# Serial Communication with Rsapberry Pi using Bluetooth Module and Xbee.

e-Yantra Team

June 25, 2016

# Contents

# 1 Objective

To study Serial communication of Raspberry Pi with other devices using Bluetooth Module(HC-05) and then by using 2 XBee modules.

# 2 Prerequisites

- A basic knowledge of Serial Communication.

# 3 Hardware Requirement

1. Raspberry Pi (I will be using Version 2 Model-B)
2. Adapter (To power up an R-pi)
3. PC
4. Bluetooth Module(HC-05)
5. 2 Xbee Modules

# 4 Software Requirement

1. XCTU
2. Blueterm

# 5 Theory and Description

The universal asynchronous receiver/transmitter (UART) takes bytes of data and transmits the individual bits in a sequential fashion. At the destination, a second UART re-assembles the bits into complete bytes. Each UART contains a shift register, which is the fundamental method of conversion between serial and parallel forms. Serial transmission of digital information (bits) through a single wire or other medium is less costly than parallel transmission through multiple wires. Serial communication is the process of sending data one bit at a time, sequentially, over a communication channel.Asynchronous serial data communication is widely used for character-oriented transmissions, while block-oriented data transfers use the synchronous method. In the asynchronous method, each character is placed between start and stop bits. This is called framing. In data framing for asynchronous communications, the data, such as ASCII characters, are packed between a start bit and a stop bit. The start bit is always one bit, but the stop bit can be one or two bits. The start bit is always a 0 (low) and the stop bit(s) is 1 (high).For example, ASCII character A (8-bit binary 0100 0001) is framed between the start bit and a single stop bit. Notice that the LSB is sent out first. when there is no transfer, the signal is 1 (high),

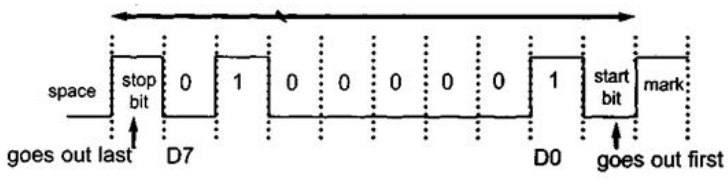


Figure 1: [3]

which is referred to as mark. The 0 (low) is referred to as space. Notice that the transmission begins with a start bit followed by DO, which is the LSB, then the rest of the bits until the MSB (D7), and finally, the one stop bit indicating the end of the character A. The rate of data transfer in serial data communication is stated in bps (bits per second). Another widely used terminology for bps is baud rate.

## 5.1 Bluetooth Module HC-05:

HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup.
**Hardware Specifications:**

- UART interface with programmable baud rate
- Vcc :to power the bluetooth module (It should be in the range of 3.3V to 6V)

- Ground pin
- RxD :connected to the TXD pin of Raspberry Pi(receives the data from Raspberry Pi)
- TXD :connected to the RXD pin of Rsapberry pi(transmits the data to raspberry pi which it receives from other connected device)


Figure 2: [3]

### 5.2 XBee

Xbee can be used for wireless communication with low power consumption. A 3.6V 600mA Lithium battery may last 6 - 12 months for powering up an Xbee while the wireless range can up to 1 mile. It talks with well known UART interface and makes it easy to use. It is simple and straight forward if you only use 2 Xbee for communication. Xbee is the module using Zigbee protocol. Zigbee is a wireless communication protocol like wifi and bluetooth. we have to just connect 4 pins of Xbee to raspberry pi which are:

1. pin 1(3.3V) to Raspberry Pi pin 1 which is 3.3V.
2. pin 2(UART DATA OUT) to Rasberry Pi pin 10(UART_RXD)

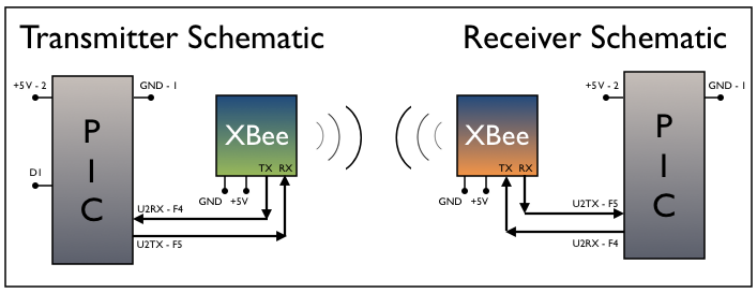


Figure 3: [3]

3. pin 3(UART DATA IN) to Raspberry Pi pin 8(UART_TXD)
4. pin 10(Ground) to Raspberry Pi Ground

**Pin Diagram of XBee:**

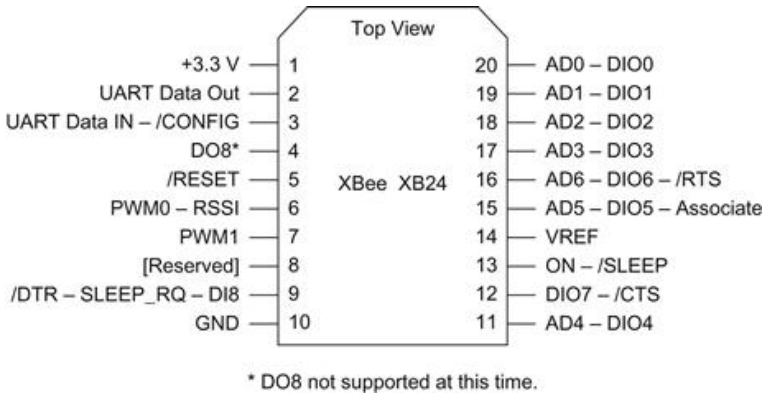


Figure 4: [3]

### 5.3 Enabling UART Communication on Raspberry Pi:

To make Raspberry Pi Compatible for Serial communication we need to follow the following steps:

- We have to make changes to the configuration file cmdline.txt
- Before making any changes to this configuration file, save the file in the case you make any mistake.
- File /boot/cmdline.txt contains the kernel options that are used to boot the system.
  In the Terminal window type the command:
  **sudo nano /boot/cmdline.txt**
- A window will open showing the following options:
  **dwc_otg.lpm_enable=0 console=ttyAMA0,115200 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline rootwait**
  The interesting option is console, because this configure the serial port device /dev/ttyAMA0 to 115200 bps.Remove this serial reference by taking away the option **console=ttyAMA0,115200** . After you make these changes the file should read:
  **dwc_otg.lpm_enable=0 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline rootwait**
- If you are using older version of Raspbian OS then you will find another option in the above mentioned line
  **kgdboc=ttyAMA0,115200**
  Just remove the option from cmdline.txt.
- Also there is another configuration file **/etc/inittab**. Just open the file and inside this file you have to locate the following line:
  **T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100**
  We will comment out the serial console task:
  **#T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100**
- After making these changes reboot the system so that they take effect. With these changes the RPi will not use the serial port at all, so the port will be free to any application that wants to use it as /dev/ttyAMA0.
- If you want to play with sending and receiving data through the serial port then you can install minicom into your RPi using the following command:
  **sudo apt-get install minicom**

And then you can open a terminal on the serial port with this command:
**minicom -b 9600 -o -D /dev/ttyAMA0**
Anything you type inside the minicom terminal will be sent to the serial port, and anything the other side sends will be displayed.

# 6 Experiment

## 6.1 Communication between Rpi and Android device using HC-05 (Bluetooth Module):

- Connect the pins of HC-05 to Raspberry Pi as shown in below figure:

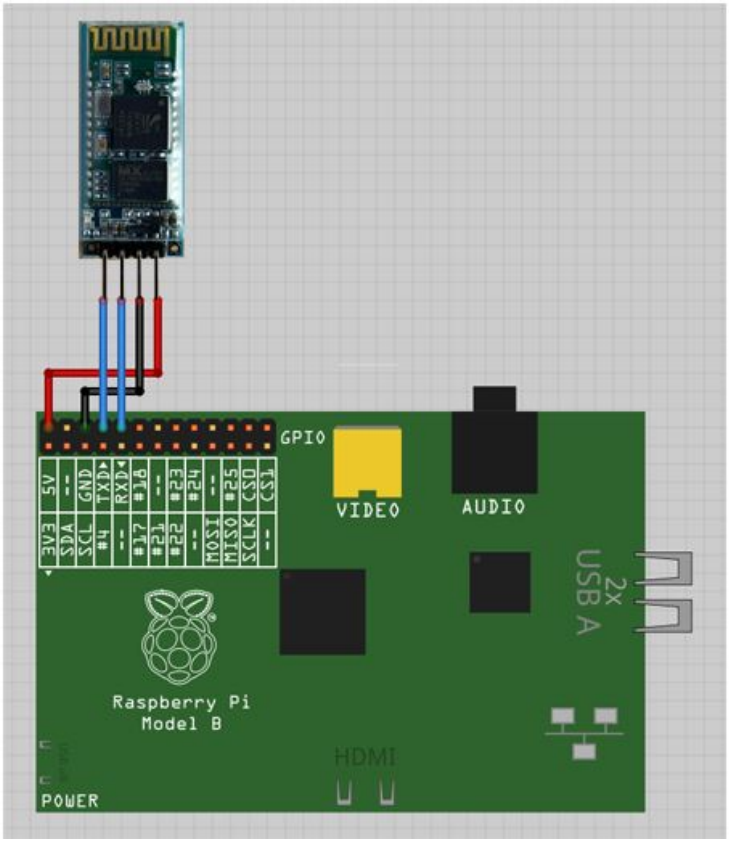


Figure 5: [3]

| RPi GPIO pin | HC-05(bluetooth) module pin |
|---|---|
| 5V (Pin 2) | Vcc |
| GND (Pin 6) | GND |
| TXD (Pin 8) | RXD |
| RXD (Pin 10) | TXD |

- Configure the bluetooth module as the Bluetooth module comes preconfigured from the factory with a set of defaults, which are:

  1. Baud rate: 9600
  2. Bluetooth ID: linvor
  3. PIN: 1234

  If you want to change any of the above options then you can change them by using AT commands.

- Download the Blueterm app on your Android Device and connect to the HC-05 via bluetooth obviously.

- Now we will write a program in python to serially transmit and receive data as:
  **CODE:**

```
import serial
import time

#serial Driver Initialization
ser=serial.Serial('/dev/ttyAMA0',baudrate=9600,
                  parity=serial.PARITY_NONE,
                  stopbits=serial.STOPBITS_ONE,
                  bytesize=serial.EIGHTBITS,timeout=3)

# Function to read Serial data
# Function name :read
# Input : ser
# Output : read_value
# Example call: read(ser)
def read(ser):
    read_value=""
    while True:
        ch=ser.read()          # reads the data
        read_value+=ch         # appends the string in read_value
```

```
        if ch=='\r' or ch='':
            return read_value
try:
    while True:
        ser.write("\r\nHi I am Raspbian Jessie:")
                                # writes the data to serial terminal
        re_va=read(ser)
        print("Received String :")
        ser.write("\r\n"+repr(re_va)) # writes the data received
        print (re_va)
except KeyboardInterrupt:
    pass
```

- In the Blueterm app on your Android device you will see the data transmitted by Raspberry Pi through HC-05 as:


Figure 6: [3]

- The data transmitted by the device will be received by the bluetooth module serially, which will then sent the data to the raspberry pi.

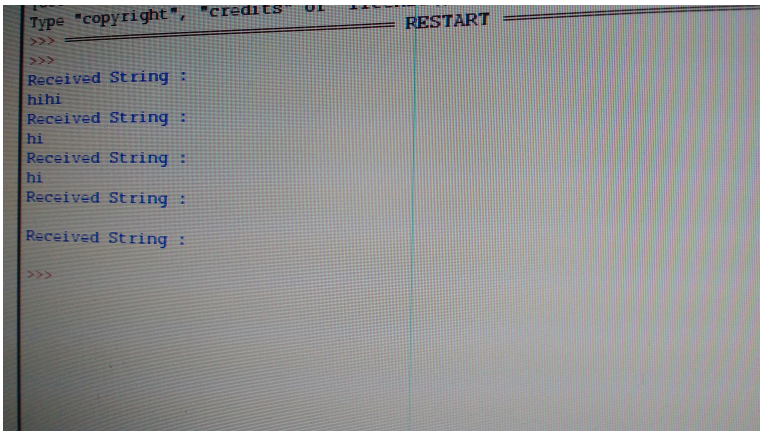


Figure 7: [3]

In this way the Data can be transmitted and received serially by the Raspberry Pi using Bluetooth module.

## 6.2 Communication between Rpi and PC using XBee Module:

- First of all download XCTU software on your PC and open it.

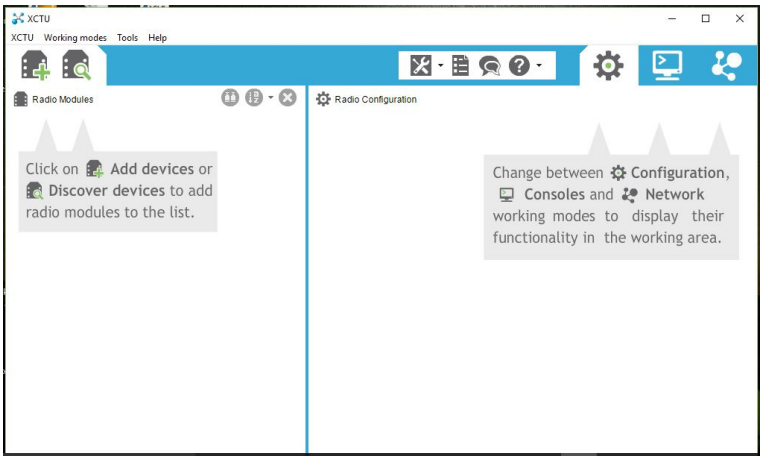


Figure 8: [3]

- Connect both the Xbee's to the PC one by one by USB jack(to configure Xbee's) and click on the Radio module on the extreme left of the window.A window named 'Add Radio Module' will occur on the screen select one Serial/USB port.

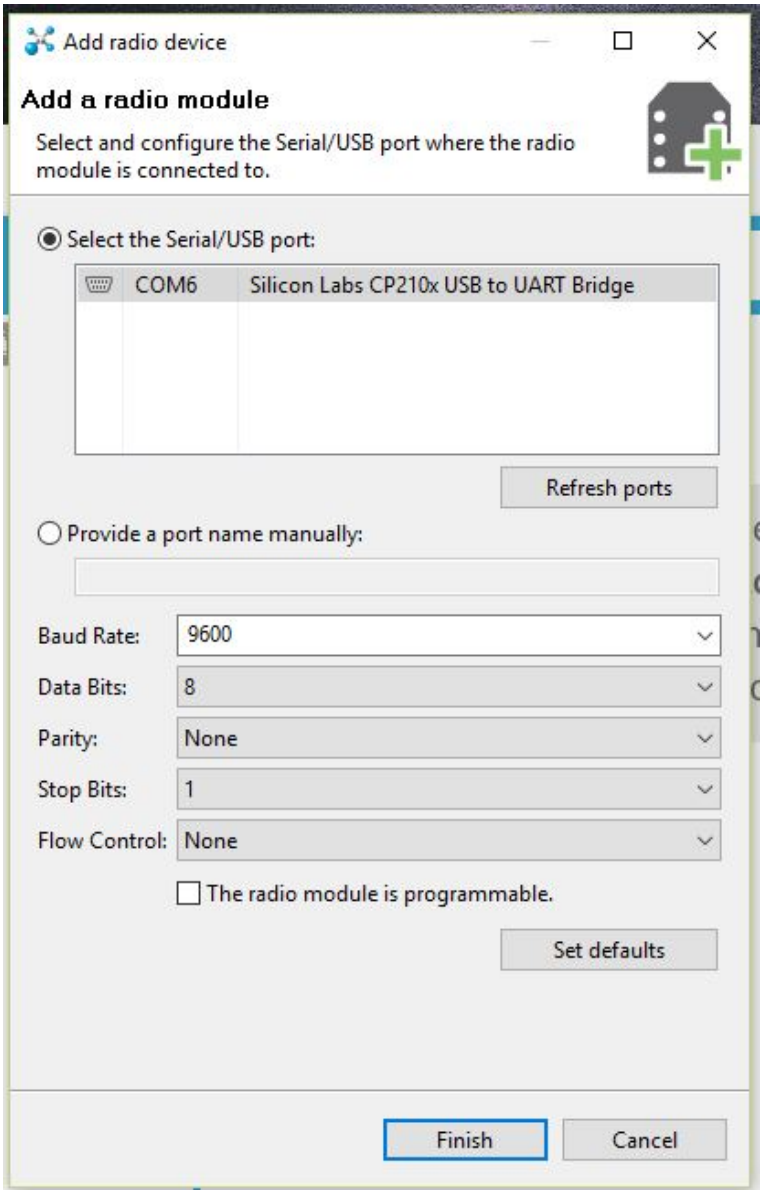


Figure 9: [3]

- On the left side of the window the radio module will be shown. Select the Radio module to display it's properties and configure it.

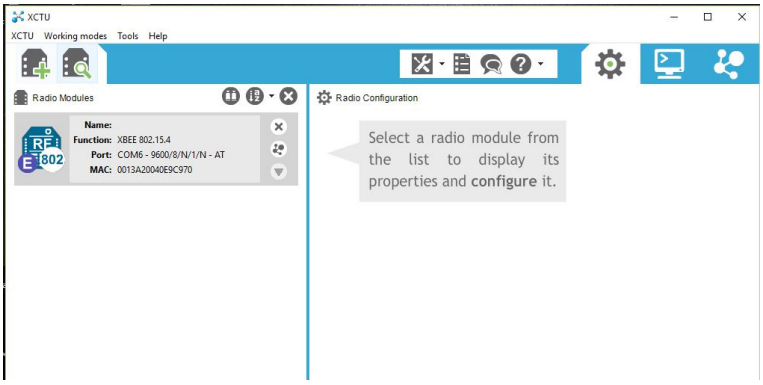


Figure 10: [3]

- Radio configuration window will open. First click on 'Default' to set all the parameters to default values.

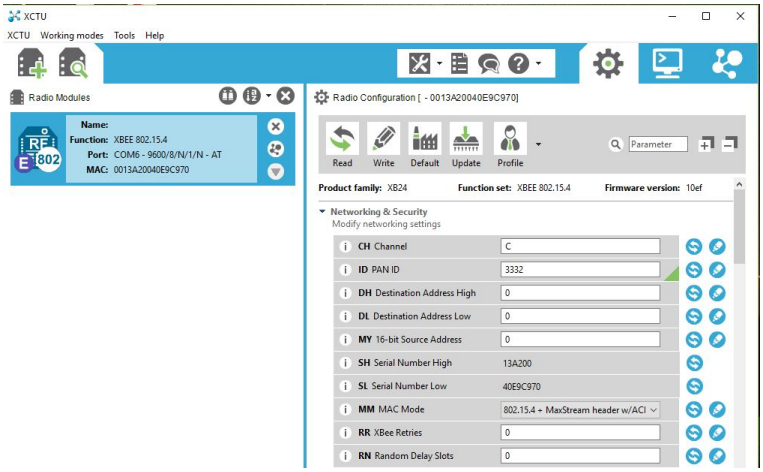


Figure 11: [3]

- Then change the PAN ID (if you want).

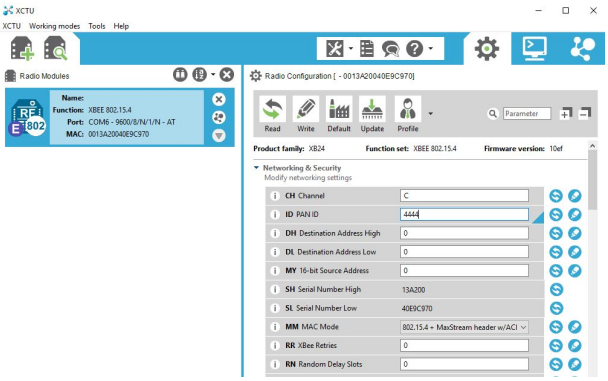


Figure 12: [3]

- Continue the same procedure with other XBee module also. Just keep in mind to set the PAN ID same as the first XBee module otherwise they will not communicate.
- Then disconnect one of the XBee from the PC and connect it to Raspberry Pi.
- Make the connections of XBee to the Raspberry Pi as shown below:

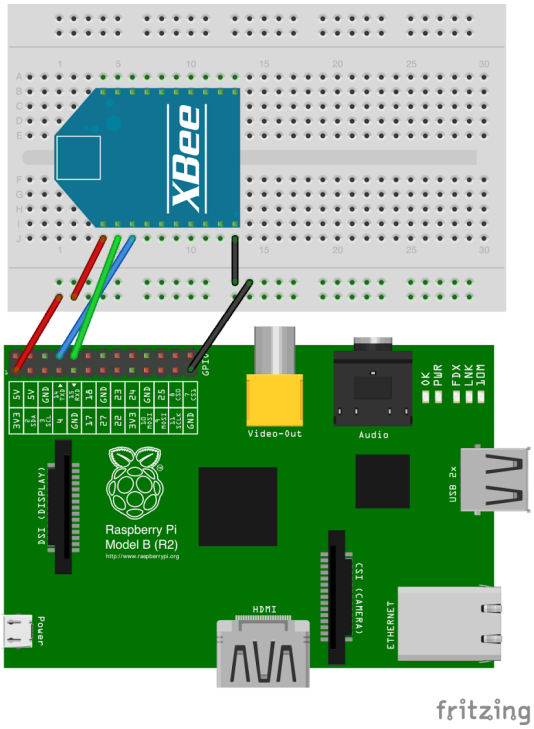


Figure 13: [3]

You can also directly use the USB jack to connect Xbee and Raspberry Pi.

- Run the same code as used for the BLuetooth module(HC-05).
- On the XCTU there is a option "close". This makes the connection between two XBee modules.
- The data transmitted by one XBee (connected to the RPi), will be received by second XBee serially and this will be shown on "Console Log" on XCTU as:

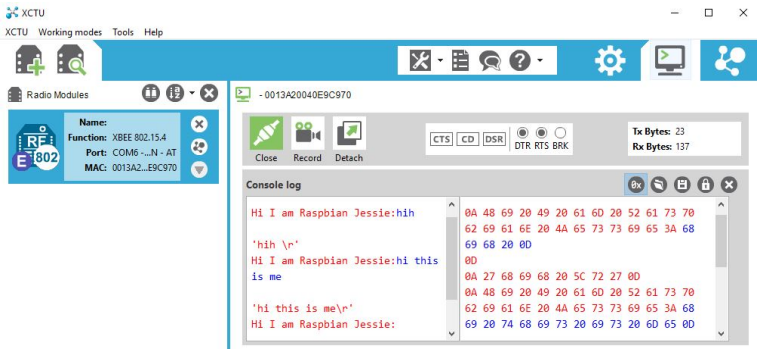


Figure 14: [3]

- Similarly the data sent by second XBee will be transmitted serially to first XBee connected to RPi.The data sent is as:

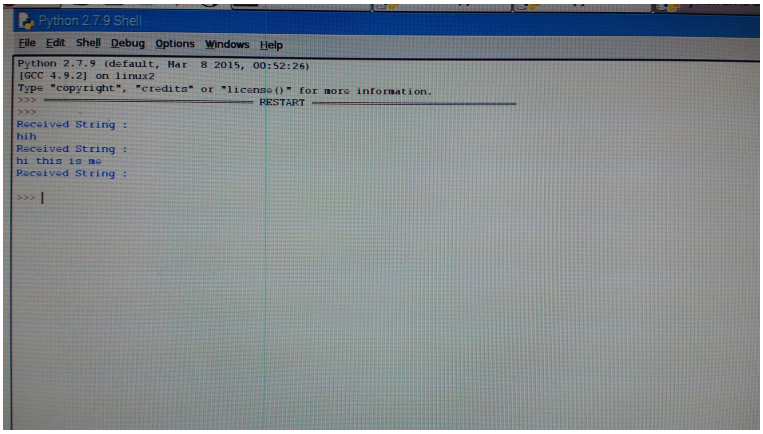


Figure 15: [3]

In this way the Data can be transmitted and received serially by the Rspberry Pi using Bluetooth module.
**NOTE:** if you are using USB jack then use **/dev/ttyUSB0** as the device path in the above given program, otherwise if you are connecting to the serial GPIO pins of Raspberry Pi(RXD & TXD) then use **/dev/ttyAMA0**

# 7 References

1. `http://www.elinux.org/Serial_port_programming`
2. `http://blog.miguelgrinberg.com/post/a-cheap-bluetooth-serial-port-for-your-raspberry-pi`
3. `https://en.wikipedia.org/wiki/XBee`