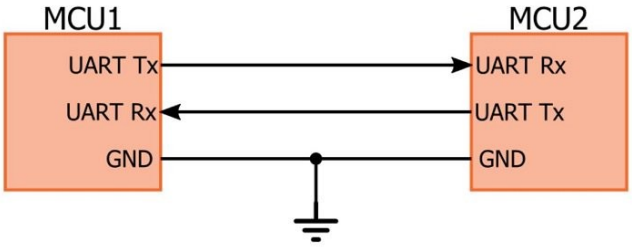
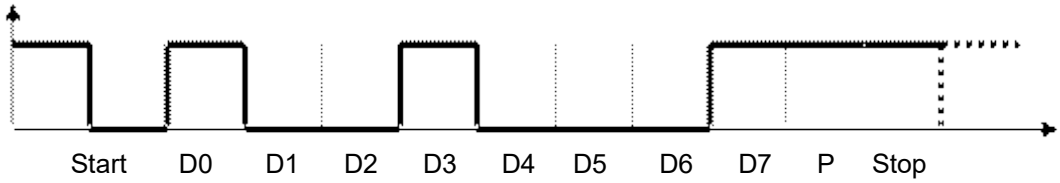
STM32 UART

Objectifs : Mise en œuvre des UART (*Universal Asynchronous Receiver Transmitter*) .
- Debug par liaison série
Matériel : Ce TP utilise une NUCLEO-F411RE, mais n'importe quelle autre carte NUCLEO convient.
Logiciel : MBED
Conserver les corrigés de tous les exercices, ils serviront pour les révisions



Les communications séries asynchrones suivent le format NZR (No Return to Zero), l'état de repos est l'état haut. Les communications s'effectuent en série et **sans horloge**.

Exemple de trame : asynchrone 1start, 8 bits, parité paire, 1 stop : nombre 10010001



Ici la lecture de l'octet transmis donne le nombre 10010001, D0 étant envoyé en premier, le nombre transmis est en fait 10001001, soit 0x8A.

La parité permet de déceler des erreurs de transmission élémentaires (inversion d'un bit). Le **bit de parité** n'est pratiquement plus employé aujourd'hui, on lui préférera le cheksum ou code CRC dans le cas de l'envoi de grands messages.

Parité paire : le bit de parité est positionné pour que l'ensemble des bits de données et le bit de parité représentent un nombre de bits à 1 pair.

Parité impaire : le bit de parité est positionné pour que l'ensemble des bits de données et le bit de parité représentent un nombre de bits à 1 impair.

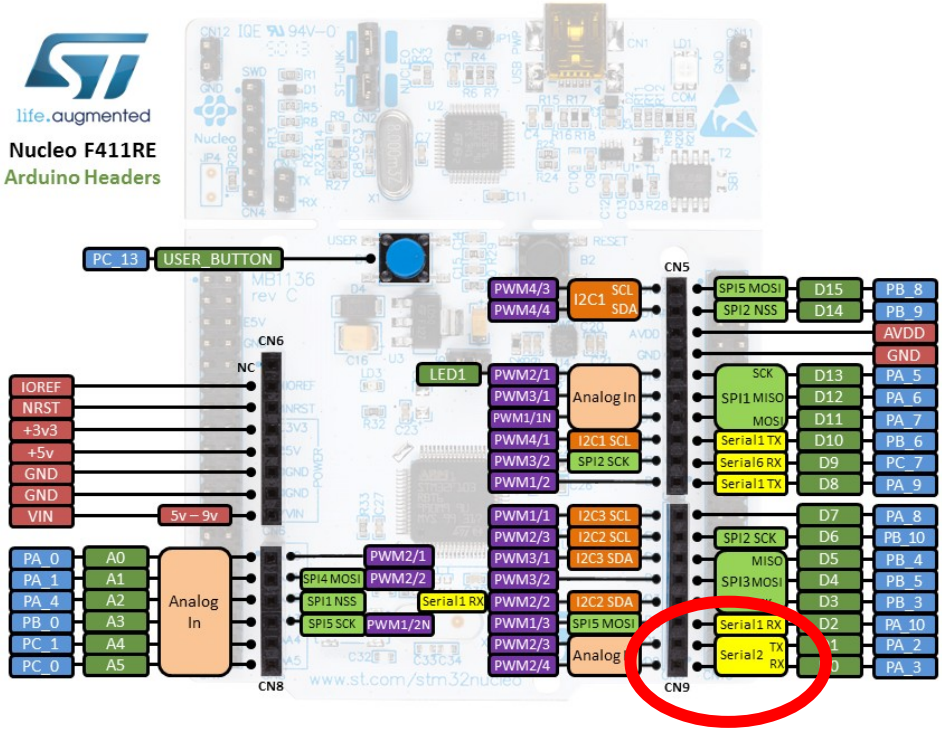
Dans ce type de communications séries l'horloge n'est pas transmise, le **bit de start** est utilisé par le récepteur pour se synchroniser avec l'émetteur. Les horloges de transmission et de réception des deux microcontrôleurs doivent être très proches. Le **bit de stop** permet de replacer la ligne à l'état de repos, après ce bit, un nouvel octet peut être transmis.

L'interface de communication asynchrone d'un microcontrôleur est l'UART (Universal Asynchronous Receiver Transceiver)

STM32 UART

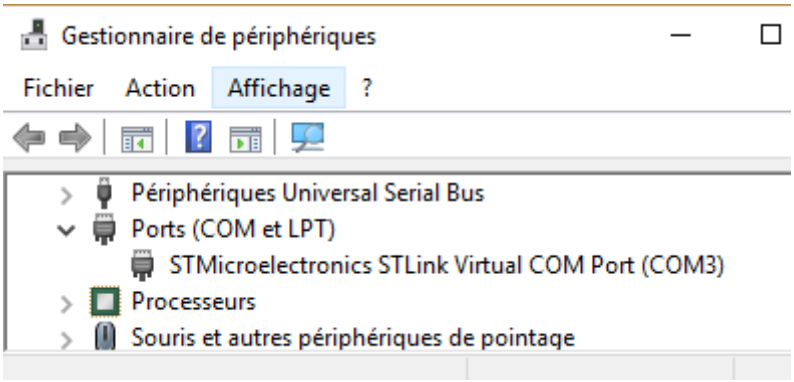
Les microcontrôleurs STM32 possèdent plusieurs UART, de 2 à 5 suivant les modèles.
L'UART2 (Serial2) est utilisé par la liaison UART over USB, qui permet d'établir des communications très simples entre un PC et la carte NUCLEO.
Attention, si l'UART over USB est utilisé, l'UART2 n'est plus disponible.

UART2 sur les connecteurs "Arduino"



La liaison USB entre le PC et la carte NUCLEO a donc une **double fonction** :
- Téléchargement et mise au point du logiciel
- Communications asynchrones entre le PC et le STM32

Vérifier la présence du driver STLINK dans le pc : ouvrir le gestionnaire de périphériques

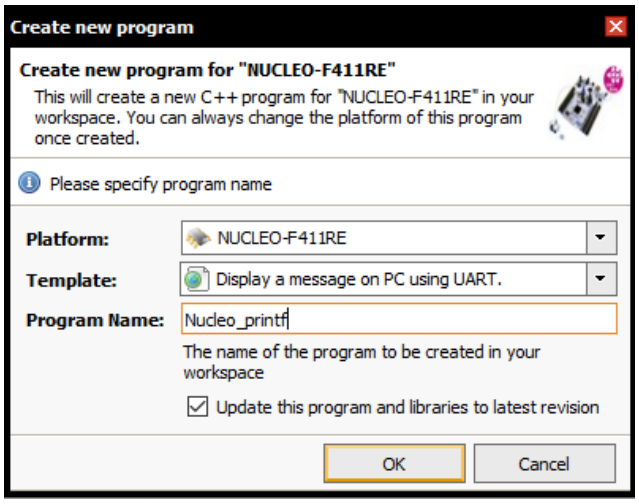


Ici l'OS a attribué le port com3 à la liaison UART over USB. Relever le numéro de Virtual COM Port attribué sur votre PC.
Sinon installer ce dernier : http://www.st.com/en/development-tools/stsw-link009.html

Un émulateur de terminal série est nécessaire, exemple , "teraterm"
https://ttssh2.osdn.jp/index.html.en

1 Émission vers l'UART

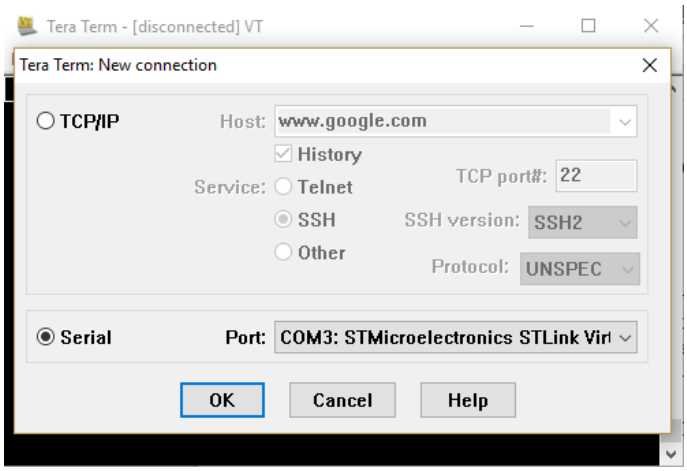
Ouvrir MBED, new program



Éditer main.c

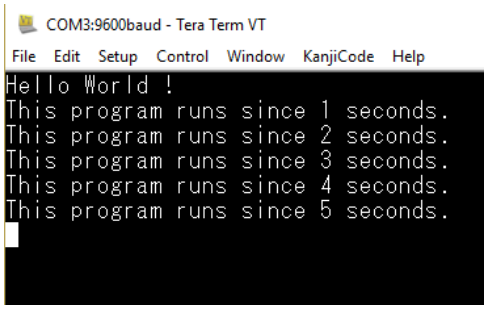
```
#include "mbed.h"
//------------------------------------
// Teraterm configuration
// 9600 bauds, 8-bit data, no parity
//------------------------------------

Serial pc(SERIAL_TX, SERIAL_RX);
DigitalOut myled(LED1);

int main()
{
    int i = 1;
    pc.printf("Hello World !\n");
    while(1) {
        wait(1);
        pc.printf("This program runs since %d seconds.\n", i++);
        myled = !myled;
    }
}
```

Ouvrir un terminal teraterm, connecté sur le port STLINK (com3 dans cet exemple) 9600 Bauds, 8bits, sans parité, sans contrôle de flux.



Onglet setup-terminal : Receive, valider LF

Compiler, télécharger



Analyse du programme :

`Serial pc(SERIAL_TX, SERIAL_RX);`
Création d'un objet "pc" instance de la classe Serial, le format par défaut est 9600,N,8,1
Rechercher dans la documentation les noms des ports correspondants à SERIAL_TX, SERIAL_RX

ON PRÉFÉRA UTILISER LES NOMS USBTX et USBRX équivalents et plus explicites.

`pc.printf("Hello World !\n");`

printf est une méthode de la classe Serial
baud est une méthode permettant de définir la vitesse de transmission en bauds ex : `pc.baud(19200);`

Serial Class Reference

`#include <Serial.h>`

Inherits mbed::SerialBase, mbed::Stream, and NonCopyable< Serial >.

Public Member Functions

- Serial (PinName tx, PinName rx, const char *name=NULL, int baud=MBED_CONF_PLATFORM_DEFAULT_SERIAL_BAUD_RATE)
  Create a **Serial** port, connected to the specified transmit and receive pins.
- Serial (PinName tx, PinName rx, int baud)
  Create a **Serial** port, connected to the specified transmit and receive pins, with the specified baud.
- bool readable ()
  Determine if there is a character available to read.
- bool writeable ()
  Determine if there is space available to write a character.
- void baud (int baudrate)
  Set the baud rate of the serial port.
- void format (int bits=8, Parity parity=SerialBase::None, int stop_bits=1)
  Set the transmission format used by the serial port.
- void attach (Callback< void()> func, IrqType type=RxIrq)
  Attach a function to call whenever a serial interrupt is generated.

Exercice 1 :
A partir du programme de démo ci dessus.
Ajouter les variables :

```
double f=3.14;
int d=1234;
char c=65;
char mess[]="Bonjour";
```

Avec printf, afficher f, d, c en décimal, c en hexadécimal, le code ASCII de c, mess.
Modifier le programme pour entrer les variables depuis le terminal (utiliser scanf).

Exercice 2 :
Afficher l'état du bouton B1

Exercice 3 :

```
#include "mbed.h"
Serial pc(SERIAL_TX, SERIAL_RX);

int main()
{
    int *i = (int *)0;
    pc.baud(19200);
    pc.printf("\n Debut \n");
    for (int j=0; j<20;j++)
        {
        pc.printf("ad %04d : ",i);
        for(int k=0;k<8;k++)
            {
                pc.printf("%08X ",*i++);
            }
        pc.printf("\n");
        }
    pc.printf("fin...");
    while(1);
}
```

Essayer ce programme (attention à la vitesse des communications)

Expliquer les lignes en rouge

2 Réception sur l'UART

La fonction C scanf permet l'acquisition de variables par le canal stdin
Essayer et analyser le programme suivant :

```
#include "mbed.h"
Serial pc(SERIAL_TX, SERIAL_RX);

int main()
{
char nom[20];
   pc.printf("\n\n Votre nom ? ");
   scanf("%s",nom);
   pc.printf("\n Bonjour %s \n",nom);
   while(1);
}
```

Pour visualiser les caractères tapés au clavier : dans teraterm, setup-terminal, cocher local echo
Reprendre l'exercice 1, mais cette fois ci les variables seront entrées au clavier du pc.

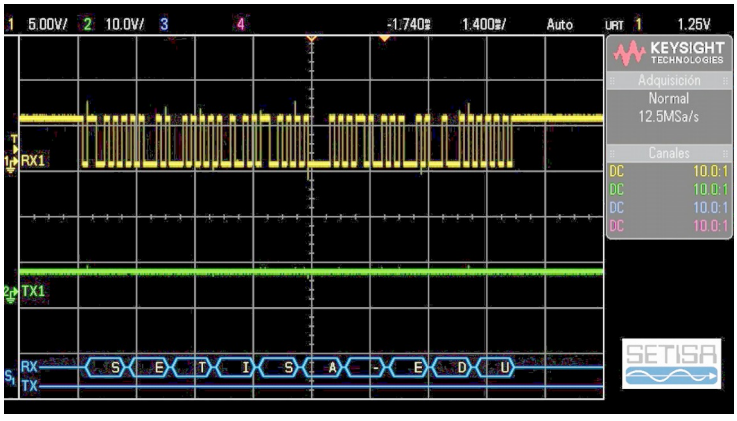
Exercice 4
La méthode "readable" retourne vrai si un caractère est disponible dans le tampon de réception.
Ex `if(nom_instance.readable());`
La méthode "getc" permet de récupérer le caractère depuis le tampon de réception de stdin.
Ex `c=nom_instance.getc();`
La méthode "putc" permet d'envoyer un caractère vers stdout
ex `nom_instance.putc(c);`

A partir de ces deux méthodes réaliser un programme renvoyant le caractère reçu plus un.
Exemple, si l'on tape 'A' sur le clavier du pc, 'B' s'affiche sur teraterm.

Si vous êtes dans l'incapacité de réaliser ce programme, rechercher une solution sur MBED.

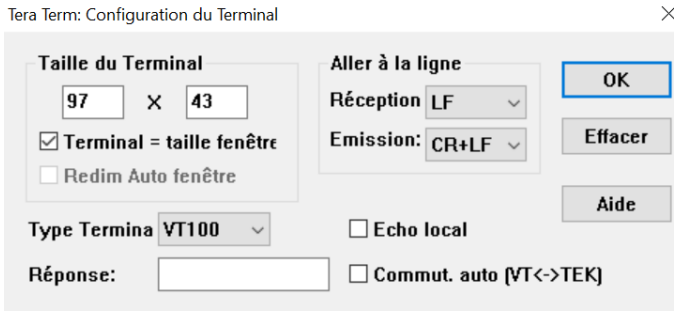
Exercice 5
Réaliser un programme allumant la LED1 lors de l'appui sur la touche 'a' du clavier du pc et éteignant la LED1 lors de l'appui sur la touche 'e'

Exercice 6
En utilisant l'exercice 4. Ajouter une nouvelle instance « Serial » que vous nommerez « test »afin de disposer d'une sortie tx sur un port physique de votre choix sur la carte Nucleo. Dans le programme envoyez également le caractère reçu sur cette nouvelle sortie.
Avec un oscilloscope, relever et analyser les messages sur la broches TX de « test » de la carte NUCLEO.
Utiliser le mode mono coup ainsi que l'analyseur de trames intégré à l'oscilloscope.
Analyser le code NRZ (Non Return to Zero) du caractère transmis, comparer avec une table ASCII, mesurer la vitesse de transfert et comparer avec celle attendue.



Gestion du terminal VT100

Sur Teraterm, Configuration – Terminal, assurez vous que le type de terminal est VT100 :



La liste des commandes VT100 est ici : http://www.termsys.demon.co.uk/vtansi.htm

Par exemple pour effacer l'écran envoyer la séquence **<ESC>[2J**
Le code ASCII <ESC> est 27, la séquence d'effacement s'envoie par `printf("%c[2J",27);`
La réinitialisation du terminal **<ESC>c** doit être suivie d'un temporisation de 100mS , wait(0.1);

Exercice 7 :
Afficher un texte de votre choix sur le terminal (quelques lignes)
L'entrée (scanf) d'un caractère quelconque provoquera l'effacement de l'écran et le déplacement du curseur en haut à gauche (position 0,0)

Exercice 8
Afficher le texte avec les attributs suivants :