

```

// ADF4251 and Arduino
// By Alain Fort F1CJN feb 2,2016
//
//*****
// * Programme minimum pour déterminer la fréquence de sortie du synthé *
// * Source de F1CJN modifiée *
// * Michel Vonlanthen HB9AFO 12.4.2017 *
//*****
//
// ***** HARDWARE IMPORTANT *****
// Avec un Arduino UN0 : utilise un pont de résistances pour réduire la tension, MOSI (pin 11) vers
// ADF DATA, SCK (pin13) vers CLK ADF, Select (PIN 3) vers LE
// Resistances de 560 Ohm avec 1000 Ohm à la masse sur les pins 11, 13 et 3 de l'Arduino UNO pour
// que les signaux envoyés DATA, CLK et LE vers l'ADF4351 ne dépassent pas 3,3 Volt.
// Pin 2 de l'Arduino (pour la detection de lock) connectee directement à la sortie MUXOUT de la
// carte ADF4351
// La carte ADF est alimentée en 5V par la carte Arduino (les pins +5V et GND sont proches de la
// LED Arduino).
//*****

#include <SPI.h>
#define ADF4351_LE 3

uint32_t registers[6] = {0x4580A8, 0x80080C9, 0x4E42, 0x4B3, 0xBC803C, 0x580005} ;
// 437 MHz avec ref à 25 MHz

int address,modif=0,WEE=0;

double RFout, REFin, INT, PFDRFout, OutputChannelSpacing, FRACF;
double RFoutMin = 35, RFoutMax = 4400, REFinMax = 250, PDFMax = 32;
unsigned int long RFint,RFintold,INTA,RFcalc,PDRFout, MOD, FRAC;
byte OutputDivider;byte lock=2;
unsigned int long reg0, reg1;

```



```

RFintold=1234;//pour que RFintold soit different de RFout lors de l'init
RFout = RFint/100 ;// fréquence de sortie
OutputChannelSpacing = 0.01;// Pas de fréquence = 10kHz

} // Fin setup

//*****Loop*****
void loop()
{
  RFout=RFint;
  RFout=RFout/100;
  if ((RFint != RFintold) || (modif==1)) {
    //Serial.print(RFout,DEC);Serial.print("\r\n");
    if (RFout >= 2200) {
      OutputDivider = 1;
      bitWrite (registers[4], 22, 0);
      bitWrite (registers[4], 21, 0);
      bitWrite (registers[4], 20, 0);
    }
    if (RFout < 2200) {
      OutputDivider = 2;
      bitWrite (registers[4], 22, 0);
      bitWrite (registers[4], 21, 0);
      bitWrite (registers[4], 20, 1);
    }
    if (RFout < 1100) {
      OutputDivider = 4;
      bitWrite (registers[4], 22, 0);
      bitWrite (registers[4], 21, 1);
      bitWrite (registers[4], 20, 0);
    }
    if (RFout < 550) {
      OutputDivider = 8;

```

```

    bitWrite (registers[4], 22, 0);
    bitWrite (registers[4], 21, 1);
    bitWrite (registers[4], 20, 1);
}
if (RFout < 275) {
    OutputDivider = 16;
    bitWrite (registers[4], 22, 1);
    bitWrite (registers[4], 21, 0);
    bitWrite (registers[4], 20, 0);
}
if (RFout < 137.5) {
    OutputDivider = 32;
    bitWrite (registers[4], 22, 1);
    bitWrite (registers[4], 21, 0);
    bitWrite (registers[4], 20, 1);
}
if (RFout < 68.75) {
    OutputDivider = 64;
    bitWrite (registers[4], 22, 1);
    bitWrite (registers[4], 21, 1);
    bitWrite (registers[4], 20, 0);
}

INTA = (RFout * OutputDivider) / PFDRFout;
MOD = (PFDRFout / OutputChannelSpacing);
FRACF = ((RFout * OutputDivider) / PFDRFout) - INTA) * MOD;
FRAC =round(FRACF); // On arrondit le résultat

registers[0] = 0;
registers[0] = INTA << 15;// OK
FRAC = FRAC << 3;
registers[0] = registers[0] + FRAC;

registers[1] = 0;

```

```
registers[1] = MOD << 3;
registers[1] = registers[1] + 1 ;// ajout de l'adresse "001"
bitSet (registers[1], 27);// Prescaler sur 8/9

bitSet (registers[2], 28);// Digital lock == "110" sur b28 b27 b26
bitSet (registers[2], 27);// digital lock
bitClear (registers[2], 26);// digital lock

    SetADF4351();// Programme tous les registres de l'ADF4351
RFintold=RFint;modif=0;

}

} // fin loop
```