

**HANDBUCH**

**für den**

**BSB-LPB-LAN-Adapter v2**  
**(Arduino-Version)**

**und die**

**BSB-LAN-Software**

*Stand: Dezember 2017*

## **ACHTUNG:**

***Es gibt KEINE GARANTIE oder Gewährleistung jeglicher Art, dass dieser Adapter dein Heizungssystem NICHT beschädigt!***

***Jegliche Umsetzung der hier beschriebenen Schritte, jeder Nachbau des Adapters sowie jede Verwendung der beschriebenen Hard- und Software erfolgt auf eigene Verantwortung und eigenes Risiko!***

***Keiner der Mitwirkenden oder Autoren kann für etwaige Schäden jeglicher Art haftbar gemacht werden!***

*Dieses Handbuch ist eine Zusammenstellung und Erweiterung der Dokumentationen README, FAQ und HOWTO.*

*Es wurde geschrieben, um den Einstieg in die Benutzung des BSB-LPB-LAN-Adapters und der dazugehörigen Software zu vereinfachen und um als Nachschlagewerk zu dienen.*

*Es wird empfohlen, dieses Handbuch vor einer initialen Verwendung des BSB-LPB-LAN-Adapters komplett zu lesen.*

### **Lizenz:**

Es steht dir frei, diese Software auf dein eigenes Risiko hin zu benutzen.  
Bitte beachte die Lizenzbedingungen der genutzten Bibliotheken und Software.

### **Autoren:**

- Software, Schaltplan v1, Dokumentationen EN, Ideenfindung, Support bis v0.16: *Gero Schumacher (gero.schumacher [ät] gmail.com)*
- Software, Platinenlayout v1 & v2, Dokumentationen EN, Ideenfindung, Support ab v0.17: *Frederik Holst (bsb [ät] code-it.de)*
- Debugging, Übersetzung EN-DE, Handbuch, Ideenfindung, Support ab v0.17: *Ulf Dieckmann (stille [ät] quantentunnel.de)*

*Basierend auf dem Code und der Mitarbeit von vielen anderen Entwicklern! Vielen Dank!*

# Inhaltsverzeichnis

<b>1. Der BSB-LPB-LAN-Adapter und die BSB-LAN-Software.....</b>	<b>5</b>
<b>2. Unterstützte Regler / Heizungssysteme.....</b>	<b>8</b>
<b>3. Grundsätzliches zum BSB und LPB.....</b>	<b>9</b>
<b>4. Installation der Arduino IDE und Konfiguration des Adapters.....</b>	<b>13</b>
<b>5. Einstellungsrelevante Parameter der BSB-LAN-Software.....</b>	<b>15</b>
<b>6. Funktionsüberprüfung und erste Nutzung des Adapters.....</b>	<b>19</b>
<b>7. BSB-LAN Web - das Webinterface des Adapters.....</b>	<b>20</b>
<b>8. Auflistung und Beschreibung der URL-Befehle.....</b>	<b>25</b>
<b>9. Etwaige Fehlermeldungen und deren mögliche Ursachen.....</b>	<b>30</b>
9.1 Fehlermeldung „unknown type <xxxxxxxx>“ .....	30
9.2 Fehlermeldung "error7 - parameter not supported" .....	30
9.3 Fehlermeldung "query failed" .....	30
<b>10. Etwaige Probleme und deren mögliche Ursachen.....</b>	<b>31</b>
10.1 Die rote LED des Adapters leuchtet nicht.....	31
10.2 Zugriff auf das Webinterface nicht möglich.....	31
10.3 Keine Parameterabfrage möglich.....	31
10.4 Regler wird nicht korrekt erkannt.....	31
10.5 HK1 kann nicht bedient werden.....	31
10.6 Es kann keine Raumtemperatur an einen HK1 gesendet werden.....	31
10.7 HK2 kann nicht bedient werden.....	31
10.8 Es kann keine Raumtemperatur an einen HK2 gesendet werden.....	32
10.9 Einstellungen des Reglers können nicht via Adapter verändert werden.....	32
10.10 Der Adapter reagiert manchmal nicht auf Abfragen.....	32
10.11 Bei der Abfrage der Logdatei passiert ‚nichts‘ .....	32
10.12 Es werden keine 24h-Durchschnittswerte angezeigt.....	32
10.13 Bei der Abfrage der Daten von DS18B20-/DHT22-Sensoren passiert ‚nichts‘ .....	32
10.14 Die DS18B20-Sensoren zeigen falsche Werte an.....	32
10.15 Der ‚Serielle Monitor‘ der Arduino IDE liefert keine Daten.....	32
<b>11. FAQ.....</b>	<b>33</b>
11.1 Kann ich Adapter & Software mit einem Raspberry Pi nutzen? .....	33
11.2 Kann ich einen Adapter an zwei Regler gleichzeitig anschließen? .....	33
11.3 Kann ich einen Adapter via LPB anschließen und mehrere Regler abfragen? .....	33
11.4 Gibt es einen einfachen Weg, um Parameter zu loggen? .....	33
11.4.1 Verwendung des Adapters als Standalone-Logger.....	33
11.4.2 Verwendung des Adapters als Remote-Logger.....	34
11.5 Wie kann ich die Daten bspw. in FHEM weiter verarbeiten? .....	34
11.6 Ist ein multifunktionaler Eingang am Regler direkt via Adapter schaltbar? .....	35
11.7 Relaisboard am Arduino Mega2560.....	36
11.8 Kann ich behilflich sein, um bisher nicht unterstützte Parameter hinzuzufügen? .....	36
11.9 Warum erscheinen bei einer Komplettabfrage einige Parameter doppelt? .....	39
11.10 Warum werden manchmal bestimmte Parameter nicht angezeigt? .....	39
11.11 Kann ich mehrere DHT22-Sensoren an einen DATA-Pin anschließen? .....	39
11.12 Warum ist kein Zugriff auf angeschlossene Sensoren möglich? .....	40
11.13 Ich nutze ein W5500-LAN-Shield, was muss ich tun? .....	40
11.14 Können Stati oder Werte als Push-Mitteilungen abgesetzt werden? .....	40
11.15 Kann bspw. FHEM auf bestimmte Broadcasts ‚lauschen‘? .....	40
11.16 Warum werden unter /B bei Stufe 2 keine Werte angezeigt? .....	40
11.17 Ich habe den Eindruck, die angezeigten Werte bei /B sind nicht korrekt.....	40

11.18 Kann ich die Funktion der Präsenztaste mit BSB-LAN simulieren?.....	41
11.19 Kann ich eine Raumtemperatur senden, um ‚Raumeinfluss‘ zu verwenden?.....	41
11.20 Was ist der genaue Unterschied zwischen /M1 und /V1?.....	41
11.21 Ich habe weitere Fragen, an wen kann ich mich wenden?.....	42
<b>12. Optionale Sensoren.....</b>	<b>43</b>
<b>13. Offene Punkte.....</b>	<b>44</b>
<b>14. Weiterführende Informationen und Quellen.....</b>	<b>45</b>
<b>Anhang.....</b>	<b>46</b>

## 1. Der BSB-LPB-LAN-Adapter und die BSB-LAN-Software

Der BSB-LPB-LAN-Adapter und die dazugehörige BSB-LAN-Software wurden entwickelt, um eine Anbindung von Heizungssystemen bzw. -reglern ans LAN und somit auch einen entsprechenden Fernzugriff zu ermöglichen.

Darüber hinaus ist es u.a. möglich, Parameter (bspw. Laufzeiten, Temperaturen) auf eine microSD-Karte zu loggen und sowohl DS18B20- als auch DHT22-Sensoren zusätzlich am Adapter anzuschließen.

Die hier vorgestellte Lösung stellt eine Alternative zu den bisherigen kommerziellen Lösungen dar, die nicht nur hinsichtlich des Kostenfaktors, sondern auch in Bezug auf die vielfältigen Einsatzmöglichkeiten mehr als nur ‚interessant‘ ist.

Für den Einsatz des Adapters müssen die Kessel-, Solar- oder auch Wärmepumpen-Regler<sup>1</sup> einen BSB<sup>2</sup> oder LPB<sup>3</sup> aufweisen. Dies ist i.d.R. bei allen (aktuelleren) Reglern der Fall, die von der Firma SIEMENS hergestellt wurden. Solche Regler werden bspw. von Heizungsherstellern wie Brötje oder Elco verbaut. Ob dein Heizungssystem über einen solchen Bus verfügt, ist den spezifischen technischen Unterlagen zu entnehmen.

Diesbezügliche Anfragen können von den Autoren nicht beantwortet werden.

Die hier vorgestellte und beschriebene Kombination aus Hard- und Software wurde u.a. an verschiedenen Heizungssystemen von Brötje und Elco getestet. Die Kommunikation sollte jedoch prinzipiell mit allen Systemen möglich sein, die einen der o.g. Bus-Typen aufweisen. Eine Auflistung der bisher erfolgreich getesteten Systeme ist im nachfolgenden Kapitel zu finden, dennoch kann niemals ein voller Funktionsumfang garantiert werden. Bei nicht gelisteten Systemen ist u.U. ein erhöhter eigener Einsatz nötig, um die Software in vollem Umfang nutzen zu können (siehe u.a. Kapitel 11.8).

Die Software ist auf einem Arduino Mega2560 samt Ethernet-Shield des Typs W5100<sup>4</sup> lauffähig, erfahrungsgemäß ‚out of the box‘. Aufgrund des geringeren Speichers ist die Verwendung von bspw. Arduino UNO, Arduino Nano o.ä. nicht möglich.

Da es teilweise unterschiedliche Pinbelegungen bei den verschiedenen LAN-Shields gibt, ist es u.U. nötig, den BSB-LPB-LAN-Adapter an andere Pins anzuschließen und die entsprechenden Änderungen hinsichtlich der Pinbelegung in der Software (Datei *BSB\_lan\_config.h*) anzupassen (siehe Kapitel 4 & 5).

Speziell in den Kapiteln 5, 7 und 8 findest du genauere Angaben zur BSB-LAN-Software.

In der Vergangenheit wurden immer mal wieder Sammelbestellungen für die Platinen durchgeführt, bei Interesse kann man sich im Forum<sup>5</sup> melden oder direkt Frederik Holst (bsb [ät] code-it.de) kontaktieren.

Der Schaltplan des Adapters ist im Anhang zu finden.

---

1 Im Folgenden nur als „Regler“ bezeichnet.

2 BSB = Boiler System Bus

3 LPB = Local Process Bus

4 Der Chip-Typ W5100 wird ohne Probleme unterstützt und wurde ausgiebig getestet, er ist einem W5500 daher immer vorzuziehen. Bzgl. Verwendung eines W5500-Boards siehe Kapitel 5 & 11.12.

5 <https://forum.fhem.de/index.php/topic,29762.0.html>

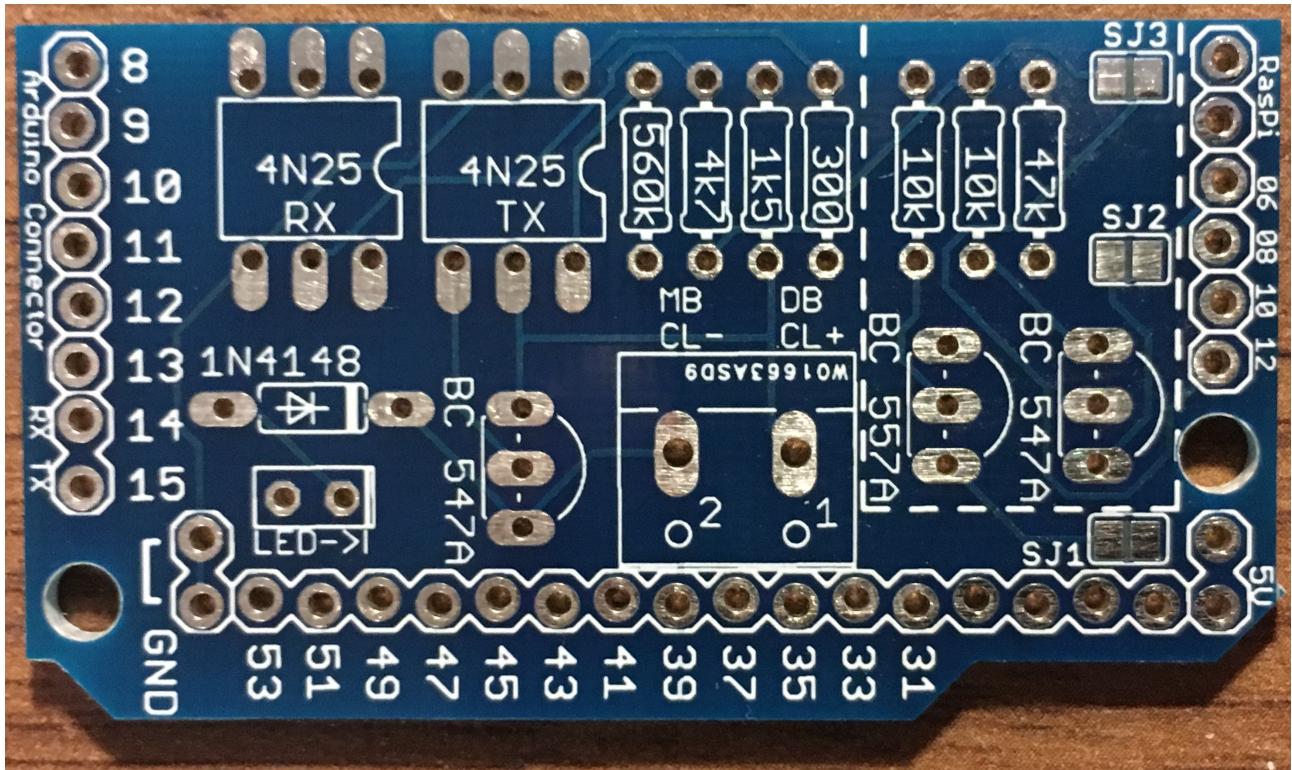


Abbildung 1: BSB-LPB-LAN-Adapter v2, unbestückt

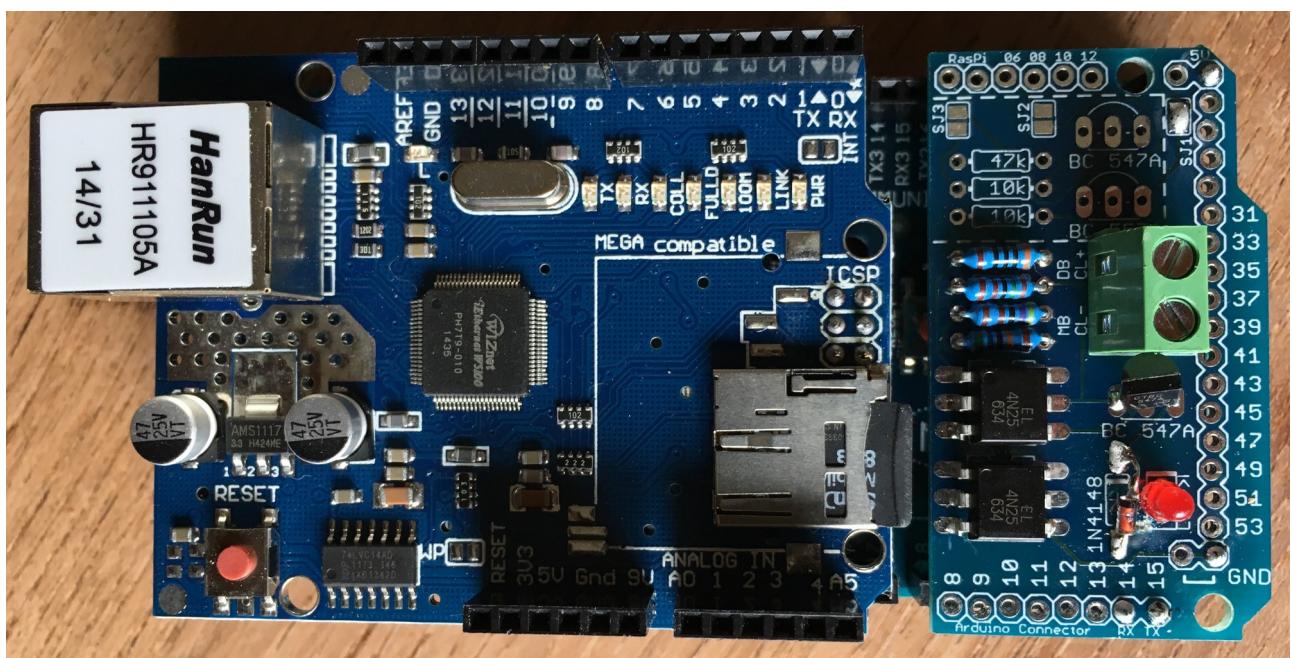


Abbildung 2: BSB-LPB-LAN-Adapter v2, bestückt, auf einem Arduino Mega2560 samt Ethernet-Shield

**Hinweis:**

Der aktuelle Adapter (v2) kann auch an einem Raspberry Pi 2 genutzt werden, jedoch nur unter Verwendung einer gänzlich anderen Software („bsb\_gateway“). Diese kann unter folgender URL gefunden werden:

<https://github.com/loehnertj/bsbgateway>

Für jeglichen Support in Zusammenhang mit der bsb\_gateway-Software kontaktiere bitte direkt den Autor von bsb\_gateway.

***Hier kann kein Support für bsb\_gateway gegeben werden!***

***Alle Informationen in diesem Handbuch beziehen sich nur auf die Arduino-Version!***

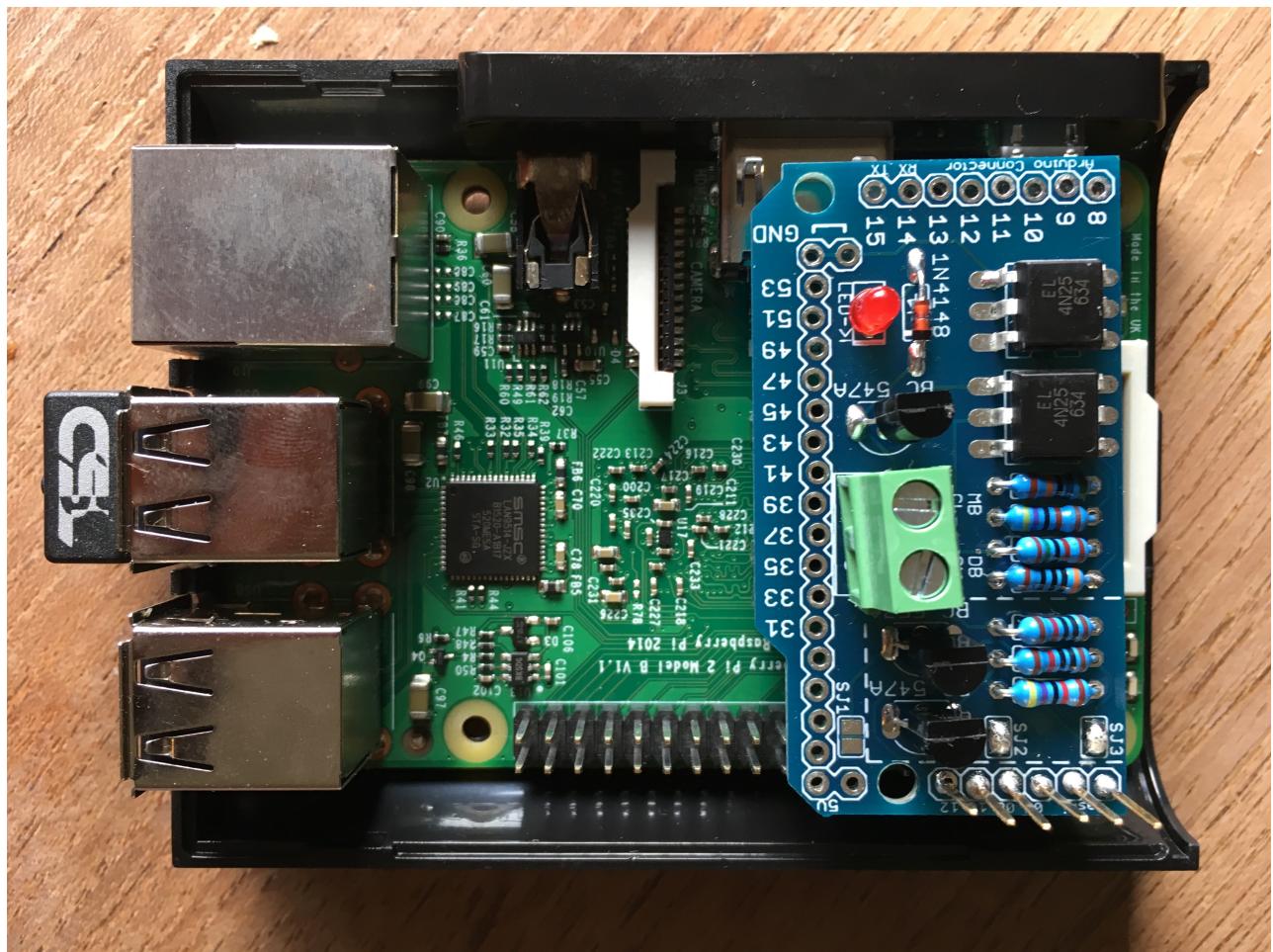


Abbildung 3: BSB-LPB-LAN-Adapter v2, bestückt, auf einem Raspberry Pi 2

## 2. Unterstützte Regler / Heizungssysteme

Die Verwendung des Adapters und der Software wurde bisher an folgenden Systemen erfolgreich getestet:

- Brötje ISR-SSR [RVS63.283/100] [RVS63.283/200] (Solar-System-Regler)
- Brötje ISR-ZR1 [RVS46.530/100] (Zonen-Regler)
- Brötje L-UB 25C [RVS43.122/100] (Ölbrenner)
- Brötje NovoCondens SOB 26 [Eurocontrol KM] {*LPB\_only*} (Ölbrenner)
- Brötje NovoCondens SOB 22C [RVS43.222/100] (Ölbrenner)
- Brötje NovoCondens SOB 26C [RVS43.222/100] + EWM [RVS75.390/109] (Ölbrenner)
- Brötje NovoCondens BOB 20 [RVS43.325/100] (Ölbrenner)
- Brötje SensoTherm BSW-K (Wärmepumpe)
- Brötje BBK 22E [LMS14.001A100] (Gasbrenner)
- Brötje WBS 22E (Gasbrenner)
- Brötje WGB-U 15H (Gasbrenner)
- Brötje WGB 15 E [LMS14] (Gasbrenner)
- Brötje WGB 20C [LMU74] (Gasbrenner)
- Brötje WGB S 17/20 E EcoTherm Plus [LMS14.001B100] (Gasbrenner)
- Brötje WGB EVO 20H [LMS15] (Gasbrenner)
- Brötje WGB Pro EVO 20C [LMU75] (Gasbrenner)
- Elco Stratton (Ölbrenner)
- Elco Stratton 21 [LOGON B G2Z2 + RVS63.283/160] (Ölbrenner)
- Elco Thision S Plus 13 [LMS14.002A167] (Gasbrenner)
- Elco Thision S 17.1 [LMU74.100A136 + AVS37.394/136] (Gasbrenner)
- Elco Thision S 25.1 (LOGON B G2Z2 [RSV63.283/360] + LOGON B MM [AVS75.390/260]) (Gasbrenner)
- Elco Aquatop 8es [RVS51.843/169] (entspricht CTA Optihead OH1-8es) (Wärmepumpe)
- Fujitsu Waterstage WSYK 160 DC 9 [RVS21.827/127] (Wärmepumpe)
- Siemens RVS63.283/109 (entspr. Brötje SSR)
- Weishaupt WTU-25 G mit WRS-CPU B2/E [RVS23.220/320] {*LPB\_only*} (Ölbrenner)

### **3. Grundsätzliches zum BSB und LPB**

BSB (Boiler System Bus) und LPB (Local Process Bus) sind zwei verschiedene Bus-Typen, die sich vereinfacht in zwei Nutzungsklassen unterscheiden lassen:

1. BSB ist ein ‚lokaler‘ Bus zur Nutzung des lokal angeschlossenen Reglers.

Angeschlossene Geräte wie bspw. die Bedieneinheit, ein Raumgerät / Fernbedienung oder auch der (via BSB) angeschlossene Adapter können nur auf den Regler zugreifen, an dem sie angeschlossen sind.

2. LPB ist ein ‚übergreifender‘ Bus zur Nutzung mehrerer angeschlossener Regler in einem kommunikationsfähigen Verbund.

Über den LPB können mehrere Regler miteinander verbunden werden und bei korrekter Parametrierung (Stichwort Gerät- und Segmentadressen) gewisse Werte/Einstellungen/Parameter miteinander teilen bzw. sich gegenseitig beeinflussen.

Auf diese Weise kann bspw. eine Kaskadenschaltung von mehreren Brennern realisiert werden oder eine Gas- oder Öl-Heizung mit einer Solaranlage und einem Feststoffkessel regelungstechnisch ‚verbunden‘ werden.

Der korrekte Anschluss der einzelnen Komponenten sowie die korrekte Parametrierung der jeweiligen Regler sollte im Normalfall bereits bei der Installation der Anlage durch den Heizungsinstallateur erfolgt sein.

#### *Beispiel:*

Vorhanden sind eine Öl- oder Gasheizung, ein wasserführender Kamin und eine thermische Solaranlage zur Unterstützung des Heizkreises oder der Warmwasserbereitung.

Alle drei Wärmeerzeuger sind hydraulisch an einem Pufferspeicher angeschlossen. Die Wärme für den Heizkreis soll vom Pufferspeicher bezogen werden.

Die Regelung der Solaranlage und des Feststoffkessels (also bspw. der jeweiligen Pumpen) übernimmt ein SSR, der in einem externen Gehäuse samt Bedieneinheit vorhanden ist. Dieser ist wiederum mit dem (Heizungs-)Regler via LPB verbunden.

Die Kesselsteuerung der Heizung übernimmt in diesem Beispiel der interne Heizungsregler. Alle weiteren Sensoren, Pumpen, Mischer etc. sind aber am SSR angeschlossen, welcher wiederum die Steuerung der Solaranlage und des Feststoffkessels übernimmt.

Durch die Verbindung der beiden Regler (Heizungsregler und SSR) via LPB kann somit bspw. eine Pufferspeicherladung geregelt werden, bei der die Heizung nur aktiv wird, wenn weder Solar noch Feststoffkessel den Puffer laden / geladen haben.

Wenn ein Adapter via BSB an einem der beiden Regler aus oben genanntem Beispiel angeschlossen ist, kann er folglich nur auf den jeweiligen Regler ‚lokal‘ zugreifen, an dem er angeschlossen ist (also bspw. Heizungsregler oder SSR). Ebenso verhält es sich mit den jeweiligen Bedieneinheiten der Regler, die über den ‚Bus BE‘ angeschlossen sind.

Wenn ein Adapter via LPB an einem der beiden Regler aus oben genanntem Beispiel angeschlossen ist, müssen

1. die Geräte- und Segmentadressen entsprechend der LPB-Konfigurationsanforderungen eingestellt werden, und
2. beim Adapter eine Zieladresse eingestellt werden, an die die jeweiligen Anfragen des Adapters geschickt werden.

Die spezifischen technischen Daten, Leistungsmerkmale und Anforderungen an entsprechende Installationen und Parametrierungen hinsichtlich der Geräte- und Segmentadressen sind den jeweiligen technischen Dokumentationen der Hersteller zu entnehmen. Hinsichtlich des LPB seien insbesondere die Dokumentationen „LPB Systemgrundlagen“<sup>6</sup> und „LPB Projektierungsgrundlagen“<sup>7</sup> empfohlen.

Bei einigen Reglern sind die entsprechenden Anschlüsse teilweise unterschiedlich gekennzeichnet:

- Der BSB ist nicht auf allen Reglern als solcher bezeichnet, weitere Bezeichnungen sind „FB“ (Fernbedienung) sowie „CL+“ und „CL-“. Der zusätzliche Anschluss „G+“ führt 12V und ist für die Hintergrundbeleuchtung der entsprechenden Raumgeräte vorgesehen. Dieser ist für den Anschluss des Adapters NICHT zu verwenden!
- Der LPB ist bei einigen Reglern mit „DB“(+) und „MB“(–) gekennzeichnet.

Die folgenden Abbildungen zeigen exemplarisch die verschiedenen Anschlüsse.

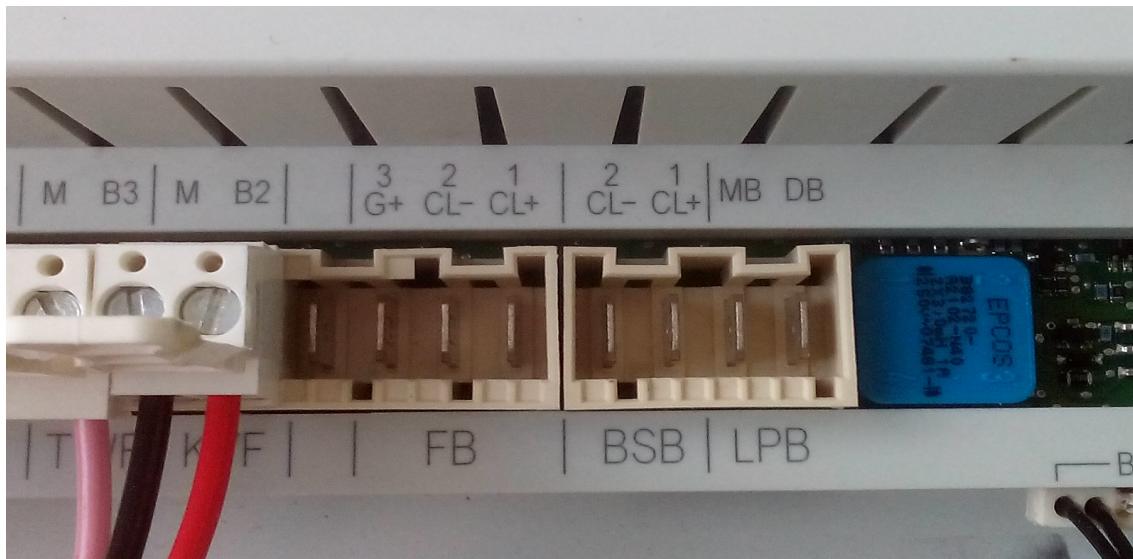


Abbildung 4: BSB & FB (CL+/CL-) und LPB (DB/MB) bei einem Brotje ISR-RVS43.222-Regler

6 Siemens Building Technologies - Landis & Staefa Division: CE1N2030D

7 Siemens Building Technologies - Landis & Staefa Division: CE1N2032D

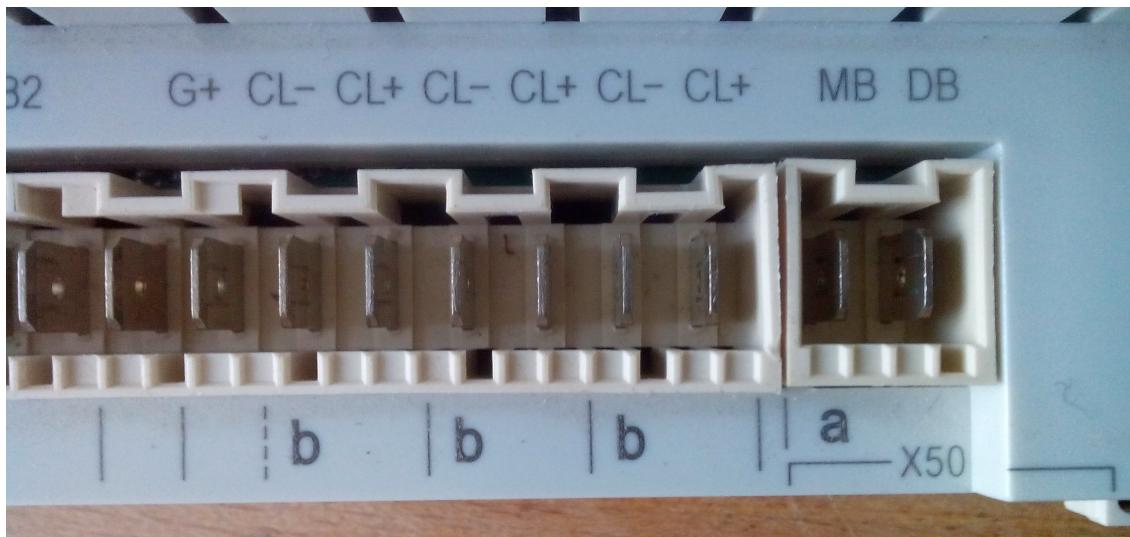


Abbildung 5: b = BSB (CL+/CL-) und a = LPB (DB/MB) bei einem Siemens RVS63.283-Regler

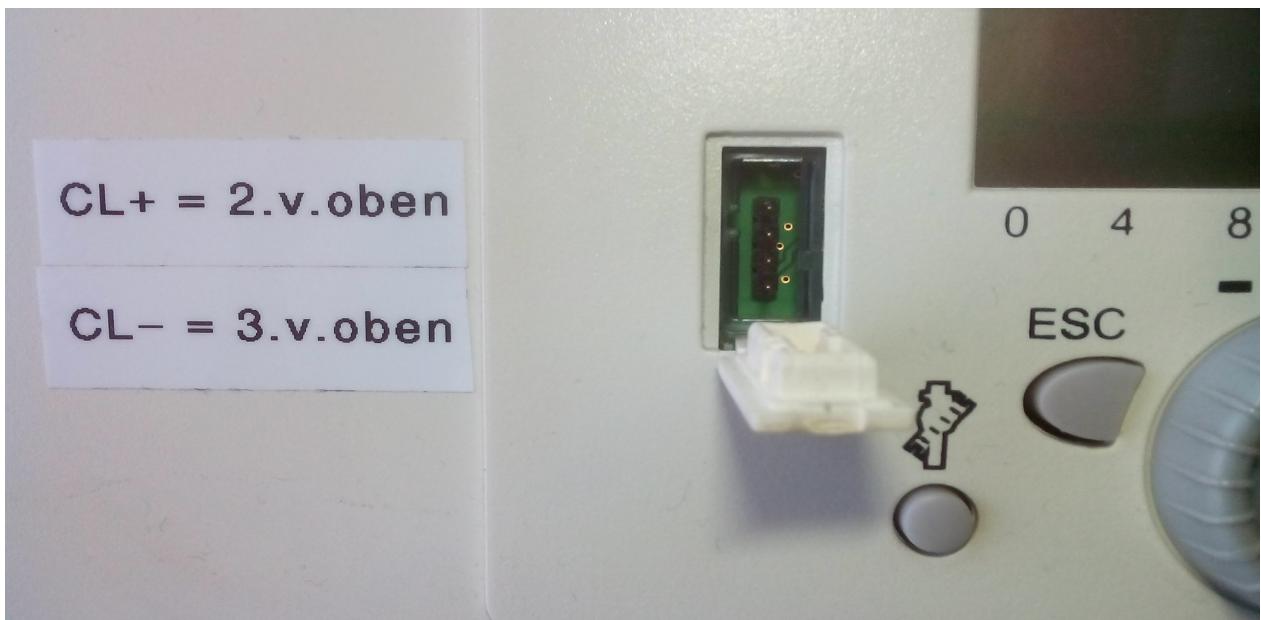


Abbildung 6: BSB (CL+/CL-) an der vierpoligen Servicebuchse vorne in der Bedieneinheit eines ISR Plus  
→ Die (dauerhafte) Verwendung dieses Anschlusses ist jedoch nicht zu empfehlen.

Bei Anschluss des Adapters sollte der betreffende Regler stets ausgeschaltet sein, ebenso bei einem Entfernen des Adapters.

Es ist unbedingt darauf zu achten, dass der Regler polrichtig angeschlossen wird!  
Ein verkehrter Anschluss kann eine Beschädigung des Reglers und/oder Adapters zur Folge haben!

Die entsprechende Polung bzw. Bezeichnung der Anschlüsse ist auf der Adapterplatine gekennzeichnet. Bei einem Nachbau ist der Schaltplan zu beachten.

Wenn du mehrere Regler (bspw. wie im obigen Beispiel) hast, bietet es sich derzeit noch an, pro Regler jeweils einen Adapter via BSB anzuschließen, um den jeweiligen Zugriff zu realisieren.

Eine übergreifende Abfrage von Werten oder Parametern zweier verschiedener Regler im LPB-Verbund kann via Adapter momentan noch nicht erfolgen, da aktuell noch kein 'fliegender' Wechsel der Zieladressen für Abfragen realisiert ist.

Da die Software ursprünglich für die Nutzung des Adapters via BSB geschrieben und erst nachträglich von Frederik um die LPB-Funktionalität erweitert wurde, ist es außerdem möglich, dass via LPB nicht alle Parameter verfügbar sind, die via BSB verfügbar wären. Die Software wird zwar stetig weiter entwickelt, derzeit bietet es sich jedoch noch an, als Anschluss den BSB zu präferieren, wenn dieser vorhanden ist.

#### *Tipps:*

Um vor Störeinflüssen möglichst geschützt zu sein, sollten die Anschlusskabel für den LPB-Anschluss gemäß LPB-Projektierungsgrundlagen<sup>8</sup> einen Querschnitt von 1,5mm<sup>2</sup> aufweisen, zweiseitig verdrillt und geschirmt sein (Leitungslänge max. 250m pro Busteilnehmer, max. Gesamtlänge 1000m).

Für den BSB-Anschluss sind Cu-Leitungen mit mindestens 0,8mm<sup>2</sup> (bis 20m) Querschnitt zu wählen, bspw. LiYY oder LiYCY 2 x 0,8. Bei Leitungslängen bis 80m sollte 1mm<sup>2</sup>, bis 120m sollten 1,5mm<sup>2</sup> Querschnitt gewählt werden<sup>9</sup>.

Generell ist eine parallele Verlegung mit Netzleitungen zu vermeiden (Störsignale), geschirmte Leitungen sind ungeschirmten Leitungen immer vorzuziehen.

Der Anschluss der Leitungen an die jeweiligen Kontakte sollte prinzipiell immer mit den spezifischen Steckern erfolgen.

Sollten diese nicht unmittelbar erhältlich oder verfügbar sein, können im Ausnahmefall (vorübergehend) auch isolierte 6,3mm-Kabelschuhe verwendet werden.

---

<sup>8</sup> Siemens Building Technologies - Landis & Staefa Division: CE1N2032D

<sup>9</sup> Siehe „Brötje Montageanleitung für Raumgerät RGT/RGTK“

## 4. Installation der Arduino IDE und Konfiguration des Adapters

- Downloade und installiere die aktuelle Version der Arduino IDE von <https://www.arduino.cc/en/Main/Software> (Windows-, Mac- und Linux-Version verfügbar).
- Schließe den Arduino (samt Ethernet-Shield und Adapter) via USB an deinem PC an.
- Downloade die aktuelle BSB-LAN-Version von [https://github.com/fredlcore/bsb\\_lan](https://github.com/fredlcore/bsb_lan) und entpacke die heruntergeladene Datei „bsb\_lan-master.zip“. Benenne den Ordner in „BSB\_lan“ um.
- Öffne den BSB\_lan-Sketch mittels eines Doppelklicks auf die Datei *BSB\_lan.ino* im BSB\_lan-Ordner. Die dazugehörigen Dateien *BSB\_lan\_config.h* und *BSB\_lan\_defs.h* werden automatisch mit geladen.
- Konfiguriere die IP-Adresse in *BSB\_lan\_config.h* deinem Netzwerk entsprechend. Die voreingestellte IP 192.168.178.88 funktioniert mit den meisten Standard-Routern wie bspw. Fritz!Box, aber prüfe, ob die IP bereits anderweitig vergeben ist, damit es nicht zu einer Adresskollision kommt.
- *Wichtig:* Passe nun die weiteren Einstellungen in der Datei *BSB\_lan\_config.h* deinen Wünschen und Hardwaregegebenheiten (Pinbelegungen, angeschlossene DS18B20- und/oder DHT22-Sensoren, zu loggende Parameter etc.) entsprechend an!
- Wähle "Arduino/Genuino Mega or Mega 2560" unter Tools/Board bzw. Werkzeuge/Board.
- Wähle "ATmega 2560" unter Tools/Processor bzw. Werkzeuge/Prozessor.
- Wähle "AVRISP mkII" unter Tools/Programmer bzw. Werkzeuge/Programmer.
- Lade den Sketch mittels Sketch/Upload bzw. Sketch/Hochladen auf den Arduino.
- Verbinde den Arduino mittels LAN-Kabel mit deinem Router/Switch. Stelle dabei sicher, dass eine Stromversorgung für den Arduino via USB oder externem Netzteil besteht!
- Öffne die Seite `http://<IP-Adresse>/` (oder `http://<IP-Adresse>/<passkey>/` wenn die Passkey-Funktion (s.u.) genutzt wird) um zu sehen, ob alles korrekt kompiliert und hochgeladen wurde.  
Die Startseite des Webinterface des Adapters sollte erscheinen.  
Sollte sie nicht erscheinen, drücke einmal kurz auf den Reset-Knopf des Arduino und rufe die Startseite erneut auf.

Unter `http://<IP-Adresse>/C` (bzw. Menüpunkt „Konfiguration“ im Webinterface) kannst du deine Konfiguration überprüfen.

*Wenn alle einstellungsrelevanten Parameter in der Datei BSB\_lan\_config.h angepasst sind (siehe nächstes Kapitel) und der Zugriff auf das Webinterface möglich ist, fahre mit der Funktionsüberprüfung des Adapters fort.*

## 5. Einstellungsrelevante Parameter der BSB-LAN-Software

Folgende Parameter in der Datei *BSB\_lan\_config.h* können bzw. sollten vor der Verwendung des Adapters angepasst werden:

- Typ des LAN-Shield-Chips (W5100 oder W5500):

Standardmäßig wird die Verwendung eines LAN-Shields mit dem Chip des Typs W5100 vorausgesetzt (s. Aufkleber auf der LAN-Buchse des Shields). Soll aber ein Shield mit Chip-Typ W5500 verwendet werden, ist das entsprechende Definement zu aktivieren:

```
#define ETHERNET_W5500
```

Zusätzlich muss die Datei *Ethernet2.zip* im Unterverzeichnis *src* entpackt werden.

- IP-Adresse:

```
#define IPAddress 192,168,178,88
```

- Ethernet-Port:

```
#define Port 80
```

- Um das System vor einem ungewollten Zugriff von außen zu schützen, kann die Funktion des Sicherheitsschlüssels (PASSKEY) aktiviert werden (sehr einfach und nicht wirklich sicher!):

```
#define PASSKEY "1234"
```

Falls die PASSKEY-Funktion aktiviert ist, muss die URL bei einem Aufruf des Webinterfaces den definierten Schlüssel als erstes Element enthalten, bspw.

`http://<IP-Adresse>/<passkey>/`

um die Hilfeseite zu sehen.

*Bitte nicht den Slash hinter dem Passkey vergessen!*

*Die URLs in den folgenden Beispielen müssen um die PASSKEY-Definition erweitert werden, falls die Funktion aktiviert wurde.*

- Darüber hinaus gibt es zwei weitere Sicherheitsfunktionen: TRUSTED\_IP und USER\_PASS\_B64

TRUSTED\_IP kann man auf das letzte Segment einer vertrauenswürdigen IP setzen (z.B. des FHEM-Servers), dann ist der Zugriff nur über die IP mit dieser Endung möglich. Lautet die vertrauenswürdige IP des Clients bspw. 192.168.178.20, so ist `#define TRUSTED_IP 20` einzustellen.

Mit USER\_PASS\_B64 kann ein in Base64-codierter String nach dem Muster *username:password* als Zugangssperre gesetzt werden. Voreingestellt ist hier der Benutzername "atari" und das Passwort "800xl" (codiert: YXRhcmk6ODAwGw=). Um eine andere Kombination zu nutzen, gehe bspw. auf die Website <https://www.base64encode.org>, lasse dein neues Passwort im Format *username:password* erstellen und füge die Codierung hinter dem Definement ein:  
`#define USER_PASS_B64 "YXRhcmk6ODAwGw="`

- Konfiguration des Heizungssystems:

```
int fixed_device_id = 0
```

Wenn der Wert auf 0 gesetzt ist, ist die automatische Erkennung des angeschlossenen Reglers beim Starten des Arduino aktiviert.

Alternativ kann hier die Ausgabe von `http://<IP-Adresse>/6225` eingetragen werden.

Ein fest eingestellter Wert (laut Ausgabe von Parameter 6225) stellt sicher, dass BSB\_LAN auch dann noch korrekt arbeitet, wenn die Heizung erst nach dem Starten des Arduino eingeschaltet wird (da in dem Fall die automatische Erkennung des angeschlossenen Reglers nicht funktionieren kann, da ja keine Rückmeldung vom Regler kommt).

- Die Sprache der Benutzeroberfläche des Webinterface des Adapters ist auf Deutsch voreingestellt. Möchte man die Sprache auf Englisch umstellen, muss das entsprechende Definement deaktiviert werden:

```
//#define LANG_DE;
```

- Sollen OneWire-Temperatursensoren (DS18B20) verwendet werden, muss die entsprechende Pinbelegung (DATA-Anschluss des Sensors am Adapterboard) definiert werden (<n> = Pinnummer):

```
#define ONE_WIRE_BUS <n>
```

- Sollen DHT22-Sensoren (Temperatur & Feuchtigkeit) verwendet werden, muss die entsprechende Pinbelegung (DATA-Anschluss des Sensors am Adapterboard) definiert werden (<n> = Pinnummer):

```
#define DHT_BUS <n>
```

- Sollen 24h-Durchschnittswerte von bestimmten Parametern berechnet werden, müssen diese bei der entsprechenden Variable eingetragen werden, bspw.:

```
int avg_parameters[20] = {
    8700,      // Außentemperatur
    8740      // Raumtemperatur-Ist
};
```

- Sollen bestimmte Werte/Parameter auf eine microSD-Karte geloggt werden, ist eine FAT32-formatierte Karte im entsprechenden Slot auf dem Ethernet-Shield einzulegen und das entsprechende Definement zu aktivieren:

```
#define LOGGER
```

Die zu loggenden Parameter müssen dann bei der entsprechenden Variable eingetragen werden, bspw.:

```
int log_parameters[20] = {
    8700,      // Außentemperatur
    8740      // Raumtemperatur-Ist
};
```

Wenn mehrere DS18B20- oder DHT22-Sensoren geloggt werden sollen, müssen diese bei den Log-Parametern entsprechend einzeln untereinander aufgeführt werden, bspw.:

[ ... ]

20200, // Spezialparameter 20200-20299: DS18B20-Sensoren 1-100 (/T)

20201,

20202,

[ ... ]

loggt die Werte der DS18B20-Sensoren 1-3.

*Hinweis:*

*Zum Loggen der Brennerstarts und -laufzeiten müssen die Spezialparameter 2001 und 2002 aufgeführt werden (siehe auch die Beschreibung in der Datei BSB\_lan\_config.h).*

*Bei einem zweistufiger Öl**brenner**, dessen Regler die entsprechenden Broadcasts schickt und bei dem eine Differenzierung der Brennerstufen möglich ist<sup>10</sup> müssen hier zusätzlich 2003 und 2004 mit aufgeführt werden!*

- Soll die IPWE-Erweiterung aktiviert werden, ist das entsprechende Definement  
`#define IPWE`

zu aktivieren, die gewünschten Parameter sind wie gewohnt einzutragen.

*Tipp:*

*Werden DS18B20- und/oder DHT22-Sensoren verwendet, werden diese hier standardmäßig mit angezeigt (<http://<IP-Adresse>/ipwe.cgi>). Dabei wird neben den gemessenen Werten auch die jeweils spezifische Hardwarekennung der Sensoren aufgeführt. Dies ist besonders bei einer Ersteinrichtung für eine eindeutige Unterscheidung der einzelnen Sensoren hilfreich.*

- Soll der Arduino per URL-Befehl mittels `http://<IP-Adresse>/X` resettet werden können, muss das entsprechende Definement aktiviert werden:

`#define RESET`

- MAC-Adresse des Ethernet-Shields:

`byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xEA};`

Üblicherweise befindet sie sich auf einem Aufkleber auf dem Ethernet-Shield, diese ist dann einzutragen. Sollte kein Aufkleber mit einer spezifischen Adresse vorhanden sein, kann die voreingestellte Adresse beibehalten werden. Eine Änderung ist i.d.R. nur nötig, wenn mehr als ein Adapter verwendet wird.

- Konfiguration des Adapters:

`BSB bus(68,69,<my_addr>,<dest_addr>);`

RX-Pin, TX-Pin, eigene Bus-Adresse (voreingestellt auf 0x06=RGT1), Bus-Adresse des Zielsystems (voreingestellt auf 0x00=Heizungsregler).

Voreingestellt wird der Adapter mit `BSB bus(68,69)` am angeschlossenen Regler als RGT1 angemeldet.

---

<sup>10</sup> Derzeit nur Brötje BOB, siehe Hinweis in Kapitel 8 unter „/B“.

Wenn bereits ein Raumgerät (RGT1) vorhanden ist, kann bzw. sollte der Adapter als RGT2 am angeschlossenen Regler angemeldet werden, um etwaige Adresskollisionen zu vermeiden:

```
BSB bus(68,69,7);
```

*Hinweis:*

*Bitte beachte, dass ein als RGT1 angemeldeter Adapter keine Temperaturen an einen HK2 senden kann, und ein als RGT2 angemeldeter Adapter keine Temperaturen an einen HK1 senden kann!*

*Ob eine Bedienung eines HK1 mit einem als RGT2 (oder HK2 mit einem als RGT1) angemeldeten Adapter in vollem Umfang möglich ist, ist bisher noch nicht ausgiebig getestet.*

- Bus-Protokoll:

```
uint8_t bus_type = bus.setBusType(0);
```

Voreingestellt ist 0 für BSB, für LPB ist 1 einzustellen.

- In der Voreinstellung ist der Zugriff des Adapters auf den Regler auf Lesen beschränkt, d.h. ein Setzen bzw. Verändern von Parametern der Heizungssteuerung per Adapter ist standardmäßig nicht möglich.

Das betreffende Definement lautet:

```
#define DEFAULT_FLAG FL_RDONLY;
```

Wer den Status ändern will, um generell Werte und Einstellungen des Reglers per Adapter verändern zu können, muss das Flag auf 0 setzen:

```
#define DEFAULT_FLAG 0;
```

Ist diese Funktion nur bei *ausgewählten* Parametern (z.B. 10000 oder 710) gewünscht, muss bei dem genannten Definement nach wie vor das Flag generell auf FL\_RDONLY gesetzt sein und dann in der Datei *BSB\_lan\_defs.h* der Wert DEFAULT\_FLAG des gewünschten Parameters durch 0 ersetzt werden.

## 6. Funktionsüberprüfung und erste Nutzung des Adapters

Um zu überprüfen, ob der angeschlossene Regler korrekt erkannt wird, sollte bei der Ersteinrichtung eine Funktionsüberprüfung vorgenommen werden.

Dazu bietet sich folgende Vorgehensweise an:

1. Den Regler ausschalten und mit dem Adapter via BSB verbinden.

*Bitte achte dabei auf den polrichtigen Anschluss von CL+ und CL- !*

*Hinweis:* Wenn (später) der LPB genutzt werden soll, muss der Bus-Typ in der Datei *BSB\_lan\_config.h* geändert werden! *Bitte achte auch hier auf die richtige Polung!*

2. Den Regler einschalten und überprüfen, ob die rote LED auf dem Adapter leuchtet.

Sollte die LED in unregelmäßigen Abständen flackern, ist dies keine Fehlfunktion, sondern zeigt Aktivität auf dem Bus an.

3. Den Adapter nun via USB mit dem PC und via LAN mit dem Netzwerk verbinden.

4. Nun die Arduino IDE starten, den korrekten Anschluss des Arduino Mega2560 auswählen (COM-Port) und dann unter „Werkzeuge“ den „Seriellen Monitor“ starten.

5. In der Weboberfläche `http://<IP-Adresse>/6224-6226` aufrufen.

Dabei sollte die rote LED des Adapters kurzzeitig flackern und Bus-Aktivität anzeigen.

- a) Wird der angeschlossene Regler bereits automatisch korrekt erkannt, steht direkt am Anfang der SerMo-Ausgabe bei „Device family“ ein Wert, der nicht „0“ ist. Bei „Device ID“ wird eine zwei- oder dreistellige Zahl angegeben. Beispiel:

```
[...]
Device family: 96
Device ID: 32
[...]
```

- b) Wird der angeschlossene Regler nicht automatisch korrekt erkannt, steht bei „Device family“ eine „0“, bei „Device ID“ wird bspw. „4294967295“ angezeigt. Vor „Device family“ stehen in dem Fall drei Zeilen „query failed“. Beispiel:

```
[...
query failed
query failed
query failed
Device family: 0
Device ID: 4294967295
[...]
```

In diesem Fall sollten die Ausgaben von `http://<IP-Adresse>/6223-6228` unter Angabe des verwendeten Heizungstyps bitte gemeldet werden, damit ein entsprechender Eintrag für dieses System in der Datei *BSB\_lan\_defs.h* und im README erstellt werden kann.

Somit wird sichergestellt, dass Regler dieses Typs künftig automatisch erkannt werden und der volle Funktionsumfang zur Verfügung steht.

## 7. BSB-LAN Web - das Webinterface des Adapters

Die Startseite des Webinterface wird angezeigt, wenn ohne weitere Parameter auf die URL des Servers zugegriffen wird:

`http://<IP-Adresse>`

Bei Verwendung der Passkey-Sicherheitsfunktion muss die URL um den definierten Sicherheitsschlüssel erweitert werden, bspw.:

`http://<IP-Adresse>/<passkey>/`

*Bitte den Slash hinter dem Passkey nicht vergessen!*

## BSB-LAN Web

<a href="#">Heizungsfunktionen</a>	<a href="#">DS18B20-Sensoren</a>	<a href="#">DHT22-Sensoren</a>	<a href="#">Anzeige Logdatei</a>
<a href="#">Konfiguration</a>	<a href="#">URL-Befehle</a>	<a href="#">HowTo</a>	<a href="#">FAQ</a>

BSB-LAN Web, Version 0.38

**Heizungsfunktionen:** Hiermit können Sie die Funktionen Ihres Heizungssystems abfragen bzw. steuern. Die einzelnen Parameter sind in entsprechende Kategorien aufgeteilt, die Sie anklicken können.

**Konfiguration anzeigen:** Hier sehen Sie, mit welchen Werten BSB-LAN konfiguriert ist. Eine Änderung dieser Parameter ist über die erweiterten URL-Befehle möglich.

**Erweiterte URL-Befehle:** Zeigt Ihnen eine Übersicht der Befehle an, die Sie über die direkte Eingabe in der Adresszeile des Browsers absenden können. Diese Befehle sind auch für die Anbindung an Hausautomations-Systeme wie FHEM nötig. |

Abbildung 7: Screenshot der Startseite des Webinterface

Prinzipiell sind alle Parameter in Kategorien zusammengefasst, die den im Display dargestellten Untermenükategorien entsprechen, wenn auf den Regler des Heizungssystems vom integrierten Bedienteil aus zugegriffen wird.

Ein Klick auf den Menüpunkt „Heizungsfunktionen“ zeigt eine vollständige Übersicht der Kategorien, die wiederum ebenfalls anwählbar sind (s. Abb. 8).

Ein Klick auf eine der gezeigten Kategorien (bspw. Heizkreis 1) startet eine Komplettabfrage der jeweiligen Kategorie, also aller Parameter, die in dieser Kategorie verfügbar sind (s. Abb. 9).

# BSB-LAN Web

<a href="#">Heizungsfunktionen</a>	<a href="#">DS18B20-Sensoren</a>	<a href="#">DHT22-Sensoren</a>	<a href="#">Anzeige Logdatei</a>
<a href="#">Konfiguration</a>	<a href="#">URL-Befehle</a>	<a href="#">HowTo</a>	<a href="#">FAQ</a>

- Brennerstatistik  
24h Durchschnittswerte
- 0 - Uhrzeit und Datum  
1 - Bedieneinheit  
2 - Funk  
3 - Zeitprogramm Heizkreis 1  
4 - Zeitprogramm Heizkreis 2  
5 - Zeitprogramm 3/HKP  
6 - Zeitprogramm 4/TWW  
7 - Zeitprogramm 5  
8 - Ferien Heizkreis 1  
9 - Ferien Heizkreis 2  
10 - Ferien Heizkreis P  
11 - Heizkreis 1  
12 - Kühlkreis 1  
13 - Heizkreis 2  
14 - Heizkreis 3/P  
15 - Trinkwasser  
16 - Hx-Pumpe  
17 - Schwimmbad  
18 - Vorregler/Zubringerpumpe  
19 - Kessel

Abbildung 8: Screenshot Menü "Heizungsfunktionen" mit den Parameter-Kategorien (Auszug)

# BSB-LAN Web

<a href="#">Heizungsfunktionen</a>	<a href="#">DS18B20-Sensoren</a>	<a href="#">DHT22-Sensoren</a>	<a href="#">Anzeige Logdatei</a>
<a href="#">Konfiguration</a>	<a href="#">URL-Befehle</a>	<a href="#">HowTo</a>	<a href="#">FAQ</a>
<p>700 Heizkreis 1 - Betriebsart: 1 - Automatik      710 Heizkreis 1 - Komfortsollwert: 19.0 °C      711 Heizkreis 1 - Komfortsollwert Maximum: 35.0 °C      712 Heizkreis 1 - Reduziertsollwert: 19.0 °C      714 Heizkreis 1 - Frostschutzsollwert: 4.0 °C      720 Heizkreis 1 - Kennlinie Steilheit: 1.30      721 Heizkreis 1 - Kennlinie Verschiebung: 0.0 °C      726 Heizkreis 1 - Kennlinie Adaption: 0 - Aus      730 Heizkreis 1 - Sommer-/ Winterheizgrenze: 20.0 °C      732 Heizkreis 1 - Tagesheizgrenze: -3.0 °C      740 Heizkreis 1 - Vorlauftollwert Minimum: 8.0 °C      741 Heizkreis 1 - Vorlauftollwert Maximum: 80.0 °C      750 Heizkreis 1 - Raumeinfluss: 75 %      760 Heizkreis 1 - Raumtemperaturbegrenzung: 0.5 °C      770 Heizkreis 1 - Schnellaufheizung: 0.0 °C      780 Heizkreis 1 - Schnellabsenkung: 0 - Aus</p>		<input type="text" value="Automatik"/> <input checked="" type="checkbox"/> Set <input type="text" value="19.00"/> <input checked="" type="checkbox"/> Set <input type="text" value="35.00"/> <input checked="" type="checkbox"/> Set <input type="text" value="19.00"/> <input checked="" type="checkbox"/> Set <input type="text" value="4.00"/> <input checked="" type="checkbox"/> Set <input type="text" value="1.30"/> <input checked="" type="checkbox"/> Set <input type="text" value="0.00"/> <input checked="" type="checkbox"/> Set <input type="text" value="0.00"/> <input checked="" type="checkbox"/> Set <input type="text" value="20.00"/> <input checked="" type="checkbox"/> Set <input type="text" value="-3.00"/> <input checked="" type="checkbox"/> Set <input type="text" value="8.00"/> <input checked="" type="checkbox"/> Set <input type="text" value="80.00"/> <input checked="" type="checkbox"/> Set <input type="text" value="75.00"/> <input checked="" type="checkbox"/> Set <input type="text" value="0.50"/> <input checked="" type="checkbox"/> Set <input type="text" value="0.00"/> <input checked="" type="checkbox"/> Set <input type="text" value="Aus"/> <input checked="" type="checkbox"/> Set	

Abbildung 9: Screenshot Parameterauflistung Kategorie "Heizkreis 1" (Auszug) bei aktivierter Set-Funktion

Unter der URL

<http://<IP-Adresse>/C>

wird eine Übersicht der Konfiguration dargestellt. Dort sind u.a. der Monitor-Modus, der Verbositäts-Level, die definierten Pins für optional angeschlossene Sensoren, die zu loggenden Parameter und Parameter, von denen 24h-Mittelwerte berechnet werden sollen, auf einen Blick überprüfbar.

Eine schwarze Schrift bei den Schaltflächen für DS18B20- und DHT22-Sensoren zeigt an, dass diese nicht definiert sind.

Bei dem folgenden abgebildeten Beispiel sind lediglich DS18B20-Sensoren definiert und es werden keinerlei 24h-Mittelwerte berechnet. Alle zehn Minuten (→ 600 Sekunden) werden Temperaturwerte von insgesamt sechs DS18B20-Sensoren geloggt.

## BSB-LAN Web

<a href="#">Heizungsfunktionen</a>	<a href="#">DS18B20-Sensoren</a>	<a href="#">DHT22-Sensoren</a>	<a href="#">Anzeige Logdatei</a>
<a href="#">Konfiguration</a>	<a href="#">URL-Befehle</a>	<a href="#">HowTo</a>	<a href="#">FAQ</a>

### Konfiguration

RAM: 1492Bytes

Bus-System: BSB

Monitor Modus: 0 |

Verbositäts-Level: 0

1-Wire bus pins: 31

Excluded GPIO pins: 10 11 12 13 50 51 52 53 62 63 64 65 66 67 68 69

MAC address: DA DE AB CD EF FA

Berechnung von 24h-Mittelwerten für die folgenden Parameter:

Loggen der folgenden Parameter alle 600 Sekunden:

8730 - Heizkreispumpe Q2

20000 - Brennerlaufzeit

20001 - Brennertakte

20200 - 1-Wire-Sensor

20201 - 1-Wire-Sensor

20202 - 1-Wire-Sensor

20203 - 1-Wire-Sensor

20204 - 1-Wire-Sensor

20205 - 1-Wire-Sensor

Abbildung 10: Screenshot der Übersichtsseite "Konfiguration" (<IP-Adresse>/C)

Der Menüpunkt „URL-Befehle“ zeigt eine Auflistung und kurze Erklärung der URL-Befehle. Grau hinterlegte Zeilen zeigen an, dass der Befehl nicht verfügbar ist. Im nachfolgend gezeigten Beispiel betrifft dies die nicht-installierten DHT22-Sensoren.

## BSB-LAN Web

Heizungsfunktionen	DS18B20-Sensoren	DHT22-Sensoren	Anzeige Logdatei
Konfiguration	URL-Befehle	HowTo	FAQ

Erweiterte Befehle:

- /K Alle verfügbaren Kategorien auflisten.
  - /Kx Alle Werte von Kategorie x abfragen.
  - /x-y Alle Werte eines Zeilenbereichs abfragen (von Zeile x bis Zeile y).
  - /Sx=v Setze Wert v (value) für den Parameter x (Leerzeichen nach = deaktiviert den Wert).
  - /lx=v Sende eine INF Nachricht für den Parameter x mit dem Wert v.
  - /Ex Alle enum-Werte für Parameter x auflisten.
  - /Rx Frage den Reset-Wert für Parameter x ab.
  - /Px Setzen des Bus-Protokolls (0=BSB, 1=LPB).
  - /Vn Setze den Verbositäts-Level auf n.
  - /Mn Bus-Monitor aktivieren/deaktivieren (n=0 deaktivieren, n=1 aktivieren).
  - /Gxx Abfragen des GPIO Pins xx.
  - /Gxx=y Setzen des GPIO Pins xx auf high (y=1) oder low (y=0).
  - /A 24h-Durchschnittswerte von ausgewählten Parametern anzeigen (in BSB\_lan\_config.h definieren).
  - /A=x,y,z Ändern der 24h-Durchschnittswerte in x,y,z (bis zu 20 Parameter).
  - /B Anzeige der akkumulierten Broadcast-Telegramme zu Brenner- und TWW-Aktivitäten.
  - /B0 Zurücksetzen der akkumulierten Broadcast-Telegramme.
  - /T Abfrage von angeschlossenen DS18B20 Temperatursensoren (optional).  
Aktiviere das Definement #define DHT\_BUS in BSB\_lan\_config.h für den folgenden Befehl:
  - /H Abfrage von DHT22 Feuchtigkeits-/Temperatursensoren (optional).
  - /D Darstellung des Logfiles datalog.txt auf der microSD-Karte.
  - /D0 Löschen bzw. Zurücksetzen des Logfiles datalog.txt auf der microSD-Karte.
  - /L=x,y,z Setzt das Logging-Intervall auf x Sekunden und (optional) die Logging-Parameter auf y und z (bis zu 20 Parameter). Um das Loggen zu deaktivieren, kann L=0,0 genutzt werden.
  - /LU=x Wenn Bus-Telegramme geloggt werden (Logging-Parameter 30000 als einzigen Parameter setzen!), logge nur unbekannte commandIDs (x=1) oder alle Telegramme (x=0).
  - /LU=x Wenn Bus-Telegramme geloggt werden (Logging-Parameter 30000 als einzigen Parameter setzen!), logge nur Broadcast-Telegramme (x=1) oder alle Telegramme (x=0).
  - /X Reset des Arduino durchführen.
- Mehrere Abfragen können miteinander verkettet werden, z.B. /K0/710/8000-8999/T

Abbildung 11: Screenshot Menü "URL-Befehle"

Zusätzlich zum Webinterface kann somit auf alle Funktionen mittels Eingabe des entsprechenden Befehls direkt zugegriffen werden. Dies ist nützlich, wenn der Adapter in Verbindung mit Heimautomationssystemen wie bspw. FHEM genutzt wird.

Eine Übersicht und ausführlichere Beschreibung der URL-Befehle findet sich im Kapitel 8.

Generell werden alle Heizungsparameter anhand ihrer Zeilennummern abgefragt. Eine nahezu vollständige Übersicht findet sich bspw. im Systemhandbuch des Brötje ISR Plus.

Einige Zeilen sind 'virtuell' und wurden hinzugefügt, um bspw. den Zugang zu komplexen Einstellungen wie den Tagesprogrammen zu erleichtern.

Eine grafische Darstellung des Logfiles erfolgt bei Klick auf „Anzeige Logdatei“.

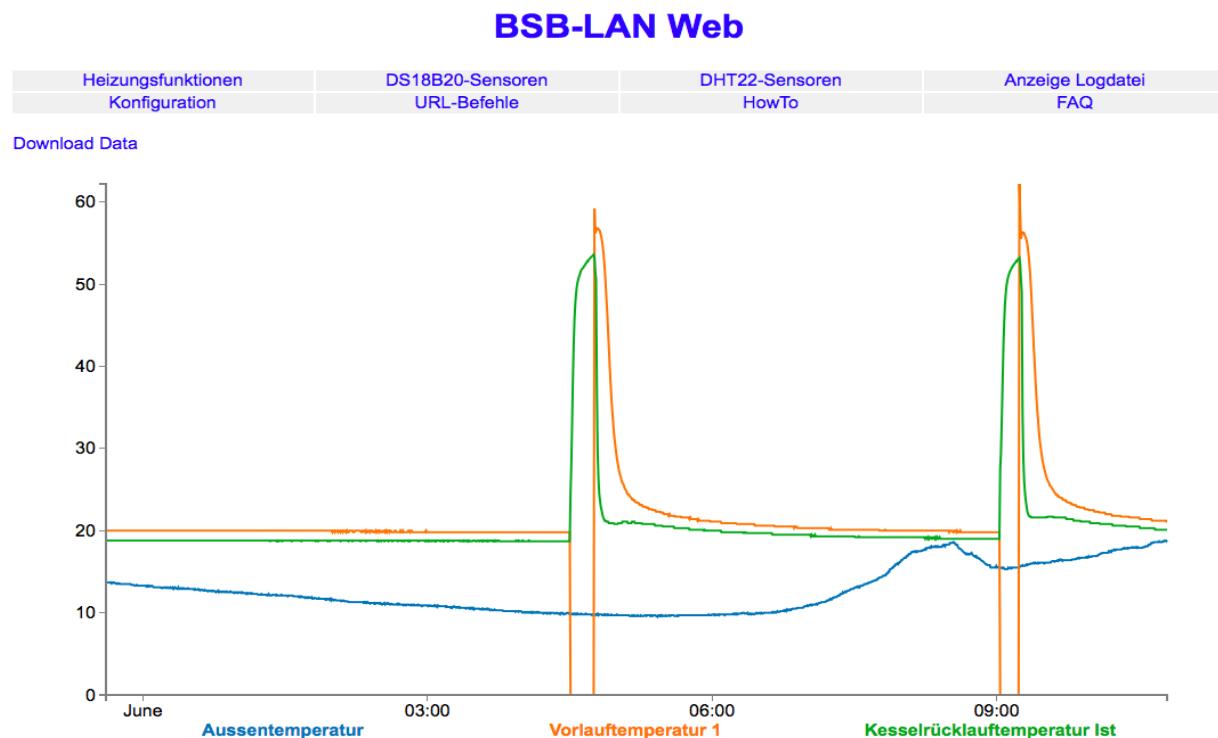


Abbildung 12: Screenshot der grafischen Darstellung der Logdatei

*Bitte beachte, dass der Arduino nicht multitaskingfähig ist. Eine neue Abfrage kann erst erfolgen, nachdem die vorhergehende Abfrage komplett beendet ist. Speziell die Abfrage mehrerer Parameter, ganzer Kategorien oder auch des SD-Logfiles kann u.U. eine längere Zeit in Anspruch nehmen, während dieser der Adapter nicht ‚ansprechbar‘ ist.*

## 8. Auflistung und Beschreibung der URL-Befehle

Da das Webinterface prinzipiell nur „aufgesetzt“ ist, um eine Bedienung ohne weitere Programme wie bspw. FHEM zu ermöglichen, ist ein direkter Zugriff auf die einzelnen Funktionen und Parameter mittels anderer Programme grundsätzlich möglich.

- Alle Kategorien auflisten:

`http://<IP-Adresse>/K`

Bei diesem Befehl kommuniziert der Adapter nicht mit dem Heizungssystem.

Es ist eine softwareseitige, interne Funktion.

- Alle enum-Werte für Parameter <x> auflisten:

`http://<IP-Adresse>/E<x>`

Bei diesem Befehl kommuniziert der Adapter nicht mit dem Heizungssystem.

Es ist eine softwareseitige, interne Funktion. Dieser Befehl ist nur für Parameter des Typs VT\_ENUM verfügbar.

- Alle Werte von Kategorie <x> abfragen:

`http://<IP-Adresse>/K<x>`

- Alle Werte von Parameter <x> abfragen:

`http://<IP-Adresse>/<x>`

- Alle Werte eines Parameterbereichs abfragen (von Parameter <x> bis Parameter <y>):

`http://<IP-Adresse>/<x>-<y>`

- Mehrere Abfragen können miteinander verkettet werden, z.B.:

`http://<IP-Adresse>/K11/8000/8003/8005/8300/8301/8730-8732/8820`

- Frage den Reset-Wert für Parameter <x> ab:

`http://<IP-Adresse>/R<x>`

Im Display der integrierten Heizungssteuerung gibt es für einige Parameter eine Reset-Option. Ein Reset wird vorgenommen, indem das System nach dem Reset-Wert gefragt wird und dieser anschließend gesetzt wird.

- Setze Wert <v> (value) für den Parameter <x> mit optionalem Ziel <z>:

`http://<IP-Adresse>/S<x>=<v !z>`

Die gewünschte Gerätezieladresse ist als <z> einzufügen, wenn <!z> nicht eingegeben wird, wird die Standardzieladresse verwendet.

*ACHTUNG: Diese Funktion ist nicht ausgiebig getestet! Bitte sei vorsichtig mit dieser Funktion und nutze sie ausschließlich auf dein eigenes Risiko hin. Das Format des Wertes hängt von seinem Typ ab. Einige Parameter können abgeschaltet werden.*

Um einen Parameter auf 'abgeschaltet/deaktiviert' zu setzen, muss lediglich ein leerer Wert eingefügt werden:

`http://<IP-Adresse>/S<x>=`

*Die Beschreibung der Wertformatierungen wird hier hinzugefügt. Bis dahin sieh dir den Quellcode an (Funktion 'set').*

*Hinweis: Voreingestellt ist nur der Lesezugriff erlaubt, siehe Kapitel 5 letzter Punkt.*

- Sende eine INF-Nachricht für den Parameter <x> mit dem Wert <v>:  
`http://<IP-Adresse>/I<x>=<v>`
- Setze das Bus-Protokoll vorübergehend auf BSB (<x>=0) oder LPB (<x>=1):  
`http://<IP-Adresse>/P<x>`
- Um zusätzlich die eigene oder die Zieladresse mittels URL-Befehl zu wechseln, muss  
`http://<IP-Adresse>/P<x,y,z>`  
genutzt werden, wobei  
<x> = Bus (0 = BSB, 1 = LPB),  
<y> = eigene Adresse (default 0x06 = RGT1) und  
<z> = Zieladresse (default 0x00 = Geräteadresse 1) sind.  
Leerwerte bei den Adressen belassen den bisherigen Wert (= Adresse).  
**ACHTUNG: Diese Funktion wurde noch nicht ausgiebig getestet!**
- Einige Werte können nicht direkt gesetzt werden. Das Heizungssystem wird mit einer TYPE\_INF-Nachricht informiert, bspw. die Raumtemperatur:  
`http://<IP-Adresse>/I10000=19.5 // Raumtemperatur beträgt 19.5°C`
- Setze den Verbositäts-Level auf <n>:  
`http://<IP-Adresse>/V<n>`  
Der voreingestellte Verbositäts-Level ist 0. Wenn <n> auf 1 gesetzt wird, werden der Bus überwacht und alle Daten zusätzlich im Raw-Hex-Format dargestellt.  
Der Verbositäts-Level betrifft sowohl die serielle Konsole des Arduino als auch (optional) das Loggen der Bus-Daten auf die microSD-Karte, so dass die Karte u.U. sehr schnell voll wird! Die html-Ausgabe bleibt unverändert.
- Bus-Monitor aktivieren:  
`http://<IP-Adresse>/M<n>`  
Wenn <n> auf 1 gesetzt wird, werden alle Bytes auf dem Bus überwacht. Telegramme werden durch Umbruchzeichen als solche erkannt. Jedes Telegramm wird im Hex-Format auf der seriellen Konsole mit einem Zeitstempel in Millisekunden dargestellt.  
Die Ausgabe der Überwachung betrifft nur die serielle Konsole des Arduino.  
Die html-Ausgabe bleibt unverändert.
- Setzen/Abfragen der GPIO Pins:  
`http://<IP-Adresse>/G<xx>[=<y>]`  
Gibt den momentanen Status von GPIO Pin <xx> an (0 oder 1). Kann ebenfalls benutzt werden, um den Pin auf <y>=0 (LOW) oder <y>=1 (HIGH) zu setzen.  
Reservierte Pins, die nicht gesetzt werden dürfen, können in der *BSB\_lan\_config.h* unter dem Parameter *GPIO\_exclude* gesperrt werden.
- 24h-Durchschnittswerte von ausgewählten Parametern anzeigen:  
`http://<IP-Adresse>/A[=parameter1,...,parameter20]`  
Zeigt rollierende 24h-Durchschnittswerte ausgewählter Parameter an. Initiale Festlegung dieser Parameter in *BSB\_lan\_config.h* in der Variable *avg\_parameters*.  
Während der Laufzeit kann "/A=[parameter1],...,[parameter20]" verwendet werden, um (bis zu 20) neue Parameter zu definieren.

- Abfrage von DS18B20-Temperatursensoren:  
`http://<IP-Adresse>/T`  
Gibt die Temperaturwerte von optional angeschlossenen DS18B20-Sensoren aus.
- Abfrage von DHT22-Feuchtigkeits-/Temperatursensoren:  
`http://<IP-Adresse>/H`  
Gibt die Temperatur- & Feuchtigkeitswerte von optional angeschlossenen DHT22-Sensoren aus.
- Akkumulierte Brennerlaufzeit anzeigen:  
`http://<IP-Adresse>/B`  
Fragt sowohl die akkumulierte Brennerlaufzeit (in Sekunden) und die Brennerstarts/takte als auch die Anzahl und die Dauer der Ladungen (in Sekunden) des Trinkwasserspeichers ab, die anhand von Broadcast-Nachrichten ermittelt wurden;  
/B0 setzt den Zähler zurück.

Bei zweistufigen Ölbrennern wird zudem *eventuell* zwischen Stufe 1 und 2 differenziert und die jeweiligen Starts und Laufzeiten werden angezeigt.

Beispiel: Brenner läuft 100 Sekunden, davon zuerst 20 Sekunden in Stufe 2, dann 40 Sekunden in Stufe 1 und dann noch mal 40 Sekunden in Stufe 2.

Takte Stufe 1:	1
Takte Stufe 2:	2
Laufzeit Stufe 1:	40
Laufzeit Stufe 2:	60

#### *Hinweis:*

*Diese Unterscheidung der Brennerstufen findet aufgrund von spezifischen Broadcasts (BC) statt, also vom Regler automatisch gesendete Meldungen. Diese stufen-spezifischen BCs werden jedoch nicht von allen Reglern gesendet, die bei zweistufigen Ölbrennern verbaut wurden. Derzeit scheint es, als wenn die entsprechenden BCs nur beim Brötje BOB gesendet werden. Beim Brötje SOB bspw. scheint nur ein Status-BC gesendet zu werden: Brenner AN oder AUS. In dem Fall funktioniert die Unterscheidung der Brennerstufen leider nicht. Die Brennerstarts und -laufzeiten werden dann kumuliert bei Stufe 1 angezeigt (beinhalten also auch Stufe 2!).*

*Mittels Abfrage der Parameter 8330-8333 können die Starts und Betriebsstunden der beiden Stufen aber nach wie vor einzeln abgerufen werden.*

*Sollen die Brennerstarts und -laufzeiten von zweistufigen (Öl-)Brennern geloggt werden, beachte bitte den entsprechenden Hinweis in Kapitel 5!*

- Aktivieren/Deaktivieren des Loggens auf die microSD-Karte:

Prinzipiell erfolgt das Aktivieren/Deaktivieren der Log-Funktion durch das entsprechende Definement in der Datei *BSB\_lan\_config.h* vor dem Flashen. Während des Betriebes kann jedoch das Loggen deaktiviert werden, indem man folgende Parameter definiert:  
`http://<IP-Adresse>/L=0,0`

Zum Aktivieren werden dann wieder das Intervall und die gewünschten Parameter eingetragen. Bei einem Reset/Neustart des Arduino werden die Einstellungen aus der Datei *BSB\_lan\_config.h* verwendet – eine dauerhafte Umstellung der Logging-Parameter sollte also dort erfolgen.

- Konfiguration des Logfiles:

`http://<IP-Adresse>/L=<x>[,<parameter1>,<...>,<parameter20>]`

Setzt während der Laufzeit das Logging-Intervall auf `<x>` Sekunden und (optional) die Logging-Parameter auf [parameter1], [parameter2] etc.

Das Logging muss durch das Definement `#define LOGGING` in der Datei *BSB\_lan\_config.h* aktiviert werden und kann initial anhand der Variablen `log_parameters` und `log_interval` konfiguriert werden (s.o.).

- Konfiguration des Loggens von Bus-Telegrammen:

`http://<IP-Adresse>/LU=<x>`

Wenn Bus-Telegramme geloggt werden (Parameter 30000 als einzigen Parameter loggen), logge nur die unbekannten Command IDs (`<x>=1`) oder alle (`<x>=0`) Telegramme.

`http://<IP-Adresse>/LB=<x>`

Wenn Bus-Telegramme geloggt werden (Parameter 30000 als einzigen Parameter loggen), logge nur die Broadcasts (`<x>=1`) oder alle (`<x>=0`) Telegramme.

- Darstellung des Logfiles:

`http://<IP-Adresse>/D`

Zeigt den Inhalt der Datei *datalog.txt*, die sich auf der microSD-Karte im Slot des Ethernet-Shields befindet.

Mittels `/D0` kann die Datei *datalog.txt* zurückgesetzt werden, gleichzeitig wird eine korrekte CSV-Header-Datei generiert (dieser Schritt wird zudem für die erste Benutzung empfohlen, bevor das Loggen startet).

Wer Parameter auf SD-Karte loggt, hat neben der reinen Textform auch die Möglichkeit, unter

`http://<IP-Adresse>/DG`

einen Graphen angezeigt zu bekommen.

Dafür muss bei Javascript-Blockern die Domain [d3js.org](https://d3js.org) freigegeben werden, da der Arduino weiterhin nur die CSV-Datei in den Browser lädt und diese dann mit dem D3-Framework grafisch aufbereitet wird. Wird die Log-Datei via Webinterface mittels Klick auf „Anzeige Logdatei“ aufgerufen, erfolgt standardmäßig zuerst die grafische Darstellung.

- Bus-Typ (BSB oder LPB) vorübergehend ändern:

`http://<IP-Adresse>/P<x>`

Wechselt zwischen BSB ( $<\text{x}>=0$ ) und LPB ( $<\text{x}>=1$ ). Nach einem Reset/Neustart des Arduino wird die Einstellung aus der Datei *BSB\_lan\_config.h* verwendet. Um den Bus-Typ dauerhaft festzulegen, sollte die Option `setBusType config` in der Datei *BSB\_lan\_config.h* entsprechend angepasst werden.

- Resetten/Restarten des Arduino:

`http://<IP-Adresse>/X`

Reset/Restart des Arduino nach einem Pausieren für 8 Sekunden (siehe Kapitel 5:  
`#define RESET in BSB_lan_config.h`).

## **9. Etwaige Fehlermeldungen und deren mögliche Ursachen**

### **9.1 Fehlermeldung „unknown type <xxxxxxxx>“**

Dieser Fehler sagt aus, dass für diesen Parameter keine Umrechnungsanweisung vorliegt, um die Rohdaten in eine entsprechende Einheit (Zeit, Temperatur, Prozent, Druck etc.) umzuwandeln.

Um den Fehler zu beheben, sollte das jeweilige Telegramm / die Command ID des betreffenden Parameters sowie der zugehörige Wert ausgelesen und gemeldet werden. Sollten mehrere Einstellungsoptionen für einen Parameter verfügbar sein, muss zusätzlich jede Option ausgelesen werden, damit eine eindeutige Zuordnung stattfinden kann.

### **9.2 Fehlermeldung "error7 - parameter not supported"**

Die zugehörige Command ID wird nicht erkannt oder der entsprechende Parameter wird vom Regler nicht unterstützt (bspw. spezifische Parameter, die eine Gasheizung betreffen und bei einer Ölheizung dementsprechend nicht verfügbar sind).

### **9.3 Fehlermeldung "query failed"**

Diese Meldung erscheint, wenn auf Anfrage des Adapters keine (sinnvolle) Antwort des Reglers kommt.

Mögliche Ursachen sind u.a. ein Fehler in der RX- und/oder TX-Verbindung oder auch ein timeout aufgrund eines ausgeschalteten oder nicht angeschlossenen Reglers.

## **10. Etwaige Probleme und deren mögliche Ursachen**

### **10.1 Die rote LED des Adapters leuchtet nicht**

- Regler ist ausgeschaltet
- Adapter ist nicht mit dem Regler via BSB oder LPB verbunden
- Adapter ist falsch mit dem Regler verbunden (CL+/CL- bzw. DB/MB vertauscht)
- Evtl. Hardwarefehler des Adapters (bspw. defektes Bauteil, Fehler im Aufbau)

### **10.2 Zugriff auf das Webinterface nicht möglich**

- Adapter hat keine Stromversorgung
- Adapter ist nicht mit dem LAN verbunden
- IP- und/oder MAC-Adresse des Adapters ist nicht korrekt
- Sicherheitsfunktionen „Passkey“, „TRUSTED\_IP“ und/oder „USER\_PASS\_B64“ aktiviert/deaktiviert → URL nicht angepasst, Zugriff von falscher IP etc.
- → Testweise Reset-Knopf des Arduino bzw. LAN-Shields drücken
- LAN-Shield mit W5100-Chip bestückt? Sollte ein W5500 zum Einsatz kommen, sind die entsprechenden Punkte bei der Installation zu berücksichtigen (s. Kap. 5 & 11.12)

### **10.3 Keine Parameterabfrage möglich**

- Siehe Punkt „Die rote LED des Adapters leuchtet nicht“
- Siehe Punkt „Zugriff auf das Webinterface nicht möglich“
- Rx- und/oder Tx-Belegung nicht korrekt, Pinbelegung und/oder Adapteranschluss stimmt nicht mit der Angabe in der Datei *BSB\_lan\_config.h* überein
- Falscher Bus-Typ (BSB/LPB)

### **10.4 Regler wird nicht korrekt erkannt**

- Regler ist ausgeschaltet
- Regler wurde erst nach dem Aruino angeschaltet (automatische Reglererkennung funktioniert dann nicht)
- Regler ist nicht oder falsch mit dem Adapter verbunden
- Gerätefamilie (<http://<IP-Adresse>/6225>) des Reglers unbekannt

### **10.5 HK1 kann nicht bedient werden**

- Adapter ist als RGT2 konfiguriert

### **10.6 Es kann keine Raumtemperatur an einen HK1 gesendet werden**

- Entsprechender Parameter ist ‚read-only‘
- Adapter ist als RGT2 konfiguriert

### **10.7 HK2 kann nicht bedient werden**

- Adapter ist als RGT1 konfiguriert

## **10.8 Es kann keine Raumtemperatur an einen HK2 gesendet werden**

- Entsprechender Parameter ist „read-only“
- Adapter ist als RGT1 konfiguriert

## **10.9 Einstellungen des Reglers können nicht via Adapter verändert werden**

- Zugriff des Adapters ist auf Lesen beschränkt

## **10.10 Der Adapter reagiert manchmal nicht auf Abfragen**

- Der Arduino ist nicht multitaskingfähig - warte, bis eine Abfrage abgeschlossen ist (insbesondere umfangreichere Abfragen wie bspw. ganze Kategorien oder auch die Darstellung des Logfiles dauern u.U. recht lange)

## **10.11 Bei der Abfrage der Logdatei passiert „nichts“**

- Es ist keine microSD-Karte eingelegt
- Das Loggen auf microSD-Karte war oder ist deaktiviert
- Die Logdatei ist sehr groß, die (graphische) Darstellung dauert entsprechend länger

## **10.12 Es werden keine 24h-Durchschnittswerte angezeigt**

- Das entsprechende Definement ist nicht aktiviert
- Es sind keine zu berechnenden Parameter angegeben

## **10.13 Bei der Abfrage der Daten von DS18B20-/DHT22-Sensoren passiert „nichts“**

- Es sind keine Sensoren angeschlossen
- Die entsprechenden Definements sind nicht aktiviert
- Die Pinbelegung ist nicht korrekt eingestellt
- Die Sensoren sind fehlerhaft installiert oder defekt

## **10.14 Die DS18B20-Sensoren zeigen falsche Werte an**

- Die Stromversorgung und Installation prüfen (Größe des PullUp-Widerstands prüfen, Kondensatoren verbauen, Verkabelung prüfen, richtige Topologie verwenden etc.)

## **10.15 Der „Serielle Monitor“ der Arduino IDE liefert keine Daten**

- Der Adapter ist nicht zusätzlich via USB angeschlossen
- Falscher Anschluss (COM-Port) oder falsches Board in der Arduino IDE ausgewählt
- Falsche Baudrate eingestellt → auf 115200 Baud einstellen
- Adapter nicht am Regler angeschlossen, Regler ist ausgeschaltet → siehe o.g. Punkte

## 11. FAQ

### 11.1 Kann ich Adapter & Software mit einem Raspberry Pi nutzen?

Ja und nein.

Der Adapter kann mit einem Raspberry Pi verwendet werden, wenn andere Pinheader genutzt werden (weibliche statt männliche) und die Platine entsprechend mit den Komponenten für die Raspi-Nutzung bestückt ist (R11-13, Q11+12, SJ2+3).

Die Software kann nicht mit einem Raspi verwendet werden, sie ist ausschließlich auf dem hier vorgestellten Arduino-System lauffähig. Zur Nutzung des Adapters mit einem Raspi muss eine vollkommen andere Software genutzt werden. Weitere Informationen diesbezüglich sind am Ende des Kapitels 1 zu finden.

### 11.2 Kann ich einen Adapter an zwei Regler gleichzeitig anschließen?

Nein, das geht leider nicht.

Derzeit brauchst du für jeden Regler einen Adapter bzw. ein komplettes Hardware-Setup (Arduino, Ethernet-Shield, Adapter), um die jeweiligen reglerspezifischen Parameter via BSB oder LPB abrufen zu können.

### 11.3 Kann ich einen Adapter via LPB anschließen und mehrere Regler abfragen?

Nein, Abfragen fallweise an unterschiedliche Regler zu senden, ist bisher mit der hier vorgestellten Software leider noch nicht möglich.

### 11.4 Gibt es einen einfachen Weg, um Parameter zu loggen?

Ja, den gibt es - zum Einen als Standalone-Variante, zum Anderen als Remote-Variante.

#### 11.4.1 Verwendung des Adapters als Standalone-Logger

Stecke eine FAT32-formatierte microSD-Karte in den Speicherkartenplatz des Ethernet-Shields, bevor du den Arduino einschaltest. Einige Geräte erkennen u.U. keine Speicherkarten, die größer als 2GB sind; nutze in diesem Fall eine kleinere Karte und formatiere sie mit FAT16.

Ändere dann die Datei *BSB\_lan\_config.h* und aktiviere das Definement `#define LOGGER`. Füge dann die zu loggenden Parameter zur Variable `log_parameters` hinzu und bestimme das Logintervall mit der Variable `log_interval`. Später können während der Laufzeit sowohl das Intervall als auch die Logging-Parameter mittels des Befehls `"/L=[Intervall],[Parameter1],..., [Parameter20]"` geändert werden.

Wenn das Setup abgeschlossen ist, schalte das System ein und warte auf ankommende Daten. Sämtliche Daten werden auf der Karte in der Datei *datalog.txt* im CSV-Format gespeichert und können somit leicht in Excel oder OpenOffice Calc importiert werden. Den Dateiinhalt kannst du mit dem URL-Befehl `"/D"` einsehen.

Um die Datei zurückzusetzen, benutze den Befehl `"/D0"`.

Dies sollte ebenfalls bei der ersten Benutzung erfolgen, da hierdurch die Datei mit dem passenden CSV-Header initiiert wird.

Bitte beachte, dass der Arduino keine exakte Uhr ist. Auch wenn du bspw. das Intervall auf 60 Sekunden eingestellt hast, weicht die in der Datei dargestellte Zeit (welche von der Heizungssteuerung empfangen wird) möglicherweise davon ab - dies kann bis zu einer Sekunde pro Minute betragen.

Sollte eine exakte Logzeit unbedingt erforderlich sein, kannst du die durchschnittliche Zeitabweichung zwischen der Arduino-Zeit und der wirklichen Zeit ermitteln, das Log-Intervall entsprechend anpassen und bspw. 59 Sekunden anstatt 60 Sekunden einstellen.

### **11.4.2 Verwendung des Adapters als Remote-Logger**

Für das Loggen in der remote-Variante führe folgenden Befehl periodisch aus (z.B. per cron job):

```
DATE=`date +%Y%m%d%H%M%S` ; wget -qO-
http://192.168.178.88/1234/8310/720/710 | egrep "(8310|720|710)" | sed
"s/^/$DATE /" >> log.txt
```

Das aus diesem Beispiel resultierende Logfile 'log.txt' enthält die aufgezeichneten Werte der Parameter 8310, 720 und 710. Ändere sowohl die Parameternummern in der http-Abfrage als auch die des egrep-Befehls und selbstverständlich die IP-Adresse sowie ggf. den Passkey (hier: 1234).

Später kannst du das Logfile basierend auf den Parameternummern sortieren, nutze hierfür den Befehl 'sort':

```
sort -k2 log.txt
```

### **11.5 Wie kann ich die Daten bspw. in FHEM weiter verarbeiten?**

Bitte beachte, dass FHEM eine komplexe Software ist, und dementsprechend dies nicht der Ort ist, an dem grundsätzliche Informationen oder Anleitungen zu FHEM bereitgestellt werden. Solltest du jedoch bereits andere Geräte in FHEM konfiguriert haben, so hilft dir das Folgende hoffentlich weiter.

Um auf die Webschnittstelle des Adapters zuzugreifen, kann das Modul HTTPMOD in FHEM genutzt werden.

Die folgende Beispielkonfiguration kann an deine eigenen Bedürfnisse und die entsprechenden Parameter angepasst werden. Selbstverständlich müssen sowohl die IP als auch der Passkey (falls der optionale Sicherheitsschlüssel aktiviert wurde) entsprechend angepasst werden.

Der dargestellte Code auf der nächsten Seite fragt die Parameter 8700, 8743 und 8314 alle 300 Sekunden ab und weist diese dem Gerät "THISION" (der Name meines Heizungssystems) und den Readings "Aussentemperatur", "Vorlauftemperatur" und "Ruecklauftemperatur" zu.

Darüber hinaus stellt es ein Reading "Istwert" bereit, das per FHEM gesetzt werden kann, um dem Heizungssystem die aktuelle Zimmertemperatur mitzuteilen (Parameter 10000).

Zu guter Letzt berechnet es die Differenz zwischen "Vorlauftemperatur" und "Rücklauftemperatur" und weist diese Differenz dem Reading "Spreizung" zu.

*Bitte beachte:* Die Regex-Bedingungen müssen vom Beginn des Strings an (also der Parameternummer wie bspw. 8700) matchen und nicht erst ab einem späteren Teil des Strings.

```
define THISION HTTPMOD http://192.168.178.88/1234/8700/8743/8314 300
attr THISION userattr reading0Name reading0Regex reading1Name
reading1Regex reading2Name reading2Regex reading0Expr set0Name set0URL
attr THISION event-on-change-reading .*
attr THISION reading0Name Aussentemperatur
attr THISION reading0Regex 8700 .*:[ \t]+([-]?[\d\.\.]+)
attr THISION reading1Name Vorlauftemperatur
attr THISION reading1Regex 8743 .*:[ \t]+([-]?[\d\.\.]+)
attr THISION reading2Name Ruecklauftemperatur
attr THISION reading2Regex 8314 .*:[ \t]+([-]?[\d\.\.]+)
attr THISION reading0Expr $val=~s/[\\r\\n]//g;;$val
attr THISION set0Name Istwert
attr THISION set0URL http://192.168.178.88/1234/I10000=$val
attr THISION timeout 5
attr THISION userReadings Spreizung
{ sprintf("%.1f",ReadingsVal("THISION","Vorlauftemperatur",0)-
ReadingsVal("THISION","Ruecklauftemperatur",0));; }
```

## 11.6 Ist ein multifunktionaler Eingang am Regler direkt via Adapter schaltbar?

Nein!

Die multifunktionalen Eingänge an den Reglern (bspw. H1, H2) sind nicht direkt an den Adapter anzuschließen!

Soll bspw. eine Betriebsartumschaltung oder Erzeugersperre mittels H1 als Arbeitskontakt realisiert werden, so muss der jeweilige Eingang den Herstellerangaben entsprechend parametriert und belegt werden. Eine Steuerung dieser Art muss mittels eines anzuschließenden Relais erfolgen, dessen reglerseitiger Ausgang unbedingt potentialfrei sein muss, d.h. es darf keinerlei Fremdspannung anliegen. Das Relais hat in dem Fall lediglich die Aufgabe, den Kontakt zu schließen (oder zu öffnen). Das Relais wiederum kann jedoch unter bestimmten Umständen vom Arduino gesteuert werden (bspw. mittels eines Relaisboards).

Entsprechende Relais findest du im Internet, bei Unsicherheiten solltest du deinen Elektriker und/oder Heizungsinstallateur zu Rate ziehen. Eine falsche Belegung und/oder Parametrierung kann den Regler u.U. zerstören!

## 11.7 Relaisboard am Arduino Mega2560

Wenn du zusätzlich ein Relaisboard am Arduino angeschlossen hast und die einzelnen Relais ansteuern bzw. deren Zustand abfragen möchtest, kannst du das prinzipiell mit der Funktion /G über das Webinterface realisieren (siehe entspr. Abschnitt im Kapitel 8).

Das Folgende ist ein Beispiel für eine FHEM-Konfiguration, bei dem die drei Relais-Ports namens "Heater", "Fan" und "Bell" abgefragt und gesteuert werden, die an die entsprechenden GPIO-Pins 7, 6 und 5 angeschlossen sind. Auch hier müssen sowohl die IP-Adresse als auch (optional) der Passkey (falls aktiviert) deiner Konfiguration entsprechend angepasst werden.

```
define EthRelais HTTPMOD http://192.168.178.88/1234/G05/G06/G07 30
attr EthRelais userattr reading0Name reading0Regex reading1Name
reading1Regex reading2Name reading2Regex reading0Expr reading0Map
set0Name set0URL set1Name set1URL set2Name set2URL setIMap
setParseResponse:0,1 setRegex
attr EthRelais event-on-change-reading .*
attr EthRelais reading0Name Heater
attr EthRelais reading0Regex GPIO7:[ \t](\d)
attr EthRelais reading1Name Fan
attr EthRelais reading1Regex GPIO6:[ \t](\d)
attr EthRelais reading2Name Bell
attr EthRelais reading2Regex GPIO5:[ \t](\d)
attr EthRelais room Heizung
attr EthRelais set0Name Heater
attr EthRelais set0URL http://192.168.178.88/1234/G07=$val
attr EthRelais set1Name Fan
attr EthRelais set1URL http://192.168.178.88/1234/G06=$val
attr EthRelais set2Name Bell
attr EthRelais set2URL http://192.168.178.88/1234/G05=$val
attr EthRelais setParseResponse 1
attr EthRelais setRegex GPIO[0-9]+:[ \t](\d)
attr EthRelais timeout 5
```

## 11.8 Kann ich behilflich sein, um bisher nicht unterstützte Parameter hinzuzufügen?

Ja! Wenn dein Heizungssystem über Parameter verfügt, die von der Software bisher nicht unterstützt werden, würden wir uns sehr freuen, wenn du uns unterstützt!

Dazu brauchst du nur deinen mit der Heizungssteuerung verbundenen Arduino mit einem Laptop/PC über USB zu verbinden und den folgenden Schritten zu folgen (siehe auch Kapitel 6):

1. Starte die Arduino IDE und öffne den seriellen Monitor (Menü Werkzeuge -> Serieller Monitor bzw. drücke Strg+Umschalt+M).
2. Aktiviere das Loggen zur seriellen Konsole und die Verbositäts-Ausgabe mit dem URL-Parameter /V1 auf dem Arduino.  
Alternativ dazu kannst du Bus-Telegramme auf die microSD-Karte loggen: Logge dazu (als einzigen!) Parameter 30000 (s. Logging-Abschnitt oben), setze die Variable

`log_unknown_only` auf 1 (URL-Befehl /LU=1) und beobachte die Logeinträge mit dem URL-Befehl /D.

3. Schalte beim Heizungssystem über die integrierte Steuerung zu dem Parameter, den du analysieren möchtest (mittels des Drehrades, der Pfeiltasten oder der spezifischen Eingabemöglichkeiten deiner Heizungssteuerung).
4. Warte auf 'Ruhe' auf dem Bus, dann schalte einen Parameter weiter vor und gleich wieder zurück zu dem Parameter, den du analysieren möchtest.

Nun solltest du etwas im Log bzw. in der Ausgabe des 'SeriellenMonitors' der Arduino IDE sehen, was in etwa so aussehen sollte:

```
DISP->HEIZ QUR    113D305F  
DC 8A 00 0B 06 3D 11 30 5F AB EC  
HEIZ->DISP ANS    113D305F 00 00  
DC 80 0A 0D 07 11 3D 30 5F 00 00 03 A1  
DISP->HEIZ QUR    113D3063  
DC 8A 00 0B 06 3D 11 30 63 5C 33  
HEIZ->DISP ANS    113D3063 00 00 16  
DC 80 0A 0E 07 11 3D 30 63 00 00 16 AD 0B
```

Die ersten vier Zeilen sind von dem Parameter, zu dem hingeschaltet wurde.

Die letzten vier Zeilen stammen von dem Parameter, den du analysieren möchtest (das Hin- und Herschalten soll nur sicherstellen, dass die letzte Nachricht auf dem Bus wirklich der Parameter ist, den du suchst).

Anstelle von DISP wird eventuell RGT1 angezeigt, dies ist abhängig vom jeweiligen Gerät, mit dem du die Eingaben am Heizungssystem tätigst (integrierte Steuerung oder angeschlossenes Raumgerät/Fernbedienung).

Jedes Datentelegramm hat die folgende Struktur:

Byte 1:	Immer 0xDC (Startbyte, Beginn des Telegramms)
Byte 2:	Quellgerät plus 0x80 (Quellgeräte: 0x00 = Heizungssystem, 0x06 = Raumgerät 1, 0x07 = Raumgerät 2, 0x0A = Display, 0x7F = Broadcast)
Byte 3:	Zielgerät (gleiche Adresswerte wie bei Quellgerät)
Byte 4:	Telegrammlänge (Startbyte des Telegramms wird nicht mitgezählt)
Byte 5:	Nachrichtentyp (0x02 = Info, 0x03 = Setzen, 0x04 = ACK, 0x05 = NACK, 0x06 = Abfragen, 0x07 = Antworten, 0x08 = Fehler)
Byte 6-9:	<i>Command ID</i> (→ diese ist es, die wir brauchen!)
Byte 10...:	Payload data (optional)
Letzte zwei Bytes:	CRC-C checksumme

5. Das untere Datentelegramm im obigen Beispiel hat die Command ID 0x113D3063. Bitte beachte, dass die ersten beiden Bytes der Command ID beim Nachrichtentyp "Abfragen" (0x06) vertauscht sind! Stelle daher bitte sicher, dass du stets auf das richtige Telegramm achtest (Typ "Antwort" (0x07), die letzte Zeile des o.g. Beispiels).
6. Suche den Bereich "*global command table*" in der Datei *BSB\_lan\_defs.h* und überprüfe, ob für diesen Befehl bereits ein Eintrag existiert (suche nach STRxxxx , wobei xxxx die Parameternummer darstellt). Falls es ihn bereits gibt, fahre fort mit Schritt 8.
7. Sollte der Parameter noch nicht in dem "*global command table*" gelistet sein, musst du einen Eintrag im Bereich "*menu strings*" wie folgt erstellen:  

```
const char STRxxxx[] PROGMEM = "Parameter_Name_or_Function";
```

Nun kopiere eine Zeile des Bereichs vom "*global command table*", wo der neue Parameter numerisch passt. Fahre fort mit Schritt 8, aber anstatt CMD\_UNKNOWN zu ersetzen, müssen logischerweise die Command ID und der Wertetyp der kopierten Zeile ersetzt werden.
8. Ersetze CMD\_UNKNOWN durch die Command ID, die du gerade herausgefunden hast. Falls der zurückgegebene Wertetyp (column 3) VT\_UNKNOWN ist, versuche herauszufinden, welcher Parametertyp von der Liste am Anfang der Datei passt. Beispiel: Wenn der Parameter einen Temperaturwert zurückgeben soll, kannst du VT\_TEMP , VT\_TEMP\_SHORT , VT\_TEMP\_SHORT5 oder VT\_TEMP\_WORD ausprobieren. Für Parameter, die mehrere Optionen bereitstellen, musst du eine entsprechende Zeile im Abschnitt "*ENUM tables*" hinzufügen.
9. Wenn der Adapter den gleichen Wert ausgibt wie auf dem Display der integrierten Heizungssteuerung dargestellt, hast du den richtigen Wert gefunden und der Parameter ist nun voll funktionsfähig (bspw. sollte das Abfragen und Setzen von Werten nun funktionieren). Herzlichen Glückwunsch!
10. Wenn du fertig bist, überprüfe nochmals, ob die neue Command ID nicht bereits irgendwo anders in *BSB\_lan\_defs.h* verwendet wird (wenn du bspw. nach der Command ID suchst, sollte sie nur einmal gefunden werden).  

Da das BSB-Protokoll nicht standardisiert ist und die unterschiedlichen Hersteller (zumindest bei spezifischeren Parametern) keine Rücksicht darauf zu nehmen scheinen, wie andere Hersteller die Command IDs verwenden, ist es nämlich möglich, dass Command IDs für verschiedene Parameter bei verschiedenen Heizungssystemen existieren. Sollte es passieren, dass eine Command ID nun doppelt in der Datei *BSB\_lan\_defs.h* existiert, markiere diese bitte deutlich bevor du uns das Update schickst und teile uns bitte mit, welches Heizungssystem genau du verwendest. Wir werden dann bedingungsabhängige Compiler-Flags hinzufügen, so dass das Heizungssystem X anders kompiliert wird als Heizungssystem Y und somit letztlich beide Systeme die mehrdeutigen Command IDs für die korrekten Parameter nutzen können.

11. Die neuen bzw. aktualisierten Zeilen poste bitte im u.g. Thread des FHEM-Forums.

Solltest du eine Diff-Datei nutzen, überprüfe bitte vor dem Erstellen nochmals, ob du wirklich die aktuelle *BSB\_lan\_defs.h* des GitHub-Repositoriums heruntergeladen hast, da manchmal die Dateien aktualisiert werden, ohne dass direkt eine neue Version veröffentlicht wird.

*Hinweis:*

Sollte dir die hier beschriebene Vorgehensweise zu kompliziert erscheinen und du dich der Umsetzung nicht gewachsen fühlen, kannst du evtl. auch die jeweiligen Telegramme mit eindeutiger Zuordnung und Beschreibung des betreffenden Parameters auflisten und im FHEM-Forum<sup>11</sup> posten.

Wichtig ist dabei jedoch, dass du

- a) jeden Parameter eindeutig beschreibst (also Prog.Nr. und Bezeichnung),
- b) der entsprechenden Command ID eindeutig zuordnest,
- c) den jeweiligen Wert bzw. die jeweilige Einstellung des gerade ausgewählten Parameters ebenfalls notierst und
- d) idealerweise alle Optionen eines Parameters (wenn es sich nicht um Temperaturwerte o.ä. handelt) durchgehst und die jeweiligen Command IDs bzw. Telegramme wiederum mit der eindeutigen Bezeichnung der Option notierst.

### **11.9 Warum erscheinen bei einer Komplettabfrage einige Parameter doppelt?**

Wenn du eine Komplettabfrage aller Parameter via URL-Befehl machst (<IP-Adresse>/0-10000) kann es sein, dass sich einige Parameter bzw.

Programmnummern in der Auflistung wiederholen. Dies kommt daher, dass es zwar unterschiedliche Parameter sind, diese aber die gleiche Command ID haben. Dies stellt nur einen ‚optischen Mangel‘ dar, der die Funktionalität nicht negativ beeinflusst.

### **11.10 Warum werden manchmal bestimmte Parameter nicht angezeigt?**

Wenn der Regler nach erfolgtem Adapteranschluss angeschaltet wird und der Arduino zu diesem Zeitpunkt bereits lief, funktioniert die automatische Reglererkennung nicht.

Bestimmte Parameter, die mehrfach hinterlegt sind und bei denen es somit keinen reinen "DEV\_ALL"-Eintrag gibt, erscheinen in dem Fall bei einer Abfrage nicht.

Der Arduino muss dann lediglich resettet bzw. aus- und wieder angeschaltet werden.

### **11.11 Kann ich mehrere DHT22-Sensoren an einen DATA-Pin anschließen?**

Nein, jeder DATA-Anschluß braucht einen eigenen Pin, dessen Belegung in der Datei *BSB\_lan\_config.h* zu definieren ist. DHT22-Sensoren sind keine OneWire-Sensoren, die auf einem Bus-System aufgebaut sind. Bitte informiere dich über die Verwendung von DHT22-Sensoren.

---

11 <https://forum.fhem.de/index.php/topic,29762.0.html>

## **11.12 Warum ist kein Zugriff auf angeschlossene Sensoren möglich?**

Wenn du DHT22- und/oder DS18B20-Sensoren angeschlossen hast, die entsprechenden Menüs im Webinterface jedoch nicht anwählbar sind, hast du vermutlich die Datei *BSB\_lan\_config.h* nicht entsprechend angepasst. Ansonsten siehe Kap. 5, 10 & 12.

## **11.13 Ich nutze ein W5500-LAN-Shield, was muss ich tun?**

Prinzipiell ist die Nutzung eines Shields mit dem Chip-Typ W5100 zu empfehlen, da die entsprechenden Bibliotheken bereits eingebunden sind.

Sollte dennoch ein W5500 zum Einsatz kommen, ist

1. die Datei *BSB\_lan\_config.h* entsprechend anzupassen (s. Kap. 5) und
2. die Datei *Ethernet2.zip* im Unterverzeichnis *src* entpacken.

## **11.14 Können Stati oder Werte als Push-Mitteilungen abgesetzt werden?**

Nein, nicht ohne weitere Software wie z.B. FHEM. Dafür müsste ansonsten die Therme ständig abgefragt werden, was den Bus (und die Erreichbarkeit des Arduino) stark belasten würde. Die sinnvollere Variante wäre, bestimmte Werte z.B. alle 60 Sekunden abzurufen und dann anhand bestimmter Kriterien weitere Aktionen auszulösen. Bei FHEM wäre das mit DOIF oder NOTIFY möglich.

## **11.15 Kann bspw. FHEM auf bestimmte Broadcasts „lauschen“?**

FHEM kann zwar lauschen, aber BSB\_LAN kann bisher keine eigenständigen Nachrichten absetzen. Dazu müsste ein Hintergrundprozess die auflaufenden Broadcast-Meldungen anhand konfigurierbarer Schwellenwerte auswerten und über einen HTTP-Client-Aufruf an eine definierbare Zieladresse absetzen. Ob dies mit dem begrenzten Speicherplatz des Arduino noch umsetzbar ist, wäre fraglich. Wer sich aber daran probieren möchte, ist herzlich eingeladen, dies zu tun!

## **11.16 Warum werden unter /B bei Stufe 2 keine Werte angezeigt?**

Wenn du einen Gasbrenner hast, so wird dieser höchstwahrscheinlich modulieren und generell nicht über ein zweistufiges Brennersystem verfügen; zweistufige Brenner kommen meist nur bei Ölfernern zum Einsatz. Die Unterscheidung der Brennerstufen wird mittels spezifischer Broadcasts vorgenommen, die jedoch nicht jeder Regler sendet. In dem Fall werden die Brennerstarts und -laufzeiten kumuliert unter Stufe 1 dargestellt. Bitte beachte diesbezüglich auch den Hinweis unter „/B“ in Kapitel 8.

## **11.17 Ich habe den Eindruck, die angezeigten Werte bei /B sind nicht korrekt.**

Das kann durchaus sein. Die jeweiligen Starts und Laufzeiten werden anhand von Broadcasts ermittelt, die automatisch vom Regler gesendet werden. Manchmal kann es vorkommen, dass einzelne BCs nicht ankommen, bspw. wenn zeitgleich eine Abfrage gestartet wird oder der Arduino das Logfile lädt.

## **11.18 Kann ich die Funktion der Präsenztaste mit BSB-LAN simulieren?**

Prinzipiell ja: Die Funktion der Präsenztaste ist mit dem Spezialparameter 701 implementiert und als SET-Befehl auszuführen.

Bei aktivem Automatikprogramm ist dabei /S701=1 für den Wechsel auf ‚BA Reduziert‘ und /S701=0 für den Wechsel auf ‚BA Komfort‘ zu setzen.

Der jeweilige Wechsel ist bis zur nächsten BA-Umschaltung laut Zeitprogramm gültig.

Es ist jedoch bisher noch nicht umfangreich getestet, ob diese Funktion bei jedem Regler Wirkung zeigt oder nicht. Auch ist noch nicht geklärt, ob für die Verwendung ein ‚echtes‘ Raumgerät vorhanden sein muss, bei dem man zunächst die Wirkung der Präsenztaste definieren muss (mögliche Optionen, jedoch nur via echtem RGT festzulegen: Keine, HK1, HK2, Gemeinsam).

Erfolgreich getestet wurde es bisher lediglich an einem RVS43.222 (Brötje SOB26C).

## **11.19 Kann ich eine Raumtemperatur senden, um ‚Raumeinfluss‘ zu verwenden?**

Ja, mittels einer INF-Nachricht. Für die Raumtemperatur HK1<sup>12</sup> ist dies der Spezialparameter 10000, für den HK2<sup>13</sup> ist es 10001.

Siehe auch den entsprechenden Punkt in Kapitel 8.

## **11.20 Was ist der genaue Unterschied zwischen /M1 und /V1?**

Mit dem URL-Befehl /M1 aktivierst du den Monitor-Modus, mit /V1 den Verbositäts-Modus.

Mit aktiverter Monitor-Funktion (/M1) werden alle Daten, die über den Bus gehen und nicht von BSB-LAN aus initiiert wurden, „roh“ auf dem seriellen Monitor ausgegeben. Dies kann sinnvoll sein, um Fehlfunktionen in der Datenübertragung ausfindig zu machen, da ansonsten nur Meldungen von BSB\_Lan verarbeitet werden, die von ihrem Aufbau her korrekt sind. Das schließt auch die Verarbeitung von Broadcast-Nachrichten ein, d.h. mit aktiverter Monitor-Funktion findet keine Auswertung dieser Nachrichten statt.

Die Monitor-Funktion erlaubt es z.B. bei Fehlermeldungen genauer zu sehen, ob eine Nachricht schlachtweg nicht auf dem Bus angekommen ist oder ob BSB-LAN sie wegen fehlerhafter Übertragung verworfen hat. Die volle Kontrolle hätte man mit einem zweiten BSB-LAN Adapter, der auf dem Bus lauscht und dann alle hin- und hergehenden Nachrichten protokolliert.

Mit aktiviertem Verbositäts-Modus (/V1) werden zu jedem von BSB-LAN initiierten Aufruf und der entsprechenden Antwort neben dem Klartext auch die entsprechenden Rohdaten auf dem seriellen Monitor ausgegeben, wenn die Nachricht von ihrem Aufbau her korrekt sind und fehlerfrei übertragen wurden.

Eine Auswertung von (fehlerfreien) Broadcasts findet hier weiterhin statt. Es werden hier beim Senden aber nur die Daten ausgegeben, die BSB-LAN vorbereitet hat. Dies muss nicht bedeuten, dass diese Daten - z.B. bei Hardwarefehlern - auch auf dem Bus

---

<sup>12</sup> Adapter muss als RGT1 konfiguriert sein, um HK1 zu bedienen.

<sup>13</sup> Adapter muss als RGT2 konfiguriert sein, um HK2 zu bedienen.

ankommen. Umgekehrt werden beim Auswerten der Rückmeldung auf einen Befehl zwar die Daten ausgegeben, die auf dem Bus zurück gekommen sind, aber nur dann, wenn die Nachricht auch korrekt aufgebaut war.

Eine Kombination aus beiden Parametern ist möglich und führt dazu, dass im Monitor-Modus auch bei von BSB-LAN initiierten Nachrichten die Rohdaten ausgegeben werden - mit den bereits erwähnten Einschränkungen des Verboseitäts-Modus bezüglich des Verwerfens von nicht korrekt aufgebauten Nachrichten.

### **11.21 Ich habe weitere Fragen, an wen kann ich mich wenden?**

Das Beste wäre, wenn du dich dafür im FHEM-Forum (<https://forum.fhem.de/>) anmelden würdest, da dort speziell für diesen Adapter ein eigener Thread existiert und sich dort eine nette und hilfsbereite Community findet. Hier findet ein reger Austausch über die Hard- und Software statt, Fragen werden meist zügig beantwortet und auf Updates wird hingewiesen.

Hier findest du den entsprechenden Thread:

<https://forum.fhem.de/index.php/topic,29762.0.html>

Wenn du dich mit deinen Fragen vorstellst, gib uns bitte zuerst genaue Informationen bzgl. des von dir verwendeten Heizungstyps, der Regler, verwendeter Bus-Typ etc.

Wenn du den Adapter bereits erfolgreich angeschlossen und in Verwendung hast, frage bitte außerdem die Parameter 6223-6228 (<IP-Adresse>/6223-6228) ab und schreibe die Ausgaben zusätzlich mit in deine Beschreibung.

Prinzipiell kann man sagen: Lieber erst einmal zu viele Informationen, als zu wenige.

Fragen, deren Antworten sich aus dem Lesen dieses Handbuchs ergeben, werden lediglich mit einem Verweis hierauf beantwortet. Bitte bedenke, dass dies für jeden von uns nur ein Hobby-Projekt ist und wir keinerlei Geld damit verdienen – unsere Zeit ist also schon alleine deswegen sehr begrenzt..

## 12. Optionale Sensoren

Es besteht die Möglichkeit, zusätzliche Sensoren des Typs DS18B20 (OneWire-Temperatursensor) und DHT22 (Temperatur- und Feuchtigkeitssensor) direkt an bestimmte Pins des Adapters bzw. Arduino anzuschließen. Die entsprechenden Bibliotheken für die Arduino IDE sind bereits im Softwarepaket des Adapters integriert.

Zur Nutzung dieser Sensoren muss lediglich die Konfiguration in der Datei *BSB\_lan\_config.h* entsprechend angepasst werden.

Auf die Daten der Sensoren kann nach erfolgter Installation über das Webinterface oder mittels der URL-Befehle /T (für DS18B20) und /H (für DHT22) zugegriffen werden. Sollen die gemessenen Werte geloggt werden, oder sind 24h-Mittelwertsberechnungen gewünscht, so kann dies mit den jeweiligen Konfigurationen ganz einfach realisiert werden.

Genauere Informationen über die jeweiligen Konfigurationen findest du bei den entsprechenden Einträgen in Kapitel 5.

Der Anschluss der Sensoren findet üblicherweise an GND und +5V des Adapters statt, der Anschlusspin für DATA ist wie in Kapitel 5 beschrieben zu definieren.

### Tipps:

Werden DS18B20- und/oder DHT22-Sensoren verwendet, werden diese unter <http://<IP-Adresse>/ipwe.cgi> standardmäßig mit angezeigt.

Dabei wird neben den gemessenen Werten auch die jeweils spezifische Hardwarekennung der Sensoren aufgeführt. Dies ist besonders bei einer Ersteinrichtung für eine eindeutige Unterscheidung der einzelnen Sensoren hilfreich.

Voraussetzung ist, dass das ipwe-Definiment in der Datei *BSB\_lan\_config.h* aktiviert ist (siehe Kapitel 5).

Kommen mehrere DS18B20-Sensoren und/oder größere Leitungslängen zum Einsatz, hat es sich bewährt, je einen 100nF-Keramikkondensator (und ggf. noch einen 10µF-Tantalkondensator zusätzlich) möglichst nah am Sensor in die Leitung zwischen GND und VCC (+5V) zu positionieren, um einen Spannungsabfall bei der Abfrage zu kompensieren. Von der Verwendung des sog. „parasitären Modus“ ist eher abzuraten. Der Wert des PullUp-Widerstandes am Adapterausgang zwischen DATA und VCC (+5V) ist für einen problemlosen Betrieb u.U. kleiner als die üblicherweise empfohlenen 4,7kΩ zu wählen. Die Verwendung einer geschirmten Steuerleitung ist zu empfehlen.

Kommen mehrere DHT22-Sensoren zum Einsatz, so muss für jeden DATA-Anschluss ein eigener Pin am Arduino definiert werden.

Auf die jeweiligen technischen Daten, weiteren installationsspezifischen Voraussetzungen, verschiedenen Ausführungen und Einsatzmöglichkeiten soll aufgrund des Umfangs hier nicht weiter eingegangen werden. Bitte informiere dich eigenständig, bspw. im Internet.

## 13. Offene Punkte

- Mehr Befehle (Command IDs) hinzufügen.

Nur die bekannten Befehle aus der genannten FHEM-Forendiskussion und den getesteten Heizungssystemen sind Bestandteil des Programms. Jeder Nutzer eines anderen Heizungssystems kann fehlende Parameter dekodieren und zur Weiterentwicklung der Software beitragen (siehe Kapitel 11).

Unser Ziel ist es, ein generell lauffähiges System zu entwickeln, das herstellerübergreifend mit allen Heizungssystemen verwendet werden kann, die einen BSB/LPB aufweisen. Jede Hilfe und jede Rückmeldung ist willkommen!

- Testen und Vervollständigen der Funktionalität.

Mit der gegenwärtigen Implementierung können bereits viele Werte gesetzt werden.

Jedoch sind noch immer Tests nötig und einige Parameter müssen hinzugefügt werden.

- Dekodieren der DE-Telegramme.

Möglicherweise beinhalten sie Statusinformationen, die ohne Abfragen genutzt werden können.

- Unterstützung für heizungsseitige Fehlermeldungen hinzufügen.

## 14. Weiterführende Informationen und Quellen

Ein reger Austausch bzgl. der hier vorgestellten Hard- und Software findet in folgendem Forum statt: <https://forum.fhem.de/index.php/topic,29762.0.html>

Dies ist auch eine gute Anlaufstelle für Fragen, Erfahrungsaustausch und Support, wo auch regelmäßig über Neuerungen und erfolgte Updates informiert wird.

Sämtliche Dokumentationen zur hier vorgestellten Hard- und Software sowie die verschiedenen Software-Versionen sind unter [https://github.com/fredlcore/bsb\\_lan](https://github.com/fredlcore/bsb_lan) zu finden.

Dieses Handbuch ist zudem unter <https://github.com/1coderookie/BSB-LPB-LAN> erhältlich.

Die Software und die dazugehörigen Dokumentationen für den Einsatz des hier vorgestellten Adapters in Verbindung mit einem Raspberry Pi 2 ist unter <https://github.com/loehnertj/bsbgateway> zu finden.

Die initiale Idee der Regleranbindung via BSB/LPB kann hier nachvollzogen werden:  
<http://www.mikrocontroller.net/topic/218643>

<http://blog.dest-unreach.be/2012/12/14/reverse-engineering-the-elco-heating-protocol>

Als relativ umfangreiche Quelle mit vielen Parameterbeschreibungen sei das „Brötje Systemhandbuch ISR Plus“ empfohlen. Es stellt neben zahlreichen anderen und modellspezifischen Anleitungen die zugrunde liegende ‚Referenz‘ für die Parameterdefinitionen des hier vorgestellten Projekts dar.

Tiefergehende Informationen wie Spezifikationen und technische Anforderungen der Bus-Typen sind den jeweiligen Dokumenten der Hersteller zu entnehmen.

Speziell hinsichtlich des LPB seien zwei Dokumente von „Siemens Building Technologies - Landis & Staefa Division“ empfohlen:

- CE1N2030D Local Process Bus LPB Systemgrundlagen
- CE1N2032D Local Process Bus LPB Projektierungsgrundlagen

Hinsichtlich der Installation und Verwendung von DHT22- und OneWire-Sensoren wie dem DS18B20 gibt es zahlreiche Informationsquellen. Im Internet finden sich etliche kostenlose Anleitungen, Beispielinstallationen und Skripte.

Bei Problemen mit einer umfangreicheren OneWire-Installation sei ein Blick auf die technischen Anforderungen von OneWire (speziell hinsichtlich der Bus-Typologie und der Leitungslängen) empfohlen.

## Anhang

### Schaltplan BSB-LPB-LAN-Adapter v2

