

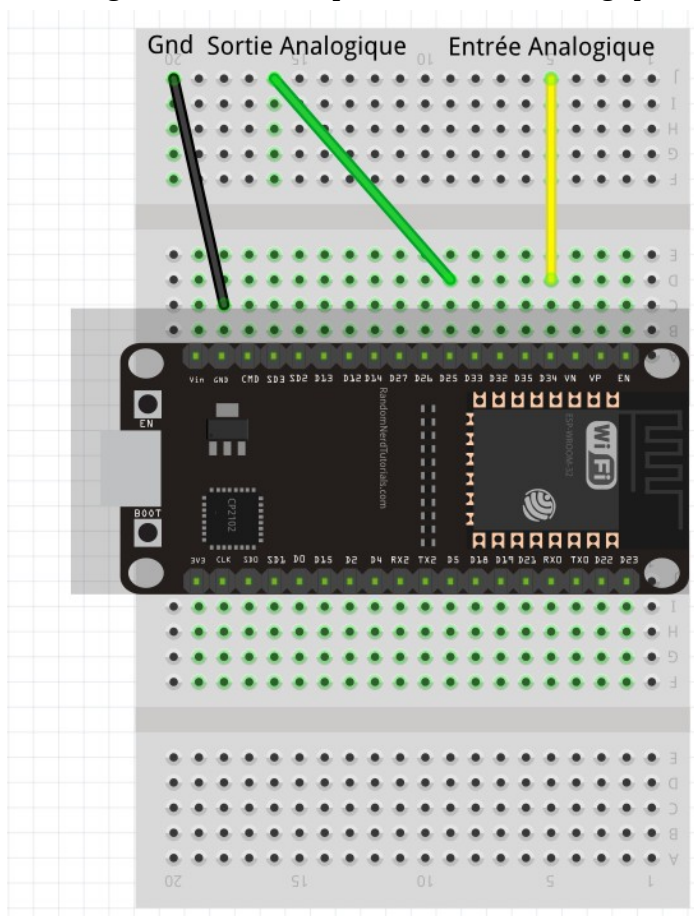
Filtrage Numérique

1 Introduction

Les applications embarquées dans les microcontrôleurs traitent des signaux provenant de différents capteurs (des capteurs de lumière, de températures, etc). Pour extraire des informations de ces signaux et déclencher des actions, ces signaux peuvent nécessiter un filtrage numérique. Par exemple, pour un capteur qui délivre un signal en principe lentement variable mais présentant un bruit important, il est préférable d'échantillonner le bruit puis de procéder à un filtrage passe-bas numérique. Un autre exemple courant est le filtre intégrateur ou le filtre dérivateur.

2 Premier montage

Le premier montage consiste à recopier une entrée analogique sur une sortie analogique.



Ce premier montage permettra de déterminer T_e .

$$y(n) = x(n)$$

```
#include <driver/dac.h>
#include <Arduino.h>

#define InAna 34

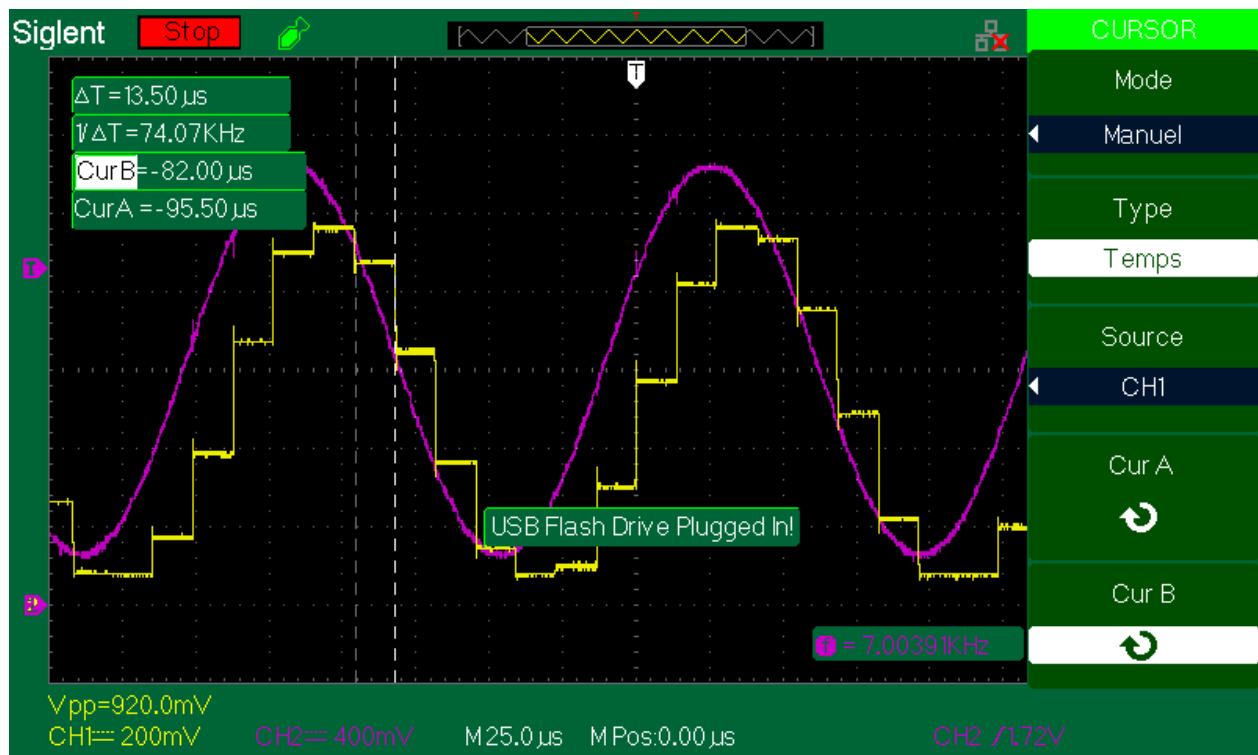
int x;
int y;

void setup() {
```

```

    dac_output_enable(DAC_CHANNEL_1);
}
void loop() {
    x = analogRead(InAna)/16;
    dac_output_voltage(DAC_CHANNEL_1, x );
}

```



On observe le signal d'entrée analogique (en **violet**) , c'est un signal sinusoïdale fréquence 7kHz. Le signal de sortie du convertisseur (en **jaune**).

La période d'échantillonnage est mesurée à **13,5 us**

3 Filtre passe-bas du premier ordre

Filtre passe bas du premier ordre $\tau = 2 T_e$

$$y(n) = 0,333 x(n) + 0,666 y(n-1)$$

Le programme correspondant

```
#include <driver/dac.h>
#include <Arduino.h>

#define InAna 34

int x[2];
int y[2];

void setup() {
    dac_output_enable(DAC_CHANNEL_1);
    y[1] = 0;
}

void loop() {
    x[0] = analogRead(InAna)/16;
    y[0] = (x[0] + 2*y[1])/3; // équation de récurrence passe bas 1er ordre
    dac_output_voltage(DAC_CHANNEL_1, y[0]);
    y[1] = y[0]; // yn-1 <- yn
}
```

Réponse indicielle : $\tau = 2 T_e = 27\mu s$

