

# Communication série asynchrone

## But de la manipulation

Nous allons transformer le voltmètre du labo 2 d'ELEC-H305 en un oscilloscope en le connectant à un PC qui s'occupera de l'affichage des mesures effectuées par le dsPIC.

Pour cela, il sera nécessaire de comprendre le principe d'une communication série asynchrone ainsi que son implémentation dans les dsPIC et dans les PC.

Les caractéristiques désirées de notre oscilloscope sont :

- Fréquence d'échantillonnage : 5kHz.
- Baudrate de la communication série : 115200bits
- Format de la trame : 8 bits de données, pas de parité, 1 stop bit (8N1)

## Prérequis

Avant d'entrer au laboratoire, il est conseillé de lire la section sur l'UART du document *Programmation d'une carte à microcontrôleur*.

## Objectifs

A la fin de ce laboratoire, vous devez être capable :

- De réaliser une connexion série asynchrone et d'en expliquer le fonctionnement.
- D'échanger des données entre un PC et un système à  $\mu\text{C}$  au travers de cette connexion.
- D'utiliser le *Peripheral Pin Select* des dsPIC.

## Schéma-bloc du dispositif

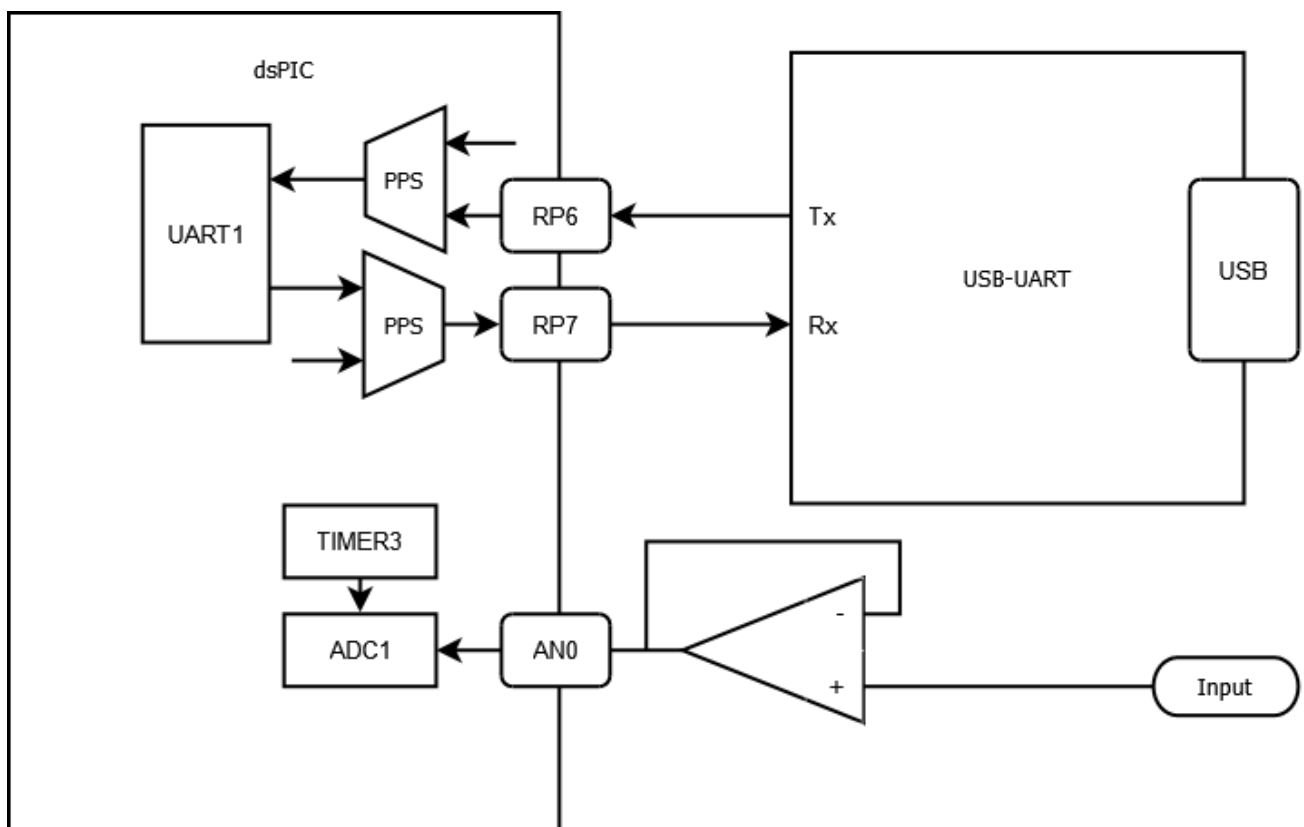


Figure 1 : Schéma-bloc de l'oscilloscope

La Figure 1 montre le schéma-bloc de notre oscilloscope. Pour introduire le projet, au lieu d'utiliser la carte Explorer16, vous allez câbler votre propre système à  $\mu\text{C}$  sur un protoboard. Le schéma des connexions nécessaires au fonctionnement du dsPIC est fourni en annexe.

Vous allez utiliser le dsPIC33FJ128GP802, qui est de la même famille que le  $\mu$ C de l'Explorer16, vous pourrez donc réutiliser du code des deux premiers labos.

En plus du dsPIC, vous aurez besoin d'un ampli-op pour interfacer l'entrée analogique. Son rôle est double : il assure l'adaptation d'impédance et protège le dsPIC d'une surtension éventuelle.

Vous aurez également besoin d'un module pour interfacer l'UART du dsPIC avec le PC. Comme il devient difficile de trouver un PC avec un port série, nous utiliserons un convertisseur USB-UART, dont le fonctionnement est donné en annexe. De plus, ce module fournit une alimentation 3,3V qui va vous permettre d'alimenter facilement votre système.

Comme le montre le schéma-bloc, les conversions de l'ADC seront déclenchées par le Timer3. AN0 sera utilisée comme entrée analogique.

L'UART sera connecté aux pattes RP6 et RP7 du dsPIC par le *Peripheral Pin Select* (PPS) du dsPIC. Le PPS permet de choisir à quelles pattes du dsPIC on veut connecter les périphériques utilisés, c'est très utile pour les  $\mu$ C qui possèdent peu de pattes. Son fonctionnement est basé sur des multiplexeurs (comme représenté dans le schéma-bloc) et est expliqué dans la datasheet du dsPIC (disponible sur l'Université Virtuelle), à la section 11.6.

## **Algorithme**

### ***Partie $\mu$ C***

Le principe de fonctionnement de notre oscilloscope est le suivant :

- Après l'initialisation, le  $\mu$ C attend de recevoir un octet envoyé par le PC. Cet octet sert à synchroniser le PC et le  $\mu$ C, sa valeur n'a pas d'importance.
- Le  $\mu$ C effectue alors 1000 conversions analogique/numérique et envoie, après chaque conversion, le résultat de cette dernière au PC.
- Après les 1000 conversions, le  $\mu$ C attend un nouvel octet pour redémarrer un cycle de 1000 conversions.

L'algorithme détaillé est donné par la Figure 2.

### ***Partie PC***

Côté PC, le fonctionnement doit être complémentaire de celui décrit ci-dessus :

- Après l'initialisation, le PC envoie un octet de synchronisation au  $\mu$ C.
- Le PC attend alors de recevoir 1000 valeurs envoyées par le  $\mu$ C.
- Après avoir reçu les 1000 valeurs, il les affiche sur un graphique.

Nous avons écrit un script Python qui implémente ce fonctionnement. Il attend des valeurs non signées codées sur 8 bits.

## **Manipulation**

- Câblez votre oscilloscope sur la plaquette protoboard fournie.
- Ecrivez un programme permettant de tester votre communication série. Pour cela, nous vous conseillons de réaliser un simple écho : lorsque votre dsPIC reçoit un octet du PC, il doit simplement le lui renvoyer. Côté PC, vous pouvez utiliser *Termite*, un terminal série qui permet d'envoyer et de recevoir du texte sur un port série.
- Ensuite, implémentez l'algorithme de la Figure 2 et testez-le avec le script Python *oscilloscope.py*
- Déterminez les limites de votre oscilloscope :
  - Qu'est-ce qui limite la fréquence d'échantillonnage, à quelle valeur ?
  - Quelles sont les contraintes sur le signal d'entrée ?
- Voici quelques améliorations possibles à apporter à votre oscilloscope :
  - Permettre au PC de définir la fréquence d'échantillonnage et/ou le nombre d'échantillons par acquisition.
  - Envoyer les données sur 10 bits au lieu de 8.
  - Ajouter un trigger
  - ...

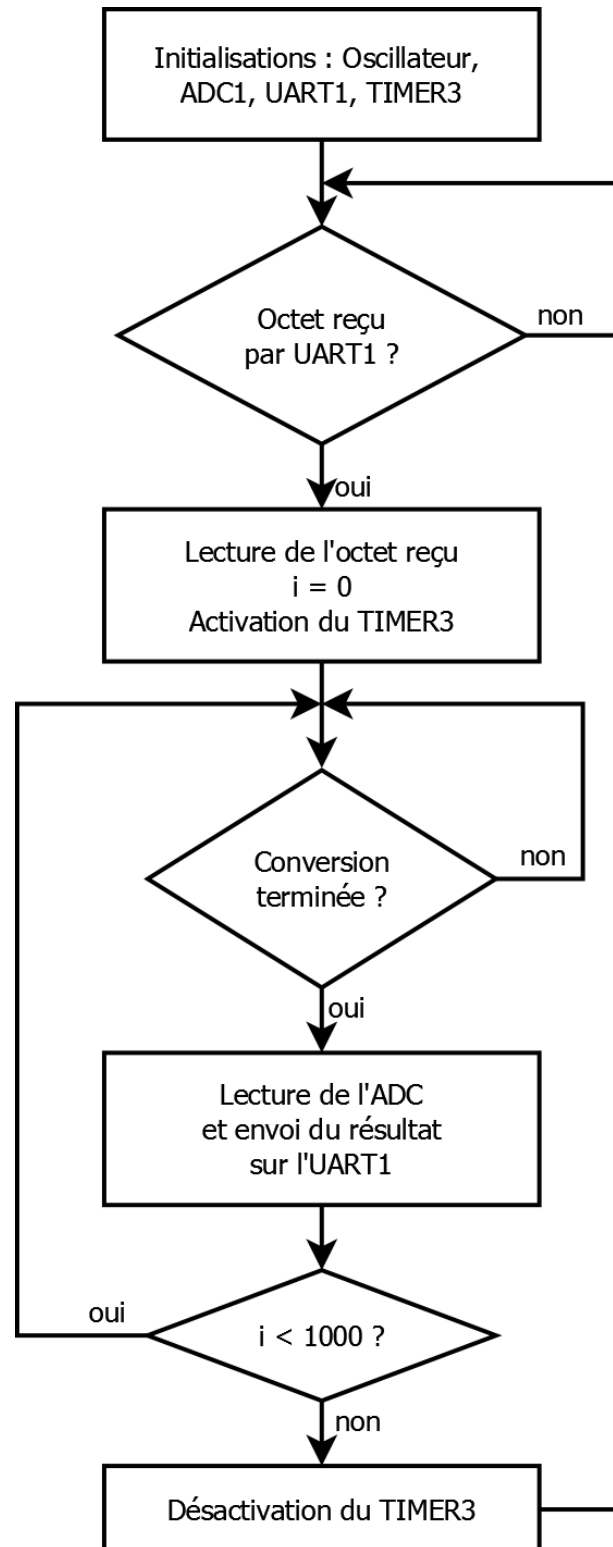


Figure 2 : Algorithme de l'oscilloscope (partie μC)

## Annexes

### Connexion du dsPIC

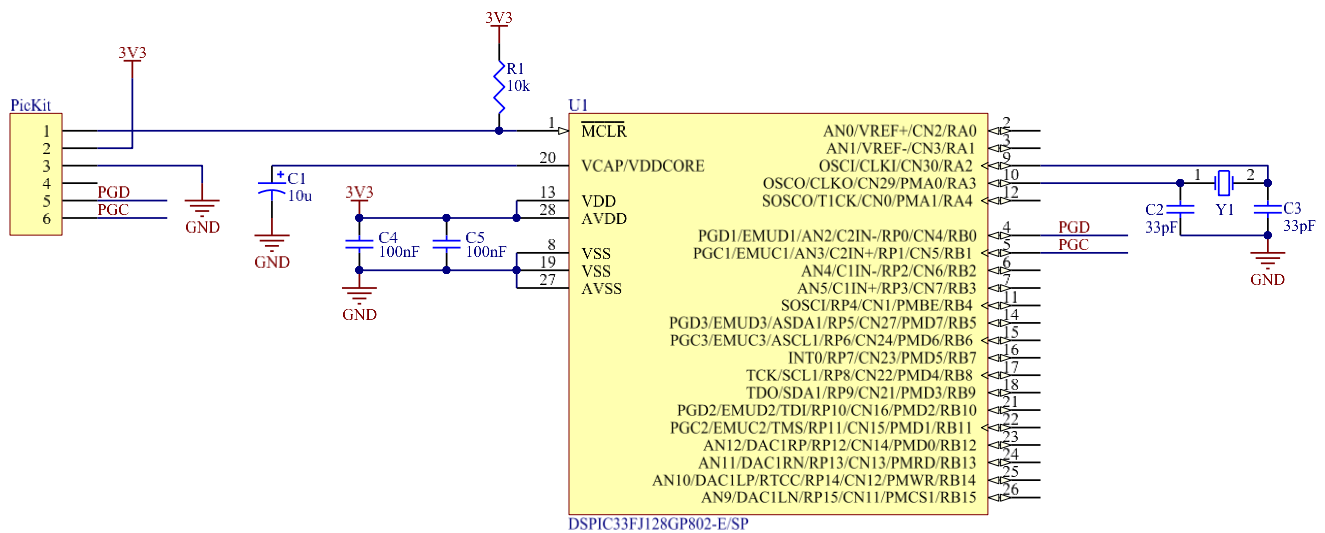
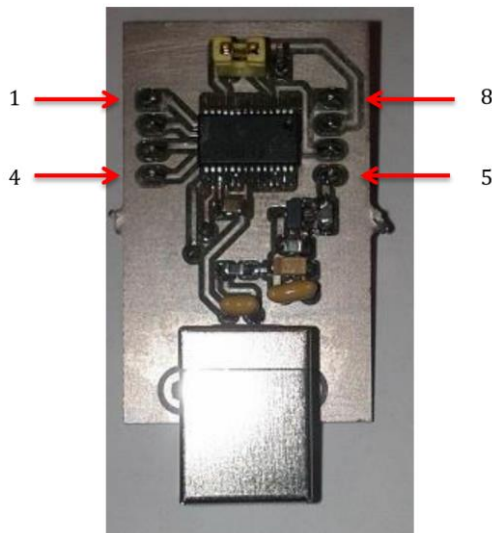


Figure 3 : Connexions minimales du dsPIC

### Le module USB-UART

Le rôle de ce module est de transformer l'UART d'un  $\mu C$  en port USB. Il est basé sur le circuit FTD232R et est vu par le PC comme un port série virtuel. La partie USB de la communication est donc transparente pour nous.

Les bornes du module sont données ci-dessous :



- 1 : masse
- 2 : Rx du module (à connecter au Tx du  $\mu C$ )
- 3 : Tx du module (à connecter au Rx du  $\mu C$ )
- 4 : 5V
- 5 : 3,3V
- 6 : RTS
- 7 : CTS
- 8 : masse

Le jumper permet de choisir la tension des signaux logiques (3,3V ou 5V) en le plaçant du côté de la tension souhaitée.

Figure 4 : Module USB-UART

Les signaux RTS et CTS permettent un contrôle hardware du débit et ne sont pas obligatoires. Les alimentations peuvent être utilisées pour alimenter le  $\mu C$  et l'ampli-op.

Le port série virtuel du PC peut être utilisé avec la plupart des langages informatiques. Il est ainsi possible d'interagir avec le module via Matlab ou via un script Python (avec le module PySerial).

Pour des tests plus courts, vous pouvez également utiliser un terminal qui vous permettra de directement échanger des données entrées au clavier. Sous Windows, le terminal Termite est un des plus simples. Il vous sera peut-être nécessaire d'installer les drivers pour le module, ceux-ci sont disponibles sur le site de FTDI (installer le driver de type VCP : Virtual Com Port).