TP2

**1- Génération (émission) d’un signal série asynchrone type « RS232 » en niveaux TTL (0V, 5V)**

**1-1- Exercice 1 :** (ExoTp2-1-1.c) **Questions :**

- Pour une liaison RS232 (niveaux +V / -V), donner la valeur standard des tensions +V et -V.

R : En RS232, les niveaux de tension +V et -V sont généralement définis comme suit :

* +V (logique 0) : Typiquement entre +3V et +25V (en générale +12)
* -V (logique 0) : Typiquement entre -3V et -25V (en générale -12)

- Donner un exemple de circuit intégré qui permet de transformer les niveaux TTL (0V, 5V) en niveaux RS232.

R : circuits spécialisés comme le MAX232 ou le LT1081

- Donner les débits possibles de la liaison série de type RS232 (avec des niveaux +V / -V).

- Donner la longueur maximum d’un câble RS232 pour un débit de 9600 bauds.

R: 15m

- Donner la longueur maximum d’un câble RS232 pour un débit de 56000 bauds.

R: 2.6m

- Pour le PIC utilisé, donner le nom de la broche de sortie du signal de type RS232 (en niveaux TTL) de l’UART.

"TXD" pour "Transmit Data’’

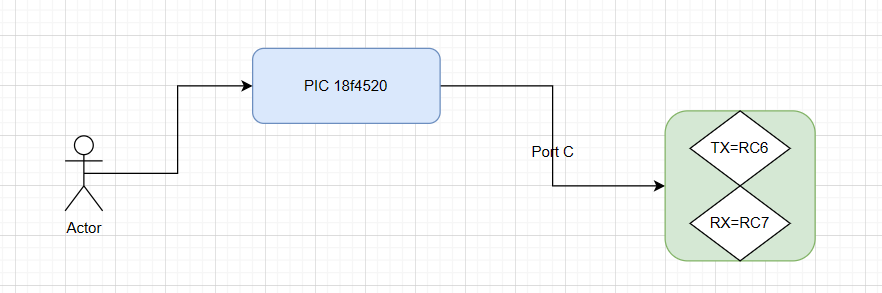
- Pour information pour la suite du TP, donner le nom de la broche d’entrée du signal de type RS232 (en niveaux TTL) de l’UART.

**R: RX** (*Receive data*)

**Exercice 1**

*On désire transmettre un mot de 8 bits en liaison série de type RS232 en niveaux TTL.*

1.1 Diagramme de block



**1.2 Code commenté**

/////////////////////////////////////////////////////////////////////////

// Emission d'une trame série de type RS232 avec l'UART

// Sortie : PortC : TX=RC6 (pour info RX=RC7)

////////////////////////////////////////////////////////////////////////

#include <18f4520.h>

#use delay (crystal=20MHz)

#use rs232(baud=19200, PARITY=N, BITS=8, STOP=1, xmit=PIN\_C6)

// Déclaration des variables

unsigned int16 TEMPO; // variable TEMPO en 16 bits

unsigned int8 DATA; // variable DATA en 8 bits

// PROGRAMME PRINCIPAL

void main()

{

TEMPO=10;

DATA=0x31; // = 0b00110001 = 49

while(TRUE)

{

printf("%c",DATA); // envoi du contenu de DATA (en binaire) sur Tx en RS232

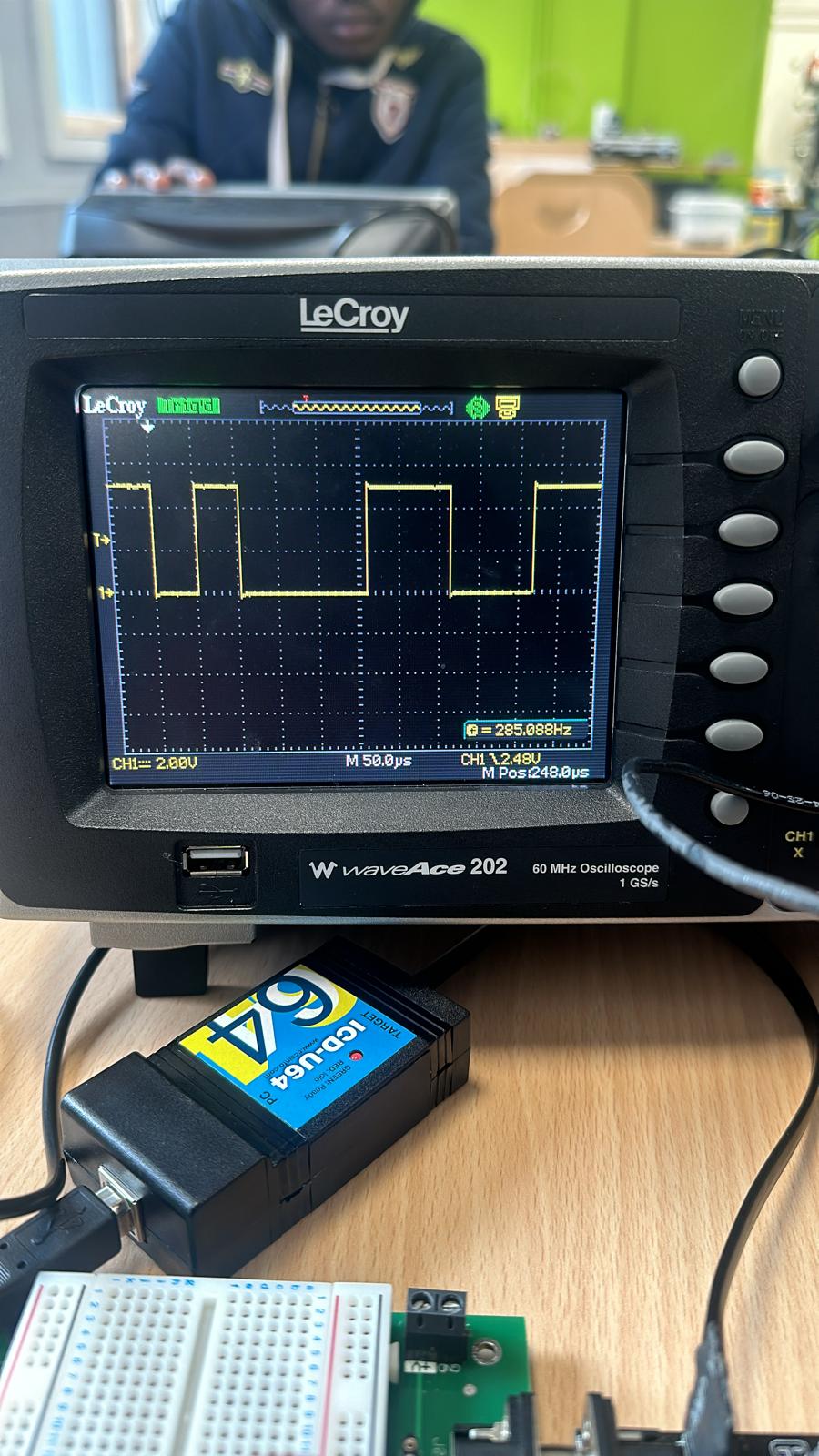
delay\_ms(TEMPO);

}

}

///////////////////////////////////////////////////////////////////////////

- Relever à l’oscilloscope le signal de sortie de type RS232 en niveau TTL.



D’apres la visualisation on trouvée : 00110001

- Sur la trame série relevée, identifier sur ce signal : Les bits de *Start = 0*, le *LSB =1*  de la *donnée*, le *MSB =0* de la *donnée*, le bit de *Stop.*

- Mesurer la durée d’un bit et la durée d’une transmission total du mot de 8 bits.

En déduire la vitesse de transmission.

Pour 1 bit on a 50µs

Pour les 8 bits on a 8.2 careau 8.2\*50µs =410µs

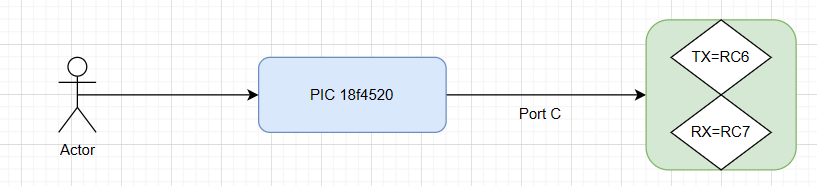
En déduire la vitesse de transmission (bit/s)

bauds

**1-2- Exercice 2 :** (ExoTp2-1-2.c)

**Cahier des charges partiel (voir le code en plus donné ci-dessous) :** *On désire transmettre un caractère (en code ASCII (Unicode UTF-8)) en liaison série de type RS232 en niveau TTL.*

*2.1 - Diagramme de block*



2.2 Mettre en œuvre le programme (petite partie à compléter).

/////////////////////////////////////////////////////////////

// Emission d'une trame série de type RS232 avec l'UART

// Sortie : PortC : TX=RC6 (pour info RX=RC7)

///////////////////////////////////////////////////////////

#include <18f4520.h>

#use delay (crystal=20MHz)

#use rs232(baud=19200, PARITY=N, BITS=8, STOP=1, xmit=PIN\_C6)

// Déclaration des variables

unsigned int16 TEMPO; // variable TEMPO en 16 bits

unsigned int8 DATA; // variable DATA en 8 bits

// PROGRAMME PRINCIPAL

void main()

{

TEMPO=10;

DATA=0x31; // = 0b00110001 = 49

while(TRUE)

{

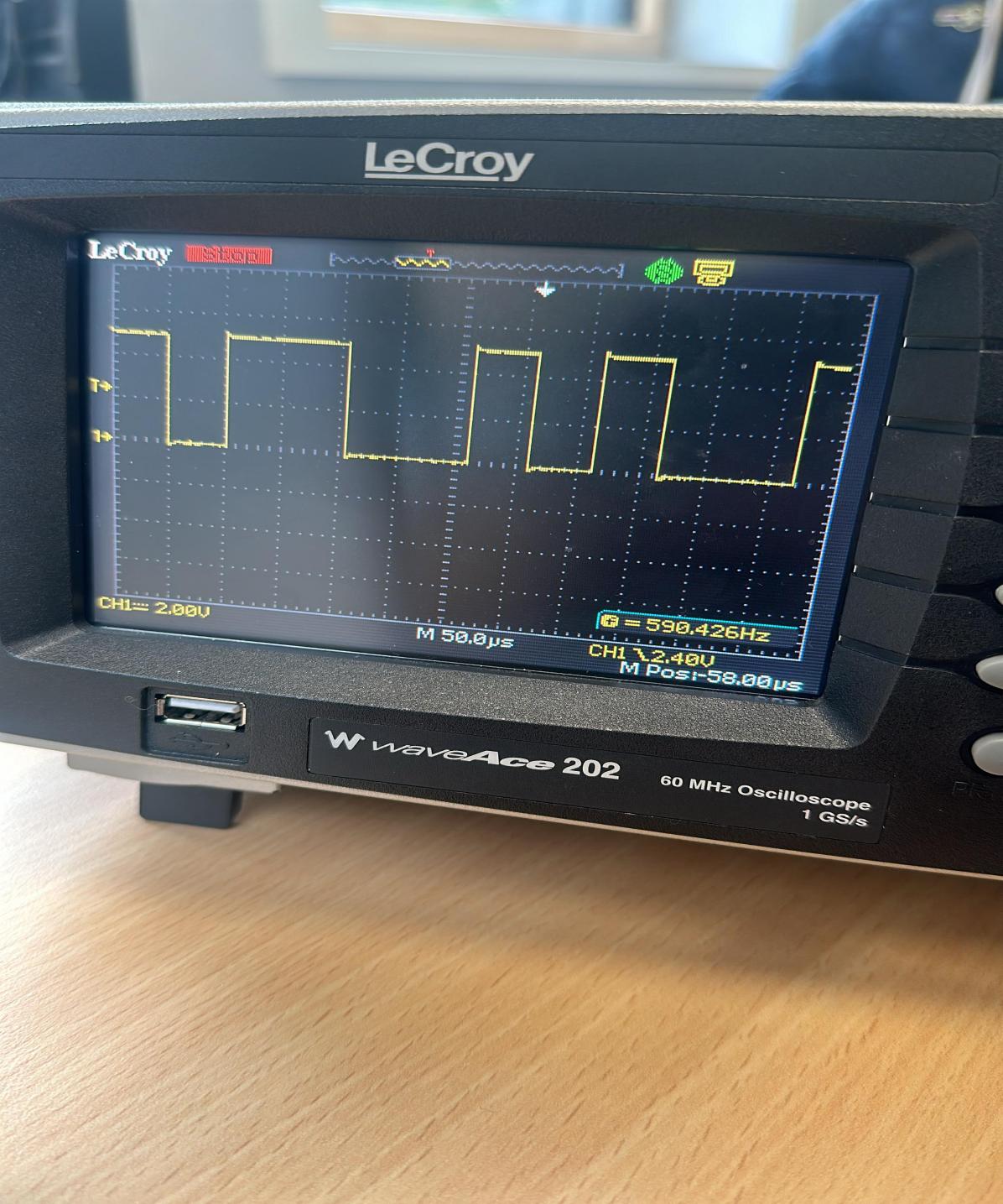
printf("%i",DATA); // envoi du contenu de DATA (en code ASCII) sur Tx en RS232

delay\_ms(TEMPO);

}

}

- Relever à l’oscilloscope le signal de sortie de type RS232 en niveau TTL.



Dans l’osciloscoppe on releve : 0, 01010011

**- Sur la trame série relevée, identifier sur ce signal : Les bits de Start, le LSB de la donnée, le MSB de la donnée, le bit de Stop**

Bit de start = 1; LSB =0; MSB = 1

**- Mesurer la durée d’un bit et la durée d’une transmission total du mot de 8 bits.**

**En déduire la vitesse de transmission.**

Pour 1 bit on a 50µs

Pour les 8 bits on a 8.2 careau 8.2\*50µs =410µs

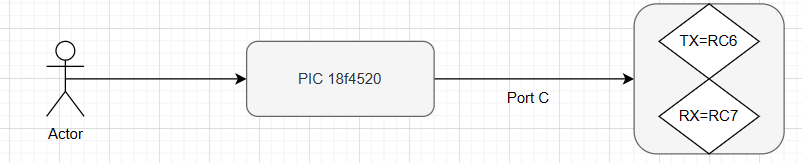
En déduire la vitesse de transmission (bit/s)

bauds

1-3- Exercice 3 : (ExoTp2-1-3.c)

Cahier des charges : On désire transmettre un mot binaire de 8 bits de valeur (010010101) 2 en liaison série de type RS232 de niveau TTL à 4800 bauds sans bit de parité et un bit de stop.

3.1 - Diagramme de block



3.2 -Programme

/////////////////////////////////////////////////////////////////////////////

// On désire transmettre un mot binaire de 8 bits de valeur

// (010010101) 2 en liaison série de type RS232 de niveau

// TTL à 4800 bauds sans bit de parité et un bit de stop.

///////////////////////////////////////////////////////////////////////////////

#include <18f4520.h>

#use delay (crystal=20MHz)

#use rs232(baud=4800, PARITY=N, BITS=8, STOP=1, xmit=PIN\_C6)

// Déclaration des variables

unsigned int16 TEMPO; // variable TEMPO en 16 bits

unsigned int8 DATA; // variable DATA en 8 bits

// PROGRAMME PRINCIPAL

void main()

{

TEMPO=10;

DATA= 0b01001010; // Mot binaire à transmettre

while(TRUE)

{

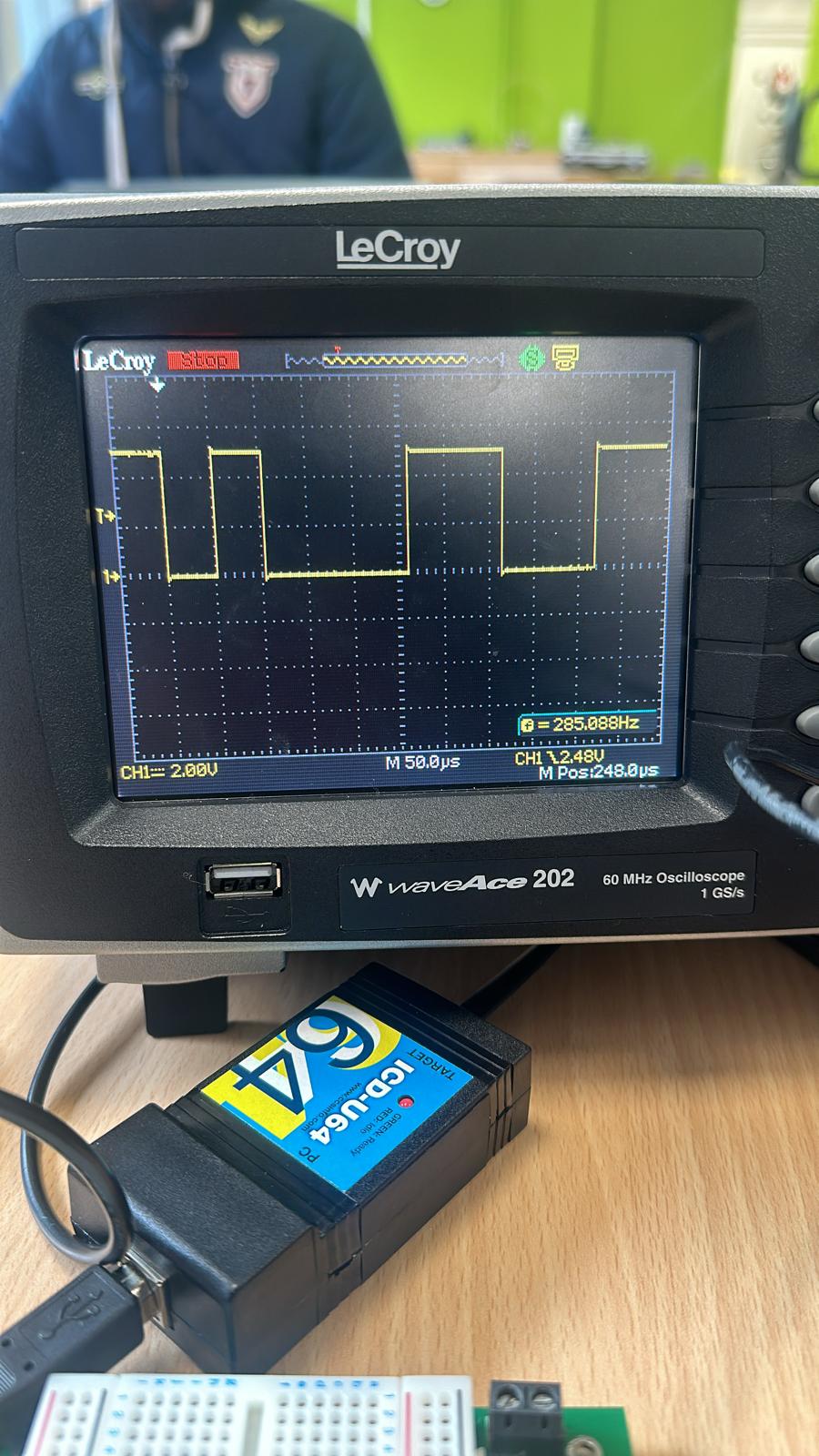
printf("%c",DATA); // envoi du contenu de DATA (en code ASCII) sur Tx en RS232

delay\_ms(TEMPO);

}

}

- - Relever à l’oscilloscope le signal de sortie de type RS232 en niveau TTL.



**2- Réception d’un signal série asynchrone type « RS232 » en niveaux TTL (0V, 5V)**

2-1- Exercice 4 : (ExoTp2-2-1.c)

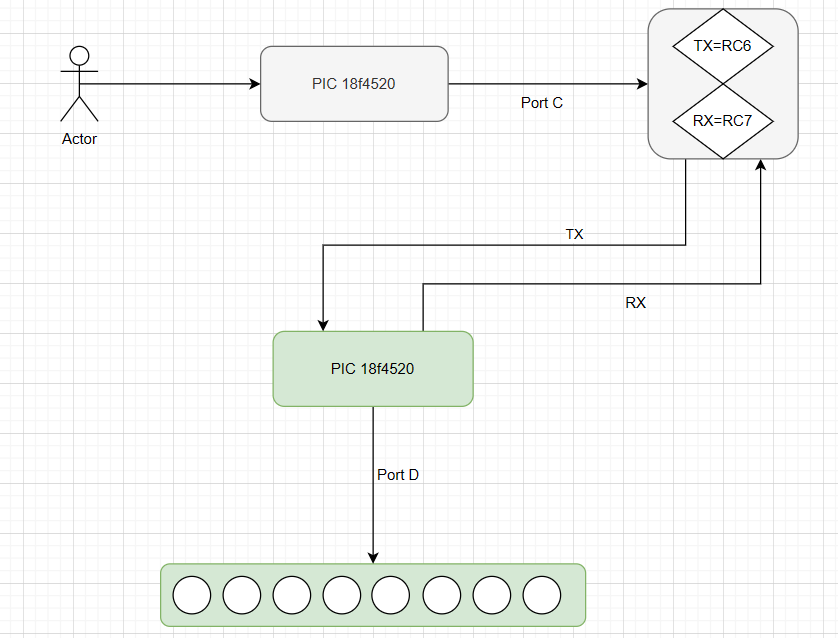
Cahier des charges : On désire recevoir et afficher sur des leds une donnée au format de 8 bits en liaison série de type RS232 de niveau TTL à 4800 bauds sans bit de parité et un bit de stop.

Remarque : Afin de pouvoir tester ce cahier des charges, on vous fournit une 2ème carte de prototypage

EB016 et une 2ème carte PIC EB006 dans laquelle on pourra implanter le programme de l’exercice 3.

Cette 2ème carte sera reliée à votre 1ère carte PIC EB006 contenant le programme du cahier des charges de l’exercice 4.

Diagramme de block interne



4.2 - Programme commenté

// On désire \*recevoir\* et afficher sur des leds une donnée au format de 8 bits

//en liaison série de type RS232 de niveau TTL à 4800 bauds sans bit de parité

//et un bit de stop.

//Entrée : PortC : RX=RC7 (pour info TX=RC6)

// Sortie : PortD avec 8 leds

#include <18f4520.h>

#use delay (crystal=20MHz)

#use rs232(baud=4800, PARITY=N, BITS=8, STOP=1, xmit=PIN\_C6, rcv=PIN\_C7)

// Déclaration des variables

unsigned int8 DATA; // variable DATA en 8 bits

unsigned int8 TEMPO; //variable TEMPO

// PROGRAMME PRINCIPAL

void main()

{

 TEMPO=10;

 //DATA=0x95;//0b10010101 Mot binaire à transmettre;

 while(TRUE)

 {

 DATA = getc(); // Lecture de la donnée reçue sur l'UART

 output\_d(DATA) ; // Envoi de DATA sur les leds du PortD

  }

}

// Emission d'une trame série de type RS232 avec l'UART

// Sortie : PortC : TX=RC6 (pour info RX=RC7)

#include <18f4520.h>

#use delay (crystal=20MHz)

#use rs232(baud=4800, PARITY=N, BITS=8, STOP=1, xmit=PIN\_C6)

// Déclaration des variables

unsigned int16 TEMPO; // variable TEMPO en 16 bits

unsigned int8 DATA; // variable DATA en 8 bits

// PROGRAMME PRINCIPAL

void main()

{

TEMPO=10;

DATA=0x95; // = 0b10010101 = 95

while(TRUE)

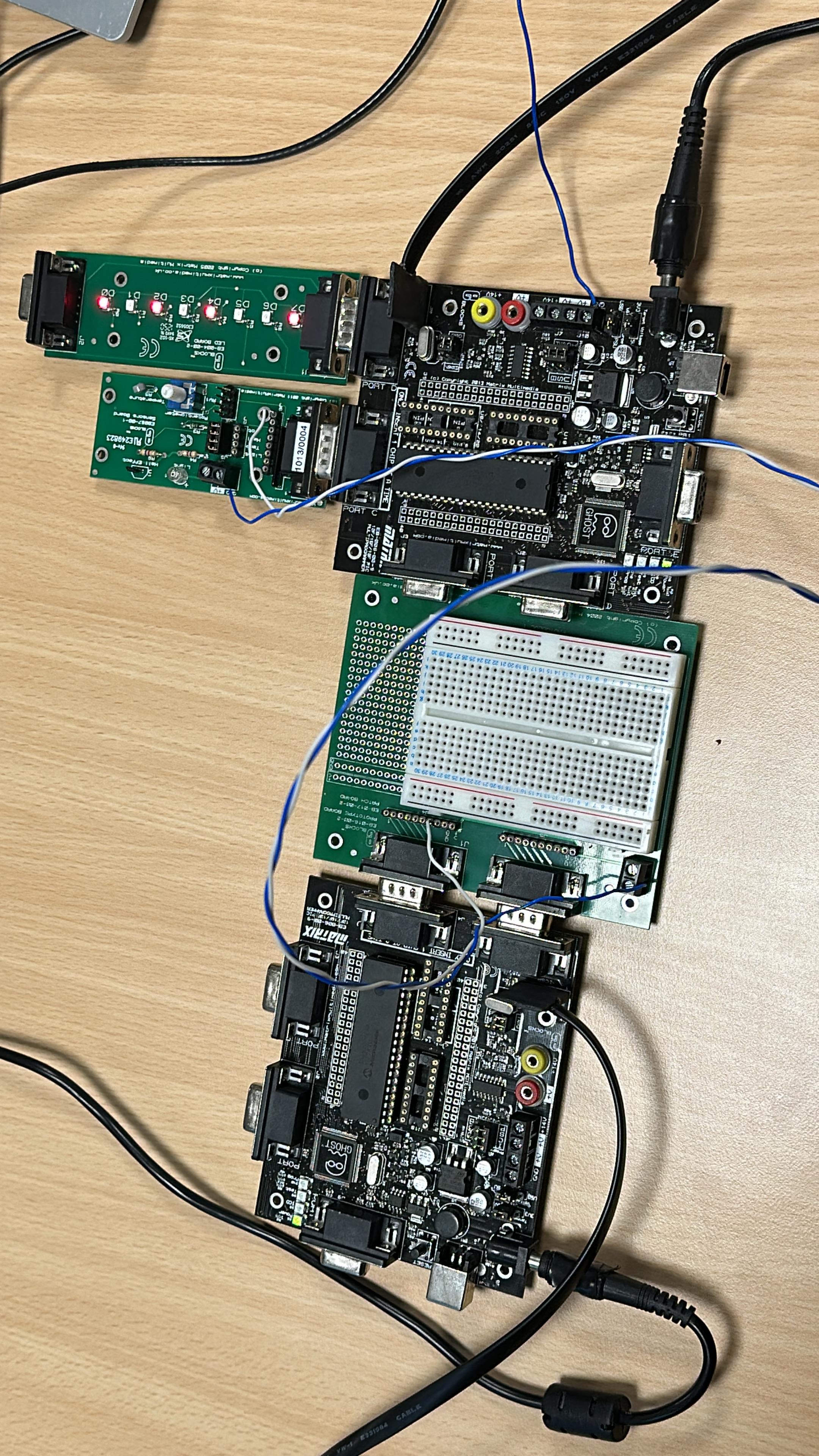
{

printf("%c",DATA);//envoi du contenu de DATA (en code ASCII) sur Tx en RS232

delay\_ms(TEMPO);

}

}

- Relever à l’oscilloscope le signal de sortie de type RS232 en niveau TTL.

