

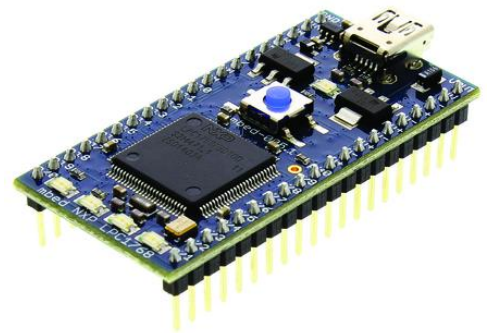
Denis Hautot – Lycée Victor-Hugo BESANÇON.

STS SN

Développement à base de cartes de prototypage rapide  
mbed

# Utilisation du Convertisseur A/N et de la liaison RS232

Prérequis : langage C et TP1 mbed.



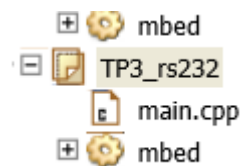
## Ce que vous saurez faire à la fin de ce TP.

Dans ce TP, vous allez apprendre à utiliser la liaison RS232, envoyer et recevoir un caractère et utiliser le convertisseur Analogique Numérique intégré.

## Création d'un nouveau projet.

Sur la carte de TP, vous disposez d'une liaison RS232 que vous pouvez connecter à un ordinateur. En utilisant Teraterm (Hyper terminal étant inutilisable au lycée, sans doute jugé trop dangereux par les administrateurs ;), vous pourrez recevoir et envoyer des caractères sur la ligne.

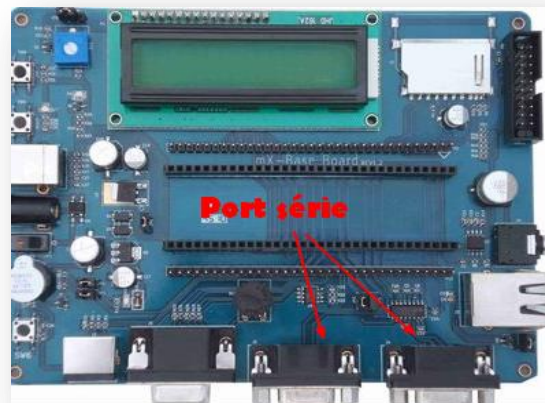
Dans votre interface de travail sur mbed.org, créez un nouveau projet que vous nommerez TP3\_RS232.



## Utilisation de la liaison série RS232, dialogue avec un ordinateur via Tera Term.

Si vous utilisez la carte Base board mx, cet exercice est adapté et vous pourrez communiquer via le port série qui est implanté dessus :

Si vous utilisez une carte ne possédant pas de connecteur DB9, un exemple sera donné plus loin dans le TP. Lisez tout de même tout ce qui concerne la recherche dans les classes, cela s'appliquera aussi.



Effectuez une recherche dans les classes afin de trouver ce qui est disponible au niveau de la liaison RS232. (Serial)

Vous y trouverez la liste des fonctions disponibles ainsi qu'un exemple qui vous permettra de tester.

Quelques fonctions qui peuvent servir :

Serial (p9, p10) ; ce sont p9 et p10 sur lesquelles sont

### Serial Class Reference

```
#include <Serial.h>
Inherits mbed::SerialBase, and mbed::Stream.

Public Member Functions
Serial (PinName tx, PinName rx, const char *name=NULL)
Create a Serial port, connected to the specified transmit and receive pins.
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.
int readable ()
Determine if there is a character available to read.
int writable ()
Determine if there is space available to write a character.
void attach (void(*fptr)(void), IrqType type=RxIrq)
Attach a function to call whenever a serial interrupt is generated.
template<typename T >
void attach (T *fptr, void(T::*mptr)(void), IrqType type=RxIrq)
Attach a member function to call whenever a serial interrupt is generated.
void send_break ()
Generate a break condition on the serial line.
void set_flow_control (Flow type, PinName flow1=NC, PinName flow2=NC)
Set the flow control type on the serial port.
```

les bornes tx et rx

Baud (9600) ; permet de définir la vitesse de communication, ici 9600 ;

Format définit le format de transmission

Readable() indique si un caractère est disponible à la lecture (est arrivé sur la ligne ).

Ecrivez ce programme:

```
#include "mbed.h"

Serial pc(p9, p10) // tx, rx

int main()
{
    while(1) {
        if(pc.readable()) {
            pc.printf("La lettre frappee est : ");
            pc.putc(pc.getc());
            pc.printf("\r\n");
        }
    }
}
```

Paramétrez Tera Term à la vitesse 9600 Bauds, 8N1, valeur par défaut de la transmission sur la carte mbed, puis testez ce programme en frappant un caractère dans Tera Term.

Pour modifier la vitesse de transmission, insérer dans le programme, avant la boucle while la commande `pc.baud(19200)` ;

Paramétrez Tera Term à la vitesse de 19200 bauds et testez à nouveau.

#### Cas d'une carte sans liaison série sur DB9 :

Nous pouvons aussi utiliser l'USB comme USB type 1 (c'est-à-dire port série). Le câble mini USB qui sert à alimenter la carte (et aussi à la connecter comme une clé pour y déposer les fichiers) fabriquera un port COM. (Il faut que le driver `mbedWinSerial.exe` soit installé sur votre poste.)

La seule chose à modifier dans le programme est de remplacer la ligne `Serial pc(p9, p10); // tx, rx` par `Serial pc(USBTX, USBRX); // tx, rx` les données seront automatiquement redirigées.

**Utilisation du convertisseur Analogique numérique et envoi des résultats vers la liaison série.**

Effectuez une recherche dans les classes afin de trouver ce qui est disponible au niveau de la liaison AnalogIn.

Comme toujours, vous y trouverez la liste des fonctions disponibles ainsi qu'un exemple qui vous permettra de tester.

AnalogIN permet de définir et donner un nom à la broche

Read() qui vous donnera un résultat entre 0.0 et 1.0 (un réel proportionnel à la tension lue)

Read\_u16 qui vous donnera un entier sur 16 bits correspondant au résultat de la conversion.

Les 6 entrées analogiques peuvent se situer entre les bornes p15 et p20 sur la carte mbed.

### AnalogIn Class Reference

```
#include <AnalogIn.h>
```

**Public Member Functions**

[AnalogIn](#) (PinName pin)  
Create an **AnalogIn**, connected to the specified pin.

float [read](#) ()  
Read the input voltage, represented as a float in the range [0.0, 1.0].

unsigned short [read\\_u16](#) ()  
Read the input voltage, represented as an unsigned short in the range [0x0, 0xFFFF].

[operator float](#) ()  
An operator shorthand for [read\(\)](#)

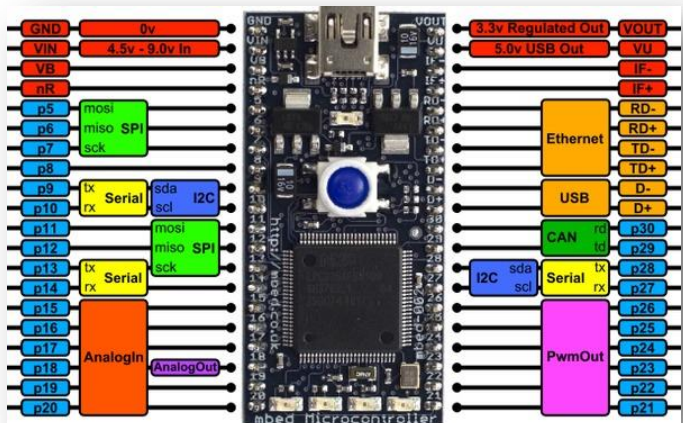
---

**Detailed Description**

An analog input, used for reading the voltage on a pin.

Example:

```
1 // Print messages when the AnalogIn is greater than 50%
2
3 #include "mbed.h"
4
5 AnalogIn temperature(p20);
6
7 int main() {
8     while(1) {
9         if(temperature > 0.5) {
10            printf("Too hot! (%f)", temperature.read());
11        }
12    }
13 }
```



Suivant la carte de TP utilisée, trouvez la résistance ajustable R14 pour la Base board mx ou bien Pot2 sur l'autre carte, (connectée en P20).

Nous allons mesurer la tension fournie par cette résistance et nous envoyer le résultat sur la liaison série sous forme de résultat « La tension lue est XX » toutes les secondes.

Nous allons créer un nouveau projet TP3\_cad et copier les travail déjà fait sur la liaison rs232 dans ce nouveau projet :

Ajoutons la définition

AnalogIn VR14(p20); // pour donner le nom VR14 à la borne d'entrée analogique p20

Modifiez le texte `pc.printf("La lettre frappee est : ");` par

```
pc.printf("La tension lue est (%f)", VR14.read());
```

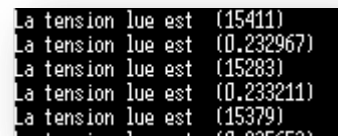
Supprimez la condition d'envoi qui disait que la carte mbed devait envoyer le résultat que si un caractère arrivait par une temporisation d'une seconde. (TP1)

Testez votre programme et faites varier le potentiomètre R14.

Votre programme doit ressembler à ceci

```
1 #include "mbed.h"
2
3 Serial pc(p9, p10); // tx, rx
4 AnalogIn VR14(p20);
5 int main()
6 {
7     while(1) {
8         pc.printf("La tension lue est (%f)", VR14.read());
9         pc.printf("\r\n");
10        wait(1);
11    }
12 }
```

Modifiez votre programme pour que l'affichage du résultat de la conversion soit aussi donné par un nombre variant de 0 à 65536 comme ceci



```
La tension lue est (15411)
La tension lue est (0.232967)
La tension lue est (15283)
La tension lue est (0.233211)
La tension lue est (15379)
```

### Bilan du TP.

Dans ce TP, vous avez utilisé la communication série, la conversion analogique digitale et le formatage de l'affichage lors de l'envoi de données (%f et %d).

Vous avez sans doute aussi remarqué que la recherche et l'utilisation des ressources en ligne, en cherchant dans les classes, permet de progresser rapidement sur des fonctions inconnues.

