

Sur chaque Raspberry :

Faire les mises à jour :

```
sudo apt update  
sudo apt upgrade
```

Activer le port série et désactiver la console série :

```
Raspi-config → Interface Options → Serial Port  
Would you like a login shell to be accessible over serial? <No>  
Would you like the serial port hardware to be enabled? <Yes>  
Puis redémarrer.
```

Installer les paquets nécessaires :

```
sudo apt install g++
```

Raspberry de réception

Sur le raspberry qui reçoit les données, créer un fichier lora_reception.cpp

```
nano lora_reception.cpp
```

Insérer les lignes suivantes :

```
#include <iostream>  
#include <fcntl.h>  
#include <termios.h>  
#include <unistd.h>  
#include <string.h>  
  
int main() {  
    int serial_port = open("/dev/serial0", O_RDWR);  
  
    if (serial_port < 0) {  
        std::cerr << "Erreur ouverture port série\n";  
        return 1;  
    }  
  
    struct termios tty;  
    memset(&tty, 0, sizeof tty);  
    tcgetattr(serial_port, &tty);  
  
    cfsetispeed(&tty, B9600);  
    cfsetospeed(&tty, B9600);  
  
    tty.c_cflag |= (CLOCAL | CREAD);  
    tty.c_cflag &= ~CSIZE;  
    tty.c_cflag |= CS8;  
    tty.c_cflag &= ~PARENB;  
    tty.c_cflag &= ~CSTOPB;  
    tty.c_cflag &= ~CRTSCTS;  
  
    tcsetattr(serial_port, TCSANOW, &tty);
```

```
char c;
std::string message = "";

std::cout << "En attente de messages LoRa...\n";

while (true) {
    int n = read(serial_port, &c, 1);

    if (n > 0) {
        if (c == '\n') {
            std::cout << "Message reçu : " << message << std::endl;
            message = "";
        } else {
            message += c;
        }
    }
}

close(serial_port);
return 0;
}
```

Compiler le fichier :

```
g++ lora_reception.cpp -o reception
```

Lancer le programme:

```
./reception
```

Raspberry d'envoi

Sur le raspberry qui envoie les données, créer un fichier lora_envoi.cpp

```
nano lora_envoi.cpp
```

```
#include <iostream>
#include <fcntl.h>
#include <termios.h>
#include <unistd.h>
#include <string.h>

int main() {
    int serial_port = open("/dev/serial0", O_RDWR);

    if (serial_port < 0) {
        std::cerr << "Erreur ouverture port série\n";
        return 1;
    }

    struct termios tty;
    memset(&tty, 0, sizeof tty);

    tcgetattr(serial_port, &tty);

    cfsetispeed(&tty, B9600);
    cfsetospeed(&tty, B9600);

    tty.c_cflag |= (CLOCAL | CREAD);
```

```
tty.c_cflag &= ~CSIZE;
tty.c_cflag |= CS8;
tty.c_cflag &= ~PARENB;
tty.c_cflag &= ~CSTOPB;
tty.c_cflag &= ~CRTSCTS;

tcsetattr(serial_port, TCSANOW, &tty);

std::string message;

std::cout << "Programme d'envoi LoRa - tape 'exit' pour quitter\n";

while (true) {
    std::cout << "Message: ";
    std::getline(std::cin, message);

    if (message == "exit") break;

    message += "\n";
    write(serial_port, message.c_str(), message.length());

    std::cout << "Message envoyé: " << message;
}

close(serial_port);
return 0;
}
```

Compiler le fichier :

```
g++ lora_envoi.cpp -o envoi
```

Lancer le programme:

```
./envoi
```