

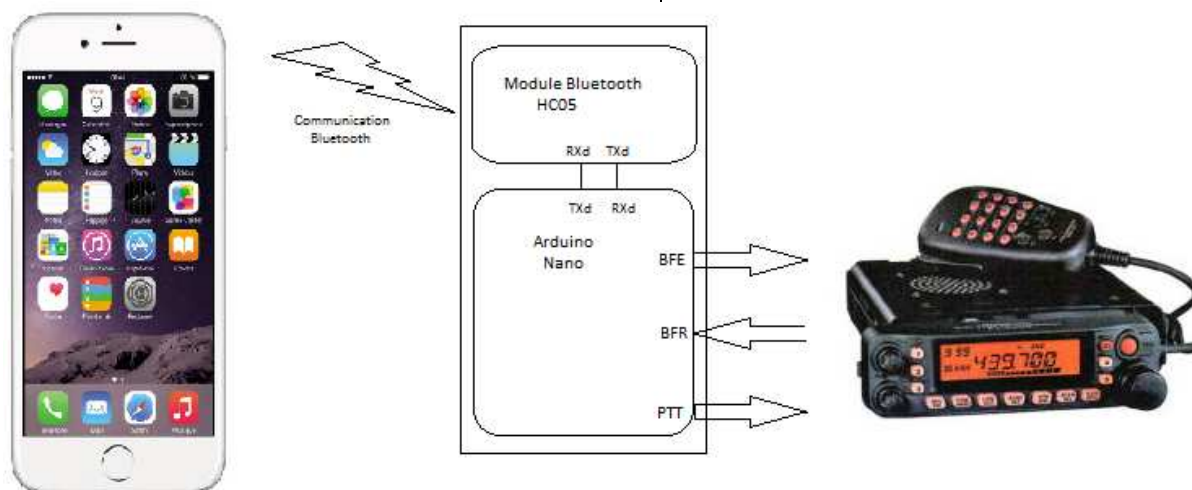
# Modem Kiss Arduino, Modilinkd

Par Jean-François de F5BCB

## Mobilinkd, Bluetooth et APRSDroid

Dans l'édition précédente du CQ44, j'ai décrit comment remplacer le modem Kiss TNC-Pi par un module Arduino Nano exécutant le code du TNC1 de Mobilinkd. Ce modem est raccordé par USB au contrôleur Raspberry Pi/Raspbian sur lequel est installé l'application APRX constituant ainsi un digipeater APRS avec (ou sans) accès à APRS-IS.

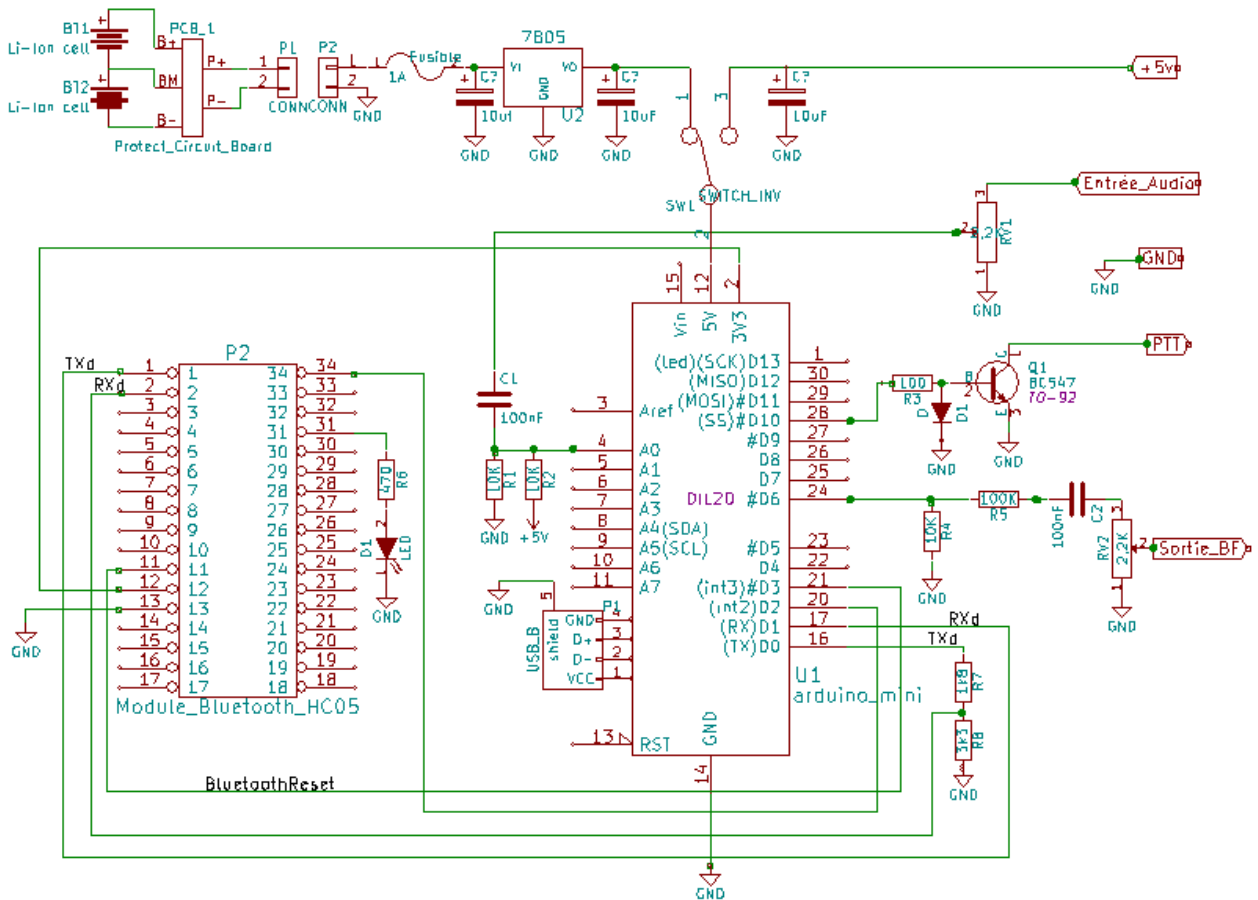
Dans cet article je vous propose de substituer l'ensemble Raspberry-Pi / Raspbian / APRX par votre Smartphone et l'application Android APRSDroid pour constituer un tracker APRS utilisant le GPS et les cartes off-line du smartphone. Les deux éléments communiquent par Bluetooth selon le principe suivant :



Pour que ceci fonctionne, il faut donc adjoindre un module Bluetooth à l'Arduino, les deux communiquant par les lignes de leur port série respectif (Txd et Rx). La vitesse du port série est de 38400 Bds sur l'interface Bluetooth et de 1200 Bds par défaut coté radio.

Le schéma de l'ensemble se présente comme ci-dessous. On reconnaîtra l'Arduino Nano chargé avec le firmware TNC1 : (<https://raw.githubusercontent.com/mobilinkd/tnc1/arduino/images/mobilinkd-473-arduino.hex>) selon la méthode indiquée dans le CQ44 du mois de décembre 2015. Les paramètres KISS sont accessibles de deux façons mais uniquement par Bluetooth :

- Par votre PC au moyen d'un utilitaire Windows Mobilinkd TNC Configuration (<http://www.mobilinkd.com/download/22/> ). Naturellement, votre PC devra posséder une interface Bluetooth (*intégré ou dongle USB*)
- Par votre smartphone sur lequel vous aurez installé l'utilitaire de configuration Mobilinkd TNC Config que vous chargerez depuis Google Play . Ces utilitaires donnent accès aux paramètres du modem KISS : TxDelay, Persistence, Slot Time & Full Duplex. Il vous permettra également de régler les niveaux audio RX et TX.



L'ensemble ainsi constitué peut se construire de façon compacte et pourra être équipé d'une alimentation autonome.

Dans le schéma ci-dessus deux batteries Li-Ion (16850) avec leur PCB de protection alimentent le tout au travers d'un régulateur 5v. Un inverseur permet de le raccorder à une alimentation 5v externe pouvant provenir de la radio. Ce montage peut aussi s'alimenter via le port USB de l'Arduino. La batterie n'est alors plus nécessaire.

Chacun fera son affaire du raccordement à la radio après avoir consulté le manuel et identifié les différents signaux PTT, entrée et sortie BF, masse et 5v éventuellement. Ce système constitue un tracker APRS pouvant être installé dans un véhicule. Il

exploite les coordonnées GPS fournies par le Smartphone et peut reporter sur une cartographie embarquée (Open Street Map par exemple) la position des stations APRS reçues.

La section Download du site de Mobilinkd vous fournira la documentation utilisateur (<http://www.mobilinkd.com/downloads/>)

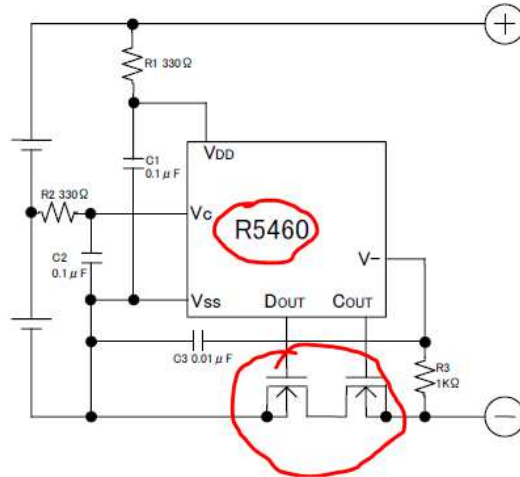
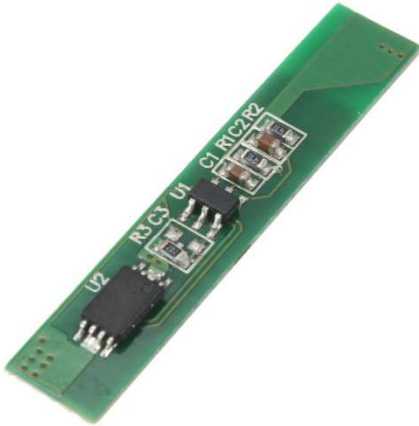
Concernant les modules Bluetooth, vous pourrez utilement consulter le site suivant qui prodigue quantité d'informations sur ces modules. <http://www.martyncurrey.com/>

Le site de l'auteur de l'application APRSdroid, Georg DO1GL, fournit également de très utiles informations : <https://aprsdroid.org/>

### Précisions concernant le module de protection des batteries Li-Ion.

J'ai choisi d'utiliser des batteries Samsung de la série SDI 18650 d'une capacité de 3Ah. Ces batteries, comme toutes celles utilisant cette chimie, nécessite une attention particulière durant la charge et la décharge : pas de surcharge ( $\max 4.3v \pm 0,03v$ ,  $\max 3A$ ), pas de décharge en dessous de la tension

spécifiée par le constructeur (2,75V), pas de débit au-delà des limites (6A). Cette protection est confiée à un module spécialisé constitué d'un processeur (R5460) et de transistors MOS de très faible RdsON (55F8205A). Ce module surveille donc ces paramètres et agit en conséquence sur franchissement de seuil en interrompant la charge ou la décharge.



La réalisation sur plaque pastillée :

