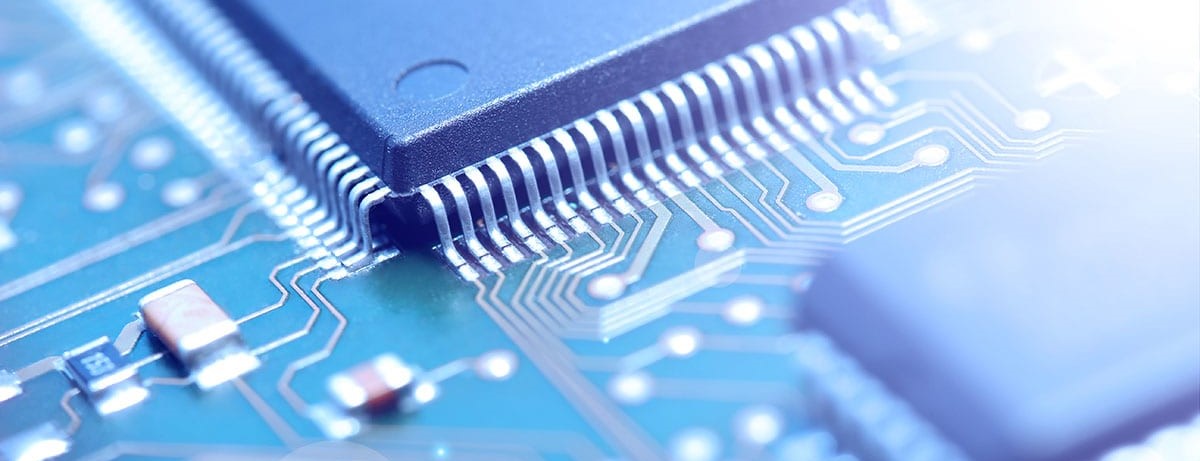


**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**TP MICROCONTROLEUR**

**TP4 – Liaisons séries asynchrone**



**Par Clément CHABANIS & Marion ESCOUTELOUP**

**CESI S3E – A4P1G1 – FIPA26**

**14 novembre 2022**

**TABLE DES MATIERES**

[I. INTRODUCTION 3](#_Toc108564732)

[II. Convertisseur Analogique/Numérique ou CAN 4](#_Toc108564733)

[1. Préparation 4](#_Toc108564734)

[2. Initialisation du CAN 5](#_Toc108564735)

[3. Lecture de la valeur du CAN **Erreur ! Signet non défini.**](#_Toc108564736)

[III. Bus I2C et Convertisseur numérique/analogique (CNA) 5](#_Toc108564737)

[1. Préparation et initialisation du bus I2C 5](#_Toc108564738)

[2. Envoie sur Vumètre (cadran) 5](#_Toc108564739)

[IV. Affichage sur écran LCD 6](#_Toc108564740)

[1. Préparation 6](#_Toc108564741)

[2. Mise en route de l’écran LCD 6](#_Toc108564742)

[V. CONCLUSION 7](#_Toc108564743)

**TABLE DES FIGURES**

[**Figure 1 : Architecture du Projet** 3](#_Toc108564744)

[**Figure 2 : Schéma électrique pour utilisation du potentiomètre et CAN** 4](#_Toc108564745)

# INTRODUCTION

Les microcontrôleurs sont des circuits intégrés qui rassemble des éléments essentiels d’un « ordinateur » : mémoires, processeurs, unités périphériques et interfaces entrées-sorties. Ils sont omniprésents dans notre vie, et sont implantés dans la majorités des objets connectés qui nous entourent. Vulgairement, ce composants est à lieu le cerveau d’un système, et permet de rassembler et de gérer toutes les éléments extérieurs.

Dans la continuité des enseignements de microcontrôleurs vu en A3-S3E, nous allons à présent compléter notre travail avec la notion de liaisons séries asynchrones. Dans notre cas, nous utiliseront l’acronyme officiel **UART (*Universal Asynchrone Receiver Transmiter*)** Afin de mieux appréhender le sujet, voici les définitions des termes :

**Liaisons séries :**

est appelé liaisons séries la méthode de transmission de données via une unique voie de circulation. Chaque transmission est effectuée sous la forme d’une chaine de bits, envoyé 1 par 1. L’intérêt de la liaison série, en comparaison de la liaison parallèle, est quelle nécessite l’utilisation de moins de câble et permet d’éliminer les perturbations entre les pistes. Ainsi, elle peut être déployés pour des débits ou des distances de communications plus importantes.

**Mode asynchrone :**

Est appelé mode asynchrone un mode de transmission ne nécessitant pas un front montant d’horloge pour transmettre un bits d’information. Cependant, il a besoin de connaitre la fréquence d’horloge pour permettre le bon cadencement de lecture afin de pouvoir réaliser un envoie ou une réception de la chaine de bits complète correctement. Sans cecadencement d’horloge, alors des bits pourrait être perdus.

# Présentation des besoins

## Liaisons

Notre but lors de ce TP est de programmer des fonctions nous permettant de réaliser des liaisons asynchrones pour piloter une imprimante à ticket présente sur la mallette d’expérimentation et de gérer une entrée HyperTerminal. Nous aurons ainsi besoins de mettre en place deux liaisons séries :

* Liaison PC ⬄ mallette
* Liaison microcontrôleur ⬄ afficheur/imprimante

Nous appliquerons ainsi les caractéristiques des liaisons séries asynchrones suivantes :

* Vitesse de transmission : 9600 bauds (théorique)
* Nombre de bits envoyés : ici 8 bits
* Trame de données : 1 bit start + 8 bits données + 1 bit stop + ~~parité/flux~~

## Positionnements des ports

Le fonctionnement des modules EUSART est contrôlé par 3 différents registres :

* Transmit Status and Control (**TXSTAx**)
* Receive Status and Control (**RCSTAx**)
* Baud Rate Control (**BAUDCONx**)

Il nous faudra configurer ces 3 registres pour chacune des liaisons UART. Le diagramme d’implantation du microcontrôleur nous indique les ports suivants:

* RX1/TX1 🡪 RC7/RC6 du PIC18
* RX2/TX2 🡪 RG2/RG1 du PIC18

# Initialisation des éléments pour permettre une liaison UART

## Eléments de configuration dans la datasheet du PIC 18

La datasheet du composant nous donne les indications suivantes sur la manières de configurés les liaisons UART :

Une image contenant table

Description générée automatiquement

**1 - Configuration du registre BAUDCONx dans la datasheet du PIC 18**

Une image contenant table

Description générée automatiquement

**2 - Configuration du registre TXSTAx dans la datasheet du PIC 18**

Une image contenant table

Description générée automatiquement

**3 - Configuration du registre RCSTAx dans la datasheet du PIC 18**

## Initialisation UART 1 et UART 2

1. void Init\_UART1(void){
2. SPBRG1 = 17;
4. TXSTA1bits.SYNC = 0;
5. TXSTA1bits.TXEN = 1;
7. RCSTA1bits.SPEN = 1;
8. RCSTA1bits.CREN = 1;
10. BAUDCON2bits.BRG16 = 0;
12. TRISCbits.TRISC6 = 0;
13. TRISCbits.TRISC7 = 1;
14. }

# CONCLUSION