

## T3 Beacon – Rev 4.2 – Programming application note

Ce document récapitule les informations pertinentes concernant la programmation du microcontrôleur Teensy 3.2 fourni avec la balise de localisation ultrasonique S&G Cie.

Toutes ces informations sont issues des schémas de la balise ainsi que du site officiel Teensy : <https://www.pjrc.com/teensy/>

Et à partir de ce point. J'arrête d'écrire en Arial.

## Assignation des pins

Pins permettant la lecture des signaux provenant des différentes balises réceptrices :

Canal interne – 80Hz :	15
Canal interne – 40 Hz :	16
Canal n°1 – 80Hz :	17
Canal n°1 – 40Hz :	18
Canal n°2 – 80Hz :	19
Canal n°2 – 40Hz :	20

Le canal est choisi sur chaque balise via l'interrupteur à 3 positions. La balise directement connectée au module d'alimentation et de communication doit impérativement utiliser le canal « interne ».

Pins de communication avec la xBee :

Il s'agit d'une communication série. Le port série n°3 de la Teensy est utilisé.

RX : 7

TX : 8

Pins des DELs de statut :

DEL power ON :	3
DEL canal n°1 :	4
DEL canal n°2 :	5
DEL canal intérieur :	6

## Paramétrage des pins

Entrées / Sorties :

Les pins 3 à 6 (contrôlant l'allumage des DELs de statut) doivent être paramétrées en **OUTPUT**, tandis que toutes les autres doivent être laissées en **INPUT**.

```
void pinMode(uint8_t pinNb, uint8_t in_out);
```

Communication série :

Le port série n°3 est utilisé pour la communication avec la xBee. L'utilisateur peut utiliser le port n°0 pour les logs et le debug. Ce dernier n'est pas disponible sur les pins de la Teensy, il l'est uniquement

à travers le port USB si ce dernier est configuré en port série (option à vérifier au moment du flashage).

```
void Serial3.begin(int baudrate); // Ouverture du port n°3  
void Serial.begin(int baudrate); // Ouverture du port n°0 (USB)
```

#### Interruptions :

Les pins 15 à 20 doivent être associées à des interruptions sur changement d'état (sur fronts montant et descendants). Le code permettant d'associer une entrée numérique à une fonction à travers une interruption est le suivant :

```
void attachInterrupt(uint8_t pin, void (*function)(), int mode);
```

pin : l'entrée numérique

function : la fonction appelée par l'interruption

mode : RISING, FALLING ou CHANGE (front montant, descendant ou bien les deux)

Pour mieux comprendre comment les interruptions doivent être utilisées :

<https://www.pjrc.com/teensy/interrupts.html>

## Programmer une Teensy 3.2

Pour pouvoir commencer à programmer il vous faudra télécharger et installer :

Le logiciel Arduino : <https://www.arduino.cc/en/Main/Software>

Le logiciel Teensyduino : [https://www.pjrc.com/teensy/td\\_download.html](https://www.pjrc.com/teensy/td_download.html)

A ce stade, vous pouvez commencer à coder et à flasher votre Teensy, mais ça se passe dans le logiciel Arduino, donc pas vraiment un IDE. Il est possible de paramétrer d'autres IDEs afin de coder et flasher des Arduinos (et donc aussi des Teensy maintenant que vous avez ajouté Teensyduino). Pour Visual Studio il suffit d'installer l'extension Visual Micro :

<https://visualstudiogallery.msdn.microsoft.com/069a905d-387d-4415-bc37-665a5ac9caba>

Pour les autres, Google est votre ami.

## Utiliser le module xBee

Pour communiquer avec le module xBee, vous pouvez vous contenter de parler et d'écouter sur le port série associé... Mais comme vous avez probablement autre chose à faire que de lire la documentation complète du module xBee, vous pouvez utiliser la librairie xBee pour Teensy :

[https://www.pjrc.com/teensy/td\\_libs\\_XBee.html](https://www.pjrc.com/teensy/td_libs_XBee.html)

La configuration préalable du module xBee est expliquée. Il n'y a pas beaucoup d'exemples de code, mais ça doit pas être sorcier !