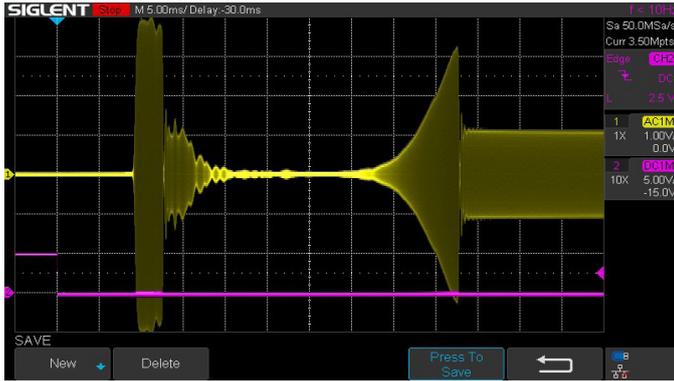


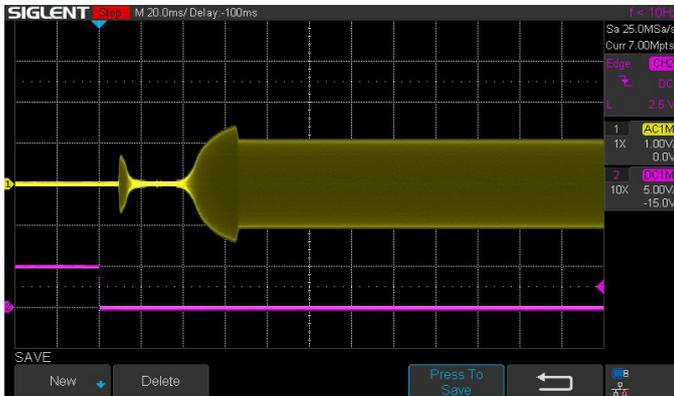
# Anti-Overshoot avec un Arduino

Jacques Pochet F6BQP

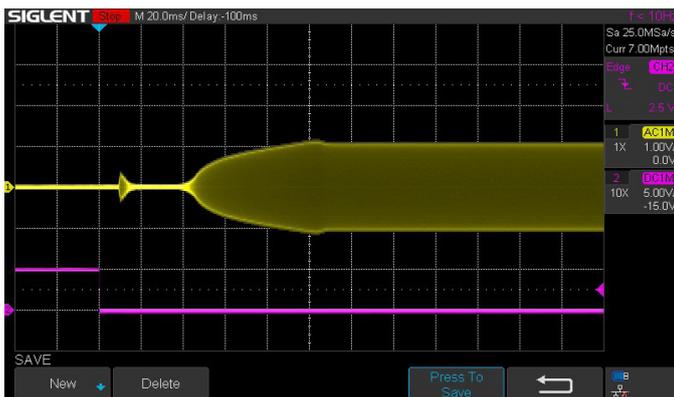
On observe sur la plupart des transceivers le défaut suivant : la présence d'un pic de puissance, qui peut être très élevé, au moment du passage en émission (appelé en anglais overshoot). Hélas, cette anomalie n'est jamais spécifiée par les constructeurs ...



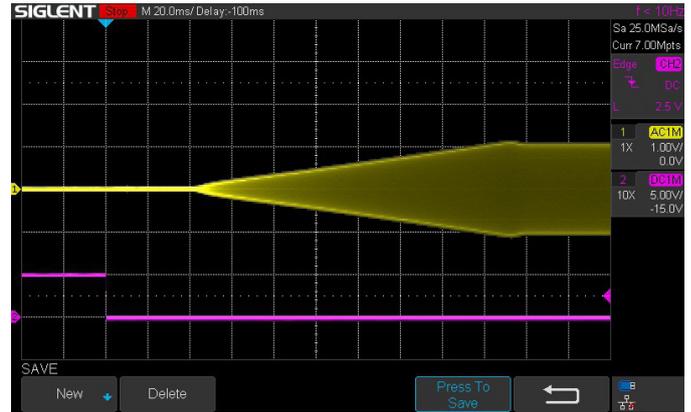
Ce défaut est très gênant dès lors que l'on attaque un amplificateur linéaire car les circuits de protection de ce dernier entrent immédiatement en jeu, provoquant la mise en sécurité de l'ampli.



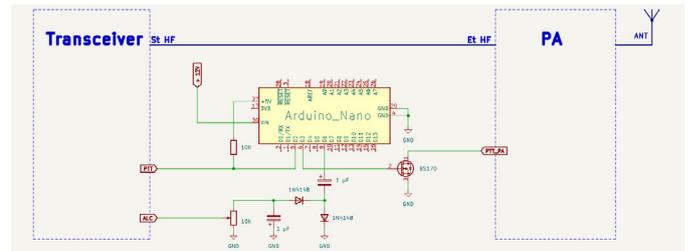
Parmi les transceivers qui n'ont pas ce défaut on peut citer (entre autres) Elecraft ou bien le Kenwood TS-890S.



Sur la figure 1, on peut voir le détail de l'overshoot du FT847 en AM pour une puissance de sortie réglée à 10 W. L'overshoot présente une tension égale à 4 fois la tension correspondant à la puissance de 10 W, soit une puissance de  $10 \times 4^2 = 160$  W, et ceci pendant environ 3 millisecondes (les mesures sont faites en AM ou FM, car en SSB il est plus difficile de visualiser l'overshoot).



Je vous propose un petit montage très simple pour résoudre ce problème avec un microcontrôleur Arduino Nano, dont voici le principe : en réception, l'Arduino génère une tension négative qui est injectée sur l'entrée ALC du transceiver.



L'Arduino surveille en permanence la sortie PTT du transceiver. Lorsqu'il y a passage en émission, le PTT passe à 0 V et ceci déclenche une interruption sur l'Arduino. Après un délai de 15 ms l'Arduino relâche la tension d'ALC qui va remonter jusqu'à 0 V avec une constante de temps de 10 ms, et la puissance de sortie du transceiver va monter progressivement pour atteindre la puissance nominale sans créer d'overshoot.

Les résultats sont visibles sur les figures 2 à 4, pour différentes valeurs de la tension d'ALC. La trace de couleur mauve représente le PTT alors que la trace de couleur jaune représente l'enveloppe du signal HF en sortie antenne du transceiver.



Le meilleur compromis est obtenu pour une tension d'ALC de -1,15 V (figure 3) avec un temps de montée en puissance de 80 ms.

Pour une tension d'ALC de -1,0 V (figure 2) il reste un peu d'overshoot.

Pour une tension d'ALC de -1,3 V (figure 4) le temps de montée en puissance s'allonge à 160 ms.

Ces valeurs sont uniquement valables pour le FT847.

Pour trouver le bon réglage :

- ▶ débrancher le cordon PTT entre le transceiver et l'Arduino ;
- ▶ passer en émission 100 W FM ;
- ▶ ajuster le potentiomètre de 10 kΩ pour avoir entre 5 et 10 W en sortie du transceiver ;
- ▶ rebrancher le cordon PTT ;
- ▶ vérifier à l'oscilloscope (à mémoire, c'est mieux) que l'overshoot a bien disparu.

```

1  /*-----*/
2  *   Anti Over Shoot pour ampli lineaire par F6BQP  V2.1 le 17/02/2022   *
3  *   EXEM pour fonctionner avec un Arduino nano   *
4  *   Attention: les temporisations sont faites avec delayMicroseconds() car
5  *   l'instruction delay() ne fonctionne pas a l'interieur d'une interruption
6  *   /*-----*/
7  //----- Affectation des Pins Arduino -----
8  #define LED 13 // sortie LED
9  #define PTT_PA 3 // sortie PTT PA
10 int PTT = 2; // entree PTT obligatoirement pin 2
11
12 //----- Routine d'interruption -----
13 void Interruption()
14 {
15   if (digitalRead(PTT) == LOW) // si emission
16   {
17     digitalWrite(PTT_PA, HIGH); // PTT PA = OV
18     digitalWrite(LED, HIGH); // allume LED
19     delayMicroseconds(15000); // 15000 microsecondes MAX
20     // delayMicroseconds(16383); // si besoin de plus de 16 ms
21     digitalWrite(LED, LOW); // eteint LED
22     digitalWrite(6, LOW); // ALC OFF
23   }
24   else // si reception
25   {
26     analogWrite(6, 127); // ALC ON
27     delayMicroseconds(5000); // attente 5ms
28     digitalWrite(PTT_PA, LOW); // PTT PA OFF
29   }
30 }
31
32 //----- Routine d'initialisation au demarrage -----
33 void setup()
34 {
35   pinMode(LED, OUTPUT); // sortie LED
36   pinMode(PTT_PA, OUTPUT); // sortie PTT PA
37   pinMode(PTT, INPUT_PULLUP); // entree PTT
38
39   digitalWrite(PTT_PA, LOW); // PTT PA OFF
40   analogWrite(6, 127); // ALC ON
41
42   attachInterrupt(0, Interruption, CHANGE); // initialise interruption sur D2
43
44
45 //----- boucle principale -----
46 void loop()
47 {
48 }
49 // FIN du programme

```

Le schéma est donné figure 5. La tension négative est générée par un signal PWM de 5 Vcc à 980 Hz sur la broche 6 de l'Arduino et qui est redressé/ filtré en négatif.

Un transistor BS170 fournit le PTT pour le PA.

Le logiciel est donné figure 7. Le programme tourne en boucle en attente d'une interruption générée par le PTT sur la broche 2.

Le fichier .ino est ici :

<http://urls.r-e-f.org/ps130gj>

Ce dispositif a été testé avec succès sur les transceivers suivants : FT847, FT857, IC706 et IC7300. Bonne réalisation !