamps, il faut faire cette opération plus tard avant de pouvoir calculer des positions.

Si vous êtes habiles, et savez reconnaître les « hack », vous devriez pouvoir lui faire renvoyer le format de données suivant :

```
[timestamp CANAL_1] ; [timestamp CANAL_2] ; [timestamp INT]
```

Voici le détail de la procédure à suivre, pour ceux qui ont moins d'instinct :

Ligne 129 : supprimer « while(true) ; //DEBUG »

Lignes 301, 344 et 386 : supprimer « Serial.printf("%u", micros()); »

Lignes 302, 324, 345, 366, 387 et 407 : supprimer « return; »

Après ces modifications, le microcontrôleur renverra des triplets de timbres temporels correctement associés. Si vous souhaitez modifier le format des données envoyées, rendez-vous ligne 156 et éditez le contenu du « Serial.printf() »

Remarquez que si vous avez supprimé les lignes comme indiqué (que vous ne les avez pas juste commentées), tous les numéros de ligne deviennent faux. Vous l'avez dans l'os.

Vous êtes libres de modifier le code comme bon vous semble, à condition que vous le compreniez.

En joie !

Ce document a été rédigé par Sylvain G. (Loli Senpai) le 27 juillet 2016, entre 9h12 et 14h53 (UTC +9:00), sous l'emprise de substances illicites (Lemon Tea KIRIN), tout en faisant semblant de travailler pour le compte de Thomas K. (Ronald Ier), chercheur à ATR. Il contient donc une quantité de troll non négligeable qu'il conviendra de distiller avant lecture. Oui, les recommandations de lecture se trouvent à la fin du document, ce n'est pas un insecte, c'est une fonctionnalité.

Pour toutes réclamations, merci de contacter le service client TechTheTroll par courrier postal :

シルヴァン・ゴーティエ

E25-304 Nakatomi-Daisan-Danchi

1-4162-1 Nakatomigaoka, Nara-shi, Nara

〒631-0003