

Construction d'un récepteur bi-bande VHF/UHF FM

F4GSC - Jean-Luc Levant - ARALA

Cet article présente la construction d'un récepteur VHF/UHF FM dont le cœur de la partie radio est le module SA828 et piloté par une carte Arduino Pro mini, muni d'un afficheur OLED 128x64 pixels et contrôlé par un codeur optique. Le SA8xxx est équipé du circuit intégré radio RDA1846S qui est aussi utilisé dans la famille des Baofeng(UV3...). Son architecture est détaillée sur la figure ci-dessous.

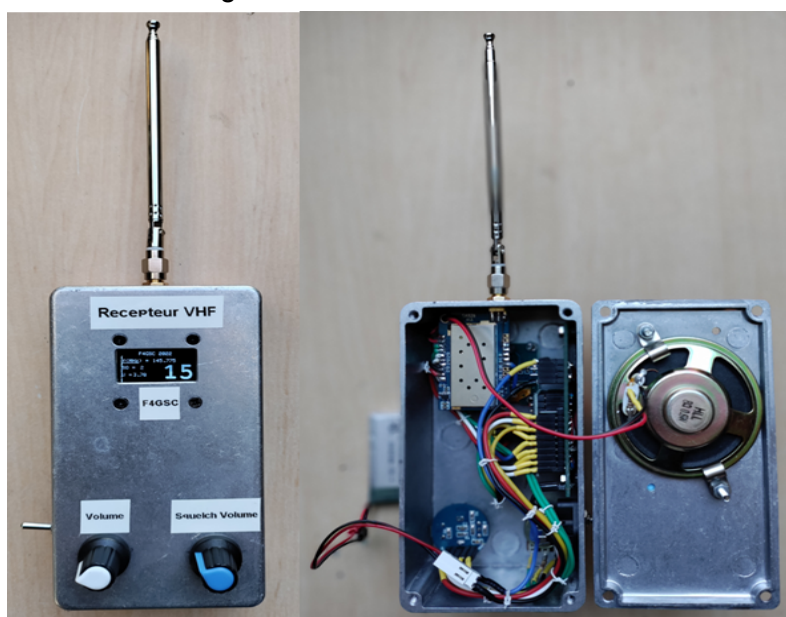


Figure 1. Récepteur FM à base de SA828.

Le SA8xxx est équipé du circuit intégré radio RDA1846S qui est aussi utilisé dans la famille des Baofeng(UV3...). Son architecture est détaillée sur la figure ci-dessous.

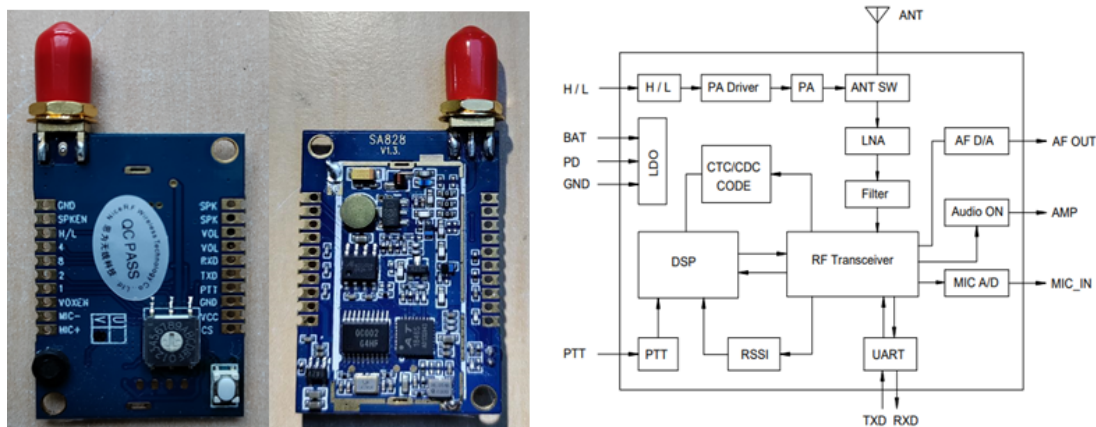


Figure 2. Module radio SA828 et schéma bloc de la puce RDA1846S.

Le SA828 intègre l'amplificateur audio (MIX2021). Il nécessite de rajouter la résistance R10 de 220 kOhms et le potentiomètre de 100 kOhms logarithmique. Le haut-parleur de 8 ohms est connecté sur les sorties SPK. Les broches RXD et TXD permettent de piloter le module à partir d'une liaison série TTL à 9600 bauds .

Schéma du récepteur

Le SA828 est prévu pour fonctionner de manière autonome en ajoutant un minimum de composants. Par exemple, les 16 fréquences de réception et d'émission enregistrées dans le module sont sélectionnables par l'intermédiaire d'un commutateur à 16 positions soudé sur la carte.

Cependant, ce module est utilisé différemment. Un afficheur est ajouté pour visualiser la fréquence sélectionnée, le niveau du signal reçu ainsi que la tension de la batterie. Le pilotage du SA828 par la carte Arduino peut se faire de deux manières: Interface série et parallèle.

Interface Série

Cette interface utilise la liaison série pour sélectionner la fréquence à recevoir et pour ajuster le squelch.

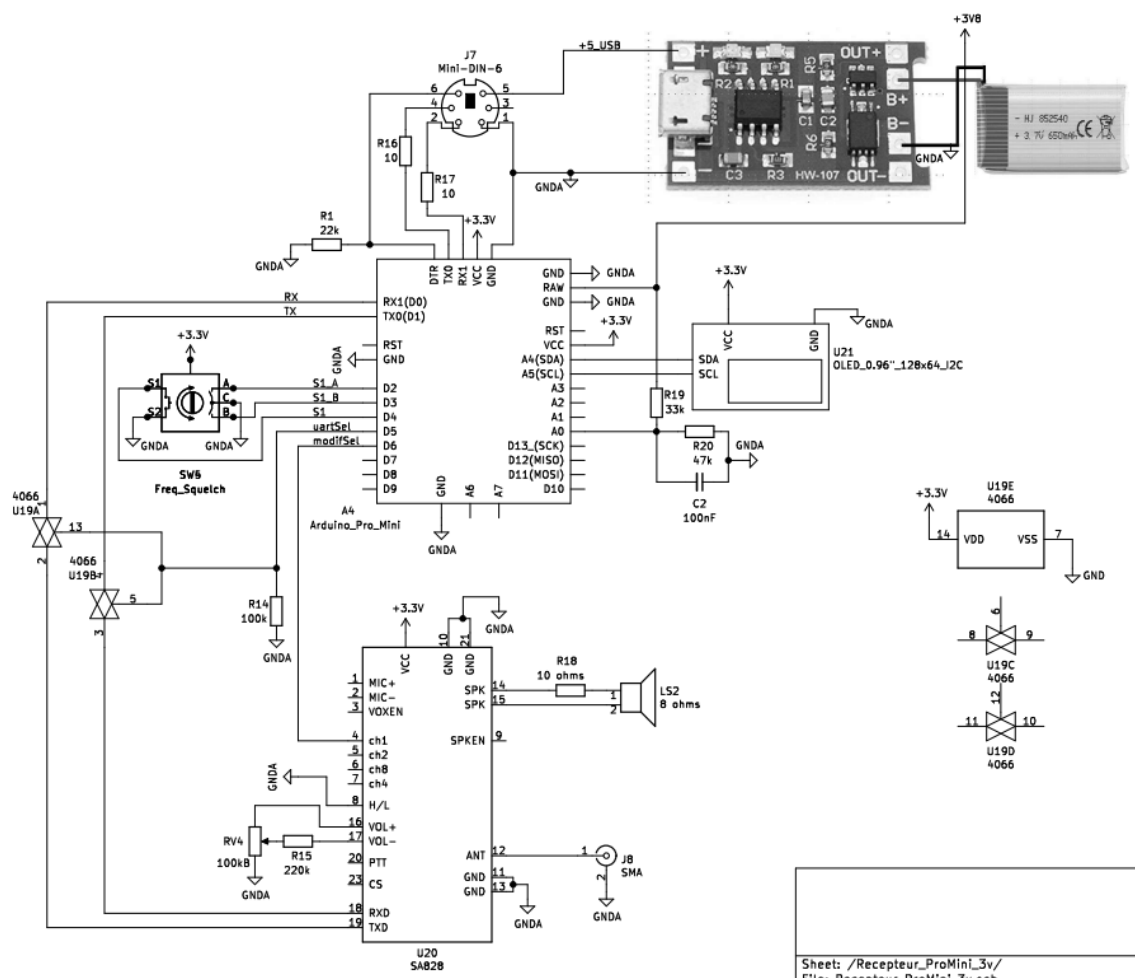


Figure 3. La liaison série gère la programmation des paramètres du module.

Pour changer la fréquence de travail et le niveau du squelch il est nécessaire de téléverser les 32 fréquences (16 en réception et 16 en émission) ainsi que le niveau du squelch à chacun des changements. Le temps mis pour transférer l'ensemble est de 416 ms.

Interface parallèle

Cette interface utilise les 4 entrées CH4, CH8, CH2 et CH1 pour sélectionner la fréquence de travail. L'ajustement du squelch est réalisé par la liaison série.

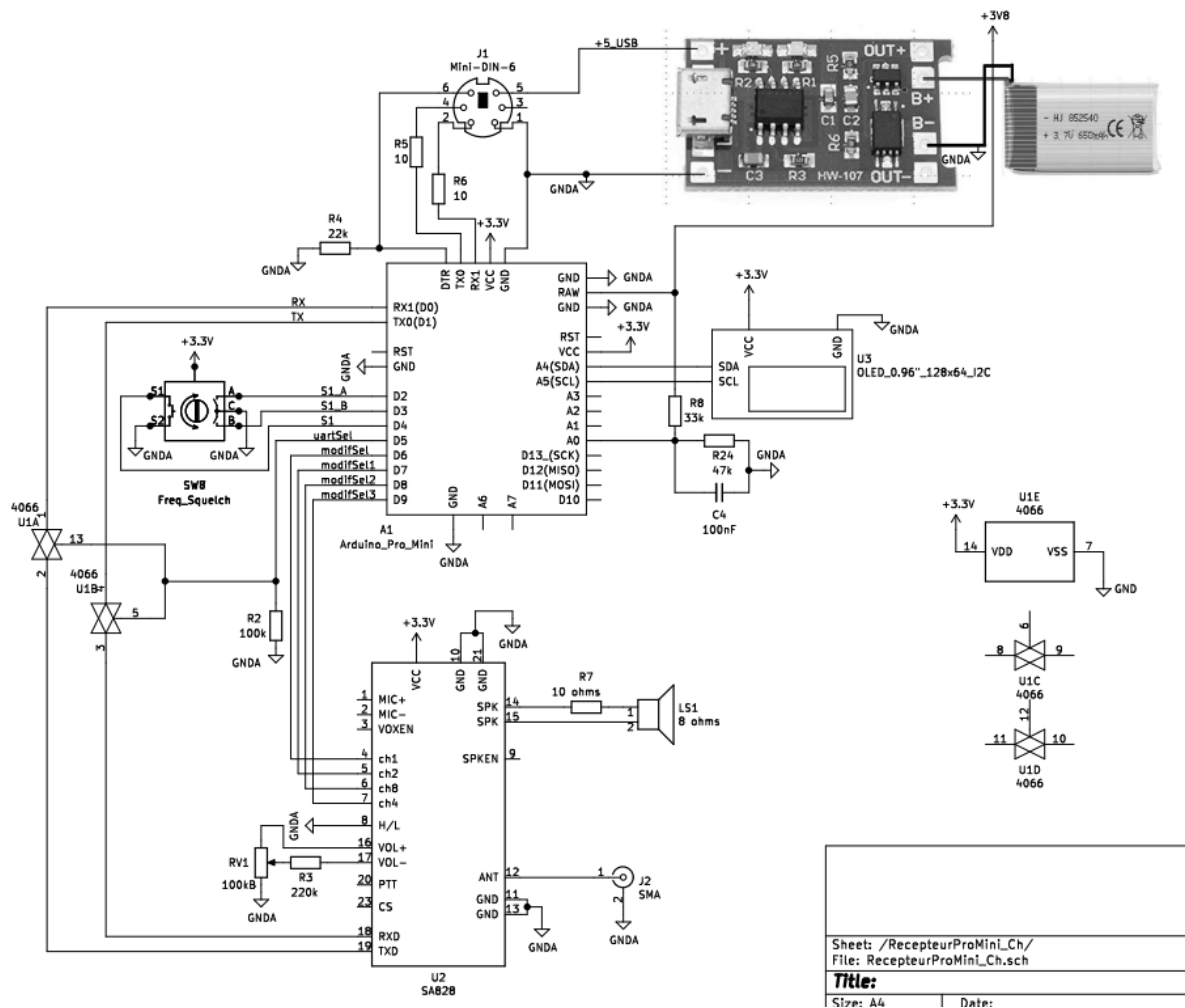


Figure 3. Les entrées CH4,CH8,CH2 et CH1 sélectionnent la fréquence.

Le changement de fréquence s'effectue en 116 ms et du niveau du squelch en 416 ms.

Quelque soit l'interface la carte Arduino Pro mini gère:

- le codeur optique qui règle la fréquence et le squelch numérique
- Le module radio SA828 est piloté par la liaison série à 9600 bauds à partir des lignes de transmission RXD et TXD
- L'afficheur OLED est contrôlé à partir du bus série synchrone I2C (SDA et SCL)
- Le connecteur mini-din 6 broches permet de connecter l'adaptateur USB/RS232

Les sorties SPK sont connectées directement au HP. Une résistance de 10 ohms a été insérée entre le HP et la sortie *SPK* pour réduire la consommation et la longévité de la batterie. Le potentiomètre de 100 kohms permet de régler le volume audio. La résistance série de 200 kohms permet de régler le gain de l'amplificateur. Le connecteur SMA 50 ohms

permet de connecter l'antenne. Les résistances R11 et R13 permettent de mesurer la tension de la batterie. Le point milieu de ce pont diviseur est connecté à l'entrée du convertisseur analogique/numérique A0. La batterie est du type Lipo avec une capacité de 650 mA.heure. Un chargeur intégré maintient la batterie à sa capacité nominale. Il fonctionne à partir d'un +5 v externe ou d'une prise USB.

Les caractéristiques du module SA828

Les principales caractéristiques du module sont données ci-dessous:

- VHF → 134-174 MHz, UHF → 400 - 480 MHz
- Squelch → réglable numérique de 1 à 8
- Lecture du niveau du signal reçu (RSSI)
- Sensibilité → -124 dBm
- CTCSS → RX_CTCSS et TX_CTCSS, 38 niveaux
- Tension → +3.3v à +5.5v
- Consommation en réception (+5v) → 60 mA en mode veille et 170 mA avec une station reçue.

Lors de l'achat de ce module, il est nécessaire de préciser la version désirée soit VHF ou UHF. Le filtre de bande de sortie est adapté à la bande de fréquence.

Fonctionnement de la liaison série

La liaison dessert les trois fonctions décrites ci-dessous:

- Téléversement du programme (sketch Arduino), PC ↔ Arduino
- Mise au point par le moniteur de l'IDE Arduino, PC ↔ Arduino
- Le pilotage du SA828 , SA828 ↔ Arduino

Pour éviter un conflit entre les données provenant de la liaison série du PC et celle provenant du SA828, des interrupteurs CMOS (CD4066,HC4066...) isolent ces deux sources. A la mise sous tension, ce sont les données provenant du PC qui accèdent à l'Arduino Pro mini. Pour cela le signal *uartSel* est maintenu au niveau zéro par la résistance de 100 kohms. Une fois le téléversement achevé, l'accès du SA828 sera possible par le en maintenant le signal *uartSel* à la valeur 1 (HIGH) dans la section *setup()* du programme.

```
digitalWrite(uartSel,HIGH);
```

L'utilisation du moniteur de mise au point nécessite de sélectionner la liaison du PC en maintenant *uartSel* à la valeur 0 (LOW).

```
digitalWrite(uartSel,LOW);
```

Principe de pilotage du SA628

Le fonctionnement du récepteur est expliqué en commentant les parties les plus importantes du programme. Le programme complet est téléchargeable sur le site de l'ARALA.

Sélection de la fréquence

Les 32 fréquences sont contenues dans la table *tabFreq*. La première fréquence est celle de l'émission et la deuxième celle de la réception. Ces fréquences sont définies par l'utilisateur.

```

//=====
//-- Table des fréquences d'émission et de réception
//-- TX1,RX1,TX2,RX2..TX16,RX16,Tx_CTCSS,RX_CTCSS,SQ
//=====
const float tabFreq[] =
    {145.775,145.775,145.755,145.755,
     145.735,145.735,145.715,145.715,
     145.695,145.695,145.675,145.675,
     145.655,145.655,145.635,145.635,
     145.615,145.615,145.595,145.595,
     145.575,145.575,145.555,145.555,
     145.535,145.535,145.515,145.515,
     145.495,145.495,145.475,145.475};

```

Le principe retenu pour sélectionner une fréquence est de reprogrammer la mémoire du SA828 en téléversant la table *tabFreq* dans la mémoire du microcontrôleur du module. La commande permettant ce transfert correspond à la syntaxe ci-dessous:

```
AAFA3 TX1,RX1.... TX16,RX16,CTCSS,CTCSS,SQ\CR\LF
```

AAFA3 est le code de la commande à envoyer pour télécharger les 32 fréquences ainsi que les codes CTCSS et la valeur du niveau de squelch (SQ). La fonction de téléversement de la table des fréquences et des deux autres paramètres est détaillée ci-dessous. Le premier paramètre *frx* est la fréquence à recevoir et le deuxième *squelch* est le niveau du squelch à changer.

```

void configModule(float frx,int squelch)
{
    int freqIndex;
    Serial.flush();

    Serial.print("AAFA3");
    Serial.print(frx,4);Serial.print(",");
    Serial.print(frx,4);Serial.print(",");
    for (freqIndex=2;freqIndex<=nbFreq;freqIndex++)
    {
        Serial.print(145.000,4);Serial.print(",");
    }
    Serial.print("000") ; Serial.print(",");
    Serial.print("000") ; Serial.print(",");
    Serial.println(squelch);
    Serial.flush();
    //--- Change la fréquence et la valeur du squelch
    digitalWrite(CH1,LOW);
    delay(100);
    digitalWrite(CH1,HIGH);
}

```

Les entrées CH2, CH8 et CH4 sont connectées à VCC. La valeur du niveau de squelch est envoyée en même temps que la table des fréquences et les codes CTCSS. Pour que le changement soit effectif, l'entrée CH1 est maintenue à zéro pendant 100 ms.

```

//--- Indique un changement de la fréquence, du squelch et des codes CTCSS
digitalWrite(CH1,LOW);
delay(100); //100 ms
digitalWrite(CH1,HIGH);

```

Réglage du squelch

La variable *squelch* est augmentée ou diminuée d'une valeur constante valeur égale à +1 ou -1 en fonction du sens de rotation du codeur optique. Sa plage de réglage est comprise entre 1 et 8.

```
squelch=squelch+valeur;  
if (squelch>8) squelch=8;  
if (squelch<1) squelch=1;
```

Lecture du niveau du signal reçu (RSSI)

La lecture du RSSI est réalisée en envoyant la commande *RSSI?\CRLF* sur la sortie TX de l'Arduino Nano. Le SA828 renvoie sur l'entrée RX un code compris entre 0 et 255 pour indiquer le niveau du signal. La fonction ci-dessous détaille comment lire le RSSI.

```
//=====  
//-- Lit la valeur de la force du signal  
//=====  
void rssi_reading(void)  
{  
  char buf[10];  
  Serial.flush();  
  Serial.println("RSSI?");  
  Serial.readBytesUntil('\n',buf,9);  
  rssi = (byte) atoi(&buf[5]);  
}
```

Le niveau du signal reçu est placé dans le tampon *buf* à l'adresse 5. La valeur est au format ASCII qui est convertie en format numérique 8-bits par la ligne de programme *rssi = (byte) atoi(&buf[5])*.

Codeur Optique

Il est représenté sur la figure ci-dessous et il permet d'ajuster les paramètres du module radio. Le codeur possède deux sorties (CLK et DT) qui sont électriquement décalées de 90° l'une par rapport à l'autre. C'est le sens de rotation de la roue qui permettra de détecter les états électriques pour déterminer si une action incrémentation ou décrémentation doit être faite.

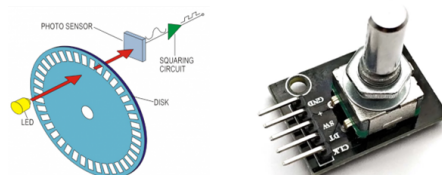


Figure 4. Exemple de codeur optique.

Les deux signaux *CLK* et *DT* sont reliés aux entrées 2 (*codeurRotatifB_2*) et 3 (*codeurRotatifA_3*) de la carte Arduino ainsi qu'un bouton poussoir (*key*) connecté sur l'entrée 4. Ces deux signaux permettent de déterminer dans quel sens le codeur est tourné et ils servent à indiquer si il faut incrémenter ou décrémentation la fréquence (*frx*) et le niveau du squelch (*squelch*). Le troisième signal (*key*) permet de sélectionner *frx* ou *squelch*. Le

choix de l'un des paramètres est indiqué par le soulignement de l'un de ces deux paramètres.

Afficheur OLED

Cet afficheur OLED comprend une matrice de points de 128 par 64. Il est mono couleur et graphique. Il est piloté par l'interface I2C dont l'adresse est 0x3C. Il est nécessaire d'utiliser les bibliothèques *Adafruit_SH1106.h* et *Adafruit_GFX.h* (Adafruit_SH1106-master) qui sont importées par le gestionnaire de librairie de l'IDE d'Arduino.

Carte Arduino

La carte Arduino Pro mini a été choisie pour des raisons d'optimisation de consommation. En effet, le convertisseur USB <-> RS232 n'est pas présent sur cette carte. De plus, la diode de protection mise en série dans l'alimentation pour prévenir des inversions de polarités et la résistance de polarisation de la diode LED ont été supprimées pour augmenter la plage de fonctionnement de la batterie (+3.3 v à +4.2 v).

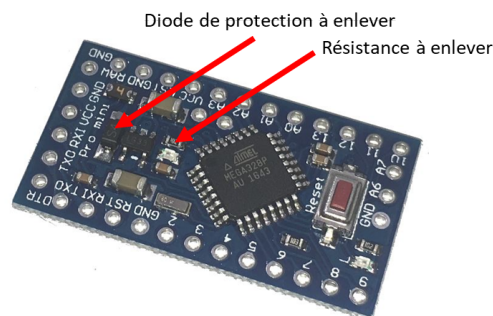


Figure 5. Arduino Pro mini, 3.3v 8 Mhz.

Adaptateur USB/RS232 TTL

Pour programmer la carte Arduino, il est nécessaire de connecter le convertisseur USB/RS232 externe sur les broches +VCC, GND, TXD, RXD et DTR. Il est à noter que le +VCC doit fournir le +5V pour recharger la batterie alors que les signaux TXD et RXD doivent être fournis en +3.3V. Pour ma part j'utilise celui de chez DSD TECH pour lequel j'ai modifié les signaux +VCC et CTS:

- CTS → DTR, j'ai coupé la piste de la broche CTS et je l'ai reconnectée sur le DTR (fil bleu)
- VCC connecté sur le +5v après avoir coupé la piste et mis le strap en position +3V3.

La figure ci-dessous montre l'adaptation faite sur cet adaptateur.

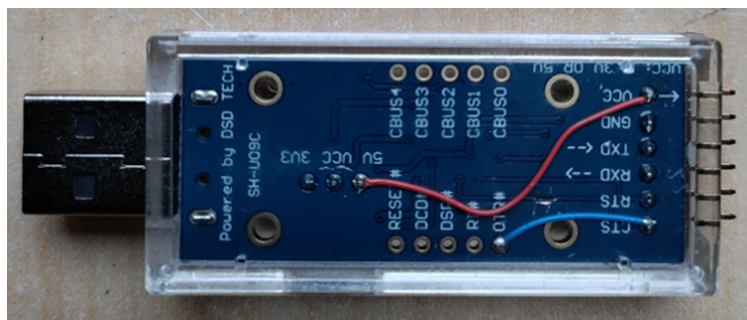


Figure 6. Adaptateur USB/RS232 TTL de chez DSD TECH

Conclusions

L'utilisation des modules radio de la famille SAXXX et d'une petite carte Arduino permettent de construire facilement un récepteur FM. Le programme pourra être trouvé sur le site de l'ARALA dans le dossier Technique/Arduino/Zone de telechargement (RecepteurVHFUHF Mars2022_SA828_V1.zip).