

Installation der Entwicklungsumgebung

bitte zunächst das Online-Dokument *Primer* durcharbeiten

- <http://www.tinkerforge.com/en/doc/Primer.html#primer>

download der TinkerForge-Software

- <http://www.tinkerforge.com/en/doc/Downloads.html>

- Sie brauchen:

- den **Brick Daemon**

sorgt dafür, dass Sie TinkerForge-Bricks über eine virtuelle Netzwerkschnittstelle mit IP-Adresse und Portnummer ansprechen können

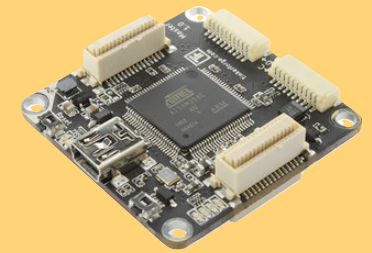
- den **Brick Viewer**

ist ein Werkzeug für Firmware Update, Konfiguration und Test von Bricks und Bricklets

- die **Bindings and Examples** für die Programmiersprache Java

- mit dem Brick Viewer können Sie feststellen, ob die Firmware auf den Bricks und Bricklets up-to-date ist

- ev. entsprechende Firmware-Dateien herunterladen und mit dem Brick Viewer installieren (siehe Flashing Tutorials)



Prüfen der Firmware-Versionen

Brick Viewer 2.0.7

Setup Master Brick 2.0 Ambient Light Bricklet Barometer Bricklet Humidity Bricklet LCD 20x4 Bricklet 1.2

Disconnect

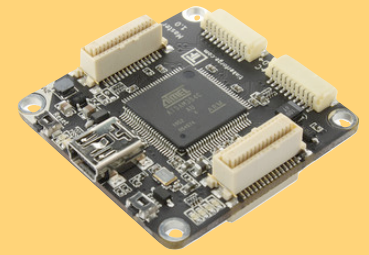
Host: localhost Port: 4223

Name	UID	FW Version
▼Master Brick 2.0	6xCYzF	2.0.6
A: Ambient Light Bricklet	eqy	2.0.1
B: Barometer Bricklet	euN	2.0.1
C: Humidity Bricklet	dej	2.0.1
D: LCD 20x4 Bricklet 1.2	eyu	2.0.3

Mit der Web-Seite vergleichen, da das automatische Aktualisieren wegen unseres Web-Proxies nicht funktioniert!

Updates / Flashing

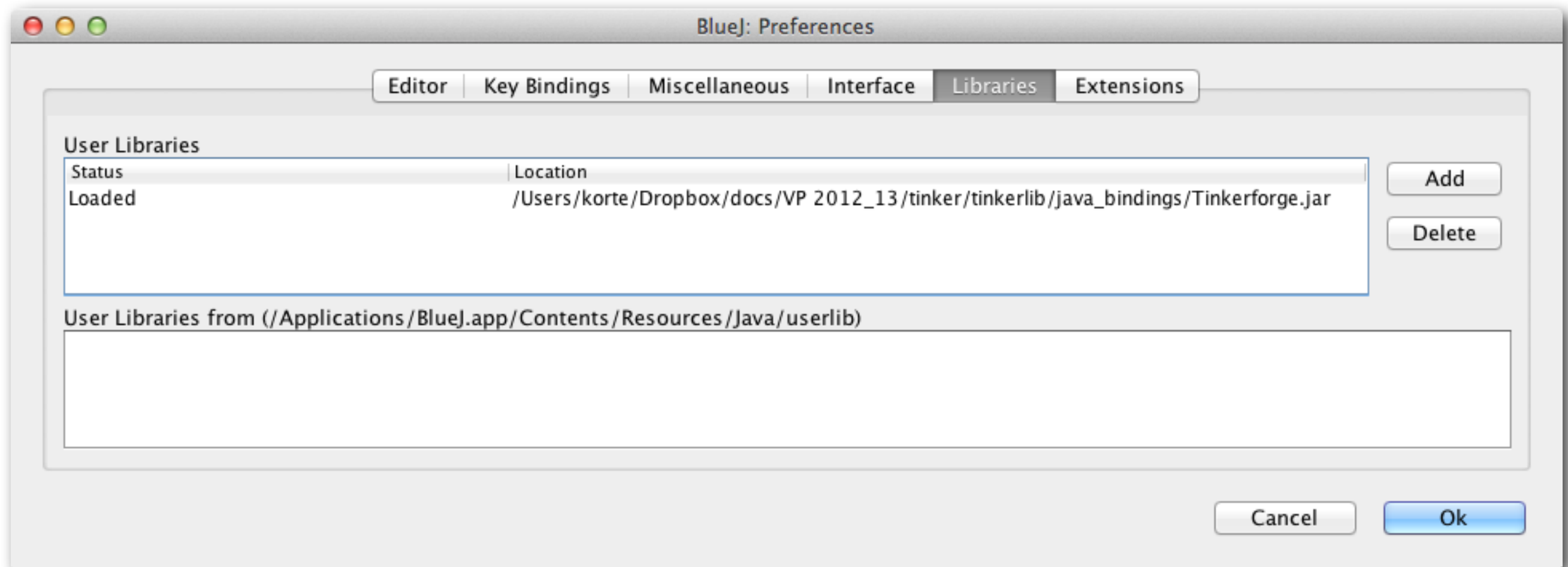
Advanced Functions



Programmierung der Tinkerforge-Module mit BlueJ

Zur Anwendungsentwicklung in Java kann **BlueJ** eingesetzt werden.

- Es muss lediglich die TinkerForge **Java**-Schnittstellenbibliothek in die BlueJ-Voreinstellungen eingetragen werden.





Programmierbeispiel: Auslesen eines Temperatur-Bricklets

1. Möglichkeit: Temperatursensor wird abgefragt (polling)

```
import com.tinkerforge.BrickletTemperature;
import com.tinkerforge.IPConnection;

public class ExampleSimple {
    private static final String HOST = "localhost";
    private static final int PORT = 4223;
    private static final String UID = "ABC"; // Change to your UID

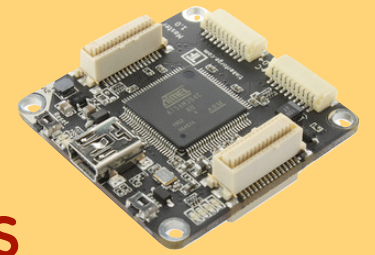
    // Note: To make the example code cleaner we do not handle exceptions. Exceptions you
    // might normally want to catch are described in the documentation
    public static void main(String args[]) throws Exception {
        IPConnection ipcon = new IPConnection(); // Create IP connection
        BrickletTemperature temp = new BrickletTemperature(UID, ipcon); // Create device object

        ipcon.connect(HOST, PORT); // Connect to brickd
        // Don't use device before ipcon is connected

        // Get current temperature (unit is °C/100)
        short temperature = temp.getTemperature(); // Can throw com.tinkerforge.TimeoutException

        System.out.println("Temperature: " + temperature/100.0 + " °C");

        System.out.println("Press key to exit"); System.in.read();
        ipcon.disconnect();
    }
}
```



Programmierbeispiel: Auslesen eines Temperatur-Bricklets

2. Möglichkeit: Temperatursensor meldet sich, wenn ein neuer Wert zur Verfügung steht (callback, event driven)

```
public class ExampleCallback {
    private static final String HOST = "localhost";
    private static final int PORT = 4223;
    private static final String UID = "ABC"; // Change to your UID

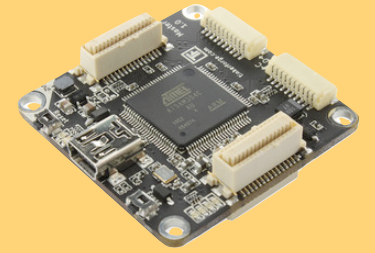
    // Note: To make the example code cleaner we do not handle exceptions. Exceptions you
    // might normally want to catch are described in the documentation
    public static void main(String args[]) throws Exception {
        IPConnection ipcon = new IPConnection(); // Create IP connection
        BrickletTemperature temp = new BrickletTemperature(UID, ipcon); // Create device object

        ipcon.connect(HOST, PORT); // Connect to brickd
        // Don't use device before ipcon is connected

        // Set Period for temperature callback to 1s (1000ms)
        // Note: The temperature callback is only called every second if the
        // temperature has changed since the last call!
        temp.setTemperatureCallbackPeriod(1000);

        // Add and implement temperature listener (called if temperature changes)
        temp.addTemperatureListener(new BrickletTemperature.TemperatureListener() {
            public void temperature(short temperature) {
                System.out.println("Temperature: " + temperature/100.0 + " °C");
            }
        });

        System.out.println("Press key to exit"); System.in.read();
        ipcon.disconnect();
    }
}
```



... nun kann ihr Praktikum beginnen

Jeder bekommt ein TinkerForge-System mit verschiedenen Sensoren und einer LCD- oder LED-Anzeigeeinheit

- Ihre Praktikumsaufgabe:
 - Programmieren Sie die TinkerForge-Module so, dass ein Sensorwert mit einer Update-Rate von 1 Sekunde auf der Anzeige dargestellt wird.
Benutzen Sie dazu die zu jedem Modul vorhandenen Programmierbeispiele!