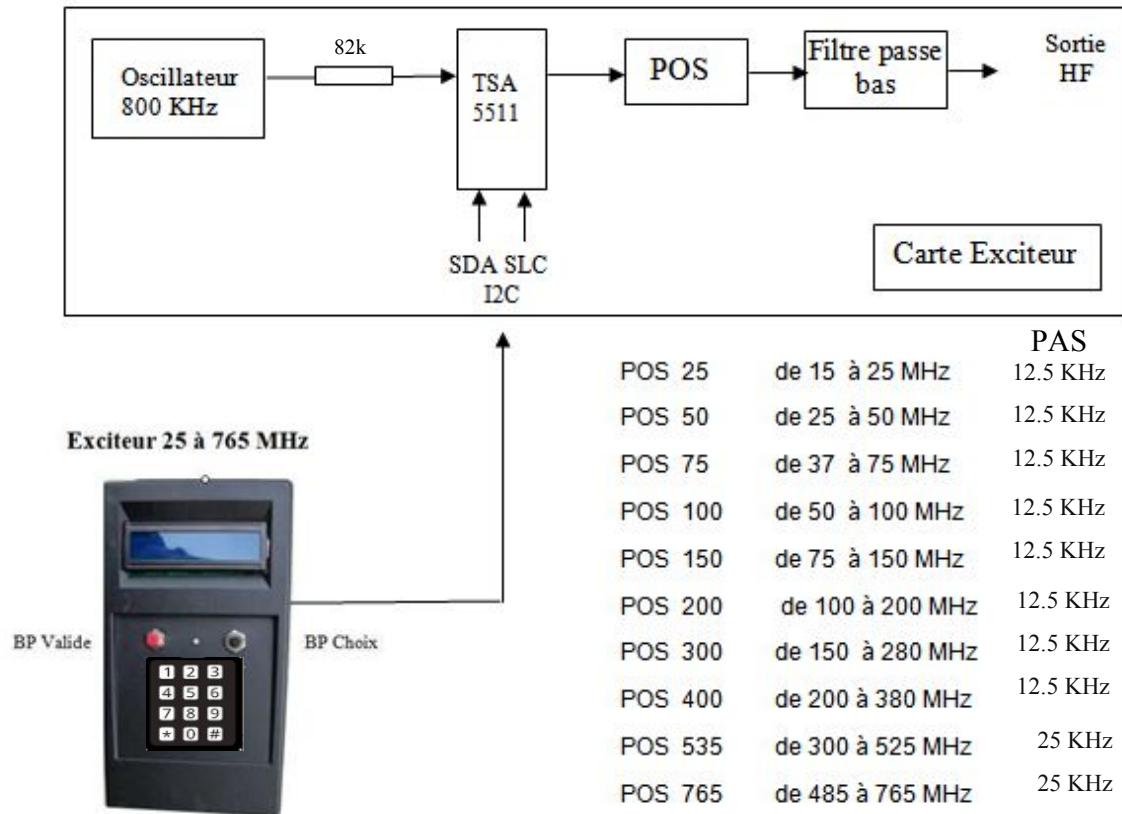


Exciteur HF – UHF

Le projet est commandé par un Arduino UNO ou NANO.

Le schéma synoptique du générateur HF



Utilisation

Brancher le câble I2C entre le contrôleur et l'exciteur

Allumer le contrôleur

A la première sous tension l'appareil passe en mode paramétrage

Vous faire le choix du POS par a touche (#) du clavier

Validation par la touche (*)

Note :

Pour des raisons techniques du TSA5511 le pas est de 12.5 KHz du POS 25 au POS 400 et de 25 KHz pour les autres.

Après validation la fréquence centrale du POS est affichée et l'exciteur émet sur cette fréquence.

Exemple pour un POS 200 la fréquence sera de 100 MHz

Ont peut ajuster la fréquence par les touches (#) pour monter d'un pas et (*) pour reculer d'un pas

Pour ajuster la fréquence plus vite appuyer sur le bouton noir et entrez la fréquence sans les décimales

Valiser par (#)

Ensuite ajuster la fréquence pas par pas.

Pour changer la Fréquence après avoir changé de POS appuyez sur le bouton rouge pour revenir au paramétrage

Matériel

1 Arduino UNO ou NANO



1 LDC I2C



1 Clavier 12 touche



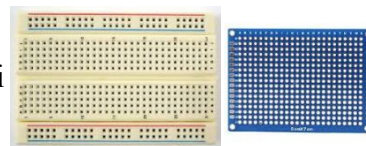
2 Boutons poussoir



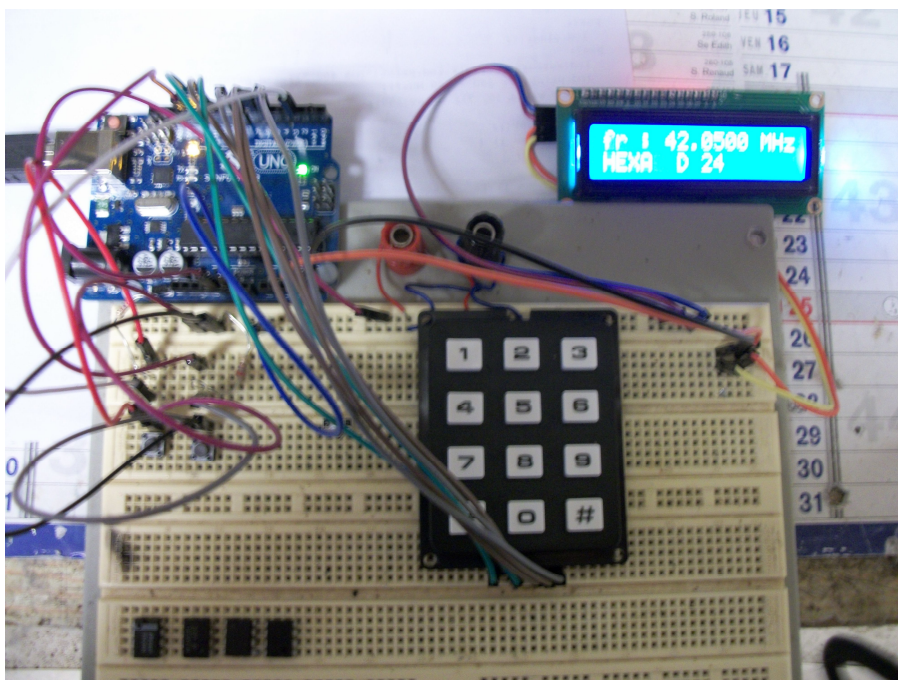
2 Résistances de 2K7



Une platine d'essai ou un Circuit imprimé d'essai



Mon montage

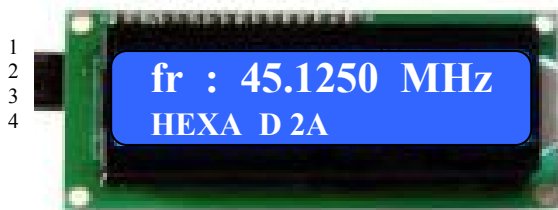


Câblage du clavier



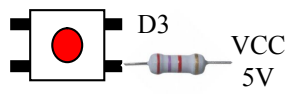
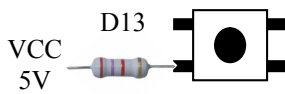
- 1 D10
- 2 D11
- 3 D12
- 4 D6
- 5 D7
- 6 D8
- 7 d9

Câblage LDC I2C

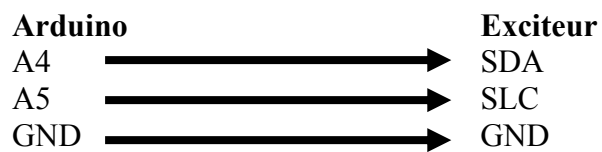


- 1 GND
- 2 VCC 5V
- 3 A4
- 4 A5

Câblage Boutons poussoir



Câblage vers Exciteur



Programme

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
#include <EEPROM.h> /* Clavier est un programme qui affiche la touche activée d'un clavier
matriciel dans le moniteur série */
#include <Keypad.h> // appel de la bibliothèque
#define L1 6 // affectation des broches du clavier
#define L2 7
#define L3 8
#define L4 9
#define C1 10
#define C2 11
#define C3 12
int touche;

const byte lignes = 4; // 4 lignes
const byte colonnes = 3; // 3 colonnes

char code[lignes][colonnes] = { // code des touches
  {'1', '2', '3'},
  {'4', '5', '6'},
  {'7', '8', '9'},
  {'*', '0', '#'}
};

byte broches_lignes[lignes] = {L1, L2, L3, L4}; // connexion des lignes
byte broches_colonnes[colonnes] = {C1, C2, C3}; // connexion des colonnes

Keypad clavier = Keypad( makeKeymap(code), broches_lignes, broches_colonnes, lignes,
colonnes); // création de l'objet clavier

#define Plus 13 // Bouton frequence + NOIR
#define Moins 3 // Bouton frequence - ROUGE
#define Led 11 // Led
int msb;
int msb2;
int lsb;
unsigned int freq ;
int pas ;
int r = 0;
float fr ; // Fréquence en cours
float pasfr ; / Pas de la fréquence
int pompe = 174; // Pompe de charge pour le TSA5511
int adress = 92; // Adresse de TSA5511
int car;
int para;
int Bmoins; int Bplus; int POS_XX;
```

```

float fmaxi; float fmini;
int flagP = 0; int flagM = 0;
unsigned long frx;
int plsb; int pmsb; int pmsb2; int cont;
unsigned long pa_2; int del = 100; int tt = 5;
String FR_clav = "";
String carn = "";
int mul [4] = {1, 10, 100, 1000};
int nb;
float FRx;
float FR;
int a;
//***** Fin de déclarations
*****

void setup()
{
  // CONFIGURATIONS /

  Serial.begin(9600); //Vitesse du port serie.
  Wire.begin();
  lcd.begin();
  lcd.backlight();
  Wire.begin(0x61);
  pinMode(Plus, INPUT);
  pinMode(Moin, INPUT);
  pinMode(Led, OUTPUT);
  lcd.clear(); // Efface le LCD
  lcd.setCursor(0, 0); // Met le curseur du LCD à la première ligne à gauche
  //zero_mem(); // Pour debug met toute les EEPROM à 0
  Lect_pasfr(); // Lecture du pas depuis l'EEPROM
  lecture_EPRM (); Lit les différents paramètres de l'EEPROM
  lect_fr_mem(); Lit la fréquence en cours sauvee sur l'EEprom pour redémarrer sur la
  fréquence
  fr = FRx;
}
//*****
//          DEBUT DU PROGRAMME          *
//*****

void loop()
{
  rr:
  if ( fr == 0 ) { //Si fr = 0 première mise en service ont va au paramètres

    Parametres();
  }

  calcul(fr); // recalcule fréquence
  imp_fr();   // Imprime résultat
  FR_I2c(); // Envoie de la fréquence au TSA551 en I2C

```

```

r = r + 1;
//***** Saisie Touche (#) *****
encore:
if (r == 1) {
  BP(); // Get la saisie au clavier
  // *****test si Plus *****
  if (Bplus == 1) { // Si la touche # est actionnée on augmente d'un PAS la fréquence
    digitalWrite(Led, HIGH);
    fr = fr + pasfr; // + 1 PAS
    if (fr <= fmaxi) { // Contrôle si on ne dépasse pas la fréquence maxi
      calcul(fr); // c'est ok alors on calcul la nouvelle fréquence
      imp_fr(); // Imprime résultat
      FR_I2c(); // Envoie de la fréquence au TSA551 en I2C
    }
    else { // On dépasse la fréquence maxi
      fr = fr - pasfr; calcul(fr); // Alors on remet la fréquence ou elle était
      Serial.println("Limite de fréquence haute atteinte"); // On signal à l'opérateur que la
fréquence maxi est dépassée
    }
  }
}
//***** Saisie Touche (*)*****
sortp:
// *****test si Moin *****
if (Bmoin == 1) { Si la touche * est actionnée on diminue d'un PAS la fréquence
  digitalWrite(Led, HIGH);
  fr = fr - pasfr;
  if (fr >= fmini) { Contrôle si on ne dépasse pas la fréquence mini
    calcul(fr);
    imp_fr(); // Imprime résultat
    FR_I2c(); // Envoie de la fréquence au TSA551 en I2C
    flagP = 0;
  }
  else { On dépasse la fréquence mini
    fr = fr + pasfr; calcul(fr); // Alors on remet la fréquence ou elle était
    Serial.println("Limite de fréquence basse atteinte");// On signal à l'opérateur que la
fréquence maxi est dépassée
  }
}
digitalWrite(Led, LOW);

//***** Saisie bouton Fréquence
*****
if (digitalRead (Plus) == 0 && digitalRead (Moin) == 1) { // On surveille si le bouton
NOIR est actionné
er:
  Entrez_FR(); // Si oui on va la la routine de saisie de la nouvelle fréquence
  //***** Contrôle si hors limite de fréquence
*****

```

```

    if (fr < fmini || fr > fmaxi) { // En sortie de la routine on Contrôle si on ne dépasse pas Mini
Maxi
        Serial.println("Erreur de saisie "); // Si oui on le signale puis on retourne à la saisie de
fréquence
        goto er;
    }

//*****
*****

    calcul(fr); // Si la saisie est bonne on calcule la nouvelle fréquence
    imp_fr(); // Imprime résultat
    FR_I2c(); // Envoie de la fréquence au TSA551 en I2C
    }
//***** Saisie bouton Paramètre
*****

    BPpara();
    if ( para == 1) {
        Parametres(); // voir si boutons rouge est appuyé pour aller aux Paramètre */
att:
        if (digitalRead (Plus) == 0 || digitalRead (Moin) == 0) { // si un des deux bouton rouge ou
noir est appuyé
                                                    // alors on attends

            goto att;
        }
    }
//*****
*****

    // lecture_EPROM ();
    goto encore;
}
ret:
    digitalWrite(Led, LOW);
    //delay(10);
}
//*****
//          FIN PROGRAMME          *
//*****

//***** Emplacement des sous programme*****

//*****
//*          Choix POS          *
//*****

void Parametres() {
    para = 0;

par:
    lcd.clear();
    lcd.setCursor(0, 0); lcd.print("POS 25 "); lcd.print(" 1");
    lcd.setCursor(0, 1); lcd.print("Fr 15 - 25 MHz");
}

```

```

Serial.print("POS 25 "); Serial.println("      1");
Serial.println("Fr 15 - 25 MHz");
pasn:
BP(); {
  if (Bplus == 1) {
    goto par2;
  }
  if (Bmoin == 1) {
    fr = 20;
    POS_XX = 1; EEPROM.write (2, POS_XX);
    fmini = 15; fmaxi = 25; Ecrire_limite (); calcul(fr); // calcul_fr();
    pas = 1; EEPROM.write (3, pas); pasfr = 0.0125; Ecrit_pasfr();
    return;
  }
}
goto pasn;

//*****
par2:
lcd.clear();
lcd.setCursor(0, 0); lcd.print("POS 50 "); lcd.print("      2");
lcd.setCursor(0, 1); lcd.print("Fr 25 - 50 MHz");
Serial.print("POS 25 "); Serial.println("      2");
Serial.println("Fr 25 - 50 MHz");
pasn2:
BP(); {
  if (Bplus == 1) {
    goto par3;
  }
  if (Bmoin == 1) {
    fr = 37;
    POS_XX = 2; EEPROM.write (2, POS_XX);
    fmini = 25; fmaxi = 50; Ecrire_limite (); calcul(fr); // calcul_fr();
    pas = 1; EEPROM.write (3, pas); pasfr = 0.0125; Ecrit_pasfr();
    return;
  }
}
goto pasn2;

//*****
par3:
lcd.clear();
lcd.setCursor(0, 0); lcd.print("POS 75 "); lcd.print("      3");
lcd.setCursor(0, 1); lcd.print("Fr 37 - 75 MHz");

Serial.print("POS 75 "); Serial.println("      3");
Serial.println("Fr 37 - 75 MHz");
pasn3:
BP(); {
  if (Bplus == 1) {

```



```

    goto par33;
}
if (Bmoin == 1) {
    fr = 56;
    POS_XX = 3; EEPROM.write (2, POS_XX);
    fmini = 37; fmaxi = 75; Ecriture_limite (); calcul(fr); // calcul_fr();
    pas = 1; EEPROM.write (3, pas); pasfr = 0.0125; Ecrit_pasfr();
    return;
}
}
goto pasn3;
//*****
par33:
    lcd.clear();
    lcd.setCursor(0, 0); lcd.print("POS 100 "); lcd.print("    4");
    lcd.setCursor(0, 1); lcd.print("Fr 50 - 100 MHz");

    Serial.print("POS 100 "); Serial.println("    4");
    Serial.println("Fr 50 - 100 MHz");
pasn3:
    BP(); {
        if (Bplus == 1) {
            goto par4;
        }
        if (Bmoin == 1) {
            fr = 75;
            POS_XX = 4; EEPROM.write (2, POS_XX);
            fmini = 50; fmaxi = 100; Ecriture_limite (); calcul(fr); // calcul_fr();
            pas = 1; EEPROM.write (3, pas); pasfr = 0.0125; Ecrit_pasfr();
            return;
        }
    }
    goto pasn3;
//*****
par4:
    lcd.clear();
    lcd.setCursor(0, 0); lcd.print("POS 150 "); lcd.print("    5");
    lcd.setCursor(0, 1); lcd.print("Fr 75 - 150 MHz");

    Serial.print("POS 150 "); Serial.println("    5");
    Serial.println("Fr 75 - 150 MHz");
pasn4:
    BP(); {
        if (Bplus == 1) {
            goto par5;
        }
        if (Bmoin == 1) {
            fr = 112;
            POS_XX = 5; EEPROM.write (2, POS_XX);
            fmini = 75; fmaxi = 150; Ecriture_limite (); calcul(fr); // calcul_fr();

```

```

    pas = 1; EEPROM.write (3, pas); pasfr = 0.0125; Ecrit_pasfr();
    return;
  }
}
goto pasn4;
//*****
par5:
  lcd.clear();
  lcd.setCursor(0, 0); lcd.print("POS 200 "); lcd.print("    6");
  lcd.setCursor(0, 1); lcd.print("Fr 100 - 200 MHz");

  Serial.print("POS 200 "); Serial.println("    6");
  Serial.println("Fr 100 - 200 MHz");
parn5:
  BP(); {
    if (Bplus == 1) {
      goto par6;
    }
    if (Bmoin == 1) {
      fr = 150;
      POS_XX = 6; EEPROM.write (2, POS_XX);
      fmini = 100; fmaxi = 200; Ecriture_limite (); calcul(fr); // calcul_fr();
      pas = 1; EEPROM.write (3, pas); pasfr = 0.0125; Ecrit_pasfr();
      return;
    }
  }
  goto pasn5;
//*****
par6:
  lcd.clear();
  lcd.setCursor(0, 0); lcd.print("POS 300 "); lcd.print("    7");
  lcd.setCursor(0, 1); lcd.print("Fr 150 - 280 MHz");

  Serial.print("POS 300 "); Serial.println("    7");
  Serial.println("Fr 150 - 280 MHz");

parn6:
  BP(); {
    if (Bplus == 1) {
      goto par7;
    }
    if (Bmoin == 1) {
      fr = 205;
      POS_XX = 7; EEPROM.write (2, POS_XX);
      fmini = 150; fmaxi = 300; Ecriture_limite (); calcul(fr); // calcul_fr();
      pas = 1; EEPROM.write (3, pas); pasfr = 0.0125; Ecrit_pasfr();
      return;
    }
  }
}

```

```

goto pasn6;
//*****
par7:
  lcd.clear();
  lcd.setCursor(0, 0); lcd.print("POS 400 "); lcd.print("      8");
  lcd.setCursor(0, 1); lcd.print("Fr 200 - 380 MHz");

  Serial.print("POS 400 "); Serial.println("      8");
  Serial.println("Fr 200 - 380 MHz");

pasn7:
  BP(); {
    if (Bplus == 1) {
      goto par8;
    }
    if (Bmoin == 1) {
      fr = 280;
      POS_XX = 8; EEPROM.write (2, POS_XX);
      fmini = 200; fmaxi = 380; Ecriture_limite (); calcul(fr); // calcul_fr();
      pas = 1; EEPROM.write (3, pas); pasfr = 0.0125; Ecrit_pasfr();
      return;
    }
  }
  goto pasn7;
//*****
par8:
  lcd.clear();
  lcd.setCursor(0, 0); lcd.print("POS 535 "); lcd.print("      9");
  lcd.setCursor(0, 1); lcd.print("Fr 300 - 525 MHz");

  Serial.print("POS 535 "); Serial.println("      9");
  Serial.println("Fr 300 - 525 MHz");

pasn8:
; BP(); {
  if (Bplus == 1) {
    goto par9;
  }
  if (Bmoin == 1) {
    fr = 375;
    POS_XX = 9; EEPROM.write (2, POS_XX);
    fmini = 300; fmaxi = 525; Ecriture_limite (); calcul(fr); // calcul_fr();
    pas = 2; EEPROM.write (3, pas); pasfr = 0.025; Ecrit_pasfr();
    return;
  }
}
goto pasn8;
//*****

```

```

par9:
  lcd.clear();
  lcd.setCursor(0, 0); lcd.print("POS 765 "); lcd.print("    10");
  lcd.setCursor(0, 1); lcd.print("Fr 485 - 765 MHz");

```

```

  Serial.print("POS 765 "); Serial.println("    10");
  Serial.println("Fr 485 - 765 MHz");

```

```

pasn9:
  BP(); {
    if (Bplus == 1) {
      goto par;
    }
    if (Bmoin == 1) {
      fr = 522.00;
      POS_XX = 10; EEPROM.write (2, POS_XX);
      fmini = 485; fmaxi = 765; Ecriture_limite (); calcul(fr); // calcul_fr();
      pas = 2; EEPROM.write (3, pas); pasfr = 0.025; Ecrit_pasfr();
      return;
    }
  }

```

```

  goto pasn9;
}

```

```

//*****
//*                               FIN Choix POS                               *
//*****

```

```

//***** data en Eprom *****
//EEPROM(0)=fr, EEPROM(1)= POS EEPROM(2)= POS_XX, EEPROM(3)= pas
//EEPROM(4)= x1, EEPROM(5)= x5, EEPROM(6) = x6, EEPROM(7) = pmsb
// EEPROM(7) = pmsb , EEPROM(8) = plsb Poids fort et poids faible du PAS fr
//EEPROM(11)=fmini, EEPROM(12) = fmaxi
//*****

```

```

//      lecture_EPROM POS PAS LIMITES en cours
//*****

```

```

void lecture_EPROM ()
{
  POS_XX = EEPROM.read(2);
  pas = EEPROM.read(3);
  fmaxi = EEPROM.read(11); fmini = EEPROM.read(12);
}

```

```

//*****
//*      Ecrit PAS fréquence en memoire  Sur 2 octets                        *
//*****

```

```

void Ecrit_pasfr()
{
  pa_2 = pasfr * 10000;
  pmsb = pa_2 / 256; EEPROM.write(7, pmsb);
  pmsb2 = 256 * pmsb;

```

```

    plsb = pa_2 - pmsb2; EEPROM.write (8, plsb);
}
//*****
//*          Lecture PAS fréquence en memoire Sur 2 octets          *
//*****
void Lect_pasfr()
{
    pmsb = EEPROM.read(7); plsb = EEPROM.read(8);
    pmsb = pmsb * 256;
    pa_2 = pmsb + plsb;
    pasfr = pa_2 * .0001;
}
//*****
//          imp_fr pour debug          *
//*****
void imp_fr2()
{
    Serial.println("-----");
    Serial.print("FR = "); Serial.println(fr, 4);
    Serial.print("FREQ = "); Serial.println(freq);
    Serial.print("pas = "); Serial.println(pas);
    Serial.print("PAS FR = "); Serial.println(pasfr, 4);
    Serial.print("POS = "); Serial.println(POS_XX);
    Serial.print("HEXA "); Serial.print(msb, HEX); Serial.print(" "); Serial.print(lsb, HEX);
    Serial.println(" ");
    Serial.println("-----");
}

//*****
//          Impression LCD et Moniteur          *
//*****

void imp_fr()
{
    lcd.setCursor(0, 0);
    lcd.print("fr : "); lcd.print(fr, 4); lcd.print(" MHz");
    lcd.setCursor(0, 1);
    lcd.print("HEXA "); lcd.print(msb, HEX); lcd.print(" ");
    lcd.print(lsb, HEX); lcd.print(" ");
    Serial.print ("Fréquence : "); Serial.print (fr, 4); Serial.print (" MHz ");
    Serial.print ("Code Hexa "); Serial.print(msb, HEX); Serial.println(lsb, HEX);
}
//*****
void receiveEvent(int bytes)
{
    car = Wire.read(); // read one character from the I2C
}
//*****
//          à zero des memoires DEBUG          *

```

```

//*****
void zero_mem() {
  EEPROM.write (0, 0);
  EEPROM.write (1, 0);
  EEPROM.write (2, 0);
  EEPROM.write (3, 0);
  EEPROM.write (4, 0);
  EEPROM.write (5, 0);
  EEPROM.write (6, 0);
  EEPROM.write (7, 0);
  EEPROM.write (8, 0);
  EEPROM.write (9, 0);
  EEPROM.write (10, 0);
  EEPROM.write (11, 0);
  EEPROM.write (12, 0);
}
//*****
//          Gestion des boutons          *
//*****
void BP()
{ Bplus = 0; Bmoin = 0; lect_clav();
  if (touche == 35) //+++++++Touche (#)+++++++
  {
    Bplus = 1; digitalWrite(Led, HIGH); return;
  }

  if (touche == 42) //+++++++Touche (*)+++++++
  {
    Bmoin = 1; digitalWrite(Led, HIGH); return;
  }
  digitalWrite(Led, LOW); return;
}
//*****
void BPpara()
{

  if (digitalRead (Moin) == 0 )
  {
    para = 1; return;
  }
  return;
}
//*****
//*          Ecriture Limite des fréquences en EEPROM          *
//*****
void Ecriture_limite ()
{
  EEPROM.write (11, fmaxi); EEPROM.write (12, fmini);
}
//*****

```

```

void lect_mem ()
{
  for ( int i = 0; i < 13; i++) {
    Serial.print (i); Serial.print (" ");
    Serial.println( EEPROM.read(i));
  }
}
//++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
void lect_clav() {
  touche = clavier.getKey();      // acquisition de la touche

  if (touche != NO_KEY)           // si appui sur une touche
  {

  }

}
//-----
//      Entré de la fréquence au clavier
//-----
void Entrez_FR()

{ lcd.clear();
  Serial.print ("Entrez une fréquence : ");
  lcd.setCursor(0, 0);lcd.print("Entrez une fr");
er:
  nb = 0; carn = ""; fr = 0; int b;
  { //nb = 0; carn = ""; fr = 0; int b;
clav:
  touche = clavier.getKey();      // acquisition de la touche

  if (touche != NO_KEY)           // si appui sur une touche
  {
    if (touche > 47 && touche < 58) {
      touche = touche - 47;
    }
  }
  if (touche == 0 || touche == 42)
  {
    goto clav;
  }
  if (touche == 35)
  {
    for (int i = nb - 1; i > -1; i = i - 1)
    {
      a = carn. charAt (i); a = a - 48;
      fr = fr + (a * mul[b]);
      b = b + 1;
    }
    Serial.println();
  }
}

```

```

    return;
}
//lcd.setCursor(0, 1);
if (nb < 3) {
    carn = carn + (touche - 1 ); Serial.print (touche - 1);
    lcd.setCursor(nb, 1);
    lcd.print (touche - 1);
    nb = nb + 1;
    goto clav;
}
goto clav;
}
}
//-----
//      Calcul du code Hexa à partir de la fréquence
//-----
void calcul(float a)
{
    float Mixeur;
    if (POS_XX < 9) {
        Mixeur = 0.8 / 512;
    }
    else {
        Mixeur = 1.6 / 512;
    }

    FR = a / 8;
    freq = FR / Mixeur;
    //Serial.println(freq);
    calculHEX(freq);// Serial.print(msb, HEX); Serial.println(lsb, HEX);
    EEPROM.write(0, msb); EEPROM.write(1, lsb);
}
//-----

int calculHEX(unsigned int a ) // Calcul le fréquence en HEXA msb et lsb
{
    int msb2;
    msb = freq / 256;
    msb2 = msb * 256;
    lsb = freq - msb2;
    //Serial.print(msb, HEX); Serial.println(lsb, HEX);
    EEPROM.write(0, msb); EEPROM.write(1, lsb);
    return ;
}
//-----
//      Calcul de la fréquence à partir du code Hexa
//-----
void calcul_fr()
{

```



```

float Mixeur;
if (POS_XX < 9) {
  Mixeur = 0.8 / 512;
}
else {
  Mixeur = 1.6 / 512;
}
FR = freq * Mixeur;
FRx = FR * 8;
}
//-----
void lect_fr_mem()
{
  msb = EEPROM.read(0); lsb = EEPROM.read(1);
  msb = msb * 256;
  freq = msb + lsb; // Serial.println(freq);
  calcul_fr();
}
//-----
// Envoie de la fréquence au TSA551 en I2C
//-----
void FR_I2c()
{
  Wire.beginTransmission (0x60); // transmettre au dispositif # 96
  Wire.write (msb);
  Wire.write (lsb);
  Wire.write (pompe);
  Wire.onReceive(receiveEvent);
  Wire.endTransmission ();
  delay(del);
}

```

Schéma d'un émetteur 10 MW

