

**Défi Technique 13 au 17 Décembre 2021**

**Utilisation IrriTrace**





Contact : [jsisto@sudexpe.net](mailto:jsisto@sudexpe.net)

Sommaire

[Nomenclature 3](#_Toc88837273)

[Définition 3](#_Toc88837274)

[a) Vitesse de transmission en bauds : 3](#_Toc88837275)

[1. Présentation IrriTrace 4](#_Toc88837276)

[2. Prototype IrriTrace 5](#_Toc88837277)

[a) Carte principale : 5](#_Toc88837278)

[b) Carte extension : 5](#_Toc88837279)

[c) Schéma de principe : 6](#_Toc88837280)

[d) Fonctionnement : 6](#_Toc88837281)

[3. Configuration de l’application Serial Bluetooth Terminal (SBT) 7](#_Toc88837282)

[4. Mise en fonctionnement d’IrriTrace 8](#_Toc88837283)

[5. Activation/Appairage du Bluetooth 8](#_Toc88837284)

[a) IrriTrace 8](#_Toc88837285)

[b) Application SBT 8](#_Toc88837286)

[6. Communication Bluetooth 10](#_Toc88837287)

[a) Récupération des données 10](#_Toc88837288)

[b) Renommage du Module Bluetooth 11](#_Toc88837289)

## Nomenclature

RTC = Real Time Clock

Bluetooth = BLE

L’Application Serial Bluetooth Terminal = SBT

## Définition

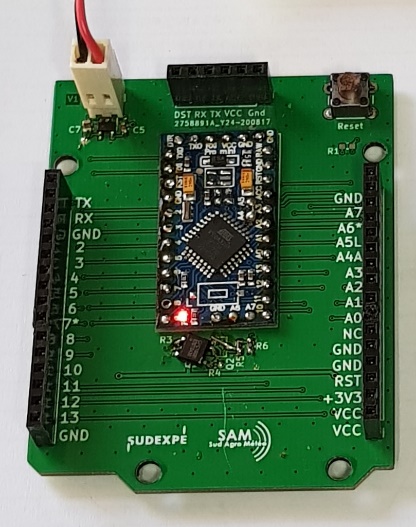
### Vitesse de transmission en bauds :

Afin de faciliter l'interopérabilité entre périphériques (PC, microcontrôleur, modem…) des vitesses de transmission sont normalisées par multiples et sous-multiples de 9600 bauds, l'unité baud correspondant à un symbole par seconde.

Pour IrriTrace la vitesse de transmission du Bluetooth est de 9600bauds

## Présentation IrriTrace

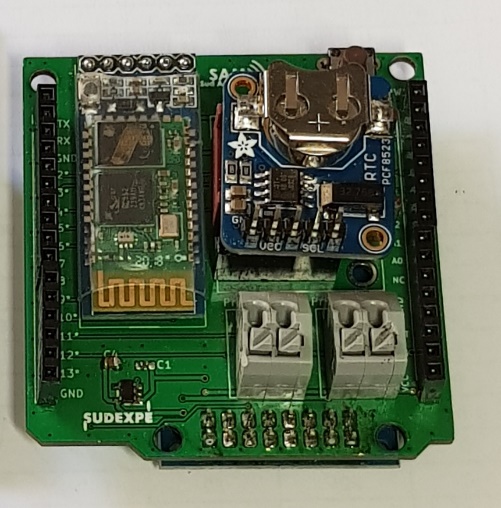
## Prototype IrriTrace



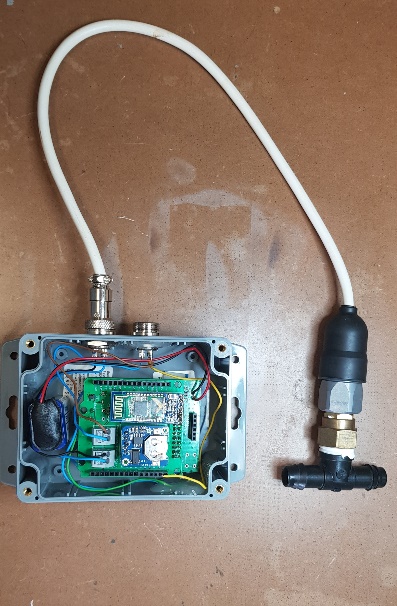
### Carte principale :

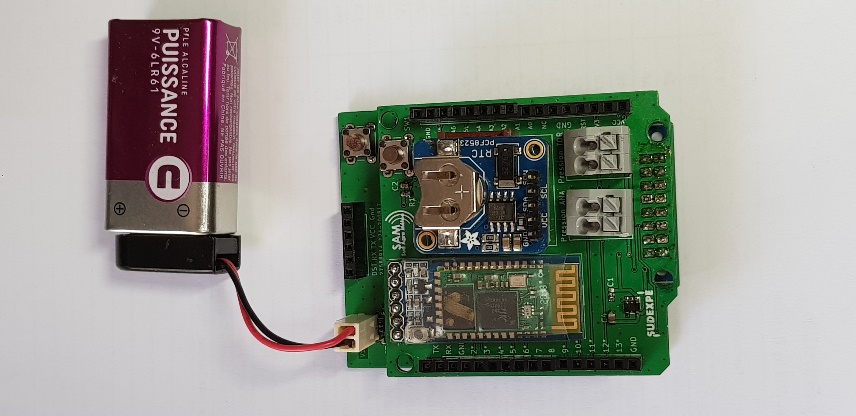
* Arduino Pro mini 3.3V 8Mhz
* Alimentation Pile 9V
* Mesure de la tension de la pile
* Connecteur pour la programmation

### Carte extension :

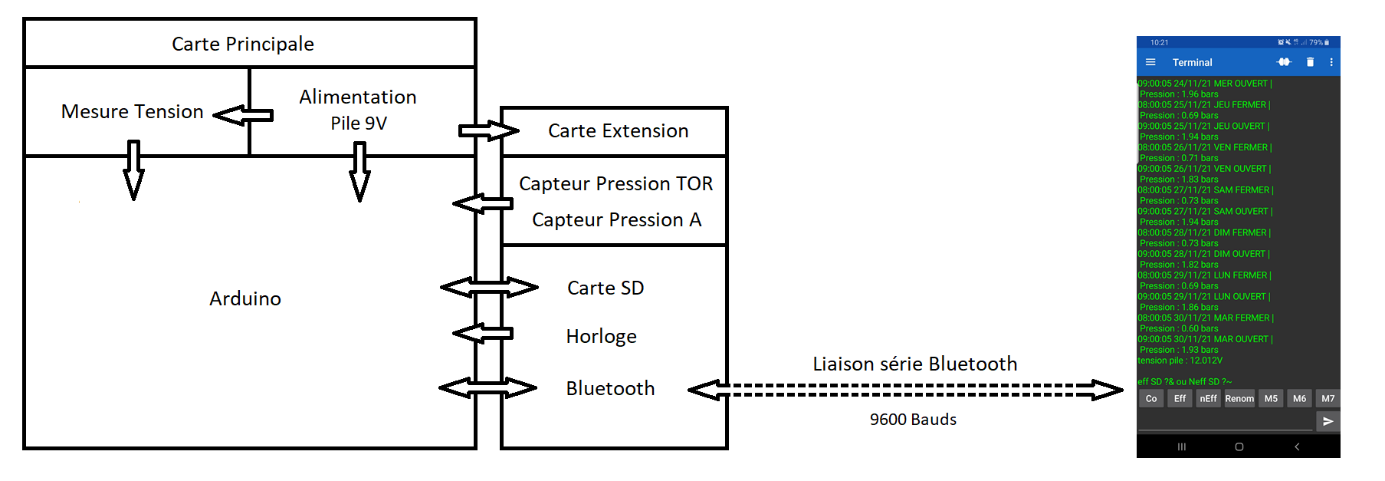


* Un connecteur pour un Capteur de pression Tout Ou Rien (TOR).
* Un connecteur pour un Capteur de pression Analogique.
* Un lecteur de carte SD pour stocker les informations.
* Un module RTC (Real Time Clock) permettant d’avoir la date et l’heure.
* Un module Bluetooth (HC-05) permettant la communication avec un smartphone.





### Schéma de principe :



IrriTrace : Carte principale / Carte Extension/Application mobile

### Fonctionnement :

* Lorsqu’il n’y a aucun évènement extérieur IrriTrace reste en mode Veille pour limiter la consommation de l’énergie.
* Le capteur de pression TOR détecte s’il y a de la pression dans la gaine d’irrigation, cet événement nous permet de dire que l’irrigation a été mise en route.
* Ce même événement permet aussi de réveiller le système.
* L’Arduino va ensuite récupère l’information de la pression dans la gaine grâce au capteur analogique ainsi que la date et l’heure de la RTC.
* Une fois ces informations récupérées l’Arduino va les stocker dans la mémoire sous cette forme :

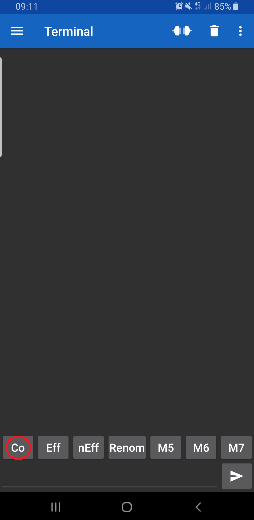
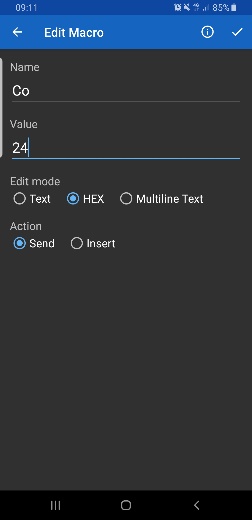
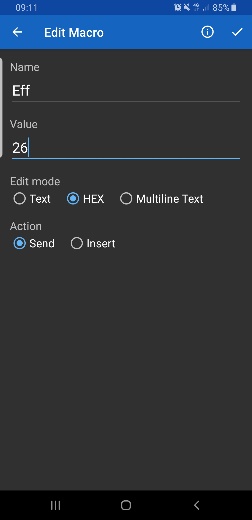
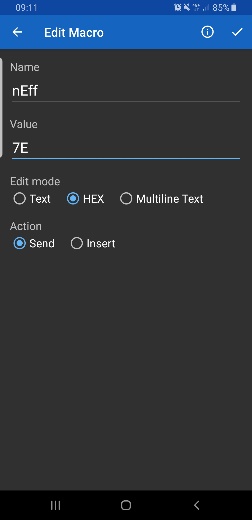
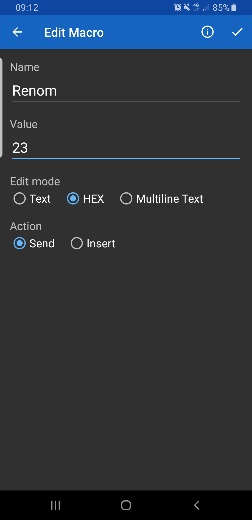
14:07:11 27/09/21 LUN OUVERT | Pression:1.93 bars

* Après ça le système repasse en mode veille jusqu’à ce que le capteur de pression TOR ne détecte plus de pression dans la gaine.
* À ce moment-là l’Arduino récupère les mêmes informations puis les stocks :

16:06:40 27/09/21 LUN FERMER | Pression:0.74 bars

* OUVERT et FERMER représente l’ouverture et la fermeture de la vanne ou l’électrovanne qui s’ouvre et qui se ferme.
* Pour récupérer les informations il faut ensuite appuyer sur un bouton qui se trouve sur le côté du boitier d’IrriTrace pour activer le Bluetooth et communiquer avec une application.

## Configuration de l’application Serial Bluetooth Terminal (SBT)

Pour pouvoir communiquer avec le module BLE il faut configurer 4 raccourcis :

Terminal

Raccourci renommer

Raccourci sauvegarde SD

Raccourci efface SD

Raccourci data

Dans l’écran du Terminal, appuyer longuement sur un raccourci pour le configurer.

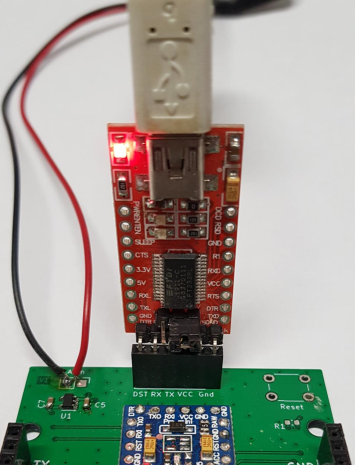
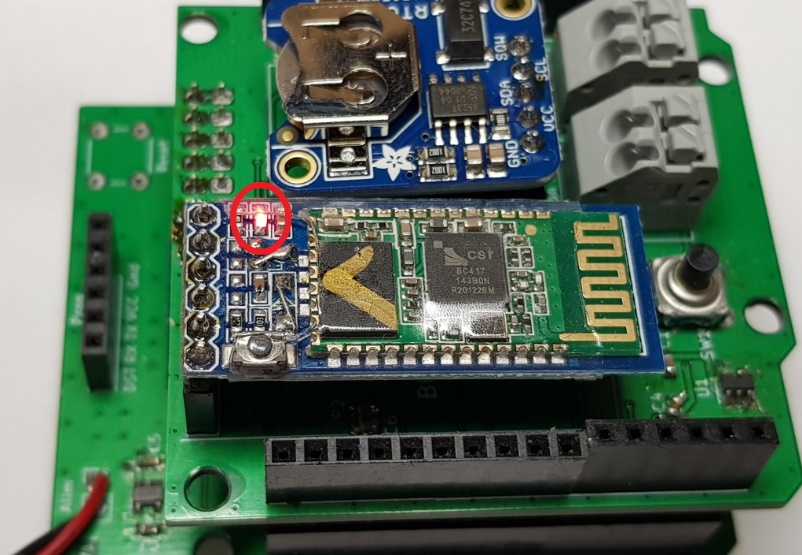
* Raccourci data : envoie le signal pour le transfert des données, sa valeur est 24 en hexadécimal ce qui correspond en ASCII au caractère « $ ».
* Raccourci efface SD : envoie le signal pour effacer les données de la carte SD après la réception des données, sa valeur est 26 en hexadécimal ce qui correspond en ASCII au caractère « & ».
* Raccourci sauvegarde SD : envoie le signal pour garder les données de la carte SD après la réception des données, sa valeur est 7E en hexadécimal ce qui correspond en ASCII au caractère « ~ ».
* Raccourci renommer : envoie le signal qui permet de lancer la procédure de renommage du module BLE, sa valeur est 23 en hexadécimal ce qui correspond en ASCII au caractère « # ».

Ces raccourcis seront à programmer dans l’application comme des boutons de sélection.

## Mise en fonctionnement d’IrriTrace

Connecter la pile 9V et attendre que la LED du module Bluetooth s’arrête de clignoter (env 10s), si le boitier est fermé on peut voir la LED clignoter à travers le couvercle.

Le programmateur peut rester connecté pour visualiser les informations sur le PC envoyé par l’Arduino.



LED Bluetooth

Programmateur

Pendant les 10 premières secondes des données sont enregistrées sur la carte SD.

## Activation/Appairage du Bluetooth

### IrriTrace

Pour activer le BLE, appuyer sur le bouton sur le côté du boitier, la LED du BLE se remet à clignoter (rapidement), le module reste alimenter pendant env 1min pour un appairage. S’il n’y a pas d’appairage après cette minute le module BLE s’éteint.

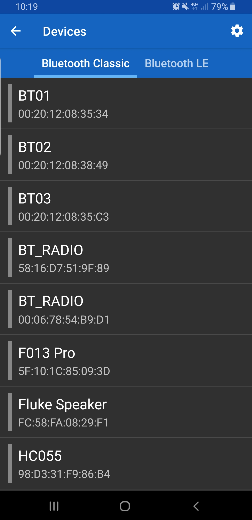
Il y a une possibilité de désactiver cette limite de temps en modifiant la variable limitB qui se trouve à la ligne 4 du programme, mais attention à la consommation d’énergie.

### Application SBT

Pour pouvoir s’appairer via l’application SBT :



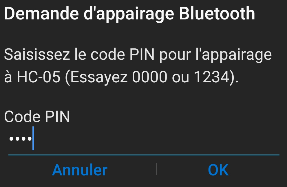
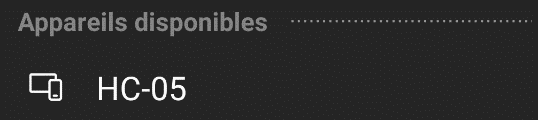
* Cliquer en haut à gauche de l’écran du Terminal :



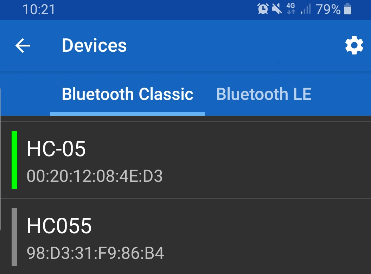
* Puis sur Devices, pour arriver dans l’interface de sélection des modules BLE :



* Cliquer ensuite sur la roue en haut à droite :
* Une fois arriver dans l’interface de recherche de BLE du smartphone, chercher le module BLE qui se nomme HC-05 (s’il n’a jamais été renommer), le code PIN pour l’appairage est par défaut « 1234 ».



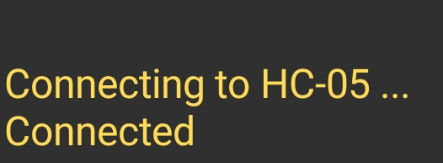
* Revenir en arrière sur l’écran Devices pour voir le module sélectionner :



* Le module BLE appairer, cette manipulation n’est à faire qu’une seule fois par module BLE.
* Revenir sur l’écran du Terminal puis se connecter au module, pour cela appuyer sur le bouton de connexion pour établir la liaison série :



* Une fois le module BLE et le smartphone connecté la LED du BLE clignote lentement.
* Au niveau du Terminal on peut voir la tentative de connexion au module BLE appairer et si la connexion s’est bien établie :



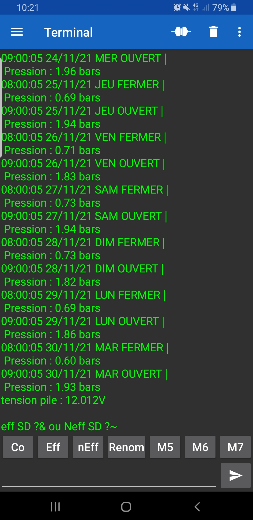
## Communication Bluetooth

Une fois la connexion établie, 2 choix possibles, soit la récupération des données, soit le renommage du module BLE.

### Récupération des données

* Pour la récupération des données, appuyer sur le « Raccourci data » (appuie court). Le raccourci envoie la valeur 24 en hexadécimal sur la liaison série qui est reçu par le module BLE qui va à son tour retransmettre le message à l’Arduino.

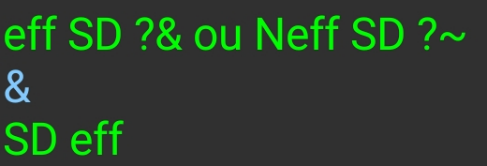
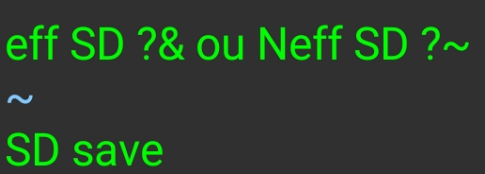
L’Arduino va reconnaitre cette valeur comme le signal pour envoyer les données sur la liaison série à travers le module BLE et être reçu par l’application.





* Une fois les données récupérées deux autres choix sont proposés :

Effacer ce qui est stocké dans la carte SD ou les garder, donc soit le « Raccourci efface SD » qui envoie la valeur 26 en hexadécimal (&), soit le « Raccourci sauvegarde SD » qui envoie la valeur 7E en hexadécimal (~).



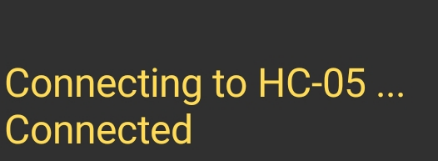
* Après cette dernière action l’Arduino éteint le module BLE et se met en veille.

### Renommage du Module Bluetooth

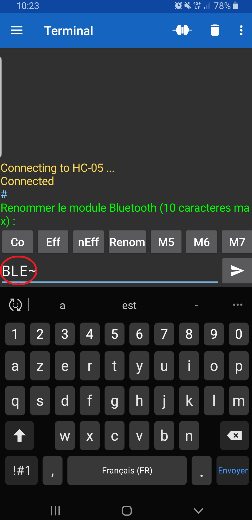
Pour le renommage du module BLE il faut établir une nouvelle connexion.



* Donc appuyer sur le bouton du boitier pour activer le BLE puis cliquer sur la connexion :



* Une fois revenue à ce niveau-là :
* Cliquer sur le « Raccourci renommer » qui envoie la valeur 23 en hexadécimal (#), le message suivant apparaît :



* « Renommer le module Bluetooth (10 caractères max) », dans l’exemple ci-dessus le module sera renommé « BLE », à noter qu’en plus du nom donné au module le caractère « ~ » est ajouté à la suite, ce caractère est obligatoire pour que l’Arduino sache à quel moment le nom du module BLE s’arrête.

Au niveau de l’application ce caractère en plus ne devra pas être ajouté par l’utilisateur, mais ajouter automatiquement de façon invisible pour l’utilisateur.



* Cliquer sur , si tout se passe bien le message retour est identique au nom envoyé.
* Après cette dernière action l’Arduino éteint le module BLE et se met en veille.