

Sommaire

- [1 Objectifs](#)
- [2 A quoi sert un signal sinus cardinal ?](#)
- [3 Programme principal \(regarde la vidéo\)](#)
- [4 Fonction Génération du signal sin\(\)](#)
- [5 Fonction Génération du signal sinc\(\)](#)

Objectifs

1. Savoir générer un signal [sinus cardinal](#)
2. Savoir générer un bruit
3. Savoir générer un signal quelconque

A quoi sert un signal sinus cardinal ?

- Un signal sinus cardinal est un signal large bande contrairement au signal sinus (bande étroite). Il contient une bande plate de fréquences
- Avec un signal sinus cardinal on peut analyser le comportement d'un système linéaire dans une bande de fréquences contrairement au signal sinus il faut balayer la bande utile



fréquence par fréquence: En une seule acquisition, on peut déduire la fonction du transfert du système!

- sinus cardinal dans une bande de fréquence $[+fc, -fc]$ équivalent à une somme infinie de sinus ayant des fréquences variantes de $-fc$ à fc !
- Synthèse des filtres numériques dans le domaine de fourrier en traitement du signal et de l'[image](#). Un filtre passe bas parfait, passe haut, etc. est équivalent à la fonction porte dans le domaine du fourrier OU BIEN au sinus cardinal dans le domaine temporel!!!
- Le sinus cardinal est une fonction spéciale très importante en traitement du signal et d'images
- Etc.

Programme principal (regarde la vidéo)



Arduino #40: le signal et le bruit - le sinus cardinal





```
#define N      512          // N=2^8, 8 bits

#define pi     3.1415926535897932384626433832795

#define f0     1.00

#define eps    1E-15

#define NLobes 4.00    // Nb lobes (primaire + secondaires)

unsigned int i_sin=0, i_med=0;

float sin_x, SigNoise[N], Sig[N];

float Median_val=0.0;

void setup()

{

    // Génération du signal original

    //getSinNoise(Sig, N, 0.0);
```



```
// Génération du signal bruité (signal + bruit)

//getSinNoise(SigNoise, N, 0.5);

// Génération du signal original

getSinNoise(Sig, N, 0.0);

// Génération du signal bruité (signal + bruit)

getSinCNoise(SigNoise, N, 0.1,NLobes);

// Initialisation du port série - Affichage des signaux

Serial.begin(9600);

}

void loop()
```



```
{  
  
  // Affichage du signal original  
  
  Serial.print( Sig[i_sin]);  
  
  Serial.print(",");  
  
  
  // Affichage du signal bruité  
  
  Serial.println( SigNoise[i_sin]);  
  
  
  
  // Mise à jour des indices  
  
  i_sin++; i_sin=i_sin%(N-1);  
  
  
  
  // Re-génération du signal bruité  
  
  //if(i_sin==0) getSinNoise(SigNoise, N, 0.5);
```



Arduino #40: le signal et le bruit - le sinus cardinal

```
}
```



Fonction Génération du signal $\text{sin}()$



Arduino #40: le signal et le bruit - le sinus cardinal





```
void getSinNoise(float *Buffer, int Taille, float AmpNoise)
{
    float noise=0.0;

    for(int i=0;i<Taille;i++)
    {
        // Bruit

        noise=random(N);    // Bruit avec une moyenne = N/2

        noise=AmpNoise*(noise-(float)N/2.00); // Bruit centré (moyenne
nulle)

        // Signal sinusoidal

        sin_x=sin(2.00*pi*f0*(float)i/((float)N-1.00));

        Buffer[i]=0.5*(sin_x+1.00)*(N-1.00);

        // Signal + bruit

        Buffer[i]+=noise;
    }
}
```



Arduino #40: le signal et le bruit - le sinus cardinal

```
}
```



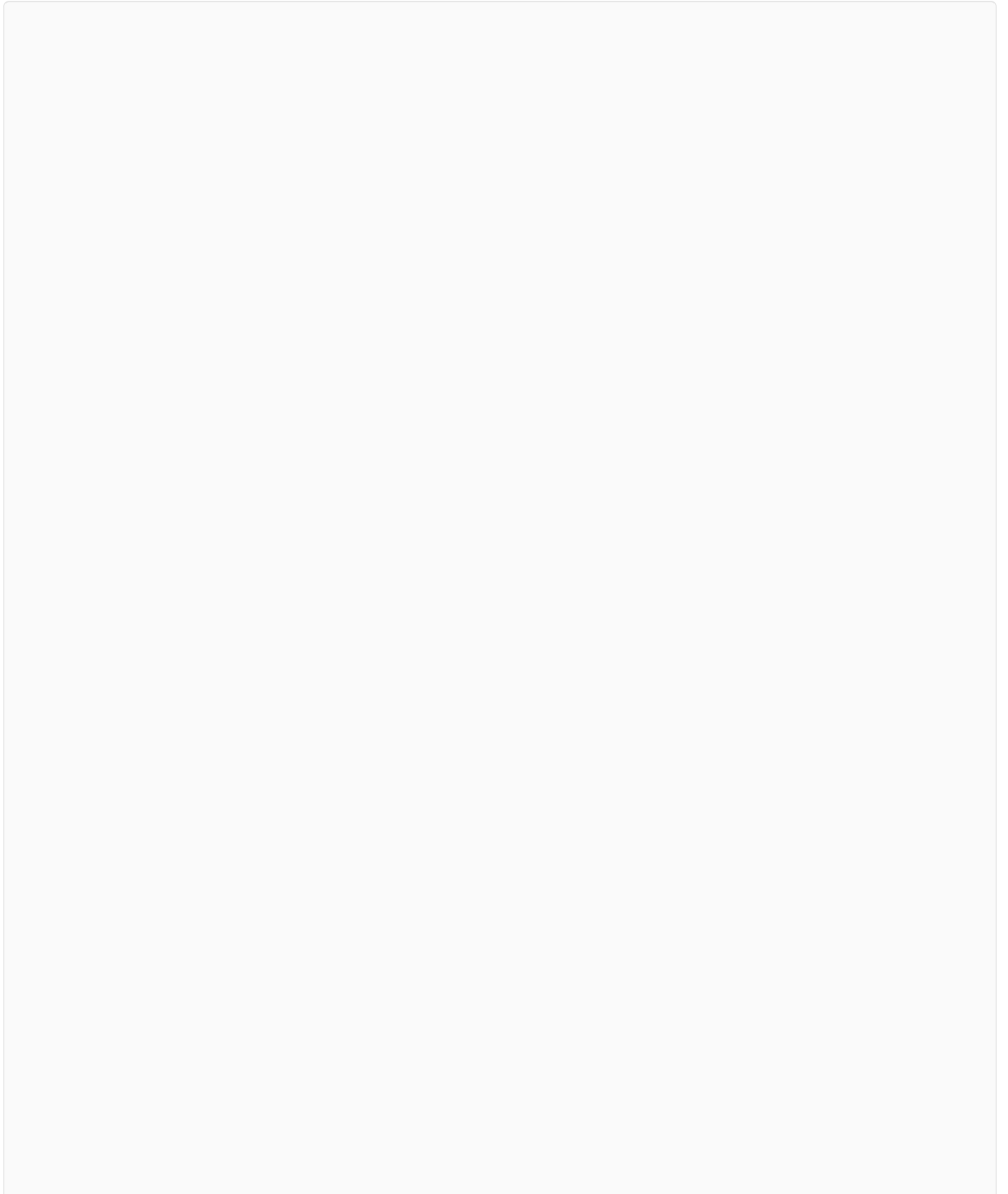
```
}
```



Fonction Génération du signal sinc()



Arduino #40: le signal et le bruit - le sinus cardinal





```
void getSinCNoise(float *Buffer, int Taille, float AmpNoise, float
NumLobes)
{
    float noise=0.0, sinc_x;

    float Step=2.0*NumLobes/(float)Taille;

    int j=0;

    float val_min;

    // Génération du signal sinc(x)
    for(float i=-NumLobes;i<NumLobes;i+=Step)
    {
        // Signal cardinal

        if (i==0.00) Buffer[j]=1.00;

        else Buffer[j]=sin(pi*i)/(pi*i+eps);

        //else Buffer[j]=cos(pi*i)/(pi*i+eps);

        j++;
    }
}
```



```
}

// Calcul de la valeur moyenne

val_min=Buffer[0];

for(int i=1;i<Taille;i++)

{

    if (Buffer[i]<=val_min) val_min=Buffer[i];

}

// Mise en échelle & ajout du bruit

for(int i=0;i<Taille;i++)

{

    // Bruit

    // Bruit

    noise=random(N);    // Bruit avec une moyenne = N/2
```




```
    noise=AmpNoise*(noise-(float)N/2.00); // Bruit centré (moyenne
nulle)

    // Signal + bruit

    Buffer[i]=Buffer[i]+abs(val_min);

    Buffer[i]/=1.00+abs(val_min);

    Buffer[i]=(float)N*Buffer[i] + noise;

}
```



```
}
```



Arduino #40: le signal et le bruit - le sinus cardinal

[Total : 1 Moyenne : 5/5]