

## ZigBee Chat

## Ziel

## Schreiben Sie einen Chat Client.

Eingaben von der seriellen Konsole sollen per Broadcast an alle ZigBee Modems im gleichen PAN (Personal Area Network) weitergeleitet werden. Broadcast Nachrichten sollen, zusammen mit der 16bit Adresse des Senders, auf der Konsole ausgegeben werden (siehe Abbildung 1).

```
Welcome to Chatbot 3000 at address 2D9F
-----
5814: Hello World!
2D9F: Hello Yourself!
0: Oh, Hi!
5814: Coordinator? You're talking?!
```

Abbildung 1) Beispielausgabe auf der seriellen Konsole

Es werden ZigBee Modems der Firma Digi verwendet (siehe Abbildung 2 für das Pinout). Diese verwenden andere Pin-Abstände als unsere Steckbretter. Zudem benutzen sie CMOS Pegel von 3,3V statt der 5V TTL Pegel des Arduinos. Auch für das Flashen der Firmware wird zusätzliche Hardware benötigt. Aus diesem Grunde befinden sich die Modems auf UartSBee Breakout-Boards (Pinout siehe Abbildung 3).

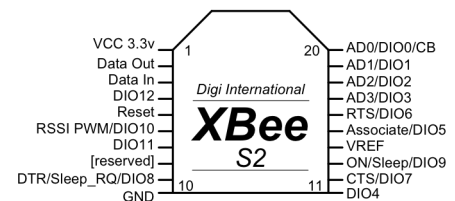


Abbildung 2) Pinout XBee

ZigBee PANs sind automatisch gebildete Netzwerke. Dazu übernimmt jedes Modem eine von drei Rollen: Coordinator, Router oder End Device. In dieser Übung ist Ihr Modem ein Router. Der Coordinator wird vom Labor gestellt und wir haben keine End Devices. Die Kommunikation zwischen Arduino und UartSBee / XBee erfolgt dabei über eine serielle Schnittstelle. Da wir für die Texteingabe und -ausgabe über USB die (Hardware) Serielle Schnittstelle des Arduinos an den Pins 0 und 1 benutzen, soll die „SoftwareSerial“ Bibliothek verwendet werden, mit der eine zweite Serielle Schnittstelle in Software implementiert werden kann. Diese Bibliothek gehört zum Standardumfang der Arduino Umgebung.

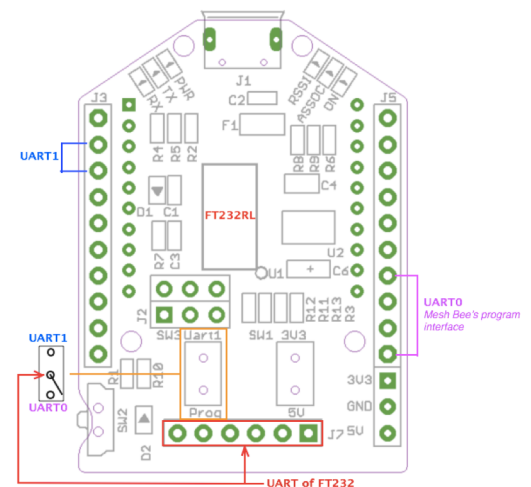


Abbildung 3) Pinout UartSBee v5

## Vorgehensweise

Bereiten Sie sich vor. Im Labor kontrollieren Sie zuerst die Schalter am UartSbee Breakout-Board: **SW3 muss auf UART1 und SW1 auf 5V** eingestellt sein. Nur in diesen Positionen kann das Board verwendet werden. Schließen Sie es nun über Micro-USB an den PC an, starten Sie XCTU und kontrollieren Sie die Konfiguration des Modems wie in Tabelle 1 zu sehen. Alle anderen Parameter sollen auf die Standardwerte g

<b>ID</b>	<b>C01A</b>
<b>JV</b>	<b>1</b> (Enabled)
<b>JN</b>	<b>1</b> (Enabled)
<b>NI</b>	<b>*XY</b> (Stern + Gruppen ID, z.B: <b>*01</b> )
<b>AP</b>	<b>2</b> (API enabled with escaping)

Tabelle 1) Parameter für XCTU

Die Verbindungseinstellungen sind 9600 Baud 8N1 - ohne Flow Control. Das Modem ist vom Typ **XB24-ZB**. Stellen Sie sicher, dass die Firmware aktuell und vom Typ **ZigBee Router API** ist.

Nach der Konfiguration können Sie das Micro-USB Kabel wieder trennen. Bauen Sie jetzt Arduino und UartSBee auf dem Steckbrett auf. Verwenden Sie dazu folgende Verbindungen: XBee DOUT / UartSBee TX1 an Arduino Pin Nr. 6 (für RX), XBee DIN / UartSBee RX1 an Arduino Pin Nr. 5 (für TX), wichtig ist auch eine gemeinsame Erdung von Arduino und UartSBee. Schließen Sie den Arduino dann an den PC an.

Installieren und verwenden Sie die XBee-Arduino Bibliothek. Schreiben Sie nun ein Programm, das sowohl mit **Newline** terminierte Eingaben von der seriellen Konsole entgegennimmt und per API Modus an das angeschlossene Modem weitergibt (siehe **Series2\_Tx** Beispiel), als auch eingehende Nachrichten auf der Konsole ausgibt (siehe **Series2\_Rx\_Nss** Beispiel). Stellen Sie vor dem Starten des Chat-Programms die Adresse des lokalen ZigBee Modems durch Absetzen eines **AT MY** Kommandos fest (siehe **AtCommand** Beispiel).

### Vorbereitung

Lesen Sie die Dokumentation der XBee-Arduino Bibliothek (in der Zip Datei). Lesen Sie die Beispiele und entwerfen Sie Quelltext, der alle notwendigen Funktionen vereint. Studieren Sie das Verhalten von Serial und SoftwareSerial hinsichtlich der Blockierung, bzw. des Rückkehrverhaltens. Beschäftigen Sie sich mit den ZigBee Grundlagen und der zu verwendenden Hardware. Beschäftigen Sie sich mit der Terminierung von Strings in C.

### Achtung

Beim Konfigurieren der XBee Modems über XCTU müssen die RX/TX Leitungen zwischen Arduino und UartSBee getrennt sein – sonst kann es zu Problemen kommen, da beide gleichzeitig auf UART1 schreiben könnten. Beim Lesen von der seriellen Schnittstelle können nur Teile einer Nachricht verfügbar sein. Sie müssen gegebenenfalls über mehrere Lesezugriffe puffern. Übertragene Chat-Nachrichten sollen maximal 40 Zeichen lang sein, stellen Sie sicher, dass es keine Fehler bei überlangen Nachrichten gibt. Achten Sie auf die richtige Adressierung Ihrer ZigBee Pakete (Broadcast). Stellen Sie die serielle Konsole auf Termination einer Zeile mit Newline ein.

### Notengebung

4,0 (Anwesend); 3,0 (Nachrichten werden empfangen); 2,3 (+ Nachrichten werden gesendet); 2,0 (+ Stabil, Integration mit Konsole); 1,7 (+ Kennt seine eigene Adresse); 1,3 (+ Code sauber geschrieben und dokumentiert); 1,0 (+ Robust, Fehler werden erkannt, keine Probleme mit überlangen Nachrichten)

### Wichtige Funktionen & Bibliotheken

- Die SoftwareSerial Bibliothek (bei Arduino mitgeliefert)
- Die XBee-Arduino Bibliothek (muss installiert werden)

### Sie brauchen

- Arduino, UartSBee mit ZigBee Modem, USB Kabel, Micro-USB Kabel, Steckbrücken
- XBee-Arduino Bibliothek: <https://github.com/andrewrapp/xbec-arduino>
- UartSBee Dokumentation: [http://www.seeedstudio.com/wiki/UartSBee\\_v5](http://www.seeedstudio.com/wiki/UartSBee_v5)
- SoftwareSerial Referenz: <https://www.arduino.cc/en/Reference/SoftwareSerial>