



BiDiB-Broker Embedded Systems

Ergänzung zum BiDiB-Broker Anwenderhandbuch



Inhaltsverzeichnis

1	Einführung.....	3
2	Voraussetzung.....	3
3	MicroSD-Karte beschreiben.....	3
4	Raspi weiter einrichten.....	4
5	Java installieren.....	5
6	Pairing-LED und -Taster.....	5
6.1	Wiring Pi installieren.....	5
6.2	Hardware.....	6
7	BiDiB-Broker.....	7
7.1	BiDiB-Broker einrichten.....	7
7.2	BiDiB-Broker starten (Beispiele).....	7
7.3	BiDiB-Broker verwalten.....	7
8	Anhang: BiDiB-Pi - Test.....	8
8.1	Zusammenfassung.....	8
8.2	Testumgebung.....	8
8.3	Testergebnis.....	8

Historie

Datum	Name	Version	Bemerkung
2023-01-29	Michael Schäfer	0.1	Initiale Version aus Notizen
2023-01-30	Michael Schäfer	0.2	Referenz auf Handbücher, Anhang eingefügt
2023-03-06	Michael Schäfer	0.3	Link zum Broker korrigiert, Kap 5.+7. verbessert

1 Einführung

Das Handbuch ist eine Erweiterung der Handbücher [Anwenderhandbuch](#) und [Entwicklerhandbuch](#).

In diesem Handbuch [Embedded Systems](#) werden Einrichtung und Inbetriebnahme in so genannte Embedded Systems am Beispiel des Raspberry PI 3/4 Model B+ beschrieben.

2 Voraussetzung

Es wurden die folgenden Komponenten verwendet:

- Raspberry Pi 3 Model B+ und Raspberry Pi 4 Model B+
(siehe auch: <https://projects.raspberrypi.org/en/projects/raspberry-pi-setting-up>)
- microSD-Karte (16 ... 32, 64 GB, Class 10)
(geeignete Karten siehe auch https://elinux.org/RPi_SD_cards)
- BiDiB-Pi (ersatzweise Standard Raspi mit IF2 oder GBMBoost)
Netzteil mit Hohlstecker ?mm, z.B. 15V/3,5A für BiDiB-Pi
- Gehäuse
- div. Befestigungsteile (Abstandshalter, Schrauben, Muttern)
- (Optional: Maus, Tastatur, Bildschirm, -kabel, Audioleitung)
- (Windows-)Rechner
(oder andere andere Systeme mit entsprechenden Software-Werkzeugen)
 - mit Internetverbindung
 - mit Kartenleser/Schreiber für microSD-Karte
 - Initiales Image für Raspi, "Paspberry Pi Imager"
(siehe auch <https://www.raspberrypi.com/software/>)
 - Fernwartung, z.B. VNC-Client (=> VNC-Server ist schon konfiguriert)
 - Datei-Manager, z.B. "WinSCP" zur grafischen Kontrolle
(siehe auch <https://winscp.net/eng/download.php>)
 - weitere Netz-Werkzeuge ... (PuTTY) ...

3 microSD-Karte beschreiben

Bei Bedarf vorher SD-Karte formatieren.

- "Paspberry Pi Imager" installieren und starten
 - Betriebssystem:
Raspberry Pi OS FULL (32bit)
A port of Debian Bullseye with Raspberry Pi Desktop (Recommended)
 - SD-Karte anschließen und auswählen
 - Einstellungen
(Zahnrad rechts unten - siehe auch Raspberry Pi Imager v.1.7.3)
 - Hostname: Pi3B-BiDiB
 - SSH aktivieren
 - Password zur Authentifizierung verwenden
 - Benutzername: pi, Passwort: bidibPi3
 - Wifi einrichten, z.B.: SSID: Xenia, Passwort: Pi3bUndGast (wird verschlüsselt)
 - Wifi-Land: DE
 - Spracheinstellungen: Europe/Berlin
 - Tastaturlayout: de
 - Tonsignal, **nicht** auswerfen (s.u. Manuelle Anpassungen ...), [Telemetry]

- "Schreiben" (c.a. 4 – 10 Min Schreiben + 4 – 10 Min Verifizieren = 8 – 20 Min)
 - Mit dem Imager wird das neueste OS geladen!
 - Eventuelle Wächter umkonfigurieren, da der MBR von der SD-Karte beschrieben wird!
 - Windows-Meldungen mit Bezug auf das/die Laufwerk(e) mit der SD-Karte ignorieren!
- Manuelle Anpassungen auf microSD-Karte
 - Unbedingt mit Unix/OSX-Format speichern (LF am Ende)! (siehe auch <https://www.raspberrypi.com/documentation/computers/configuration.html#the-kernel-command-line>)
 - cmdline.txt
 - Eintrag "console=serial0,115200" entfernen
 - config.txt
 - freigeben/einkommentieren:
dtparam=i2c_arm=on
dtparam=i2s=on
dtparam=spi=on
 - vor [cm4] anfügen:
dtoverlay=disable-bt
enable_uart=1
- SD-Karte „auswerfen“ und aus Kartenleser entnehmen

4 Raspi weiter einrichten

Im Beispiel mit PuTTY.

- Raspi zum Einrichten nur mit internetfähigem Netzwerk verbinden, weil darüber nachgeladen wird
 - Im Router neues Gerät für WLAN zulassen!
- microSD-Karte in Raspi einstecken, Spannung einschalten (ohne GUI: bis zur Anmeldung im WLAN bzw. warten)
 - jetzt wird das OS automatisch eingerichtet
 - beim "Raspberry OS FULL (32bit)" dauert der Update-Vorgang länger (etwa 2 Min)!
- Verbinden mit PuTTY (muss sich im gleichen (WLAN-)Netz wie der Raspi befinden)
 - Host Name=Pi3B-BiDiB (siehe oben), (Port=22, SSH, Telnet)
 - PuTTY Security Alert: `Accept`
 - login as: `pi`, `bidibPi3` (siehe Eingaben oben), `ENTER`
- wenn microSD-Karte über Imager beschrieben => weiter bei "!!! Software auffrischen!!!"
- Initiale Eingaben per GUI
 - Location: DE
 - Neues Kennwort: `bidibPi3`
 - WiFi mit Internet-Anschluss konfigurieren
 - System
 - Hostname=Pi3B-BiDiB
 - Schnittstellen
 - SSH, VNC, SPI, I2C, SerialPort aktiviert
 - Serielle Konsole deaktiviert
- (Alternativ zu GUI und PuTTY: `ssh pi@raspberrypi` - Hilfreich: `sudo raspi-config`)

- !!! Software auffrischen!!!

→ Funktioniert nicht mit Gastzugang und Sicherheitsabfrage!

- (GUI: Automatisch nach Anfrage)
- PuTTY
 - OS Update laden und installieren (c.a. 10 – 12 Minuten):
`sudo apt-get update && sudo apt-get dist-upgrade`
Beachte: Bei großem Download Sicherheitsabfrage mit 'j' bzw. 'y' beantworten.
 - Ungenutzte Software entfernen:
`sudo apt-get autoremove`
Beachte: Evtl. manuelle Bestätigung für Remove notwendig!
- Reboot:
`sudo reboot`
- (ODER Abschalten: `sudo shutdown -r now`)

5 Java installieren

Bei Bedarf wieder über PuTTY verbinden.

- Java Version kontrollieren (am Beispiel)
 - `java -version`
openjdk version "17.0.4" 2022-07-19
OpenJDK Runtime Environment (build 17.0.4+8-Raspbian-1deb11u1rpt1)
OpenJDK Client VM (build 17.0.4+8-Raspbian-1deb11u1rpt1, mixed mode, emulated-client)
- wenn Java 17 oder höher vorhanden => weiter bei "Wiring Pi installieren"
- wenn vorhanden, alte Version entfernen, z.B.:
`sudo apt-get purge bellsoft-jdk*`
- Lade neue Version (Stand 30.12.2022)
(siehe auch : <https://github.com/bell-sw/Liberica/releases/>)
 - Holen, z.B. (bitte Pfad an neueste Version lt. Link anpassen!):
`wget https://download.bell-sw.com/java/17.0.5+8/bellsoft-jdk17.0.5+8-linux-arm32-vfp-hflt.deb`
Beachte: Bei großem Download Sicherheitsabfrage mit 'j' bzw. 'y' beantworten.
 - Installieren, z.B. (bitte Pfad an neueste Version lt. Link anpassen!):
`sudo apt-get install ./bellsoft-jdk17.0.5+8-linux-arm32-vfp-hflt.deb`
 - Kontrolle (siehe auch oben)
 - `java -version`
openjdk version "17.0.5" 2022-10-18 LTS
OpenJDK Runtime Environment (build 17.0.5+8-LTS)
OpenJDK 32-Bit Server VM (build 17.0.5+8-LTS, mixed mode)
- bei Bedarf - set the defaults
`sudo update-alternatives --config javac`
`sudo update-alternatives --config java`

6 Pairing-LED und -Taster

Broker unterstützt das Pairing nicht nur via Pairing-GUI sondern auch über zwei GPIOs.

Soll das Pairing nur mit der GUI erfolgen, geht es weiter mit Kapitel 7.1 BiDiB-Broker einrichten.

6.1 Wiring Pi installieren

Für das Pairing mit LED und Taster wird die Bibliothek „Wiring Pi“ benötigt.

→ Wiring Pi funktioniert Stand heute (26.01.2023) nur mit Raspberry Pi OS 32bit!

Das neueste Paspberry Pi OS enthält kein wiringPi mehr! "sudo apt-get install wiringpi" führt nicht mehr zur Installation! Deshalb muss das WiringPi manuell installiert werden (siehe auch <http://wiringpi.com/wiringpi-updated-to-2-52-for-the-raspberry-pi-4b/>):

Update wiring Pi (2019-06-24) (Gordon Henderson)

```
Upgrade:  cd /tmp
          wget https://project-downloads.drogon.net/wiringpi-latest.deb
          sudo dpkg -i wiringpi-latest.deb

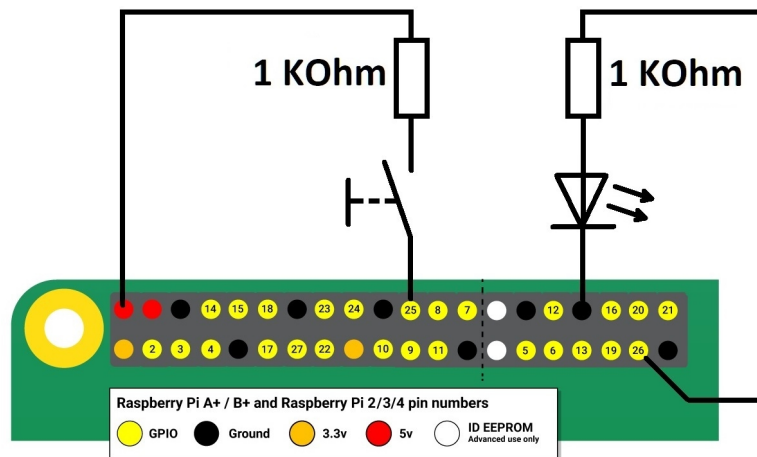
Check:   gpio -v
          gpio version: 2.52
          Copyright (c) 2012-2018 Gordon Henderson
          This is free software with ABSOLUTELY NO WARRANTY.
          For details type: gpio -warranty

Raspberry Pi Details:
          Type: Pi 3B+, Revision: 03, Memory: 1024MB, Maker: Sony
          * Device tree is enabled.
          *--> Raspberry Pi 3 Model B Plus Rev 1.3
          * This Raspberry Pi supports user-level GPIO access.
```

6.2 Hardware

Der so genannte BiDiB-Pi hat diese beiden Hardware-Elemente werkseitig eingebaut. Der Taster wird an Port BCM-25 bzw. WPI-6, die LED wird an Port BCM-26 bzw. WPI-25 erwartet. Bei einem „Raspberry Pi von der Stange“ kann man diese beiden Elemente leicht nachrüsten.

Pairing -Taster -LED



Quelle: [Raspberry Pi Documentation](#)

Broker unterstützt die zusätzliche Hardware durch den Startparameter:

```
--base-data.pairing-receipt=Button
```



7 BiDiB-Broker

Broker muss zur Zeit noch über Terminalprogramme wie z.B. PuTTY verwaltet werden.

7.1 BiDiB-Broker einrichten

- Verzeichnis BiDiB erstellen: `mkdir BiDiB`
- aktuelle BiDiB-Broker-Jar-Datei als "bidib-broker-latest.jar" nach BiDiB kopieren, z.B.:
`cd BiDiB`
`wget https://www.fichtelbahn.de/files//bidib-broker-family/broker/bidib-broker-latest.jar`
- Bei Bedarf automatischen Start Brokers beim Einschalten einrichten (siehe auch <https://forums.raspberrypi.com/viewtopic.php?t=294014>)
Edit File als „sudo“: /etc/rc.local (vor dem Ende (exit 0) einfügen:
 - `cd /home/pi/BiDiB/`
 - `sudo -u pi java -jar bidib-broker-latest.jar`Anmerkung: Zusätzlich gewünschte Startparameter anhängen

7.2 BiDiB-Broker starten (Beispiele)

Anbei Beispiele für den Start Brokers. Eine ausführliche Beschreibung der Profile und Startparameter findet sich im BiDiB-Broker_EntwicklerHandbuch.

- Broker starten standardmäßig als Hub:
`java -jar bidib-broker-latest.jar`
Bei Bedarf Skript, z.B. `run.broker` erstellen oder vom Server kopieren:
(`run.hub.broker`, `run.avatar.broker`, `run.pi.hub.broker`, `run.pi.avatar.broker`)
- Broker starten z.B. als Avatar und mit BiDiBus an der seriellen Schnittstelle:
`java -jar bidib-broker-latest.jar --bidib-system.default-role=Avatar --serial-device.preferred-ports=/dev/serial0`
- Broker starten als Hub mit überschreibbarer (Test-)Protokolldatei:
`java -jar bidib-broker-latest.jar --spring.profiles.active=test`

7.3 BiDiB-Broker verwalten

- Ob Broker läuft lässt sich feststellen. Zuerst, ob Java läuft:
`pidof java`
Überprüfung, ob Broker dabei ist:
`ps aux | grep bidib-broker-latest`
- Das Protokoll ist einsehbar entweder über das Terminal-Fenster in der GUI oder per:
`tail -f /tmp/.BiDiBBrokerFamily/broker.log`
bzw. im Testmodus:
`tail -f /tmp/.BiDiBBrokerFamily/brokerTestProfile.log`
- Broker und die verwendeten Dienste können gestoppt werden durch CTRL+C bzw. mit:
`sudo pkill -2 -f bidib-broker-latest.jar`

8 Anhang: BiDiB-Pi - Test

Komponenten: BiDiBus, Discovery, HotPlug und Pairing-Hardware (LED und Taste)

8.1 Zusammenfassung

Mit 64-Bit OS funktioniert kein HotPlug der seriellen Schnittstellen (seriell und USB). Wiring Pi ist nicht installierbar, alternativ Pi4J V2.21 nur als sudo lauffähig. Mit 32-Bit OS und 32-Bit Java sind alle Funktionen lauffähig.

Fazit: HotPlug sowie Pairing-LED und -Taster funktionieren nur mit 32-bit OS zufriedenstellend. Mit 64-Bit OS ist die serielle Schnittstelle nur per „Preferred List“ nutzbar, Pairing-LED und -Taster sind nur mit Sondersoftware verwendbar.

8.2 Testumgebung

- Raspberry Pi 4B
- Raspberry Pi OS; 64-bit / 32-bit / 32-bit full (inkl. Java 17.0.4 32-bit)
- BiDiB-Broker
 - V2: bidib-broker-0.9.4-0-LIBPI-V2.jar // mit Pi4J Release V2.21
 - V1: bidib-broker-0.9.4.jar // mit Wiring Pi 2.5.2
- Java: 17.0.5: 64-bit / 32-bit

8.3 Testergebnis

OS	Java	Broker	Device	BiDiBus	Discovery	HotPlug	HW	Bemerkung
64	64	V2	PI	J	J	N	J	V2 nur mit sudo
			USB	J	J	N	-	V2 nur mit sudo
	V1	PI	J	J	N	N		
		USB	J	J	N	-		
64	32	V2	PI	J	J	N	N	V2 nur mit sudo
			USB	J	J	N	-	V2 nur mit sudo
	V1	PI	J	J	N	N		
		USB	J	J	N	-		
32 sta nd art	32	V2	PI	J	J	J	J	V2 nur mit sudo
			USB	J	J	J	-	V2 nur mit sudo
	V1	PI	J	J	J	J		
		USB	J	J	J	-		
32 full	32	V1	PI	J	J	J	J	
			USB	J	J	J	-	

Legende: Device: PI => seriell + HW
 Bus: BiDiBus, Discovery: des Busteilnehmers, HotPlug: des USB-Interfaces,
 HW: Pairing-LED, -Button
 J: funktioniert, N: keine Funktion, -: nicht relevant