

# Projektdokumentation – Fernsteuerung AUV

-GEOMAR Helmholtz-Zentrum für Ozeanforschung Kiel-

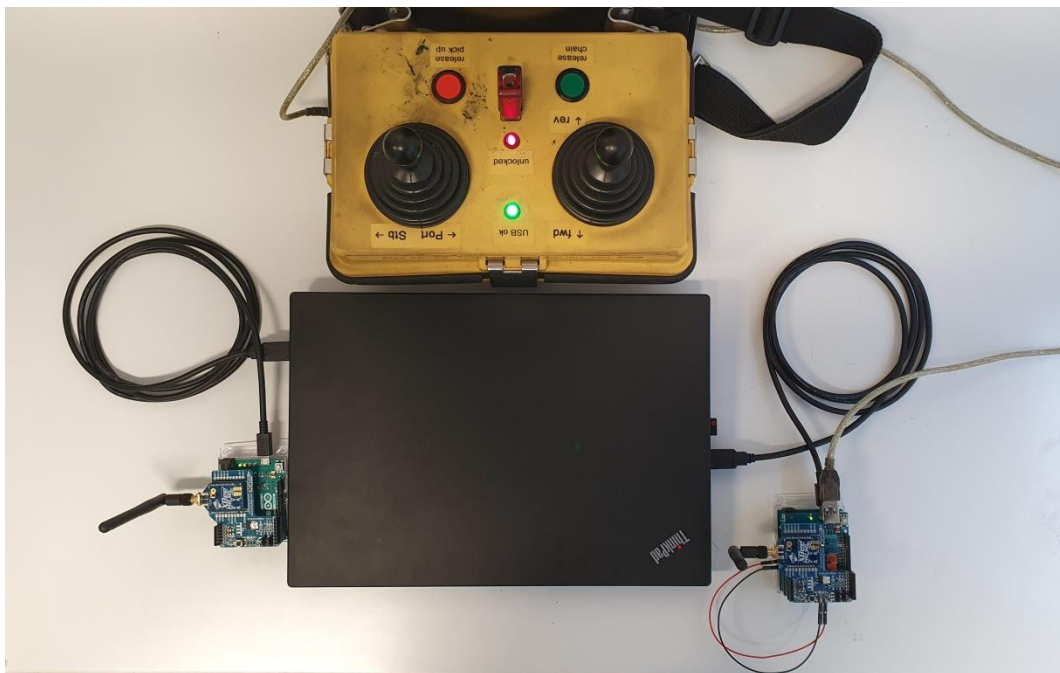


Franz Meißner

Emanuel Wenzlaff

## Idee

Ausgangspunkt für die Fernsteuerung des AUV bildet ein Bauchladen. Dieser beinhaltet zwei Joysticks für das Verfahren in X- und Y-Richtung sowie zwei weitere Taster. Die Positionen der beiden Joysticks und Taster-Stellungen werden per Funk an das AUV gesendet und dort ausgewertet.

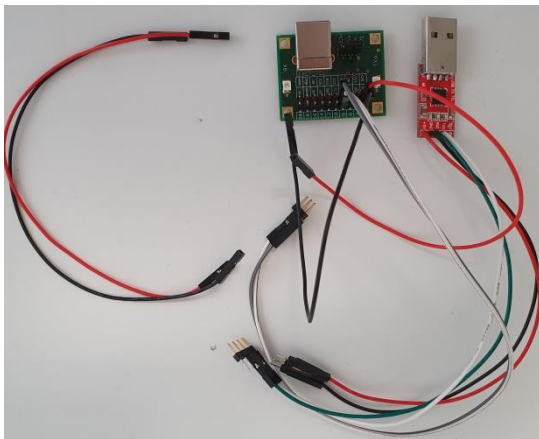


## **Inhaltverzeichnis**

1. Material .....	3
2. XBee .....	4
2.1 PC-Verbindung .....	4
2.2 Einrichtung Parameter .....	5
3. Arduinio.....	8
3.1 Software und In Betriebnahme.....	8
3.2 Bibliotheken einbinden .....	8
3.3 Programm Sender .....	9
3.4 Programm Empfänger .....	9
4. Gesamter Aufbau .....	10
4.1 Sender.....	10
4.2 Empfänger .....	12
5. Funktionsprüfung.....	13
6. Fehlerbehebung.....	14
6.1 XBee-Einrichtung.....	14
6.2 Bootloader Arduino.....	15

## 1. Material

- 2 Arduino Leonardo
- 2 XBee Shield + XBee PRO S2C
- 1 USB Host Shield 2.0
- 1 Joystick
- 2 Kabel rot/schwarz
- 2 Kabel USB zu Micro USB B
- 1 Kabel USB zu USB B
- 1 Kabel USB-Verlängerung
- 2 Converter Serielle Schnittstelle

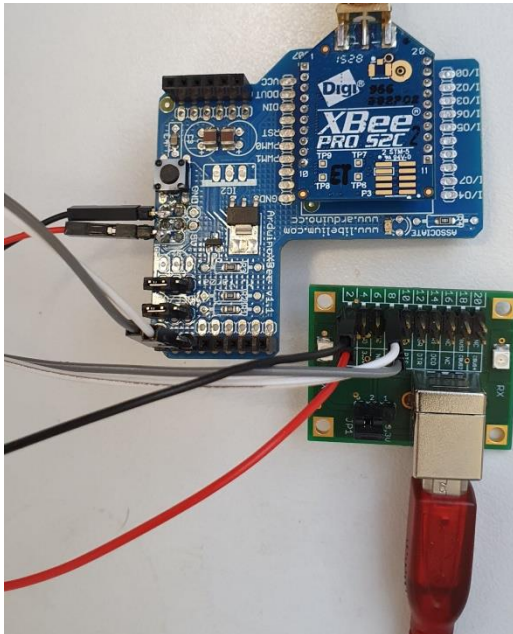


## 2. XBee

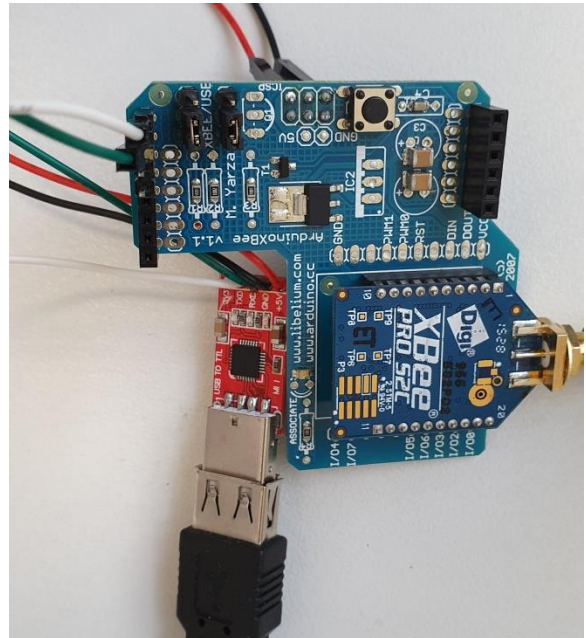
### 2.1 PC-Verbindung

Zu Beginn müssen die XBee an den PC angeschlossen werden zu dessen weiterer Konfiguration.

XBee-Sender



XBee-Empfänger

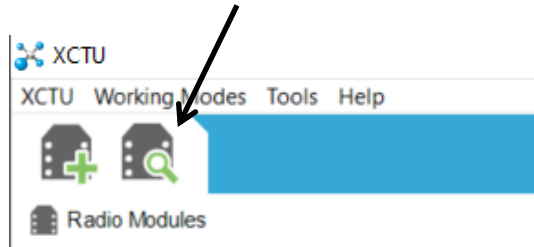


Achtung es müssen die Jumper auf die Position USB gesetzt werden

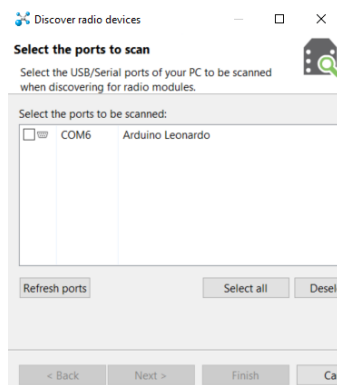


## 2.2 Einrichtung Parameter

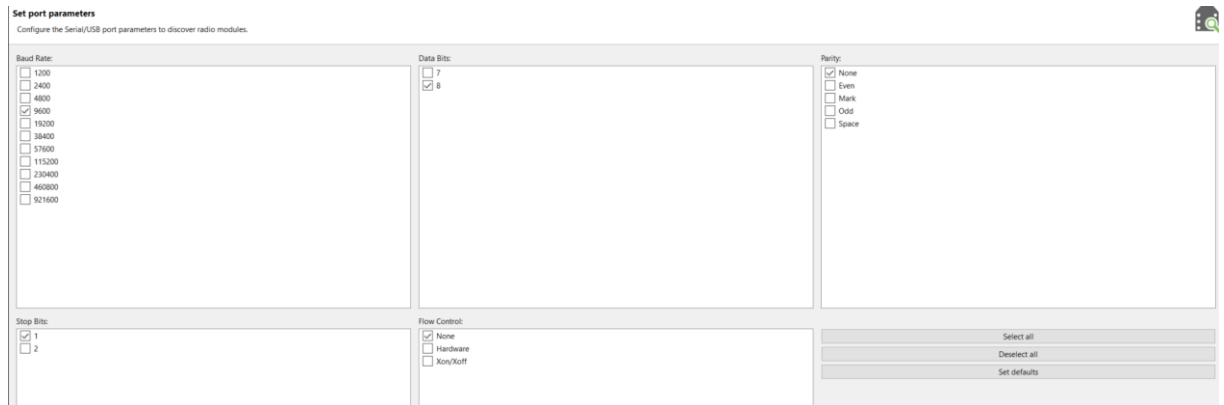
1. XBee-Software herunterladen (aktuellste Version)  
→ <https://www.digi.com/products/xbee-rf-solutions/xctu-software/xctu>
2. Anschließend auf das Symbol „Discover radio devices“ klicken



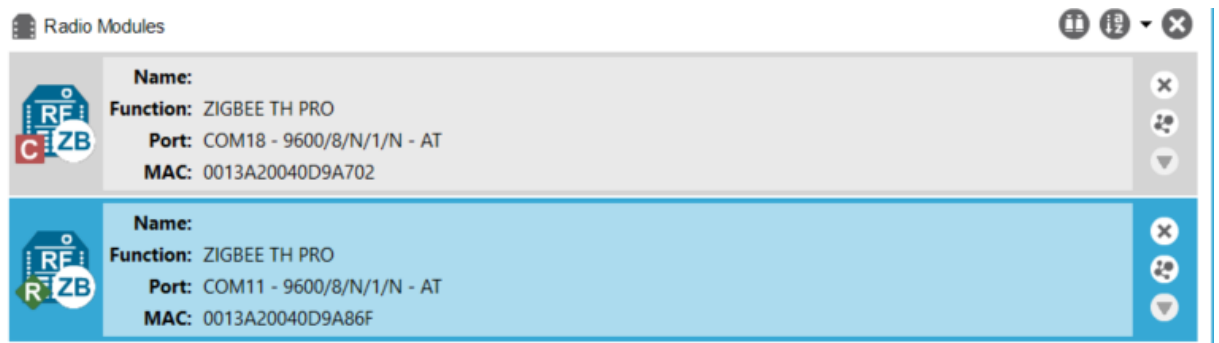
3. Es öffnet sich ein Fenster in welchem die COM-Ports auszuwählen sind



4. Anschließend auf „next“ klicken, woraufhin sich ein weiteres Fenster „Set port Parameters“ öffnet → folgende Einstellungen sind zu wählen
  - Baud Rate: 9600
  - Data Bits: 8
  - Parity: None
  - Stop Bits: 1
  - Flow Control: None



5. Die Bestätigung der Auswahl erfolgt über „finish“. Ein Fenster öffnet sich und es werden automatisch die aktuellen Geräte gesucht und können abschließend hinzugefügt werden. Es sollten daraufhin beide XBee oben links sichtbar sein.



6. Es müssen nun beide XBee konfiguriert werden. Hierzu auf den das jeweilige Modul anklicken und rechts daneben öffnet sich ein Fenster zum Setzen manueller Einstellungen.

XBee Sender:

- ➔ Es ist die PAN-ID einzustellen: 2000 (frei wählbar)
- ➔ Es muss unter CE Coordinator Enable: Enabled[1] gewählt werden
- ➔ Baud-Rate: 9600
- ➔ Änderungen werden mithilfe des Buttons „Write“ auf das Board geschrieben, zur Kontrolle anschließend mittels „Read“ Werte nochmal auslesen und gegenprüfen

The screenshot shows the configuration interface for the XBP24C module. The top bar includes icons for Read, Write, Default, Update, and Profile, along with a search bar labeled 'Parameter'. Below the top bar, the 'Product family' is XBP24C, the 'Function set' is ZIGBEE TH PRO, and the 'Firmware version' is 4061.

The 'Networking' section is expanded, showing 'Change networking settings'. The settings are as follows:

Parameter	Value	Unit/Type
ID PAN ID	2000	
SC Scan Channels	7FFF	Bitfield
SD Scan Duration	3	exponent
ZS ZigBee Stack Profile	0	
NJ Node Join Time	FF	x 1 sec
NW Network Watchdog Timeout	0	x 1 minute
JV Channel Verification	Disabled [0]	
JN Join Notification	Disabled [0]	
OP Operating PAN ID	2000	
OI Operating 16-bit PAN ID	9201	
CH Operating Channel	C	
NC Number of Remaining Children	14	
CE Coordinator Enable	Enabled [1]	
DO Device Options	0	Bitfield
DC Device Controls	0	Bitfield

The 'Serial Interfacing' section is also expanded, showing 'Change modem interfacing options':

Parameter	Value
BD Baud Rate	9600 [3]

### Achtung: Es müssen die PAN-ID bei beiden XBee-Modulen übereinstimmen

XBee-Empfänger:

- ➔ Es ist die PAN-ID einzustellen: 2000 (frei wählbar)
- ➔ Es muss unter CE Coordinator Enable: Disabled[0] gewählt werden
- ➔ Baud-Rate: 9600
- ➔ Änderungen werden mithilfe des Buttons „Write“ auf das Board geschrieben, zur Kontrolle anschließend mittels „Read“ Werte nochmal auslesen und gegenprüfen

This screenshot shows the same configuration interface as above, but with updated values for the 'Networking' settings. The 'CE Coordinator Enable' is now set to 'Disabled [0]'.

Parameter	Value	Unit/Type
ID PAN ID	2000	
SC Scan Channels	7FFF	Bitfield
SD Scan Duration	3	exponent
ZS ZigBee Stack Profile	0	
NJ Node Join Time	FF	x 1 sec
NW Network Watchdog Timeout	0	x 1 minute
JV Channel Verification	Enabled [1]	
JN Join Notification	Disabled [0]	
OP Operating PAN ID	0	
OI Operating 16-bit PAN ID	FFFF	
CH Operating Channel	0	
NC Number of Remaining Children	14	
CE Coordinator Enable	Disabled [0]	
DO Device Options	0	Bitfield
DC Device Controls	0	Bitfield

The 'Serial Interfacing' section remains the same:

Parameter	Value
BD Baud Rate	9600 [3]

### **3. Ardunio**

#### **3.1 Software und In Betriebnahme**

1. Arduino-Software herunterladen  
→ <https://www.arduino.cc/en/Main/Software>
2. Software starten und Arduino anschließen
3. In der Software unter Werkzeuge → Board das richtige Board auswählen

Wichtig: „Arduino Leonardo“

4. In der Software unter Werkzeuge → Port den richtigen COM-Port auswählen

Wichtig: Insbesondere, falls das Board zwischenzeitlich vom PC getrennt werden sollte ist hier die Richtigkeit des Ports zu prüfen, da die Software den Port nach zufälligem Schema neu benennt

5. Hochladen des aktuellen Sketches über den Button „Hochladen“

#### **3.2 Bibliotheken einbinden**

Die jeweiligen Programme der Arduinos greifen in ihrem Code auf bereits vorher importierte Bibliotheken zu. Für den Code des Senders sind alle benötigten Bibliotheken bereits bei der standardmäßigen Installation der Arduino-Software enthalten. Wichtig ist es jedoch benötigte Bibliotheken für den Empfänger im Vorhinein zusätzlich zu installieren. Hierzu folgenden Links benutzen:

→ <https://github.com/MHeironimus/ArduinoJoystickLibrary/tree/version-1.0>

Dort den Ordner „Joystick“ herunterladen

Um den Ordner erfolgreich in der Arduino-Software einzubinden gibt es zwei unterschiedliche Möglichkeiten aufgeführt in den PDF-Dateien:

→ „Bibliotheken\_als\_ZIP\_Datei\_einbinden“

→ „Bibliotheken\_Manuell\_einbinden“



Mehr Informationen unter:

- <https://www.arduino.cc/en/Guide/Libraries#how-to-install-a-library>
- <http://www.instructables.com/id/Arduino-LeonardoMicro-as-Game-ControllerJoystick/>

### **3.3 Programm Sender**

1. Das Programm „Programm\_Sender“ zum Auslesen des Joysticks öffnen und anschließend nach obigem Schema auf den ersten Arduino laden.

Wichtig: Im selben Ordner in welchem sich das Programm befindet, müssen sich zusätzlich eine CCP-Datei und eine h-Datei mit dem Namen „hidjoystickrptparser“ befinden.

2. Alle weiteren Erläuterungen sind dem Programm-Code als Kommentar beigelegt.

### **3.4 Programm Empfänger**

1. Das Programm „Programm\_Empfänger“ öffnen und auf den zweiten Arduino laden.
2. Alle weiteren Erläuterungen sind dem Programm-Code beigelegt.

**Hinweis:** Falls mehr Funktionen gewünscht sind (z.B. Schieberegler), kann dies in beiden Programmen einfach hinzugefügt werden. Die Berechnung ist bereits auf Sende -bzw. Empfangsseite implementiert.

Im Programm Sender ist dies an richtiger Stelle ausreichend kommentiert und kann einfach hinzugefügt werden.

Im Programm Empfänger sind Ergänzungen vorzunehmen, welche der PDF „Joystick\_Befehle“ zu entnehmen sind.

- <http://www.instructables.com/Arduino-LeonardoMicro-as-Game-ControllerJoystick/>

## **4. Gesamter Aufbau**

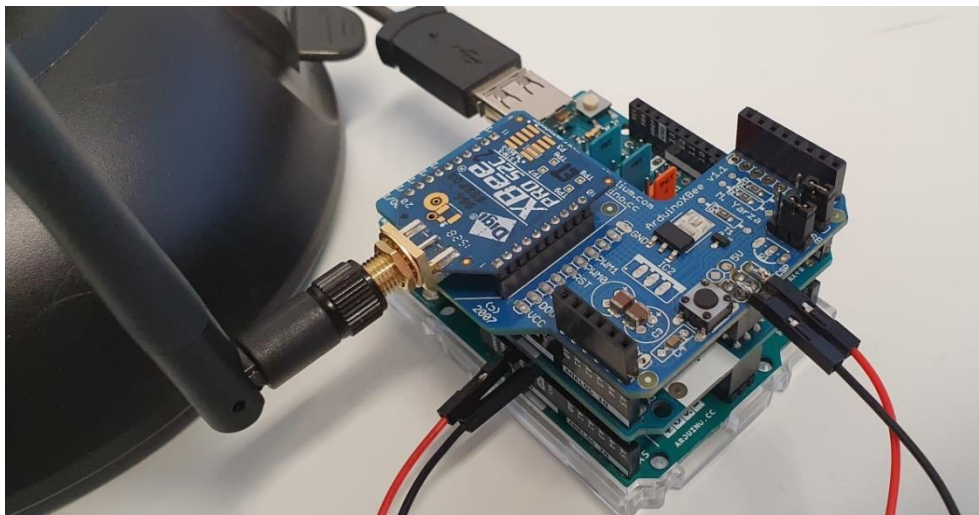
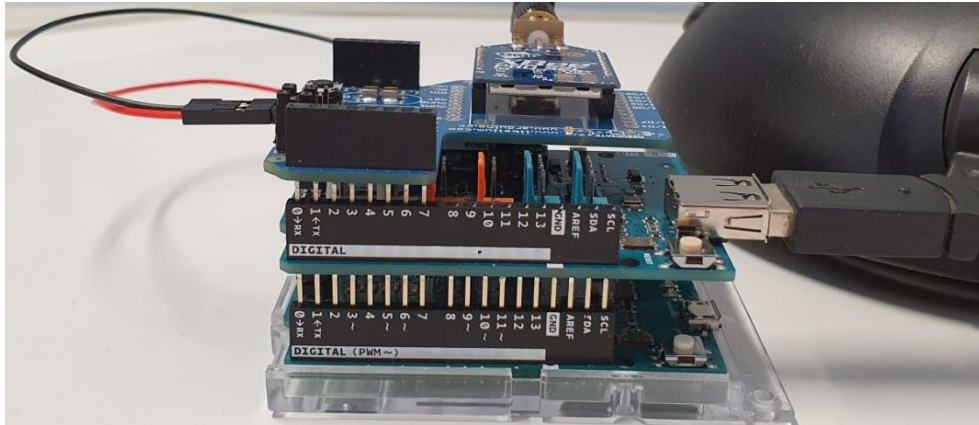
### **4.1 Jumperposition anpassen**

**Achtung:** Bevor endgültig alle Teile zusammengeführt werden, muss die Jumperposition auf den XBee-Shield verändert werden, es gilt: **USB → XBee**



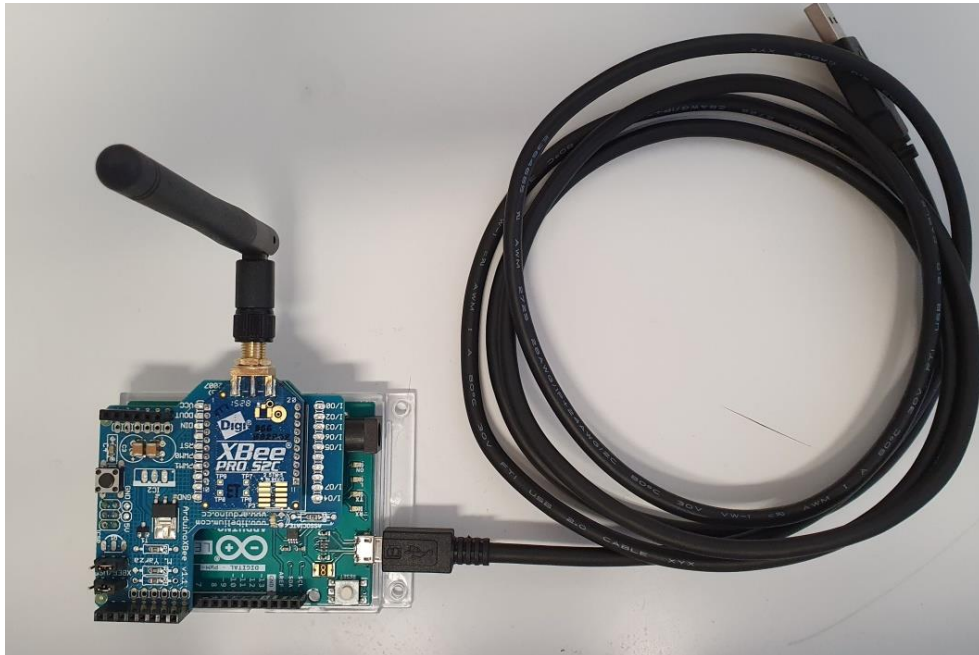
## 4.2 Sender

Senderseitig wird ein Arduino Leonardo, das USB Host Shield, ein XBee und Joystick benötigt. Der Zusammenbau ist den folgenden Bildern zu entnehmen.



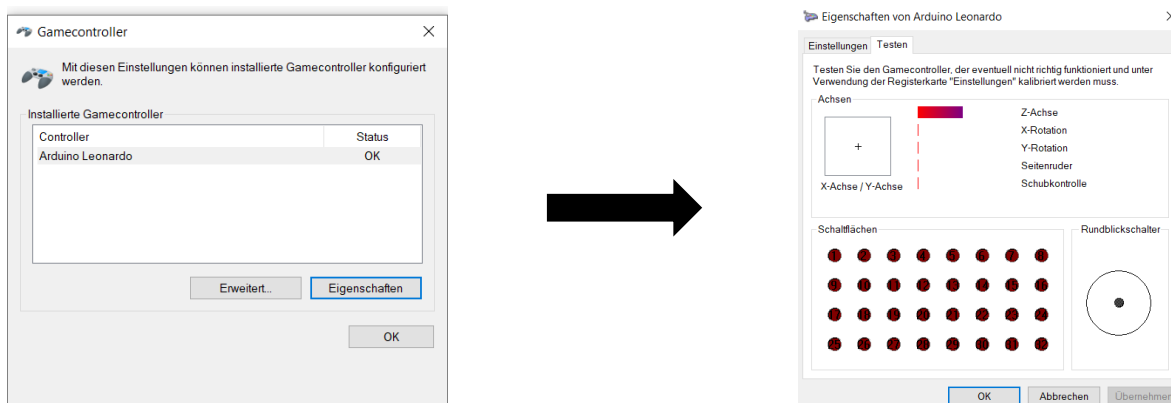
### 4.3 Empfänger

Empfangsseitig wird ein Arduino Leonardo und XBee benötigt. Der Zusammenbau ist den folgenden Bildern zu entnehmen.

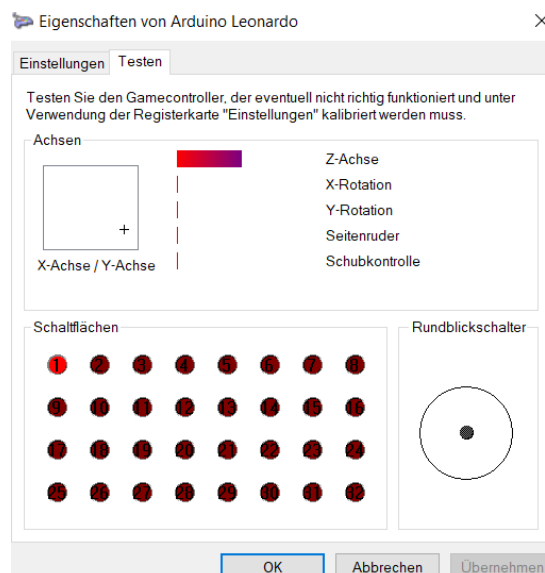


## 5. Funktionsprüfung

Nachdem sowohl Sender als auch Empfänger vollständig aufgebaut sind, sollte eine auf dem XBee-Shield integrierte LED im Sekundentakt blinken (Sender und Empfänger). Anschließend unter Windows „USB-Gamecontroller einrichten“ auswählen und in dem sich öffnenden Feld auf „Arduino Leonardo“ klicken. Anschließend wird mit einem Klick auf „Eigenschaften“ ein Joystick- und Tasterfeld sichtbar.



Jetzt kann senderseitig der Joystick bewegt und die Taster gedrückt werden. Auftretende Positions- bzw. Tasteränderungen werden nun angezeigt.

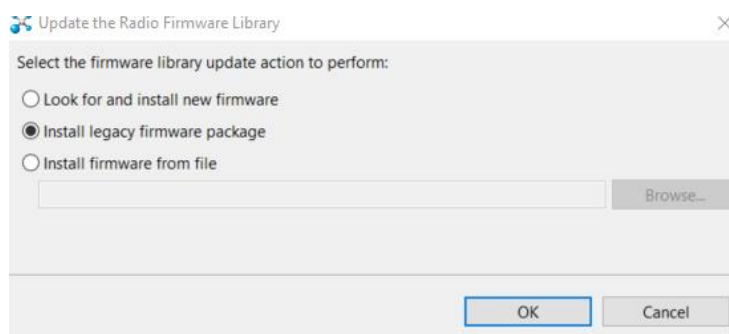


## 6. Fehlerbehebung

### 6.1 XBee-Einrichtung

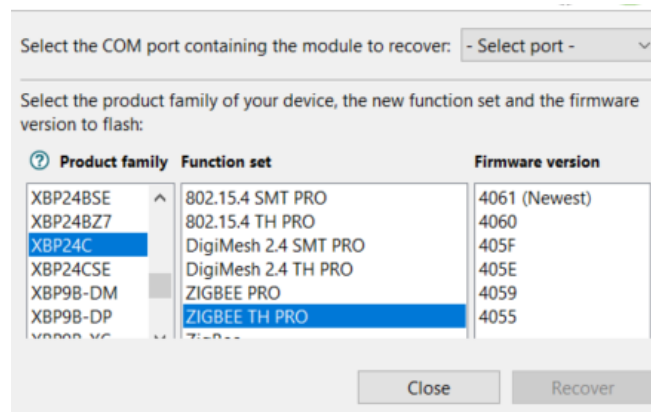
Es kann vorkommen, dass das XBee Modul zwar in XCTU erkannt wird, sich allerdings nicht öffnen lässt um Parameter wie die PAN-ID einzustellen. Zu erkennen ist dieser Fehler, wenn bei dem jeweiligen Modul unter Function „4055-Was not found“ angezeigt wird. Wird dieser Fehler erkannt, liegt ein Fehler mit der Firmware vor. Bei der Installation von XCTU wird ältere Firmware standardmäßig nicht automatisch mit eingebunden. Um auch ältere Firmware zu importieren:

„Help“ → „Install Legacy Radio Firmware“ → „Install legacy firmware package“



Nun muss geprüft werden, ob die ältere Firmware für die Installation zur Verfügung steht.

„Tools“ → „XBee Recovery“ → „XBP24C“ → „ZIGBEE TH PRO“ → „4061“ → „Recover“



Durch „Recover“ wird auf das XBee-Modul die neuste Firmware geladen und anschließend den XBee nach Abschnitt „XBee-Einrichtung Parameter“ erneut einlesen.

Folgende Firmware steht zur Verfügung:

- ➔ 401x, 402x, 403x, 404x, 405x, 406x
- ➔ Für weitere Informationen: „Anleitung XBee“ ( im Ordner)



## 6.2 Bootloader Arduino

Wird ein Arduino nicht mehr als COM-Port in der Software angezeigt (nicht mehr per USB zu erreichen), liegt die Vermutung nahe, dass etwas mit dem Bootloader nicht stimmt. Abhilfe kann es schaffen diesen neu auf den Arduino zu brennen. Hierfür wird allerdings ein Programmiergerät oder ein Arduino UNO benötigt.

### Acht Schritte

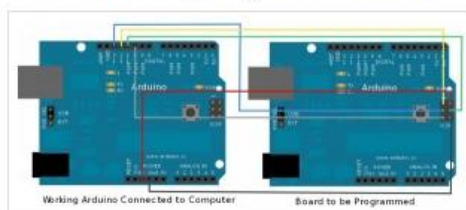
1. Wir verbinden den PC über ein USB-Kabel mit unserem Arduino Uno, den wir gleich als sogenannten ISP (In-System-Programmer) verwenden werden. In der Arduino IDE (bei mir Release 1.0.5) finden wir unter **Datei** und **Beispiele** den passenden Sketch **ArduinoISP**.



2. Als nächstes wählen wir unter **Tools** und **Board** unseren **Arduino Uno** aus, der nun in Kürze den **Serial Peripheral Interface (SPI)** Programmer spielen darf. Weiterhin muss im Menü unter **Tools** und **Serieller Port** natürlich der richtige COM-Port ausgewählt werden.

3. Als nächstes den Programmer-Sketch über **Datei** und **Upload** oder das entsprechende Upload-Symbol auf das Uno-Board hochladen.

4. Nachdem wir die USB-Verbindung und ggf. auch die Stromversorgung des Arduino UNO wieder getrennt haben, erfolgt die Verkabelung mit dem zu programmierenden LEONARDO. Dazu habe ich in den [Sparkfun Tutorials](#) nebenstehendes anschauliches Bild gefunden, das die Positionen der ggf. vorhandenen ICSP-Header eines UNOs beschreibt. Links oben ist der ICSP-Header (In-Circuit Serial Programming) des USB-Controllers zu finden, rechts außen der des embedded Controllers Atmega328 oder wie beim Leonardo der Atmega32U4. Der 6-Pin ICSP-Header des LEONARDO muss nun mit

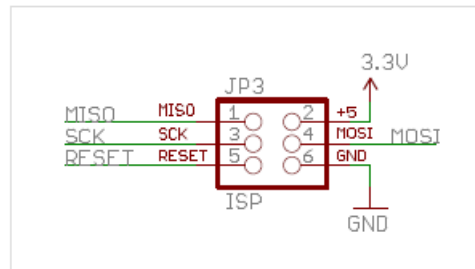


dem UNO SPI-Programmer verbunden werden. Links in nachfolgendem Bild ist der UNO als Programmer zu sehen, rechts davon wird unser LEONARDO werden, dessen Bootloader neu geflasht werden soll.

In meinem Testaufbau sieht das wie folgt aus:



Auch hier links auf dem Foto ist der UNO zu sehen, der nun als SPI-Programmer fungiert. Rechts davon der LEONARDO, der über den 6-Pin ICSP-Header mit dem Bootloader programmiert werden soll. Beim LEONARDO ist unbedingt zu beachten, dass das SPI-Interface nicht über die üblichen Digital Pins D11, D12 und D13 herausgeführt ist, sondern nur über den 6-poligen ICSP-Header!



Der 6-Pin ICSP-Header hat generell die folgende Belegung.

#### Arduino as ISP AVR Programmer ISP Header ATmega328 ATmega32U4 ATmega2560

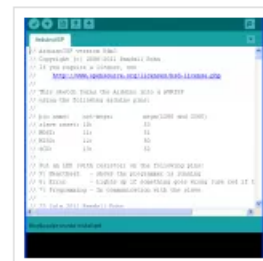
Vcc/5V	5V	Pin 2	Vcc	Vcc	Vcc
GND	GND	Pin 6	GND	GND	GND
MOSI/D11	MOSI	Pin 4	D11	D16	Pin 51
MISO/D12	MISO	Pin 1	D12	D14	Pin 50
SCK/D13	SCK/CLK	Pin 3	D13	D15	Pin 52
D10	Reset	Pin 5	Reset	Reset	Pin 53

5. Nach der SPI-Verkabelung des LEONARDO mit dem UNO stellen wir den USB-Anschluss zum PC wieder her.

6. Als nächstes müssen wir entscheiden, welcher Bootloader auf unser LEONARDO Board geladen wird. Dazu wählen wir einfach unter **Tools** und **Board** den LEONARDO aus. Zur Sicherheit überprüfen wir auch gleich noch den COM-Port. Dieser muss dem UNO ISP-Programmer entsprechen.

6. Unter **Tools** und **Programmer** wählen wir als nächstes den Menüpunkt **Arduino as ISP** aus. Damit wird der UNO nun als Programmer ausgewählt.

7. Nun müssen wir nur noch unter **Tools** den Punkt **Bootloader installieren** auswählen. Die LED an Pin 13 wie auch die beiden RX/TX LEDs auf dem UNO blinken nun eine ganze Weile. Dann erfolgt die Meldung in der Arduino IDE: „Bootloader wurde installiert“. Das war es bereits. Die LED auf dem LEONARDO sollte nun im Sekundentakt blinken.



8. Optional können an die Arduino Uno Digital Pins D7, D8 und D9 noch LEDs über entsprechende Vorwiderstände (330 Ohm) angeschlossen.

➔ <https://arduino-hannover.de/2014/05/03/kochbuch-leonardo-bootloader-flaschen/>