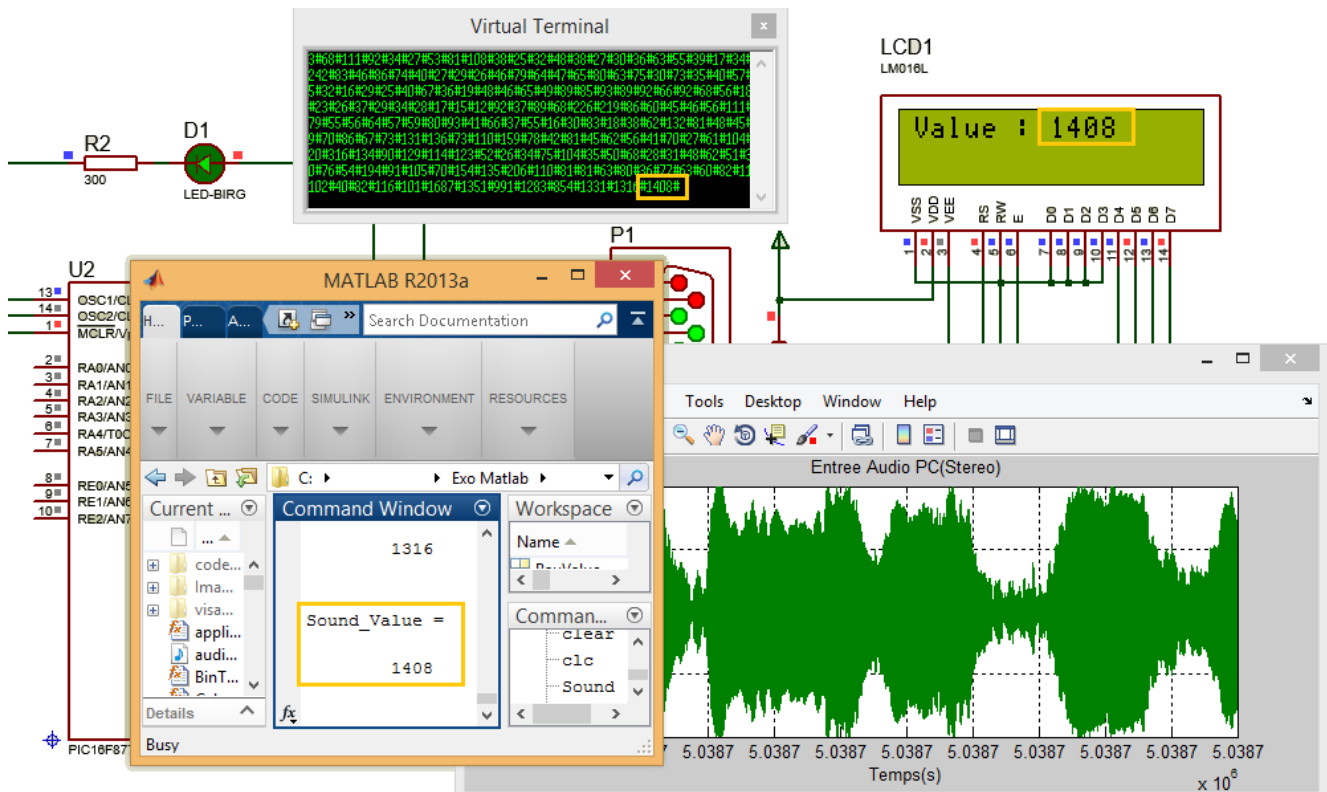


# Projet électronique : Capteur de son logiciel à base du PIC16F877 + Matlab + ISIS + MikroC

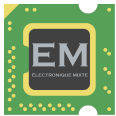


Beaucoup d'entre vous posent des questions : Comment puis-je interfacier entre Matlab et **ISIS** ? Comment lire ou écrire dans la **carte** son de mon ordinateur (microphone ou haute parleur) par Matlab ? Comment afficher les données reçues en temps réel de la carte son ? Comment créer un port COM virtuel ou des liaisons virtuels entre mes outils de développement (Matlab & ISIS) ?

Pas de problème! Ce projet vous offre les outils nécessaires (code **MikroC**, script Matlab, montage et **logiciel** gratuit pour ports virtuels) pour que vous puissiez développer vos applications vous même et booster votre esprit de créativité.

A la fin de cet article, vous trouverez un fichier RAR du projet complet à télécharger gratuitement !

N'hésitez pas de poster un commentaire, une critique ou un encouragement de votre part ☐



## Projet électronique : Capteur de son logiciel à base du PIC16F877 + Matlab + ISIS + MikroC

En peut résumé le projet en quatre étapes :

1. Lecture de la carte son par matlab
2. Calcul d'une valeur à partir des échantillons acquis, cette dernière correspond à l'intensité du son
3. Transmettre l'intensité du son au [microcontrôleur](#) via la liaison UART
4. Réception de la valeur par le microcontrôleur par la liaison UART, afficher la valeur sur [LCD](#) et allumer la [LED](#) en cas de dépassement de [seuil](#).

Les paramètres de lecture de la carte son sont définis par le script matlab :

- Période de lecture en seconde  $\gg 100\text{ms}$  (100ms est le temps nécessaire à matlab pour faire l'acquisition et l'enregistrement des échantillons), d'où l'importance de choisir une période largement supérieure à 100ms (T en seconde).
- Fréquence d'échantillonnage  $F_s$  (HZ) : Ce paramètre dépend de votre carte son (8000, 11025, 22050, 44100, 48000 ou 96000 Hz)
- Résolution [ADC](#) de la carte son de votre ordinateur en bits : 12,16 ou 24 bits (nBits)
- Nombre de canaux audio : 1 ou 2, 1 pour mono et 2 pour stéréo.

Note : Si vous ne connaissez pas les paramètres de votre carte son (résolution, fréquence), vous pouvez choisir les valeurs max et matlab fera le travail à votre place ( $F_s=96000$  et 24 bits de résolution) !

Les paramètres de la liaison série définis par le script matlab :

- Vitesse de transmission : 110 baud, 300 baud, 1 200 baud, 2 400 baud, 4 800 baud, 9 600 baud, 19 200 baud, 38 400 baud, 57 600 baud ou 115 200 baud
- Nombre de bits : 7 ou 8 bits

Note : Vous pouvez apporter des modifications sur la configuration de



la liaison série ( vitesse, nombre de bits), mais les paramètres doivent être les mêmes dans le script matlab, code mikroC et le circuit d'émulation UART sur ISIS.

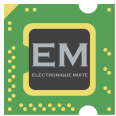
Les paramètres principales du projet sur MikroC :

- Seuil de déclenchement : Ce paramètre définit la valeur minimale de l'intensité du son qui induit l'allumage de la LED. La valeur du seuil dépend de la sensibilité du capteur. Si le seuil est trop faible, le capteur peut détecter des valeurs faibles du son
- Vitesse de réception en baud, doit être la même de celle configurée sur le script matlab.

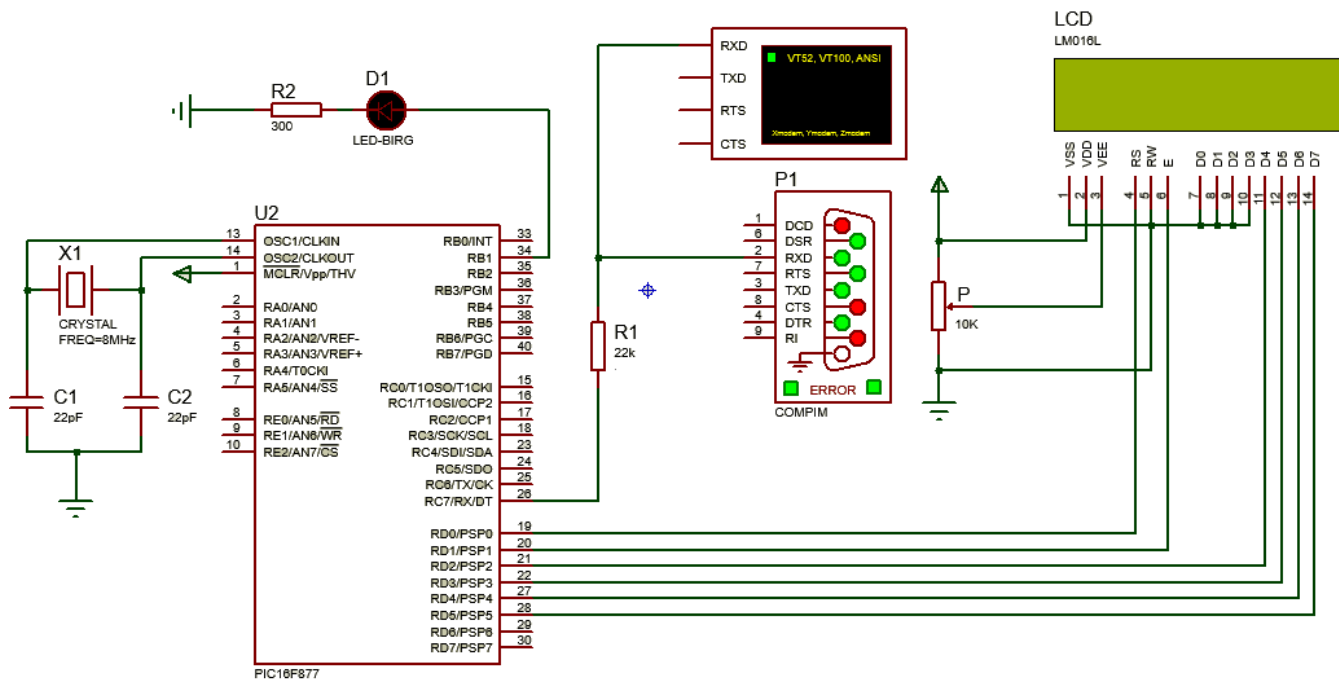
Comment calculer l'intensité du son ?

Cette mesure correspond à la variance de la tension absolue acquise par la carte son pendant une période  $T$ . C'est une mesure classique pour évaluer le changement de l'intensité du son. Vous pouvez utiliser d'autres mesures statistiques ou effectuez des post-traitements (débruitage, filtrage numérique,...) sur le son avant de calculer la valeur de l'intensité.

## Schéma électronique du projet



# Projet électronique : Capteur de son logiciel à base du PIC16F877 + Matlab + ISIS + MikroC

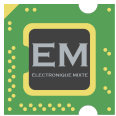


Mode de fonctionnement :

- Télécharger le fichier RAR du projet (en bas de l'article)
- Installer le logiciel de création des ports COM virtuel
- Allez dans périphérique==> Créer puis sélectionnez un périphérique de type "pair" et suivez les instructions
- Click sur Démarrer l'émulation
- Exécuter le script matlab
- Lancer le fichier ISIS et click sur lancer la simulation

Important :

- Si vous arrêtez la simulation sur ISIS vous devez relancer le fichier ISIS pour que la liaison UART sera prise en considération par l'émulateur
- Le script matlab tourne en boucle ferme, pour arrêter la simulation

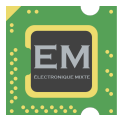


## Projet électronique : Capteur de son logiciel à base du PIC16F877 + Matlab + ISIS + MikroC

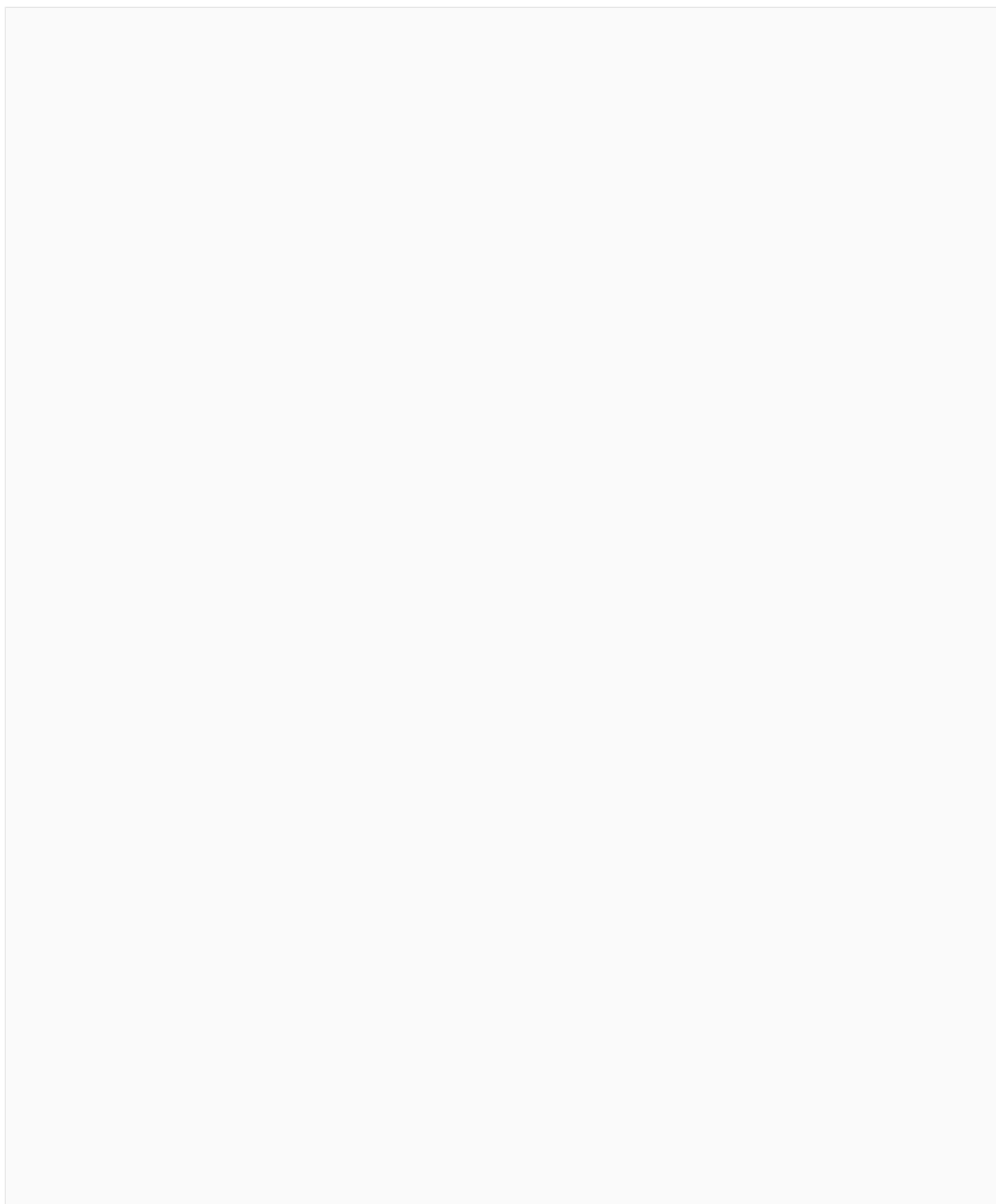
vous tapez sur l'espace workspace du matlab " Ctrl+C" puis vous exécutez le code en bas du script pour fermer le port série.

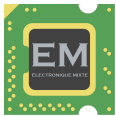
- Vous pouvez ajuster les paramètres de votre carte son pour activer le microphone ou tout simplement lire le son de votre ordinateur (musique, vidéos, ...).

### Code MicroC du projet :



Projet électronique : Capteur de son logiciel à base du PIC16F877 +  
Matlab + ISIS + MikroC



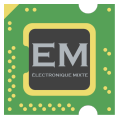


Projet électronique : Capteur de son logiciel à base du PIC16F877 +  
Matlab + ISIS + MikroC

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%                               Capteur de son logiciel                               %%  
%%                               Microphone PC & PIC 16F sous ISIS & Matlab           %%  
%%                               06/04/2015                                         %%  
%%                               https://www.electronique-mixte.fr                               %%  
%%                               https://www.facebook.com/ElectroniqueMixte                       %%  
%%                               %%  
%%                               %%  
%%                               Possibilités du projet                               %%  
%%                               - Oscilloscope numérique (Fs=96KHz, 16 bits )%%  
%%                               - Capteur de son ultra sensible Gratuit !           %%  
%%                               - Surveillance temps réel                           %%  
%%                               - Reconnaissance de la parole                       %%  
%%                               - Acquisition et traitement du signal              %%  
%%                               - ...                                               %%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
*/
```

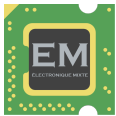
```
// Fonction conversion d'un char en int(0-9 ou -1)  
int CharToInt(unsigned char c)  
{  
    int n ;  
    switch (c)  
    {  
        case '0' :  
            n=0;  
            break;  
        case '1' :  
            n=1;  
            break;
```



Projet électronique : Capteur de son logiciel à base du PIC16F877 +  
Matlab + ISIS + MikroC

```
    case '2' :  
        n=2;  
        break;  
    case '3' :  
        n=3;  
        break;  
    case '4' :  
        n=4;  
        break;  
    case '5' :  
        n=5;  
        break;  
    case '6' :  
        n=6;  
        break;  
    case '7' :  
        n=7;  
        break;  
    case '8' :  
        n=8;  
        break;  
    case '9' :  
        n=9;  
        break;  
    default :  
        n=-1;  
}  
return n;  
}  
  
// Fonction calcul de x^y entière  
int powint(int base, int expon)
```





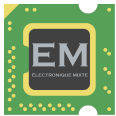
```
{
    int val =1;
    int i ;
    if (expon==0) val = 1;
    else
    {
        for(i=0;i<expon;i++)
            val=val*base;
    }

    return val;
}

// Connexions LCD
sbit LCD_RS at RD0_bit;
sbit LCD_EN at RD1_bit;
sbit LCD_D4 at RD2_bit;
sbit LCD_D5 at RD3_bit;
sbit LCD_D6 at RD4_bit;
sbit LCD_D7 at RD5_bit;

sbit LCD_RS_Direction at TRISD0_bit;
sbit LCD_EN_Direction at TRISD1_bit;
sbit LCD_D4_Direction at TRISD2_bit;
sbit LCD_D5_Direction at TRISD3_bit;
sbit LCD_D6_Direction at TRISD4_bit;
sbit LCD_D7_Direction at TRISD5_bit;

const int                                UARRT1_Baud =2400;                //Vitesse UART
1
```



Projet électronique : Capteur de son logiciel à base du PIC16F877 +  
Matlab + ISIS + MikroC

```
const int          Taille = 12;

unsigned char      SoundStr[Taille];
unsigned char      SoundStrTmp[Taille];
unsigned char      LCD_Clear[]="      ";          // Text pour E
ffacer LCD

int                Value=0;
const int          Seuil=100;                    // Seuil de d
éclenchement
int                SoundInt=0;
int                i,j=0;

void main()
{
    // Déclaration des fonctions
    int CharToInt(unsigned char );
    int powint(int , int );

    // Initialisation des périphériques
    Lcd_Init();
    Lcd_Cmd(_LCD_CLEAR);
    Lcd_Cmd(_LCD_CURSOR_OFF);

    UART1_Init(UARRT1_Baud);
    delay_ms(100);

    ADCON1 = 0x07;                                // PORTS en mo
de digital
    TRISA = 0xFF;                                // PORTA en en
trée
    TRISC = 0xFF;                                // PORTC en en
```



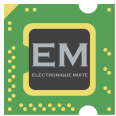
```
tée
    TRISB = 0x00; // PORTB en so
rtie
    TRISD = 0x00; // PORTD en so
rtie

    PORTB = 0x00;

do
{
    if (UART1_Data_Ready() == 1)
    {
        UART1_Read_Text(SoundStr, "#", Taille); // Lire le tex
t jusqu'à trouvé le text "#"
        Lcd_Out(1,8,LCD_Clear); // Effacer LCD
        Lcd_Out(1,8,SoundStr); // Afficher le
text sur LCD
        Lcd_Out(1,1,"Value :");

        /*Extraction de la valeur reçue (conversion String to int)

        Format de SoundStr (taille 12):
                1- .C1..... C1=('0'-'
'9') (.)='\0' Exp: SoundStr=" 1 "
                2-
.C1C2..... Exp: SoundStr=" 12 ";
                3-
.C1C2C3..... Exp: SoundStr=" 123 ";
                4-
.C1C2C3C4C5C6C7.... Exp: SoundStr=" 1234567 ";
```



```
Exemple : si SoundStr=" 478 " Après calcul SoundInt
=478!! */

SoundInt=0;
j=0;

for(i=Taille;i>0;i--)
{
    Value=CharToInt(SoundStr[i]);

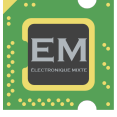
    if(Value<10 && Value>=0)
    {
        SoundInt = SoundInt +(CharToInt(SoundStr[i])*powin
t(10,j));

        j++;
    }
}

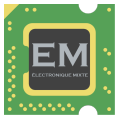
//Alimenter La LED si l'in dépasse le Seuil
if( SoundInt> Seuil) PORTB = 0x02;
else PORTB = 0x00;

delay_ms(300);

}
} while(1);
}
```

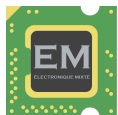


Projet électronique : Capteur de son logiciel à base du PIC16F877 +  
Matlab + ISIS + MikroC

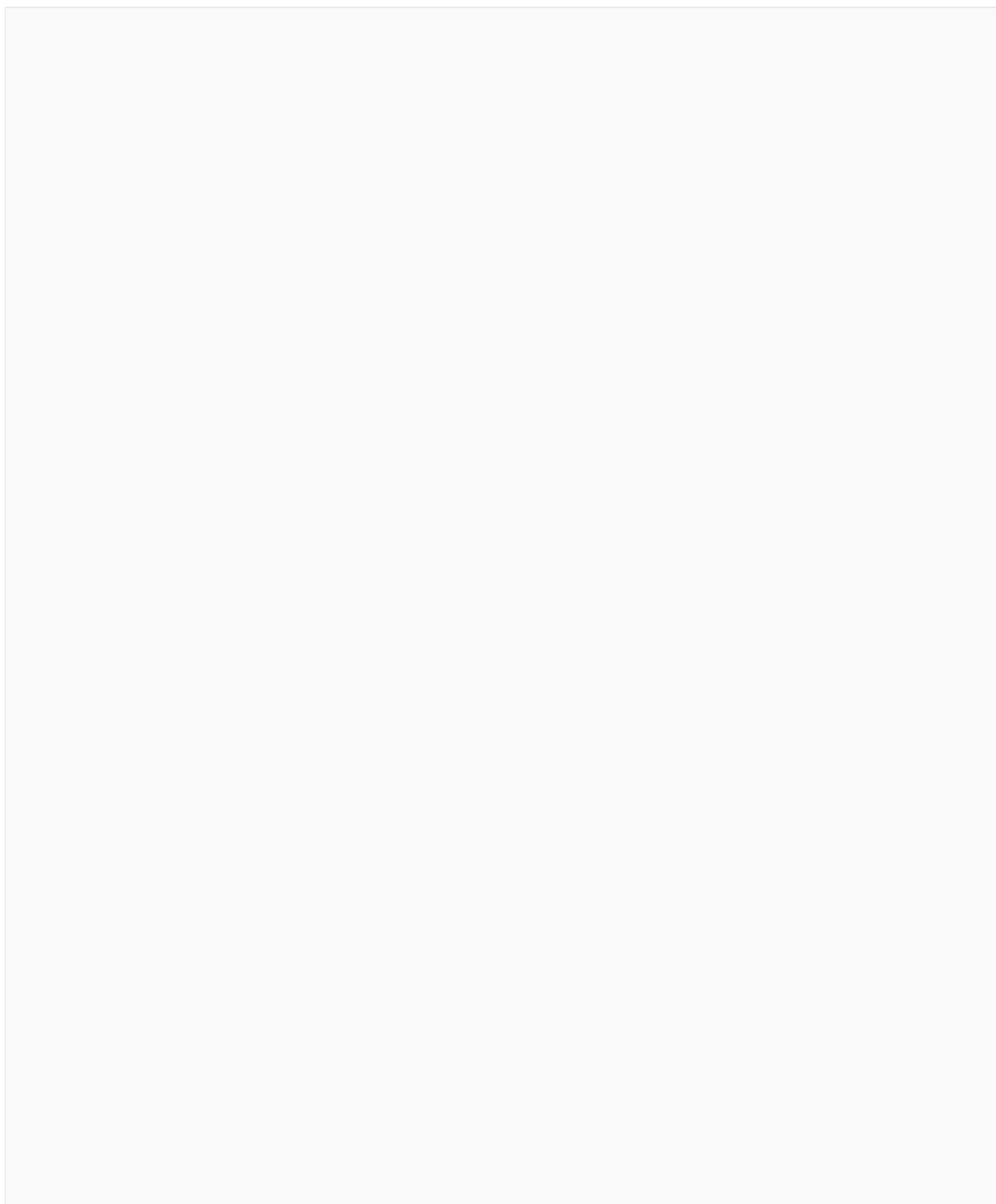


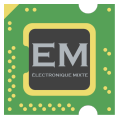
Projet électronique : Capteur de son logiciel à base du PIC16F877 +  
Matlab + ISIS + MikroC

Code Matlab :



Projet électronique : Capteur de son logiciel à base du PIC16F877 +  
Matlab + ISIS + MikroC





Projet électronique : Capteur de son logiciel à base du PIC16F877 +  
Matlab + ISIS + MikroC

```
clear all ;
close all ;
clc ;

% Paramètres du capteur du son (microphone)

Fs=48000;           % 8000, 11025, 22050, 44100, 48000 ou 96000 H
Z.
Ts=1/Fs;
nBits=16;          % Résolution ADC carte son (12,16 ou 24 bits)
nChannels=2;       % Nombre de canaux 1 (mono) 2 (stereo)
T=1;              % Période d'acquisition
i=0;
t=0:Ts:T-Ts;      % Vecteur temps pour l'affichage temporelle

% Création d'un objet Audio
recObj = audiorecorder(Fs,nBits,nChannels);

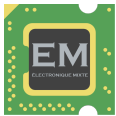
% Paramètres de la liaison série (COM)
delete(instrfind);
pause(0.1);

BaudValue=2400;    % Vitesse
NumBits=8;        % Nombre de bits

% Création d'un objet Serial Port
s = serial('COM1','BaudRate',BaudValue,'DataBits', NumBits, 'Parity', '
none');

% Connexion du port
```





```
fopen(s);

while(1)

    % Début d'enregistrement
    recordblocking(recObj, T);

    % Stockage de l'audio dans la variable Y
    Y = getaudiodata(recObj);

    % Moyenne de la valeur abs des deux canaux(son capté)
    Y_abs=abs(Y);
    Y0 = (Y_abs(:,1) + Y_abs(:,2))/2;

    % Calcul de l'intensité du son (variance)
    Y_var=sqrt(var(Y0));
    Sound_Value=round(Y_var*5e3)

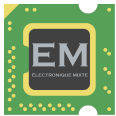
    % Conversion de la valeur flottante en chaîne de caractère
    StrValue =int2str(Sound_Value);

    % Transfert de la valeur
    fprintf(s,StrValue);

    % Transfert du caractère (fin de transmission)
    fprintf(s,'#');

    % Affichage du signal acquis

    % La base de temps
    Tmin = 1e-3;
```



```
Tmax = 10e-3;

% Tension
Vmin = -1e-4;
Vmax = 1e-1;

% Affichage temporelle
plot(t,Y);
grid on;
xlabel('Temps(s)');
ylabel('Tension(V)');
title('Entree Audio PC(Stereo)');

%     xlim([Tmin+i*T Tmax+i*T]);
%     ylim([Vmin Vmax]);

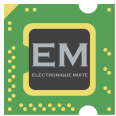
% Mise à jour de la période T
i=i+T;
t=t+i;
%     wavwrite(Y,Fs,'audio.wav'); % Enregistrement sous format v
av
end

% Destruction de l'objet (Port série)
% Note :
%     - Ctrl+C : Arrêter le programme
%     - Exécuter le code ci-dessus dans l'espace workspace
%     pour détruire l'objet Serial
```



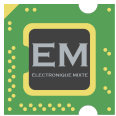
Projet électronique : Capteur de son logiciel à base du PIC16F877 +  
Matlab + ISIS + MikroC

```
fclose(instrfind);  
delete(s);
```



# Projet électronique : Capteur de son logiciel à base du PIC16F877 + Matlab + ISIS + MikroC

```
clear s;
```

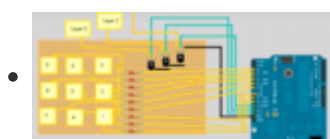


## Télécharger gratuitement le fichier du projet : Capteur de son logiciel à base du PIC16F877 + Matlab + ISIS + MikroC (Schéma ISIS + Code MikroC + Code matlab + Logiciel gratuit ) :

1. [Logiciel pour créer des ports COM virtuels](#)
2. [Montage ISIS](#)
3. [Programme MikroC](#)
4. [Script Matlab](#)

[Revenir au sommaire des projets électronique](#)

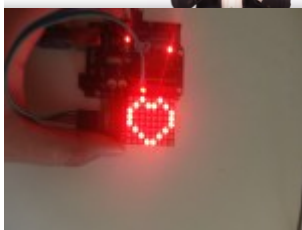
## Dernières réalisations



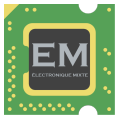
• [Projet LED Cube 3x3x3 avec Arduino](#)



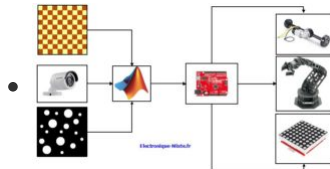
• [Quelle est la différence entre la créativité et l'innovation ?](#)



• [Projet électronique : Gestion d'une matrice des LED avec Arduino](#)



## Projet électronique : Capteur de son logiciel à base du PIC16F877 + Matlab + ISIS + MikroC



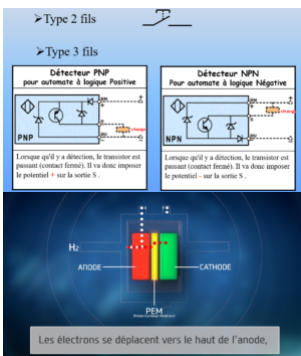
- Projets Matlab & Microcontrôleur #1: Commande graphique d'une matrice des LEDs avec Maltab et Arduino



- Projet électronique FPGA #10 : Commande factorielle d'un moteur à CC avec Arduino et FPGA



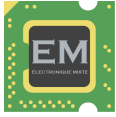
- Arduino #13 Gestion du temps - deux fonctions - partie 1
- Projet électronique FPGA 4 #2/3 : Capteur de distance ultrasonique à base du FPGA & Arduino



- GEMMA #3/3
- Comment ça marche une pile à Hydrogène (Pile à Combustible) ?

## Articles

- [Projet commander une carte Arduino avec Smartphone \(sans Modem\)](#)
- [Projet lampe tactile avec Arduino](#)



## Projet électronique : Capteur de son logiciel à base du PIC16F877 + Matlab + ISIS + MikroC

- [Arduino #18: les mémoires](#)
- [Arduino #4: la boucle for](#)
- [Matlab #10: Implémentation et Affichage d'une fonction 3D - Sinus 3D](#)
- [Matlab #6: Les 4 fonctions des Graphiques 2D](#)
- [Analyse de Fourier d'un circuit électronique - Filtre passe-bas RC - Partie 1/2](#)