

 <h1 style="text-align: center;">Le Raspberry et la liaison série</h1> <h2 style="text-align: center;">Librairie Libserial</h2>	
<b>Sommaire :</b>	
<i>I - Introduction.....</i>	<i>1</i>
I.1. Présentation.....	1
I.2. Installation de la librairie libserial.....	1
I.3. Liaison d'un projet C++ à la librairie libserial.....	1
<i>II - Mise en œuvre de la librairie libserial et de la classe SerialPort.....</i>	<i>2</i>
II.1. Présentation.....	2
II.2. Exemple d'utilisation pour lire des trames NMEA.....	3

## I - Introduction

### I.1. Présentation

On utilise le Langage **C++** afin de mettre en œuvre un **port série** sous **Linux**.

Nous allons utiliser la bibliothèque open-source **libserial** constituée du pilote (driver) nommé **libserial.so** à intégrer au système d'exploitation.

La documentation de **libserial** est disponible à l'adresse suivante :  
<http://libserial.sourceforge.net/doxygen/>.

<https://libserial.readthedocs.io/en/latest/tutorial.html>

Deux classes principales sont disponibles dans cette librairie, **SerialStream** et **SerialPort**. La classe **SerialPort** permet de lever des **exceptions** lors d'une erreur.

### I.2. Installation de la librairie libserial

Sous **Ubuntu** et **Debian** on installe facilement la librairie **libserial** à l'aide de la commande :

```
sudo apt-get install libserial-dev libserial-doc.
```

Si le paquet **libserial-dev** n'existe pas, on télécharge la librairie **libserial-0.6.0rc3** puis on l'installe à l'aide des commandes suivantes :

```
cd libserial-0.6.0rc3  
apt-get install python python-sip python-sip-dev libboost-all-dev  
./configure  
make  
make install  
ldconfig
```

### I.3. Liaison d'un projet C++ à la librairie libserial

Pour lier son application C++ à la librairie **libserial**, il faudra rajouter les options de compilation suivantes :

```
-pthread -lserial -lstdc++
```

## II - Mise en œuvre de la librairie libserial et de la classe SerialPort

### II.1. Présentation

On utilise ici uniquement la classe **SerialPort**. Ci-dessous le détail des appels des méthodes de cette classe.

#### Opening a Serial Port I/O Stream :

```
#include <libserial/SerialPort.h>
using namespace LibSerial;
// Create a serialPort instance and Open the serial port for communication
SerialPort *serialPort;
serialPort = new SerialPort();
serialPort->Open("/dev/ttyACM0");
```

#### Setting the Baud Rate :

```
serialPort->SetBaudRate(BaudRate::BAUD_9600);
```

#### Setting the Character Size :

```
// Use 8 bit wide characters.
serialPort->SetCharacterSize(CharacterSize::CHAR_SIZE_8);
```

#### Setting the Number of Stop Bits :

```
// Use one stop bit.
serialPort->SetStopBits(StopBits::STOP_BITS_1);
```

#### Setting the Parity Type :

```
// Use odd parity during serial communication.
serialPort->SetParity(Parity::PARITY_NONE);
```

#### Setting the Flow-Control Type :

```
// Use no flow-control.
serialPort->SetFlowControl(FlowControl::FLOW_CONTROL_NONE);
```

#### Reading Byte :

```
// Read one byte from the serial port.
char next_byte ;
serialPort->ReadByte(next_byte, 0); // Timeout infini
```

#### Writing Byte :

```
// Write one byte from the serial port.
char next_byte = « 0x30 » ;
serialPort->WriteByte (next_byte);
```

#### Closing the Serial Port :

```
serialPort->Close();
```

## II.2. Exemple d'utilisation pour lire des trames NMEA

Cette application permet de lire tous les caractères se présentant sur le port série nommé **/dev/ttyACM0** :

```
#include <libserial/SerialPort.h>
#include <iostream>
#include <unistd.h>
#include <cstdlib>

using namespace LibSerial;

int main( int  argc, char** argv )
{
    // Open the serial port.
    SerialPort *serialPort;
    serialPort = new SerialPort();
    // Open the serial port.
    try
    {
        serialPort->Open("/dev/ttyACM0") ;
    }
    catch(std::runtime_error &e)
    {
        printf ("Error: Could not open serial port \n");
        std::cerr << "Exception : " << e.what()<< '\n';
    }
    if ( serialPort->IsOpen() )
    {
        printf ("Serial port Open\n");
        // Set the baud rate of the serial port.
        serialPort->SetBaudRate(BaudRate::BAUD_9600);
        // Set the number of data bits.
        serialPort->SetCharacterSize(CharacterSize::CHAR_SIZE_8);
        // Disable parity.
        serialPort->SetParity(Parity::PARITY_NONE);
        // Set the number of stop bits.
        serialPort->SetStopBits(StopBits::STOP_BITS_1);
        // Turn off hardware flow control.
        serialPort->SetFlowControl(FlowControl::FLOW_CONTROL_NONE);
        // Read n bytes from the serial port.
        char next_byte ;
        do {
            try
            {
                serialPort->ReadByte(next_byte, 500); // Timeout de 0.5s
                std::cout << next_byte;
            }
            catch(std::runtime_error &e)
            {
                std::cerr << "Exception : " << e.what()<< '\n';
            }
        } while(1);
        // Close the serial port.
        serialPort->Close();
        printf ("Serial port Close\n");
    }
}
```

```
    }  
    return EXIT_SUCCESS ;  
}
```

Lors de l'exécution sur un poste **Ubuntu** comportant un port série (**/dev/ACM0**), on obtient :

```
$GPGSV,4,3,13,25,17,120,,26,62,310,34,27,16,259,20,29,31,061,30*78  
$GPGSV,4,4,13,31,46,210,17*4E  
$GPGLL,4341.87937,N,00016.60915,W,131218.00,A,A*7F  
$GPRMC,131219.00,A,4341.87925,N,00016.60922,W,0.121,,080716,,,A*6A  
$GPVTG,,T,,M,0.121,N,0.224,K,A*25  
$GPGGA,131219.00,4341.87925,N,00016.60922,W,1,08,1.33,158.7,M,48.9,M,,*4B  
$GPGSA,A,3,26,29,20,05,21,31,27,16,,,,,2.01,1.33,1.50*0F  
$GPGSV,4,1,13,04,82,352,20,05,10,043,22,09,00,326,,16,32,305,35*7D  
$GPGSV,4,2,13,18,09,141,16,20,22,089,27,21,68,135,24,23,03,302,*77  
$GPGSV,4,3,13,25,17,120,,26,62,310,32,27,16,259,21,29,31,061,29*77  
$GPGSV,4,4,13,31,46,210,17*4E  
$GPGLL,4341.87925,N,00016.60922,W,131219.00,A,A*79  
Exception : Read timeout  
$GPRMC,131220.00,A,4341.87933,N,00016.60918,W,0.506,,080716,,,A*6F  
$GPVTG,,T,,M,0.506,N,0.936,K,A*2C  
$GPGGA,131220.00,4341.87933,N,00016.60918,W,1,08,1.33,158.7,M,48.9,M,,*4F  
$GPGSA,A,3,26,29,20,05,21,31,27,16,,,,,2.01,1.33,1.50*0F  
$GPGSV,4,1,13,04,82,352,20,05,10,043,22,09,00,326,,16,32,305,34*7C  
$GPGSV,4,2,13,18,09,141,16,20,22,089,26,21,68,135,23,23,03,302,*71  
$GPGSV,4,3,13,25,17,120,,26,62,310,32,27,16,259,22,29,31,061,29*74  
$GPGSV,4,4,13,31,46,210,17*4E  
$GPGLL,4341.87933,N,00016.60918,W,131220.00,A,A*7D  
Exception : Read timeout
```

**Remarque 1** : Il est possible que les ports ne soient pas accessibles par un utilisateur quelconque, dans ce cas, il faut :

- soit exécuter l'application en tant qu'administrateur **root** ;
- soit exécuter la commande **sudo chmod o+rwx /dev/ttyACM0** avant exécution.