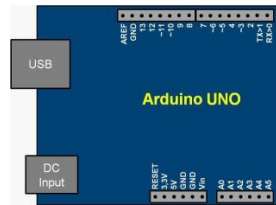
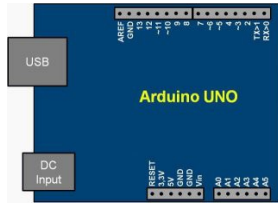


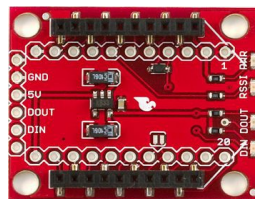
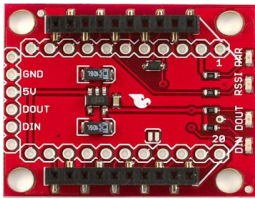
Es geht auch ohne Draht ...

Zwei Arduino´s sollten miteinander über Funk kommunizieren.

Das wurde gebraucht:



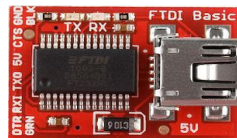
2 x Arduino UNO



2 x Explorer



2 x XBee



1 x FTDI



8 x Header

Hier die SparkFun-Liste:

DEV-11021 - Arduino Uno - R3

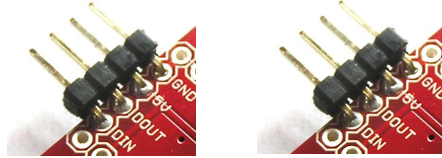
WRL-09132 - XBee Explorer Regulated

WRL-08666 - XBee 1mW U.FL Connection - Series 1 (802.15.4)

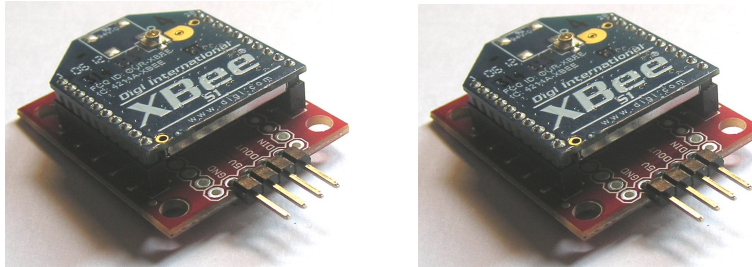
DEV-09716 - FTDI Basic Breakout - 5V

PRT-00553 - Break Away Male Headers - Right Angle

Der besseren Handhabung wegen erst einmal die Header an die Explorer löten.



Dann die XBee's mit den Explorern verbinden.



Und nun wird es interessant. Es wird noch ein Programm benötigt, mit dem die beiden XBee's so konfiguriert werden, dass sie sich miteinander verstehen.

Auf der Seite

<http://www.digi.com/support/productdetail?pid=3352&.osvid=0&type=cabling>

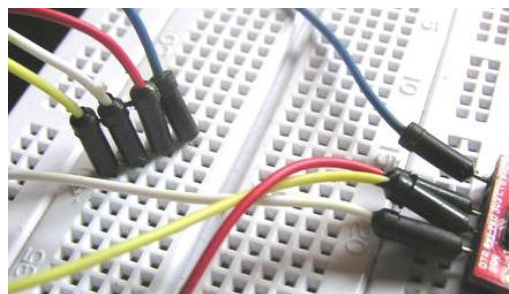
unter Drivers habe dieses ausgewählt:

[XBIB-U / PKG-U USB Drivers for Windows 98 SE - XP](#)

Nach der Installation hatte ich auf dem Desktop dieses Icon:



Damit das X-CTU-Programm ein XBee erkennen kann, muss eine Verbindung hergestellt werden zwischen dem XBee und dem PC. Dazu habe ich das FTDI-Modul, das über USB mit dem PC verbunden ist, auf ein Breadboard „herausgezogen“.

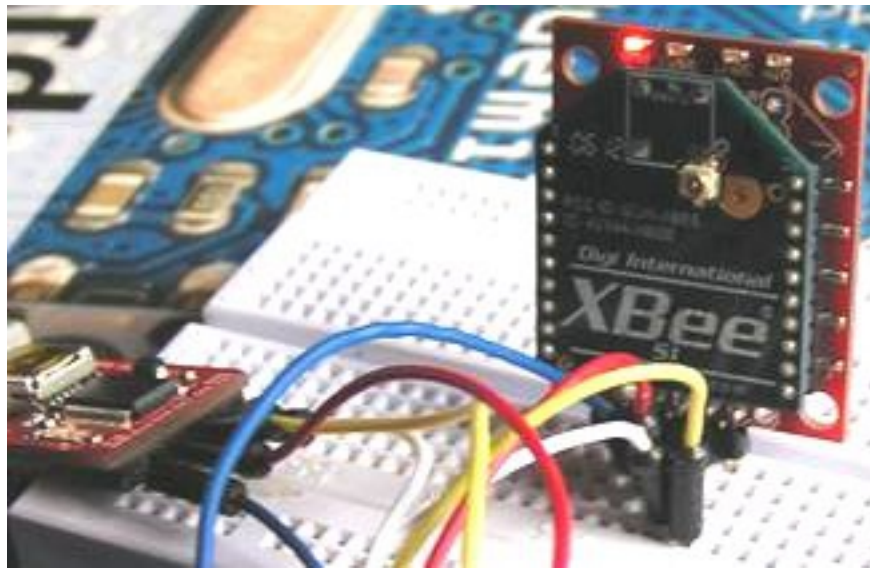


GND blau
+5V rot
TX0 gelb
RX1 weiss

Wie auf dem Bild zu erkennen ist, sind die Anschlüsse TX0 und RX1 auf dem Breadbord anders herum als am FTDI-Modul. Dies ist erforderlich, weil die Pin-Belegung am Explorer eigentlich die gleiche ist wie am FTDI-Modul, aber TX des einen Geräts immer an RX des anderen Geräts angeschlossen wird. Also:

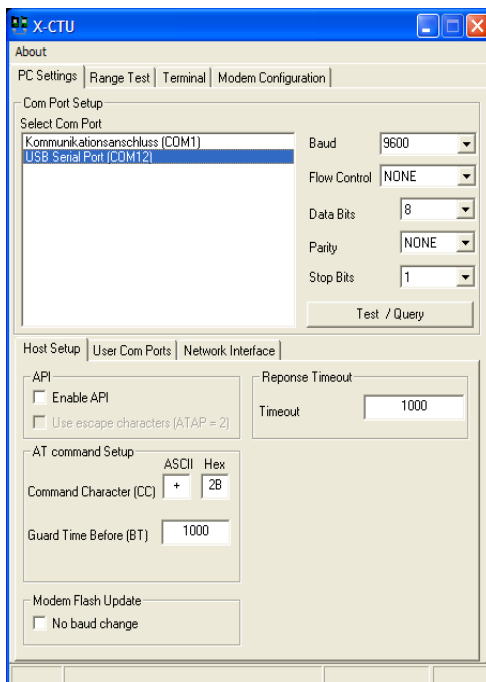
TX → IN
RX → OUT

Nachdem dieser Grundsatz erklärt und verstanden ist, kann das Gespann XBee/Explorer auf das Breadboard gesteckt werden. Das sieht dann so aus:

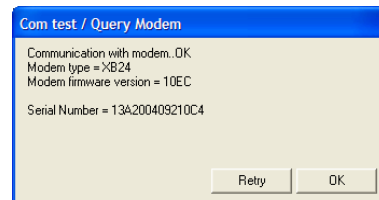


Bisschen unscharf, na ja, bin ja auch kein Industriefotograf.

Wie man sieht, leuchtet die LED. Es folgt der Klick auf



Standardmäßig verwendet X-CTU **COM12**. Das muss ausgewählt werden. Dann folgt der Klick auf **Test/Query**. Nach ein paar Sekunden sollte das gezeigt werden:



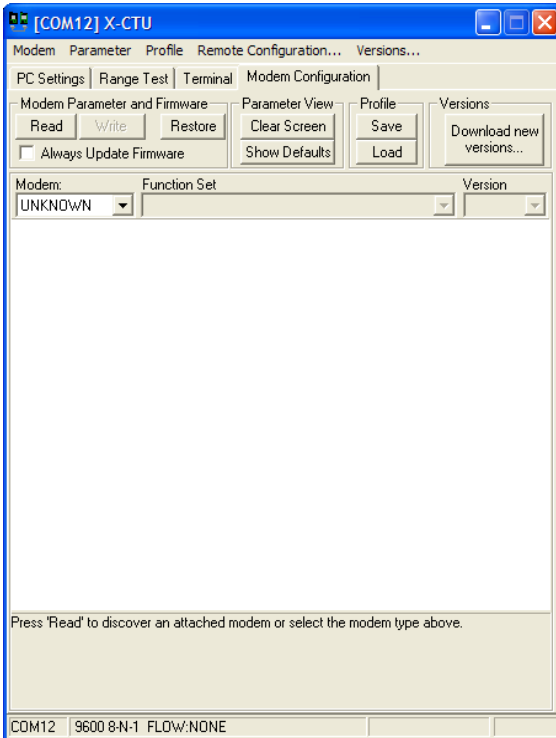
Wenn als **Modem firmware version** 10EC oder höher angezeigt wird, braucht man kein Update der Firmware vorzunehmen. Sollte dies aber nicht der Fall sein, dann gibt es am Schluss dieses Dokuments eine Anleitung für das Update.

XBee wurde also als solches erkannt und kann nun seiner Konfiguration entgegensehen.

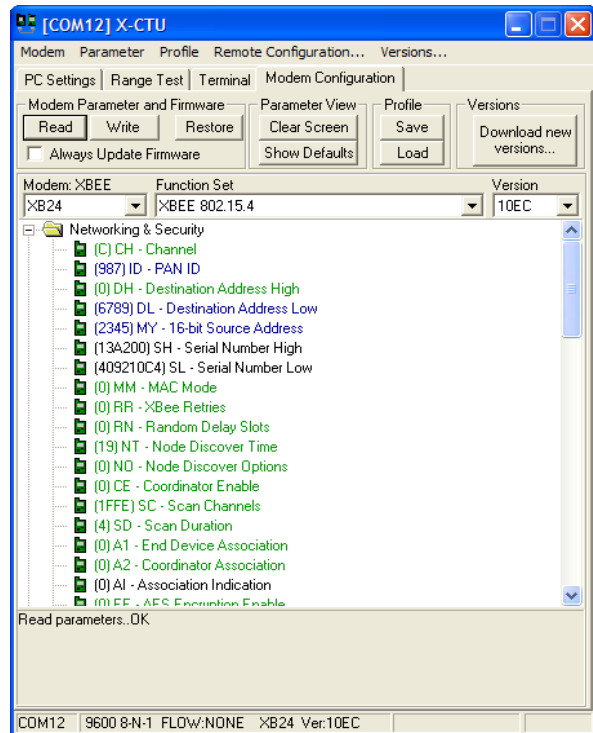
Dazu habe ich beide XBee's numeriert (Aufkleber, Edding, ...) und eine kleine Tabelle gefertigt.

	XBee 1	XBee 2
ATMY (Name des XBee)	2345	6789
ATDL (verbunden mit)	6789	2345
ATDH	0	0
ATID (Netzwerk)	0987	0987

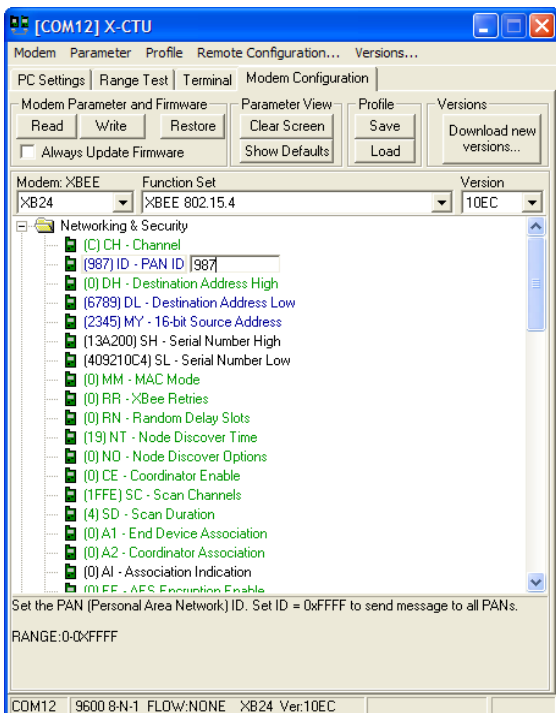
Die Werte aus der Tabelle werden nun für die Konfiguration eingetragen:



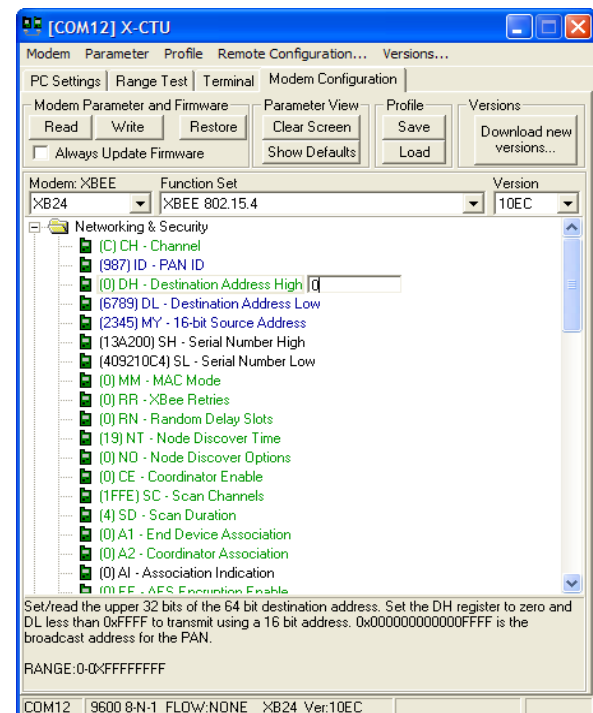
Modem Configuration anklicken.



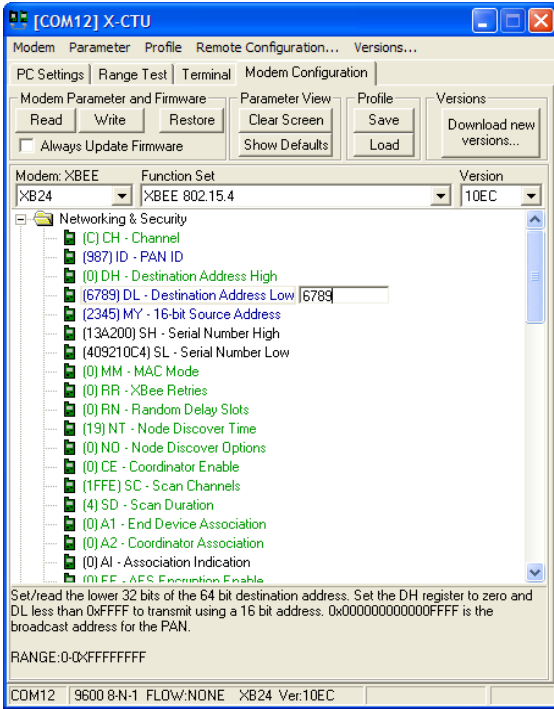
Read anklicken.



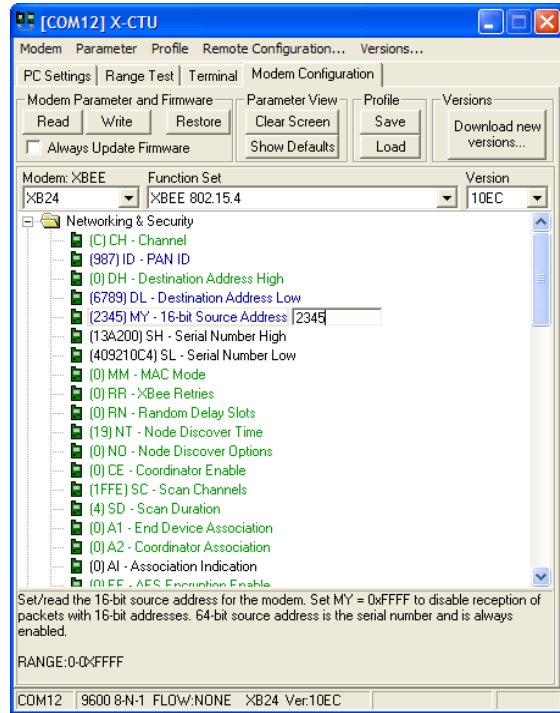
Bei PAN ID ATID eingeben.



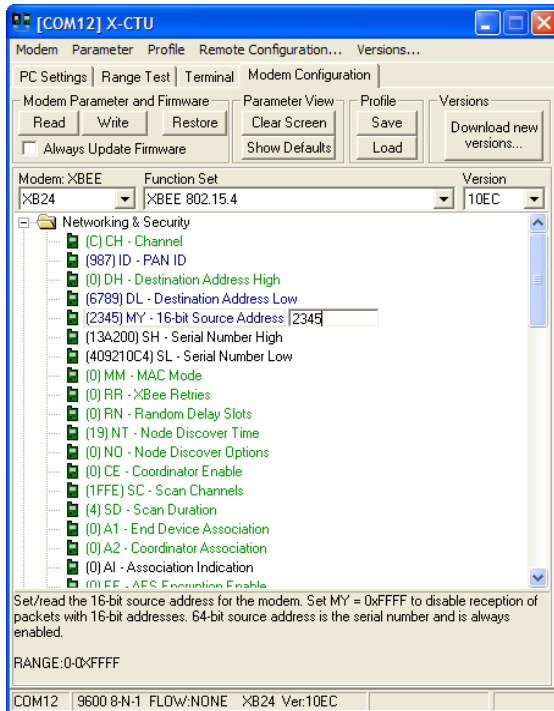
Bei Destination Address High ATDH eingeben.



Bei **Destination Adress Low** ATDL eingeben.



Bei **16-bit Source Adress** ATMY eingeben.



Nach gründlicher Kontrolle dessen, was eingetragen wurde, auf **Write** klicken. Die Parameter werden in das XBee übertragen. Das Programm X-CTU kann beendet werden.

Das war's. Jedenfalls für das eine XBee, die gleiche Prozedur muss, beginnend mit dem erneuten Start von X-CTU, für den zweiten wiederholt werden. Natürlich mit den für ihn vorgegebenen Werten.

Was noch fehlt bei der ganzen Geschichte, sind die Arduinos. Schließlich sollen die mit Hilfe der beiden XBee's Daten austauschen.

```

Sender | Arduino 0022
File Edit Sketch Tools Help
Sender$

int LED=13;
//+++++
void setup()
{
  Serial.begin(9600);
  pinMode(LED,OUTPUT);
}
//+++++
void loop()
{
  Serial.write('H');
  digitalWrite(LED,HIGH);
  delay(1000);
  Serial.write('L');
  digitalWrite(LED,LOW);
  delay(3000);
}

```

```

Empfaenger | Arduino 0022
File Edit Sketch Tools Help
Empfaenger

int LED=13;
int EMPFANG;
//+++++
void setup()
{
  Serial.begin(9600);
  pinMode(LED,OUTPUT);
}
//+++++
void loop()
{
  if(Serial.available()>0)
  {
    EMPFANG=Serial.read();
    if(EMPFANG=='H'){digitalWrite(LED,HIGH);}
    if(EMPFANG=='L'){digitalWrite(LED,LOW);}
    EMPFANG=0;
  }
}

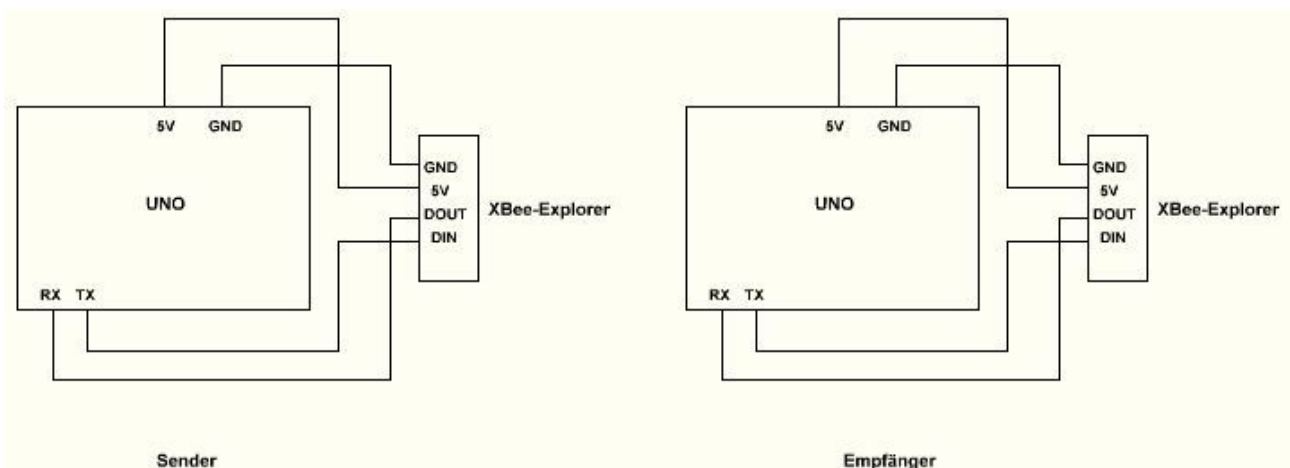
```

Das ist der Sketch für den Sender,

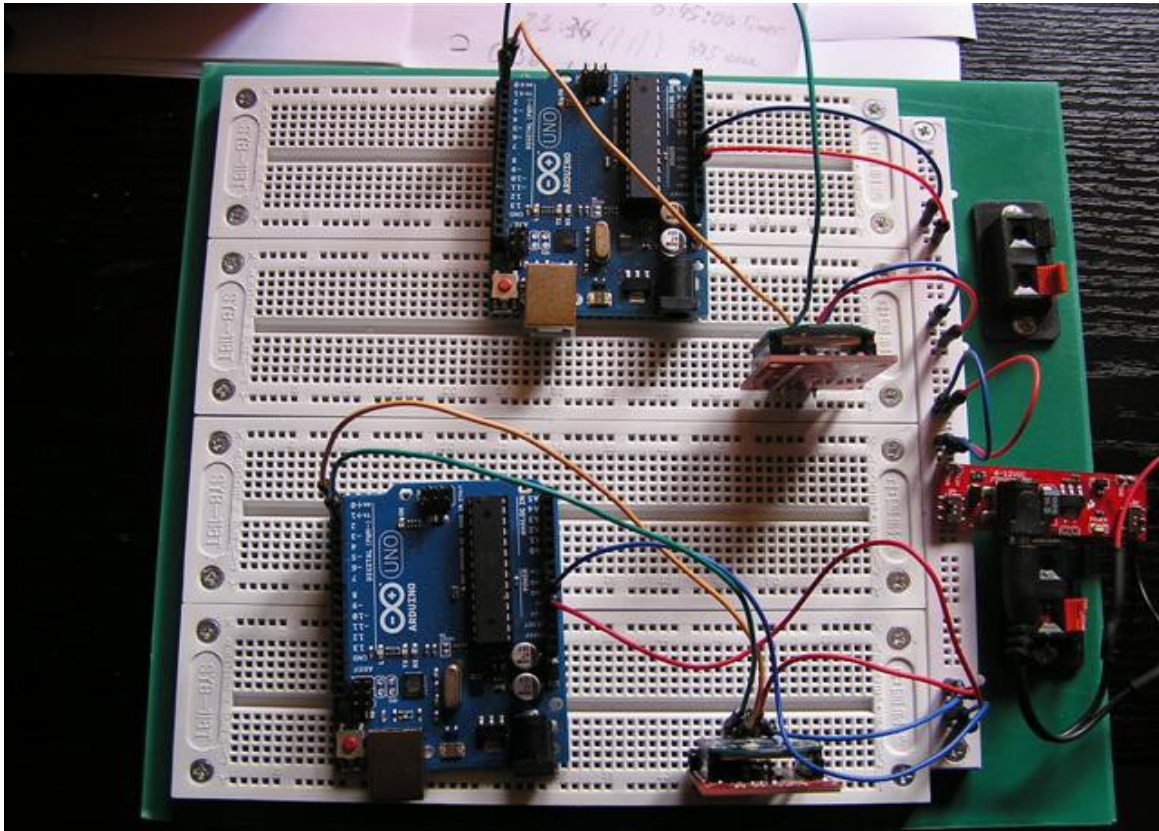
und hier der Sketch für den Empfänger.

Wenn die beiden Sketche auf die beiden Arduino's hochgeladen sind, müssen die XBee's noch mit dem jeweiligen Arduino verbunden werden.

Wichtig: Vor Upload eines Sketch müssen die Anschlüsse D0/RX und D1/TX des Arduino frei sein von jeder Beschaltung. Nach dem Upload kann das jeweilige Gerät wieder angeschlossen werden.

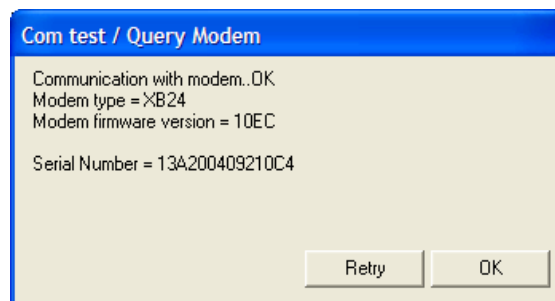


Der Versuchsaufbau



Hinweise zum Update der Firmware für XBee.

1. **Modem Configuration** anklicken.
2. **Read** anklicken.
3. Häkchen bei **Always Update Firmware** setzen.
4. **Write** anklicken. Das dauert dann eine Weile.
5. Das Häkchen bei **Always Update Firmware** entfernen.
6. **Read** anklicken. Nun sollte die aktuelle Firmware-Version vorhanden sein.



Die Firmware-Version sollte 10EC oder höher sein.