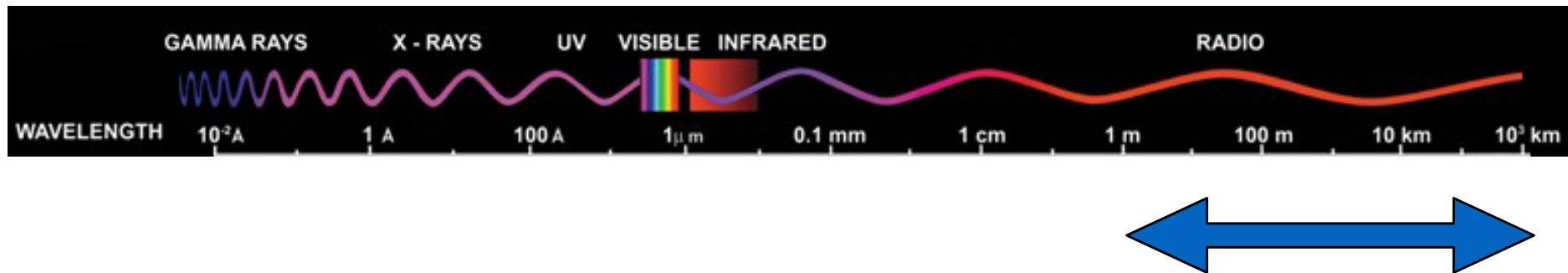


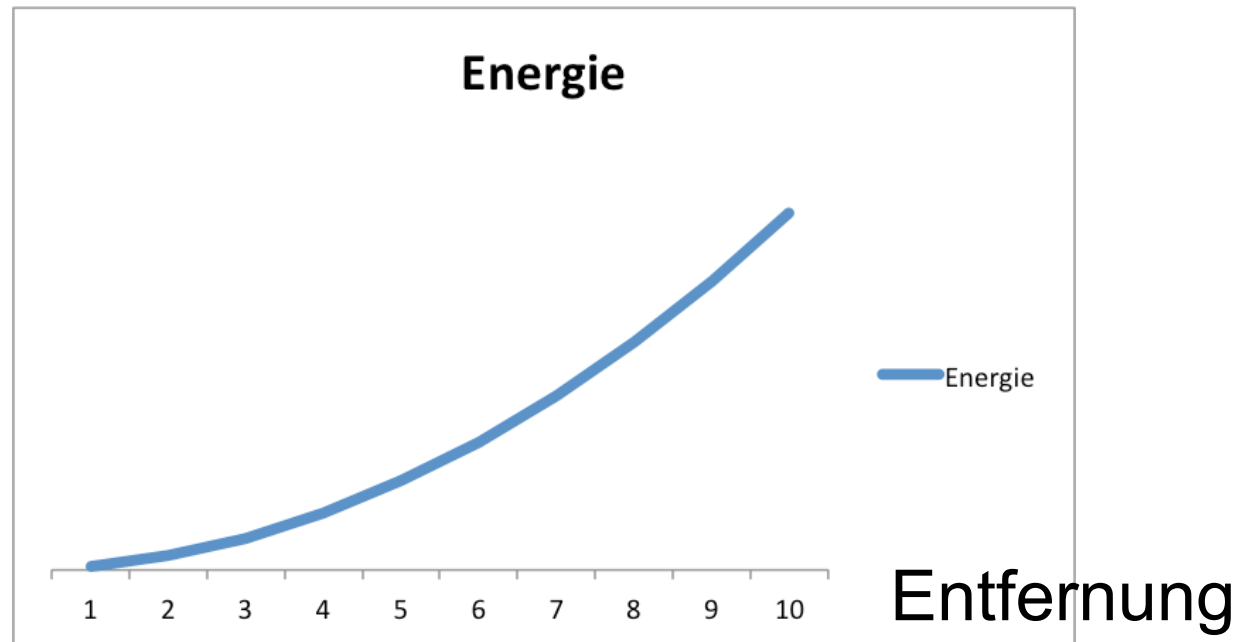
Drahtlose Kommunikation

Elektromagnetische Strahlung



Bei der drahtlosen Kommunikation hat sich vor allem der langwellige Bereich des elektromagnetischen Spektrums bewährt.

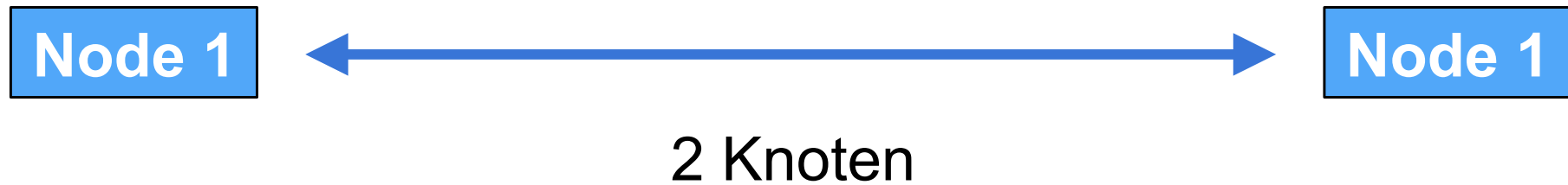
Eine wichtige Beobachtung



Bei der drahtlosen Kommunikation steigt der Energiebedarf mit zunehmender Entfernung nicht linear, sondern **mindestens** quadratisch an.

Multihop-Netzwerk

Entfernung: n



k Knoten

Entfernung: jeweils n/k

Energiebedarf (Beispiel)

1. Energiebedarf bei **direkter** Verbindung:

$$n*n$$

Energiebedarf bei Übertragung über k Knoten:

$$k(n/k)*(n/k)=\mathbf{n*n/k}$$

Der gesamte Energiebedarf lässt sich also durch Hops beliebig reduzieren.

Was ist der Nachteil?

Multihop-Netzwerk (Nachteile)

1. Bei der Übertragung müssen die Nachrichten $(k-1)$ mal wiederholt werden. Jeder Knoten muss also mehr Daten übertragen, die nicht für ihn bestimmt sind

Die Bandbreite der Knoten wird also reduziert.

2. Die Wahrscheinlichkeit, dass mindestens ein Knoten ausfällt, steigt rasant an.

Einige Standards

WiFi (IEEE-802.11)

- Keine Ad-Hoc-Vernetzung,
- Hoher Energiebedarf,
- nur Stern-Topologie,
- Access-Point ist Nadelöhr

Bluetooth (IEEE 802.15.1)

- Ad-Hoc-Vernetzung, Multihop möglich aber unüblich
- Nur Kleine Netzwerke möglich,
- Moderater Strombedarf

ZigBee (IEEE 802.15.4)

- Verschiedene Netzwerktypen (inkl. Ad-Hoc),
- Große Netzwerke möglich
- geringer Strombedarf

IEEE 802.15.4

In drahtlosen Sensornetzwerken ist Energie knapp
IEEE 802.15.4 wird häufig genutzt, da hier ein
energiegünstiger Betrieb möglich ist.

Nachteil: IEEE 802.15.4 ist unsicher, die Bandbreite ist
gering.

Produkte

Das per IEEE 802.15.4 spezifizierte Protokoll berührt nur die beiden untersten Ebenen des ANSI/OSI-Modells

Wir arbeiten hier mit dem Produkt **XBee Series 1**, das eine Implementierung dieses Standards enthält.



Verbesserungen

Aufbauend auf IEEE 802.15.4 gibt es noch weitere Protokolle, die im Industriestandard **ZigBee** definiert sind und eine komfortablere Arbeit in komplexen Szenarien mit verschiedenen Netzwerktopologien ermöglichen.

Das Produkt **XBee Series 2** ist eine Implementierung des ZigBee Standards.

Die Praxis

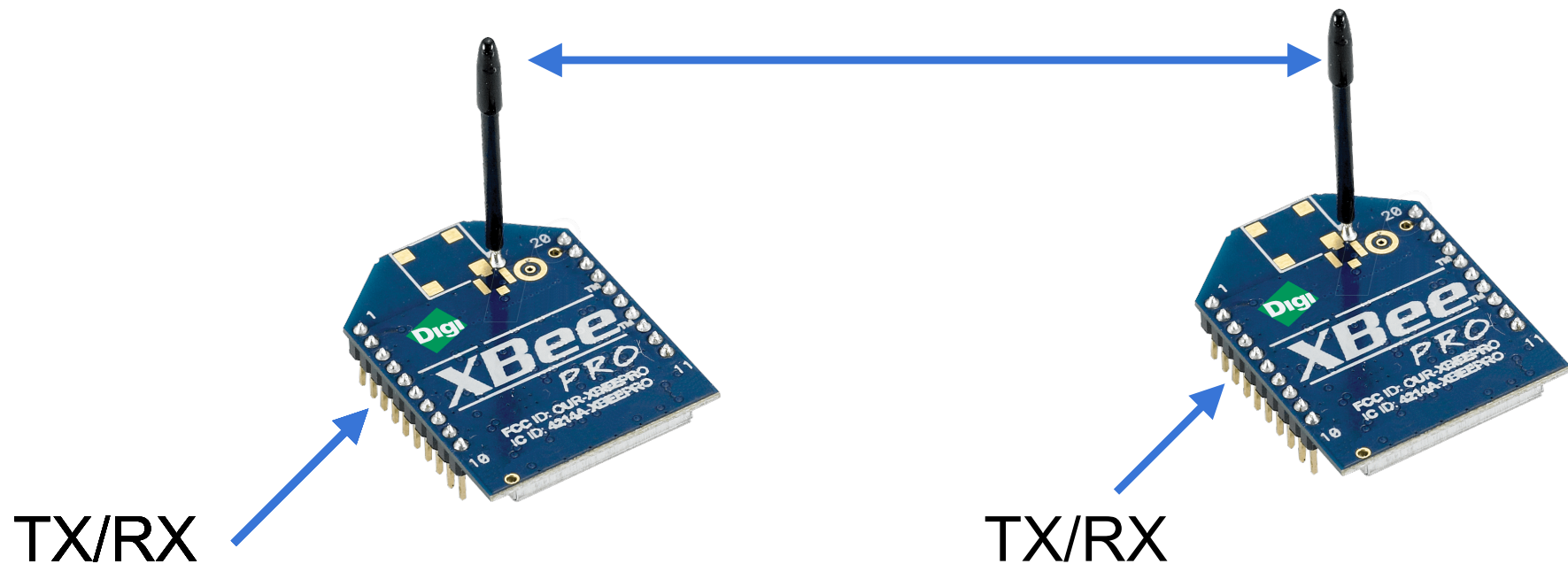
Man muss nicht viel wissen, um praktisch mit XBee-Modulen zu arbeiten.

Sie kennen zwei Betriebsmodi:

- den **Befehlsmodus** (für die Konfiguration) und
- den **Datenmodus** (zur Datenübertragung)

Datenübertragung

- Der Datenmodus ist der Standardmodus und wird für die Übertragung von Daten zu anderen XBee-Modulen genutzt.
- Die Module haben eine **serielle Schnittstelle**, die **wie gewohnt** zur Übertragung genutzt werden kann.



Der Explorer



Man kann die Module mit Hilfe eines XBee-Explorers über **USB** anschließen und dann etwa mit einem Terminal-Emulator oder eigener Software arbeiten, die die serielle Schnittstelle nutzt.

Die Konfiguration

- Der Befehlsmodus wird genutzt, um ein XBee-Modul über die serielle Schnittstelle zu konfigurieren.
- Man wechselt vom Datenmodus in den Befehlsmodus mit der Anweisung **+++**.
- **Wichtig** ist, dass diese Eingabe nicht mit der Enter-Taste abgeschlossen wird.
- Bei Erfolg sollte die Eingabe mit **OK** quittiert werden.

Die AT-Befehle

Im Befehlsmodus werden alle Befehle

- mit der Enter-Taste abgeschlossen.
- irgendwie vom XBee quittiert.

Die Syntax der Befehle orientiert sich sehr am Befehlssatz für Modems aus den 1980er Jahren. So beginnt jeder Befehl mit **AT** (attention)

Der Befehlsmodus wird verlassen

- automatisch nach 10 Sekunden (!) Inaktivität oder
- wenn der Befehl **atcn** abgesetzt wurde

Ein einfacher Use Case

Hier einige Beispiele:

Befehl	Anzeige
<code>atmy\n</code>	16-Bit-ID des XBee
<code>atid\n</code>	ID des zugehörigen Personal Area Networks (PAN)
<code>atsl\n</code>	Untere 32 Bit der 64-Bit-ID des XBee
<code>atsh\n</code>	Obere 32 Bit der 64-Bit-ID des XBee
<code>atdl\n</code>	Untere 32 Bit der 64-Bit-ID des XBee-Empfängers
<code>atdh\n</code>	Obere 32 Bit der 64-Bit-ID des XBee-Empfängers

Nachrichten adressieren

Ein XBee kann nur Nachrichten an die Adresse senden, die im ATDH/ATDL-Paar hinterlegt ist.

Dies kann sein

1. Die 16 Bit-ID des Empfängers (ATDL). In diesem Fall müssen auch die ID des PAN der beiden XBees übereinstimmen und ATDH auf 0 gesetzt sein.
2. Die 64 Bit-ID des Empfängers. Am Empfänger wird dann die 16-Bit ID auf FFFF gesetzt, um die 16-Bit Adressierung zu verhindern. Diese Variante ist vom PAN unabhängig!

IDs konfigurieren

Die Konfiguration der Module ist sehr einfach.
Ändern der 16-Bit ID auf 4711:

```
atmy4711\n
```

Ändern der ID des PAN auf 23:

```
atid 23\n
```

```
atdh 0\n
```

Die Konfiguration persistieren

Es muss nur beachtet werden, dass sämtliche Änderungen der Konfiguration mit

`atwr\n`

abschließend gespeichert werden.

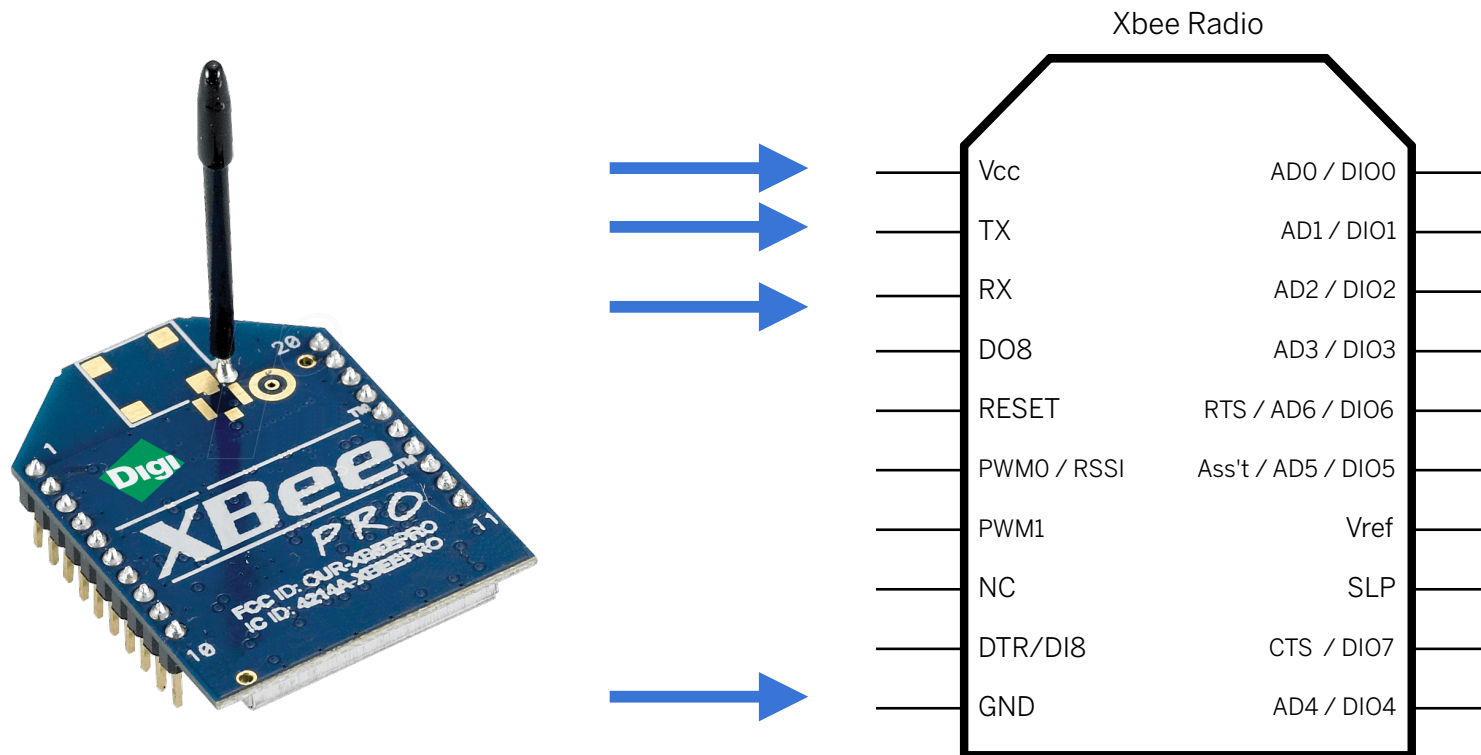
Tipp:

Nachdem Sie ein Modul vom Terminal getrennt haben, sollten Sie es erneut anschließen und die Konfiguration verifizieren!

Reset

Ein XBee-Modul wird mit
`at+re\r`
wieder auf die Werkeinstellungen gesetzt.

Die Verkabelung



Will man mit serieller Kommunikation arbeiten, müssen nur die markierten Anschlüsse verkabelt werden.

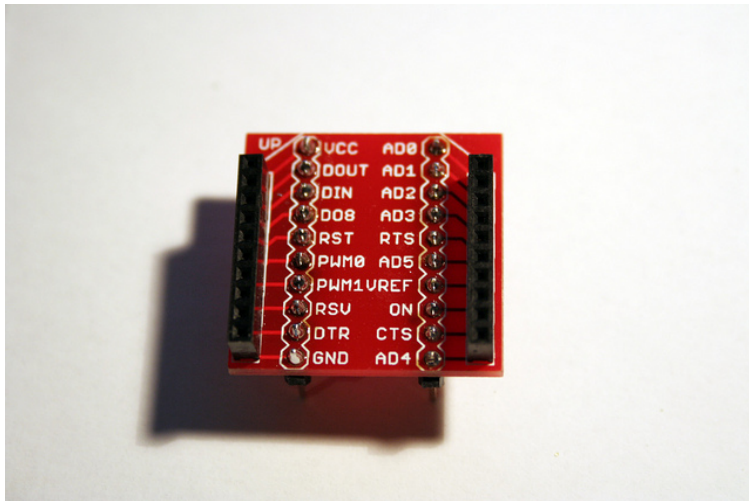
Die Verkabelung

Achtung:

Normalerweise führt RX des einen zum TX des anderen und umgekehrt.

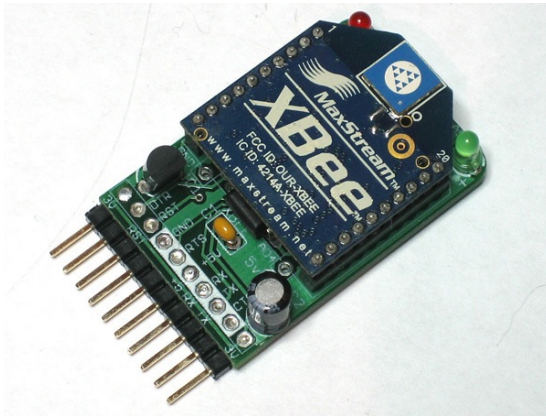
Das ist bei XBees nicht der Fall: RX führt zu RX und TX zu TX - warum auch immer!

Breakout-Boards



XBee passen nicht auf Steckbretter. Dazu verwendet man einen Adapter, ein so genanntes Breakout-Board

Adapter



Oftmals will man gar nicht sämtliche Anschlüsse des XBees nutzen. Hier bieten sich spezielle Adapter an.

Abgebildet ist ein Foca, der Stifte bietet, die auf ein Steckbrett oder in den Stecker eines passend konfektionierten Kabels passen.

Die Stifte versorgen aber nur einen Teil der Anschlüsse des XBees.

Praktikum