

## Node-Red: Verbindung von Webbrowser und serieller Schnittstelle (z.B. Xbee oder Arduino)

### Grundinstallation für Arduino und Xbee:

- **Arduino-Entwicklungsumgebung:** arduino-1.6.7-windows.exe
  - **Xbee-Treiber:** Digi\_USB\_RF\_Drivers.exe
  - **XCTU-Anwendung:** 40003026\_C.exe
- 

### Xbee-Konfiguration:

- Zwei Xbees: einer als **Router AT (oder End Device AT)** und einer als **Coordinator AT** IN DER JEWELNS AKTUELLSTEN FIRMEWARE
  - Beide Xbees in der gleichen PAN-ID,
  - Die die SH & SL von Xbee 1 wird eingetragen in die DH & DL von Xbee2 und umgekehrt
  - der Node-Name bei beiden Xbees wird gesetzt.
- 

### Node-Red-Installation

- **Python Version 2.10:** python-2.7.10.msi (**32 Bit Version**)  
,for all users' und ,add python.exe to path'
  - **Node.js:** node-v4.2.6-x86.msi (**32 Bit Version**)
  - **Node-Red als Plugin:** node-red-0.13.1.zip  
Installationsanweisung entsprechend <http://nodered.org/docs/>
    - a) Die zip-Datei im Explorer entpacken.
    - b) In das neu erstellte Verzeichnis gehen.
    - c) „Shift“-Taste halten + rechte Maustaste: *„Eingabeaufforderung hier öffnen“*
    - d) Es öffnet sich ein schwarzes cmd-Fenster. Hier eingeben: *„npm install“*
    - e) etwa 2 Minuten warten, bis die Installation fertig ist.
- 
- **Node-Red starten:**
    - a) Im Node-Red-Verzeichnis ein cmd-Fenster öffnen und eingeben *„node red“*. Bestätigen.
    - b) Die Firewall-Zugriffe alle erlauben.
    - c) In der letzten Zeile des cmd-Fensters ist zu lesen: <http://127.0.0.1:1880/> .  
Diese Adresse in den Browser eintippen:
    - d) Firefox starten, in die Adresszeile: <http://127.0.0.1:1880/>

## Grundkommunikation Xbee(Arduino) mit Xbee(Xbee-Dongle an PC):

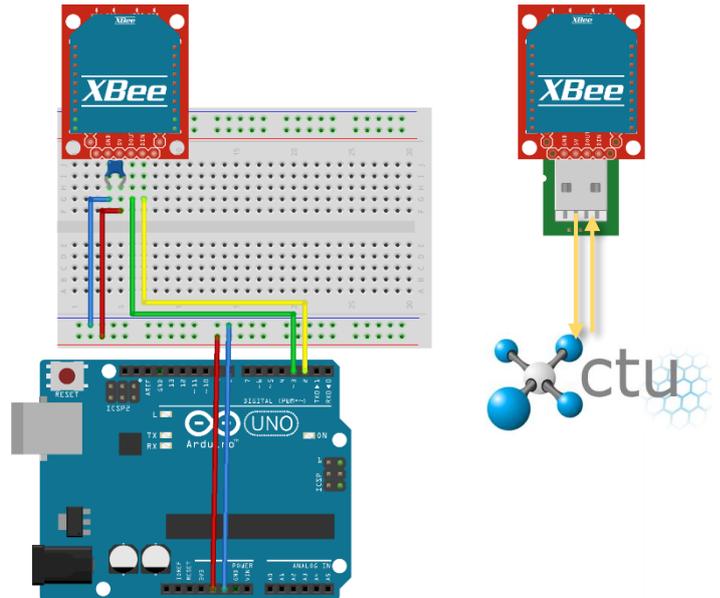
Sketch für den Arduino:

```
#include <SoftwareSerial.h>

SoftwareSerial xbeeComm(2, 3); // RX, TX
int teiler = 1;

void setup() {
  Serial.begin(9600);
  xbeeComm.begin(9600);
}

void loop() {
  if (xbeeComm.available()) {
    int xbeeInput = xbeeComm.read();
    Serial.println(xbeeInput);
    if(xbeeInput == 97){// Buchstabe a
      teiler = 1;
    }
    if(xbeeInput == 98){// Buchstabe b
      teiler = 1000;
    }
  }
  long SerialInput = millis();
  if (SerialInput % 1000 == 0) {
    xbeeComm.println(SerialInput/teiler);
  }
}
```



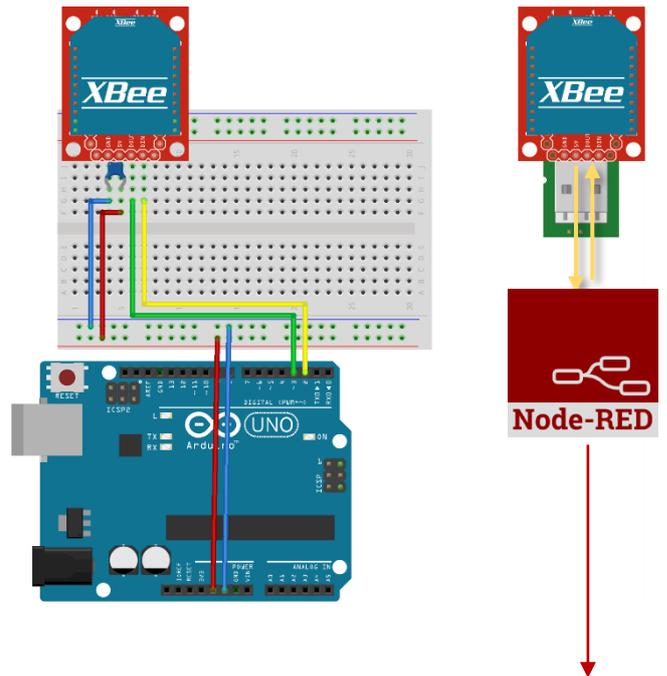
Der Xbee am Arduino sendet alle 1000 Millisekunden die Anzahl der vergangenen Millisekunden. Der XCTU-Xbee sendet per Nutzereingabe: Sendet man ein ‚a‘ wird der Befehl an den Arduino gesendet, die Millisekunden voll darzustellen, sendet man ein ‚b‘ werden die Millisekunden durch 1000 geteilt und nur volle Sekunden dargestellt.

## Grundinstallation für Xbee und NodeRed:

**NodeRed starten:** <http://127.0.0.1:1880/>

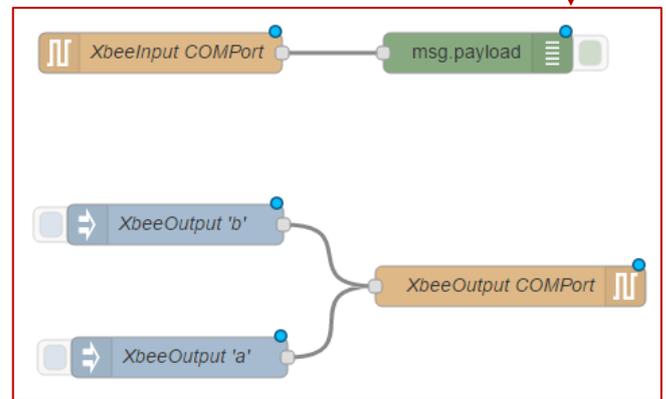
### Input-Definition:

1. Input-Serial-Node ins Fenster ziehen
2. Serial-Node konfigurieren: Com-Port suchen, Baudrate auf 9600, SplitInput auf „\n“
3. Debug-Node ins Fenster ziehen
4. Debug-Node konfigurieren: MessageProperty, msg.payload, to debugTab.



### Output-Definition:

5. Zwei Inject-Nodes ins Fenster ziehen
6. Payload „String“: **a** für die eine Node,
7. Payload „String“: **b** für die andere Node,
8. Output-Serial-Node ins Fenster ziehen



## Grundinstallation für Xbee, NodeRed und Mosquitto:

- a) Mosquitto auf Raspberry Pi: Installation und Konfiguration:

<http://www.switchdoc.com/2016/02/tutorial-installing-and-testing-mosquitto-mqtt-on-raspberry-pi/>

<http://www.kriwanek.de/homeautomation/mosquitto-mqtt/527-mqtt-broker-mosquitto-installieren.html>

- b) Subscription zum Mosquitto auf dem Raspberry:

```
mosquitto_sub -h [ipV4Adresse des Mosquitto-Servers] -t meinTopic/#
```