

Piaser-Lano : Document Technique

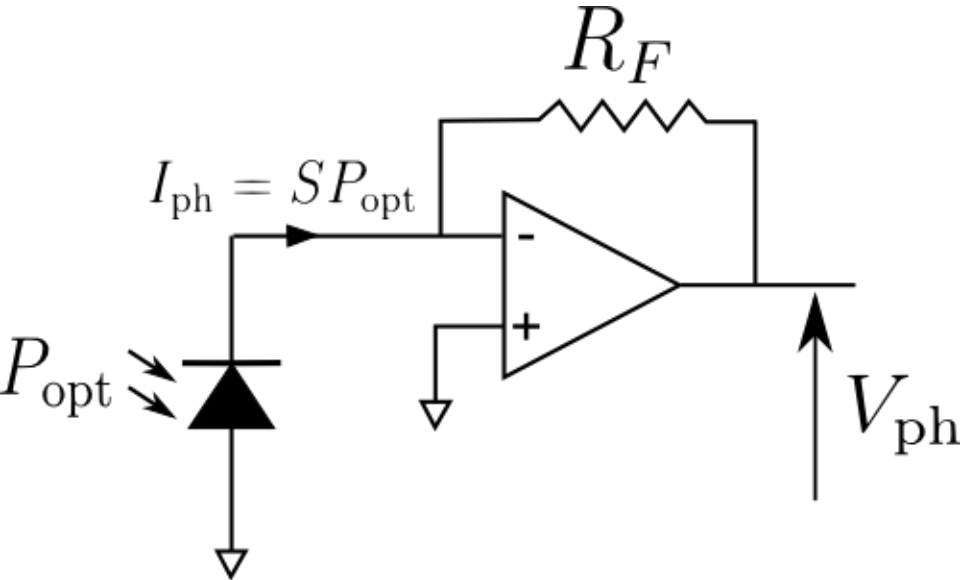
I - Schémas électriques

II - Codes informatiques

III - Difficultés rencontrées

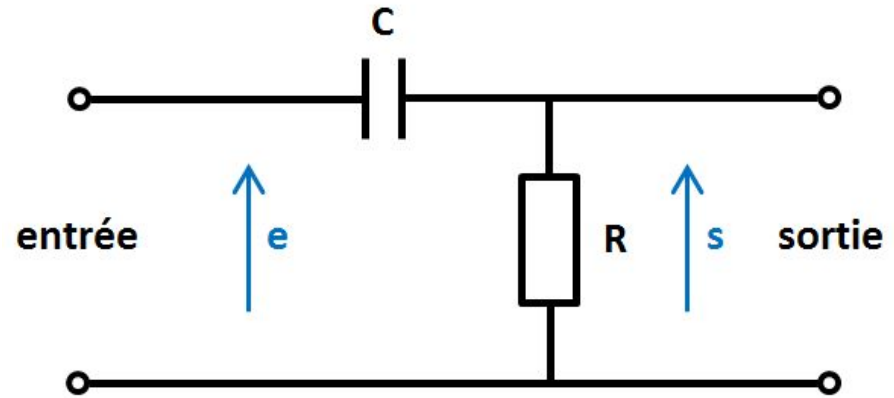
I - Schémas électriques

Montage Transimpédance



- Temps de réponse plus rapide qu'un montage de détection simple
- Permet d'augmenter le gain en jouant sur R_F nous avons choisi $R_F = 1 \text{ Mohm}$

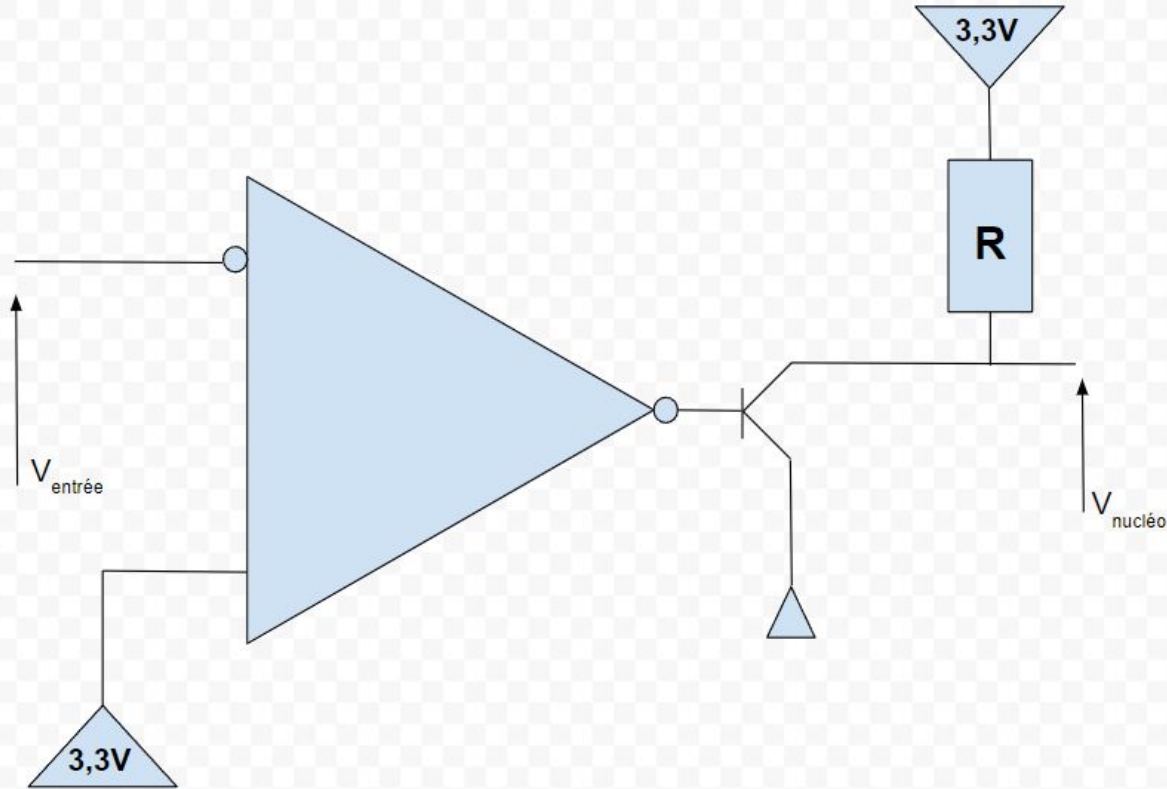
Filtre Passe-Haut



- L'entrée e correspond à V_{ph}
- Permet de filtrer des fréquences parasites comme le 50Hz habituel de EDF
- Fréquence de coupure de $f_c = 100\text{Hz}$

I - Schémas électriques

Comparateur



- Permet d'ajuster la tension d'entrée entre 0V et la tension de référence (ici 3,3V pour la nucléo)

II - Codes informatiques

```
#include "mbed.h"
#define N 20
#define STOP 10

// Partie initialisation sortie du son ---
#include "fichier_sonDo.h" // fichier contenant le tableau son_ech
AnalogOut sortie_son(A2); // DAC
Ticker tictoc;
// -----

Ticker log_timer; // Déclaration du ticker
DigitalOut sortie(D10); // Déclaration de la sortie numérique (signal à envoyer à Martin)
InterruptIn PHTD(PA_9); // Déclaration de l'entrée numérique d'intéruption
Serial pc(USBTX, USBRX); // Liaison PC

// Partie initialisation sortie du son ---
void convert_audio_emetteur();
void convert_audio_recepteur();
int indice_tab;
float FE = 9000;
float TE = 1/FE;
// -----

int timer; // variables compteurs 'Grand timer'
int compteur; // compteur qui détermine l'envoi de la note ou non (permet de modifier la valeur de 'note')
int note; // entier qui donne l'information d'envoyer la note ou non (note=1 <=> on joue la note et note=0 <=> on n'envoie pas la note)
```

Code pour le contrôle de la photodiode et de la sortie du son

```
void front_montant(void);
```

```
void toggle_cpt() {  
    if(timer != 0) timer--; // décrémentation du 'Grand timer'  
}
```

```
int main() {  
    // Initialisation des variables d'état  
    note=0;  
    compteur = 0;
```

```
    // Déclaration du 'Petit ticker(ou timer)' avec  $dT=T/N=0.00025$ ,  $T$ =fréquence de rotation du moteur  $\Leftrightarrow$  fréquence d'apparition d'un créneau au bornes de la photodiode :
```

```
    // Dans notre cas (pour le test)  $T=10\text{ms}=10^{-2}\text{s}$  et  $N=40$  (si on choisit d'avoir 2 valeurs différentes (ou 2 pics) du 'Grand timer' dans un créneau ou  $N=80$  si on en choisit 4, alors  $dT=0.000125$ )
```

```
    log_timer.attach(&toggle_cpt,0.000262);
```

```
    // Initialisation du 'Grand timer'
```

```
    timer = N;
```

```
    PHTD.rise(&front_montant); // On appel la fonction d'interruption sur les fronts montants
```

```
    while (1) {
```

```
        if(timer == 0){
```

```
            timer = N;
```

```
            compteur+=1;
```

```
            if (compteur>STOP) {
```

```
                note = 1;
```

```
            }
```

```
            pc.printf("\r\ncompteur=%d \n",compteur);
```

```
            pc.printf("\r\nnote=%d \n",note);
```

```
        }
```

```
        tictoc.attach(&convert_audio_emetteur, TE);
```

```
    }
```

```
}  
  
void front_montant(void)    // réinitialise le compteur et remet la note en position off ; note=0  
{  
    compteur=0;  
    note=0;  
}  
  
// Fonction sortie de la note ---  
void convert_audio_emetteur(){  
    if (note==1) {  
        sortie = 1;  
        if(indice_tab < NB_ECH){  
            sortie_son.write((son_ech[indice_tab]-0.5)*0.05+0.5);  
            indice_tab++;  
        }  
        else{  
            indice_tab = 0;  
        }  
    }  
    else{  
        sortie = 0;  
        sortie_son.write(0.5);  
    }  
}  
// -----
```

```
#include "mbed.h"
#include "platform/mbed_thread.h"
```

```
DigitalOut H_B_1(D12);
DigitalOut H_B_2(D11);
DigitalOut H_A_1(D8);
DigitalOut H_A_2(D7);
```

```
int main()
{
    while(1){
        H_A_1 = 1; H_A_2 = 0; H_B_1 = 0; H_B_2 = 0;
        wait(0.0004); // intervalle de temps minimal pour que le moteur tourne, cela
        correspond la la fréquence max de environ 191Hz
        H_A_1 = 1; H_A_2 = 0; H_B_1 = 1; H_B_2 = 0;
        wait(0.0004);
        H_A_1 = 0; H_A_2 = 0; H_B_1 = 1; H_B_2 = 0;
        wait(0.0004);
        H_A_1 = 0; H_A_2 = 1; H_B_1 = 1; H_B_2 = 0;
        wait(0.0004);
        H_A_1 = 0; H_A_2 = 1; H_B_1 = 0; H_B_2 = 0;
        wait(0.0004);
        H_A_1 = 0; H_A_2 = 1; H_B_1 = 0; H_B_2 = 1;
        wait(0.0004);
        H_A_1 = 0; H_A_2 = 0; H_B_1 = 0; H_B_2 = 1;
        wait(0.0004);
        H_A_1 = 1; H_A_2 = 0; H_B_1 = 1; H_B_2 = 1;
    }
}
```

Code pour le contrôle du moteur pas à pas

- Utilisation de deux ponts en H car le moteur utilisé avait 4 bobines
- Le moteur peut tourner à une vitesse maximale de 191Hz

III - Difficultés rencontrées

Les difficultés majeures furent :

- Trouver les dossiers son pour les notes de musique et les modifier si nécessaire manuellement pour ne pas avoir de silence pendant une note (se rapporter sur les documents du LEnsE pour la génération d'une tension analogique).
- Questions de précision et de méthode d'action dans la partie informatique, le plus simple que nous avons trouvé est par détection de fronts montants tout en utilisant des timer calibrés en même temps que le passage théorique du laser sur la photodiode.
- Problèmes mécaniques lors des premiers essais, les vibrations du moteur naturelles ajoutées à celles créées par la pièce en plastique créée pour poser le miroir rendent la visée sur une photodiode très compliquée, le principe du code ne permettant pas d'avoir quelques passages du laser sans qu'ils soient détectés.