

DBT03-Emulator

Testbericht



DBT03-Emulator

Inhalt:

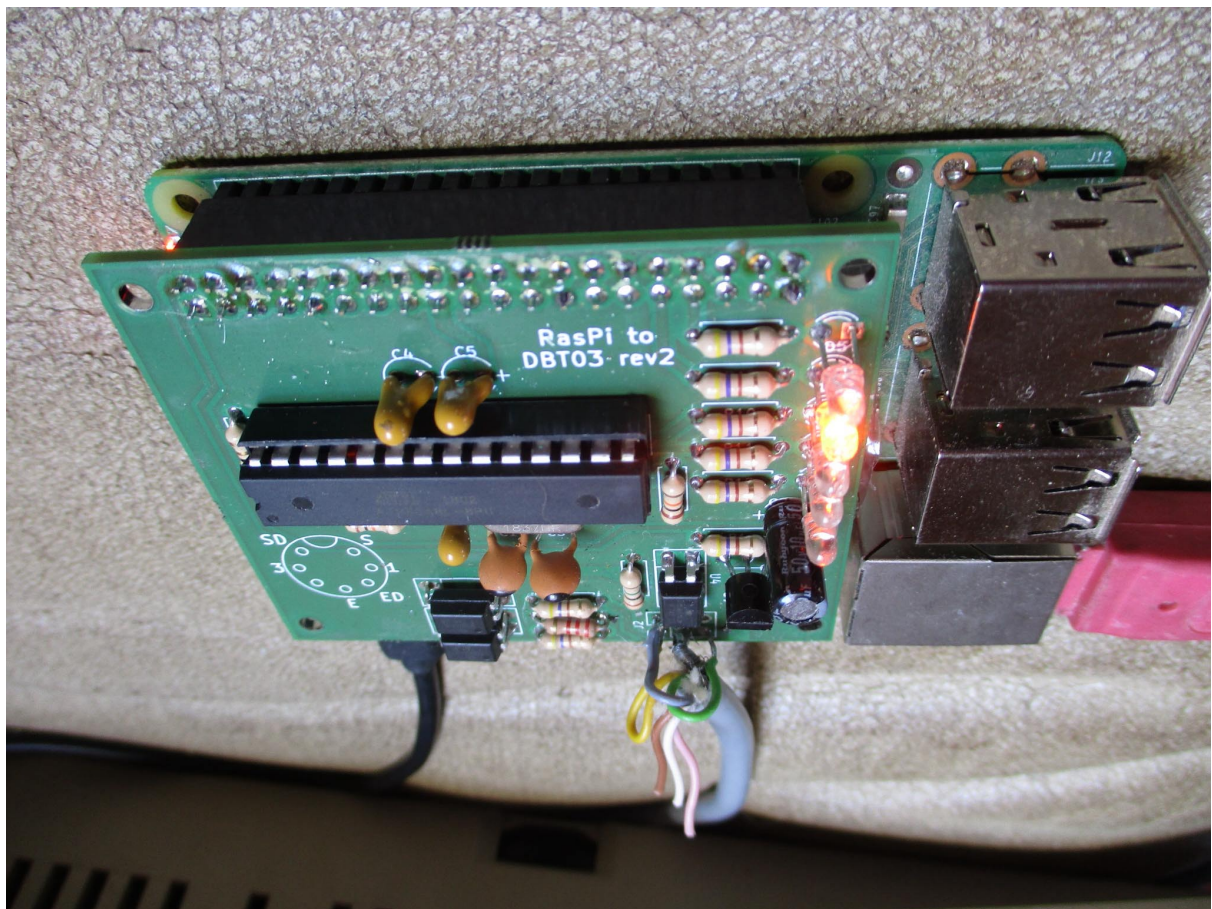
Einleitung	Seite 3
Software	Seite 4
Hardware	Seite 6
BTX-Seiten	Seite 8
Erweiterung	Seite 9
PiZero	Seite 11
Abschluss	Seite 14

DBT03-Emulator

Einleitung:

Dieser Emulator wurde mir von Thomas (Bielefeld) zugeschickt, um die Funktion an einem BTX-Terminal zu testen, das auf ein externes DBT03 angewiesen ist. Thomas hat die Platinen in Auftrag gegeben und nach Angaben des Entwicklers bestückt, den Prozessor geflashed und die Installation auf dem Raspberry durchgeführt. Mehr als die blinkende LED konnte er dann aber nicht testen.

Da ich noch zwei alte FeTap 90-1 (Bitel T3210) herumliegen hatte, habe ich mich als Tester für die gesamte Installation angeboten.



Der Emulator aufgesteckt auf dem Raspi.

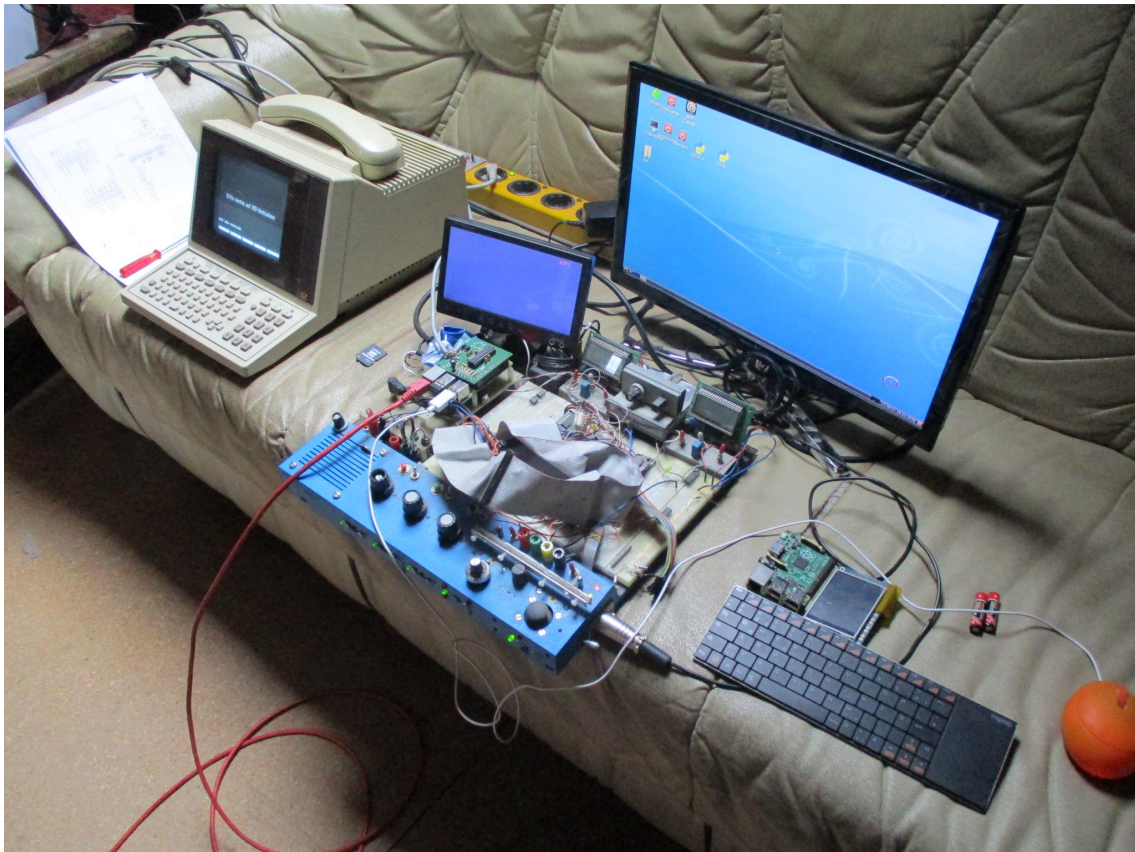
Diese Schaltung simuliert ein DBT03 mit den Leitungen SD (Senden), ED (empfangen) und S0 (steuern) Richtung Empfangsgerät. Zur Leitungsseite wird in diesem Fall eine TCP-Verbindung zum Server via dem Raspi hergestellt. Diese Idee ist ähnlich dem i-Telex System.

Im folgend wird die komplette Inbetriebnahme mit all den Herausforderungen beschrieben, die aufgetreten sind.

DBT03-Emulator

Software:

Raspberry Pi Version 2 mit 4 Core CPU lag gerade passend zur Hand, Betriebssystem Raspbian aus i-Telex Entwicklung auf SD-Karte gespeichert und aufgebaut.



Das sah dann so aus. Das Entwicklungssystem habe ich mal gebaut, um jederzeit überall kleine Hardwareentwicklungen machen zu können. Es sind alle möglichen Geber vorhanden, einfache LCD Strom/ Spannungsmesser und ein Bildschirm. Auch der grössere TV Empfänger gehört dazu, nutzt den HDMI-Eingang für den Erstkontakt.

Bei der Installation des Git-Tools ging alles wie geplant. Zur Installation des AVR-Dude fing es aber schon an zu hakeln. Den braucht man aber nur wenn der Prozessor neu geflashed werden soll. Es konnten leider nicht alle Dateien geladen werden.

Die weitere Installation lief fehlerfrei durch. Nur leider fing die eine LED nicht zu blinken an, dies signalisiert die Bereitschaft, eine Verbindung aufzubauen.

DBT03-Emulator

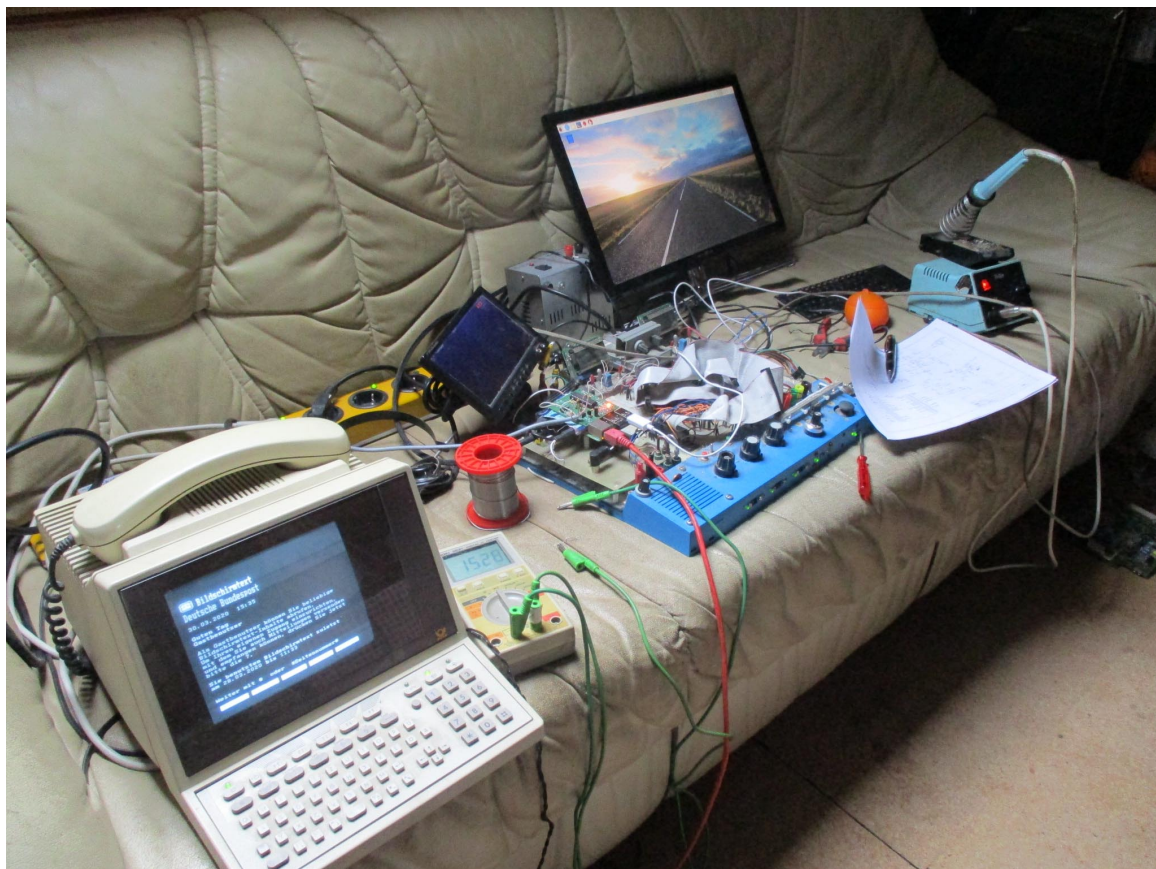
Software:

Da ich keine Vergleichsmöglichkeit hatte, also Telefonat mit Thomas.

Das Fazit aus mehreren Stunden testen:

Das Raspbian war schlicht etwas zu alt, um neue Systembefehle ausführen zu können. Wer wissen möchte ob sein OS hier mitspielt, einfach in der Konsole den Befehl „systemctl“ ausführen, wird der Befehl nicht erkannt, ist es zu alt (vor 2015). Ab OS „Jessie“ sollte es passen.

Ich habe dann von einer neuen CD das OS „Stretch“ genommen, gibt es auch als lite-Version. Damit läuft die Installation wie von Thomas beschrieben reibungslos ab. Nach Start der Datei ./do_everything.sh wird das Paket komplett (mit AVR-Dude) im Raspi integriert. Nach Neustart steht der Dienst im Hintergrund bereit und signalisiert das mit der blinkenden LED.

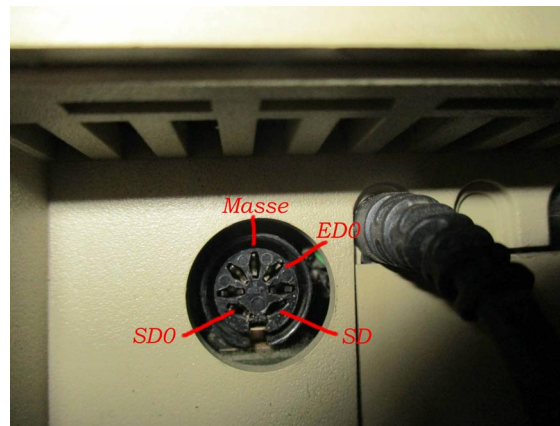
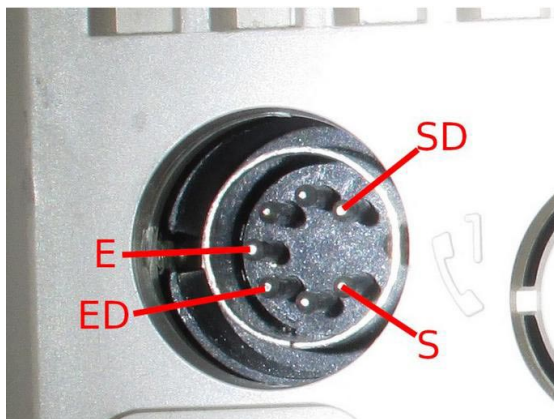


Hier ist schon „Stretch“ mit Desktop zu sehen.
Jetzt auf zur Hardware.

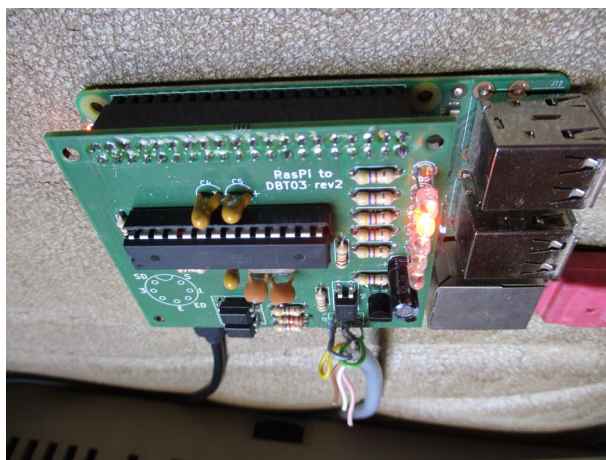
DBT03-Emulator

Hardware:

Hier geht es nun um die Verbindung vom Emulator zum Terminal. Diese 4 Leitungen wollen richtig an einem 7-poligen Stecker angeschlossen sein. Ein Hinweis vom Entwickler und ein Photo zeigen schon das erste Problem: Der Hinweis zielt auf die Spiegelproblematik bei symmetrischen Steckern /Buchsen. Der zweite Aspekt zeigt sich im Photo.



Links das Bild des Entwicklers , rechts das Bild meines Terminals, wie es sein muss um zu funktionieren. Es ist richtigerweise gespiegelt, kann man aber nicht unbedingt voraussetzen wenn schon Stecker oder Buchse am Terminal verbaut werden. Hat natürlich nicht sofort geklappt. Auch der Platinaufdruck scheint sich am Einbaustecker zu orientieren.



Ausserdem ist es nicht sicher, ob die Verbindungskabel Modem-Terminal nicht auch noch typenspezifisch verdreht waren.

DBT03-Emulator

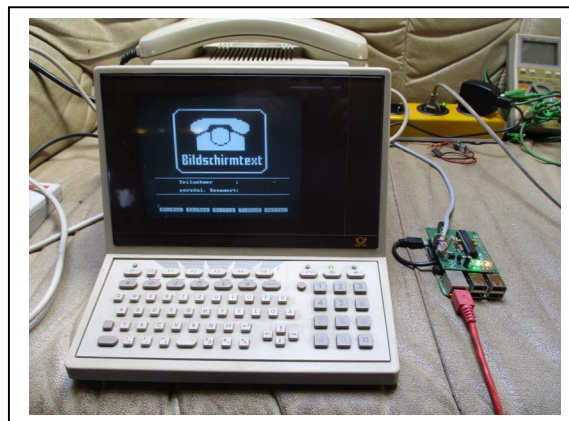
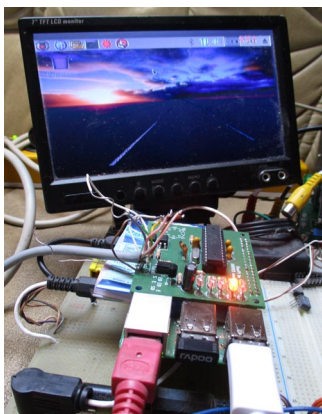
Hardware:

Daher auf Nummer sicher gehen und am zweiten Gerät (ging auch nicht) mal nachsehen, ob die Leitungen zu Schaltungen führen, die als Ein/Ausgang zu erkennen sind.



Also zweites Terminal zerlegt und mit Durchgangsprüfer die Leitungen verfolgt, soweit das ging. Danach war etwas Klarheit geschaffen. Ging aber immer noch nicht. Die Steuerleitung funktionierte und eine Verbindung zum Server wurde aufgebaut. Die ankommenden Daten wurden aber nicht beantwortet. Durch mithören an den Leitungen wurde plötzlich ein verändertes Verhalten festgestellt. Im Empfangskreis war im Emulator ein Pullup-Widerstand absichtlich nicht bestückt, da Dieser nicht nötig wäre. Bei diesem Modell (Pullup vorhanden aber wohl zu hochohmig) muss er aber zur Stabilität eingebaut sein, also alles bestücken, kann nicht schaden.

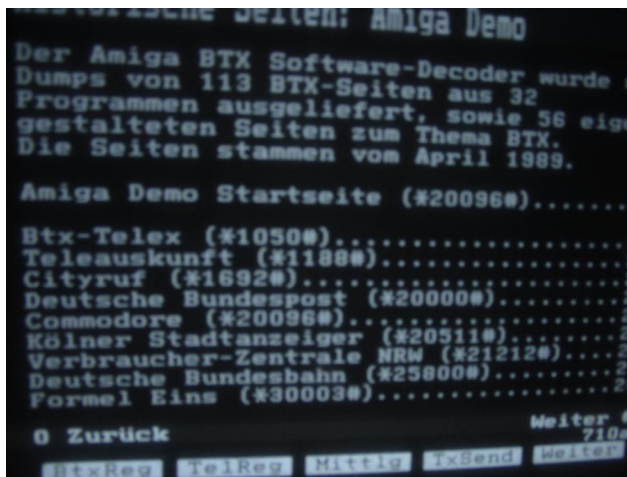
Nach diesem Ereignis wurde nach drücken der BTX-Taste der Server kontaktiert und vom Terminal bestätigt.



1 ½ Tage intensive Beschäftigung und dessen Lohn.

DBT03-Emulator

BTX-Seiten:

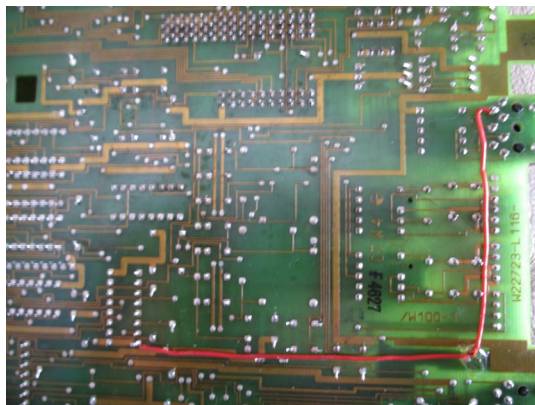


Ich habe natürlich sofort einen Account eingerichtet und etwas gestöbert.

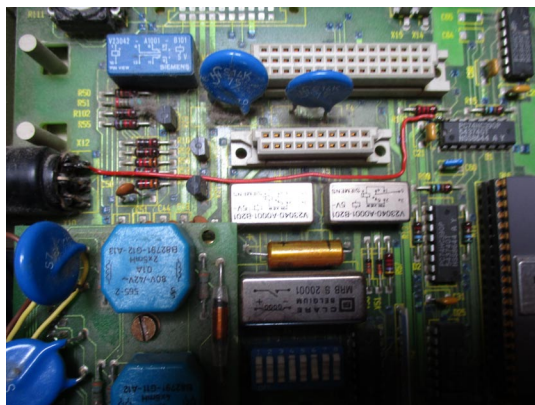
DBT03-Emulator

Erweiterung:

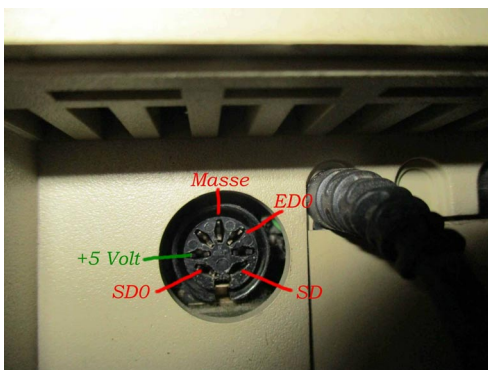
Etwas störte mich allerdings. Das DBT03 wurde aus der Telefonleitung gespeist, der Raspi als Ersatz erfordert ein zusätzliches (Stecker) Netzteil. Ein Terminal mit Bildschirm sollte doch genug Ressourcen bereitstellen können, noch mal 3-4 Watt versorgen zu können. Also beide Basisplatinen vorgenommen und einen freien Pin der Buchse mit 5 Volt vom nächstliegenden TTL-IC auflegen und am Emulator entnehmen zur Versorgung des Pi.



Bei der schon ausgebauten Platine bot es sich an, die 5V Leitung von unten zu verlegen.



Die andere Platine wollte ich nicht ganz ausbauen, Einbau dauert länger wie das Gehäuse zusammen zu bauen. Daher 5V Leitung von oben.



So sieht die Beschaltung aus.

1. +5V
2. Masse
3. frei(rechts)
4. frei(links)
5. ED Empfangsdaten
6. SD Sendedaten
7. Steuerleitung

Nummern entsprechen der Bezeichnung am Kabelstecker auf die Stifte gesehen.

DBT03-Emulator

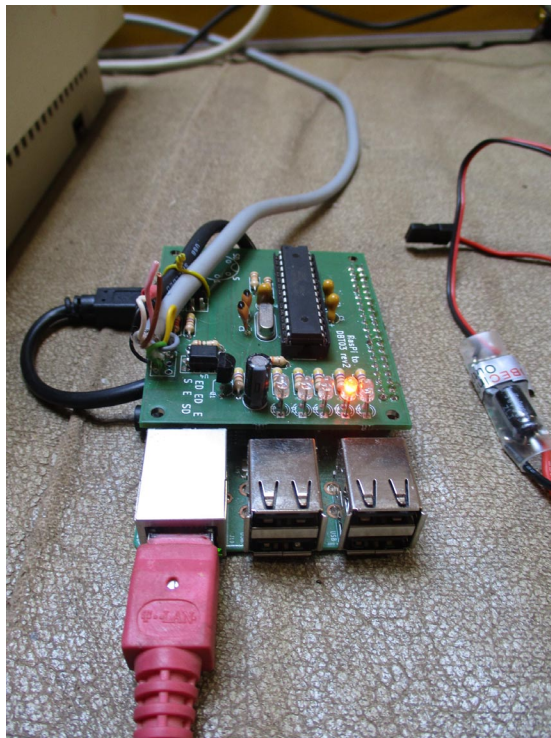
Erweiterung:

Mit dieser Spannung (direkt vom TTL-Schaltkreis) kann der Raspi direkt versorgt werden. Aber **Vorsicht:**

***Nicht direkt am GPIO-Board einspeisen, funktioniert auch hat aber Risiken
Über die MikroUSB Buchse einspeisen wie ein Netzteil***

Die Stromversorgung via USB hat den Vorteil, dass die auf der Raspiplatine enthaltenen Sicherungsmaßnahmen greifen können (Sicherung und Überspannungsschutz).

Ergebnis:



Es gibt nur noch das Netzwerkkabel und die Verbindung zum Terminal. Sobald das Netzteil des Terminals Spannung liefert, startet der Bootvorgang des Pi. Hiermit wird der Optkopplereinsatz etwas ad absurdum geführt, schützt aber weiterhin die Ports.

Wer hier Zweifel hat, kann alternativ einen DC/DC Wandler 5V in 5V out mit galvanischer Trennung als „Angstwiderstand“, zwischenschalten (im Bild rechts als Beispiel zu sehen).

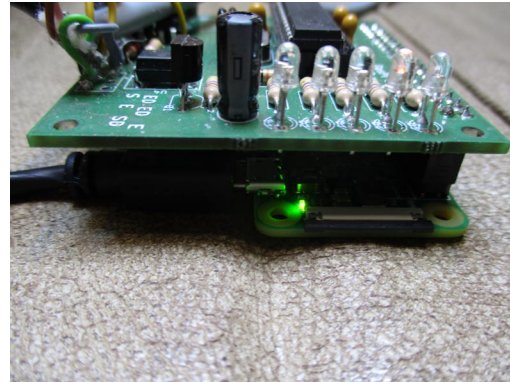
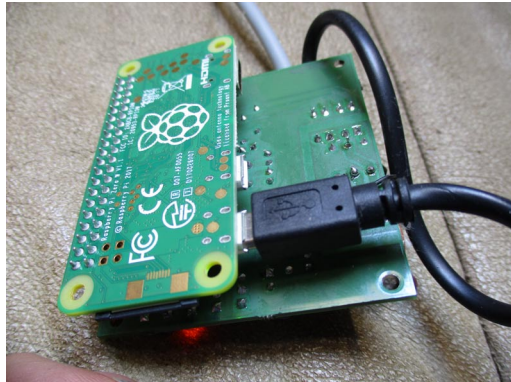
Diese Erweiterung ist ganz praktisch, man sollte aber daran denken zu prüfen ob diese Leitung bei Wiedergebrauch eines original DBT03 wirklich frei ist.

Daher ist eine kleine Schmelzsicherung (0.8A) einzuschleifen durchaus angebracht. Komplettes Setup siehe Deckblatt.

DBT03-Emulator

PiZero:

Alternativ zum normalen Raspi Pi ist drängt sich für solch einfache Zwecke der PiZero mit integriertem Wlan auf. Er ist billiger und der Stromverbrauch geringer. Leider gibt es nur 1-Core Ausführungen. Also mal testen ob es überhaupt läuft. Der Wechsel der SD-Karte und ist einfach und hier wird der PiZero auf den Emulator aufgesteckt.



Die Stretch Version mit Desktop läuft, der DBT-03 Dienst startet mit blinkender LED. Das ist schon mal die halbe Miete.

Aber noch ist kein Betrieb möglich, da die Wlanverbindung noch fehlt.

Auch hier steckt der Teufel im Detail. Der integrierte Wlan-Baustein wird nicht so aktiviert wie die evt. USB-Variante. Zweitens muss die SD-Karte (OS) so präpariert werden, das der Pi ganz von alleine ins Wlan geht. Ansonsten wäre der Erstkontakt auch nur über Bildschirm und Tastatur möglich. Das wiederum erfordert mehrere Adapter (HDMI/ USB) , die nicht jeder rumliegen hat.

Daher ist es sinnvoll, die SD-Karte für den Pi-Zero mit folgender Information zu beschreiben:

Nach Beschreiben der Karte mit dem Image des OS gibt es einen Grundordner, der auch mit Windows lesbar ist. Dort liegt ein Ordner Namens „Boot“.

In Diesem wird eine Datei angelegt mit dem Namen „wpa_supplicant.conf“.

(Diese gibt es auch unter /etc/wpa_supplicant/ für die USB-Wlaninterfaces)

Folgende Angaben müssen da stehen:

```
network={
ssid="wlan-ssid"
psk="wlan-passwort"
key_mgmt=WPA-PSK
}
```

Selbstredend muss die SSID und das Passwort individuell angepasst werden.

Das schliesst leider ein 100%iges Kartenimage aus, diese Anpassung muss jeder selber machen.

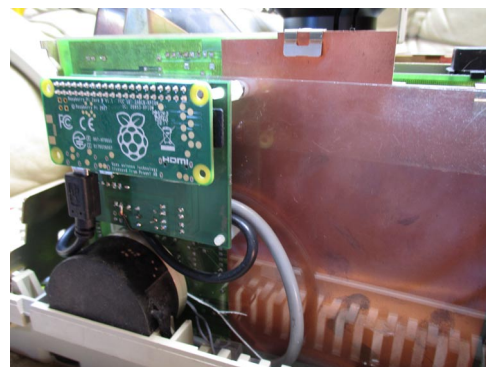
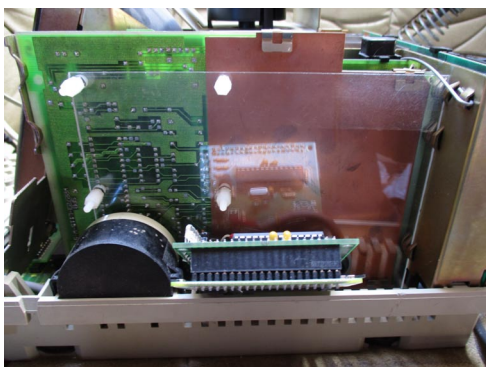
DBT03-Emulator

PiZero:

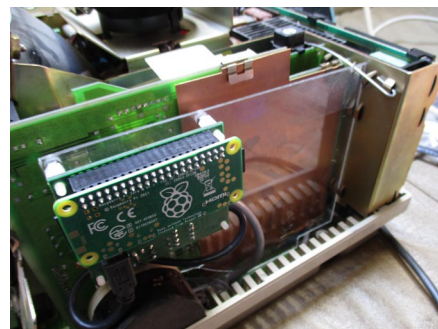
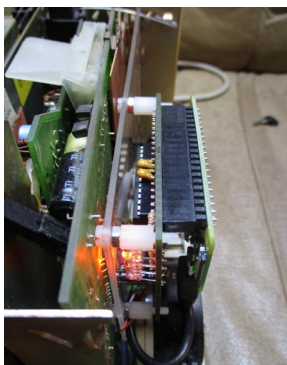
Dafür wird man mit einer eleganten smarten BTX-Modemlösung belohnt. Ich bin noch einen Schritt weiter gegangen und habe die Lösung in das Terminal integriert.



Als hätten die Designer es geahnt: das Modemkabel kann über die dritte Kabel-Ausparung wieder ins Gehäuse zurückgeführt werden.



In einen noch freien Steckplatz schnell einen Rest Plexiglas eingearbeitet und 4 Abstandshalter montiert (alles aus Kunststoff). Da die Lochbohrungen etwas kleiner 3mm sind, wird das Modul einfach aufgedrückt, hält super.

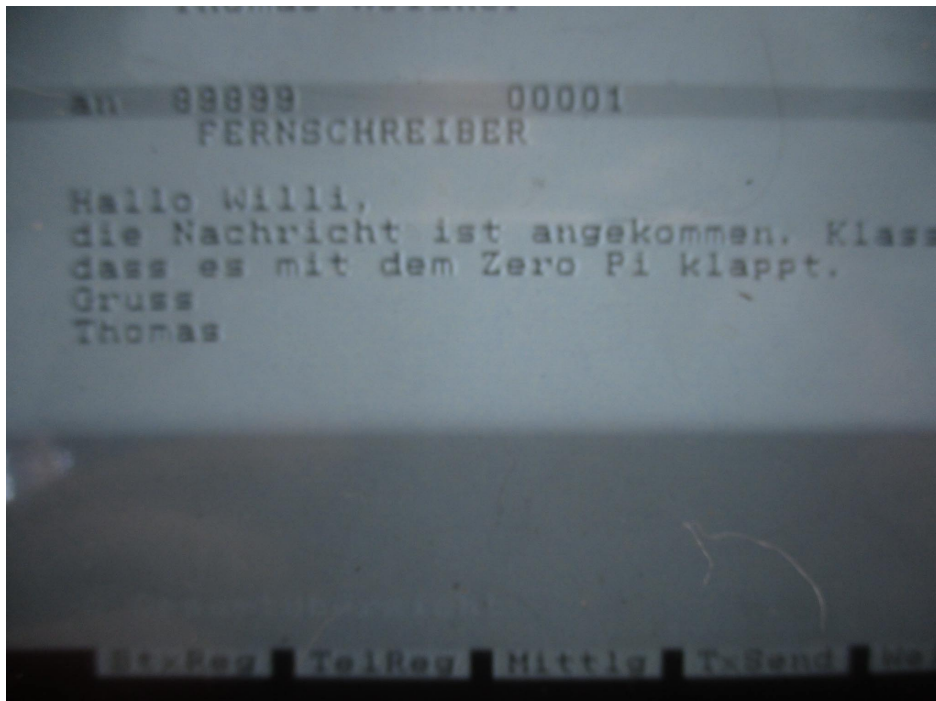


Der Befestigungsort sollte frei sein von Abschirmblechen, um Wlan zu ermöglichen.

DBT03-Emulator

PiZero:

Damit hat man ein BTX-Terminal, das nur noch den Netzstecker braucht.
Will man auch telefonieren, ist das TAE-Kabel halt noch dran, aber das könnte man ja mit DECT.....



Dann lief auch schon die erste passende Nachricht ein.

DBT03-Emulator

Abschluß:

Generell ist diese Lösung interessant, funktioniert sehr gut, die Realisierung sehr durchdacht und die eigentliche Installation völlig automatisiert,.

Als Fazit meiner Erfahrung:

OS: mindestens Jessie, Stretch oder gleich Buster, möglichst in der lite-Version

Hardware: min. Raspi Ver.2 4Core, oder Pi ZeroWlan, kleiner läuft nicht.

Emulator: alle PullUp Widerstände einlöten, Verbindung 3-6 auf Platine nicht Vorhanden (prüfen), auf Platine die Widerstandsbezeichnung lesbar positionieren. Weiterhin sollte die Platine von unten isoliert werden, sonst Berührungsgefahr mit anderen Komponenten(HDMI Vers.2) beim testen oder sofort richtig montieren. Bei Pi Zero keine Probleme

Der DBT03-Dienst ist nach Einschalten in etwa 12 Sekunden bereit (nutzbar nach etwa 20 Sekunden), schneller ist die Bildröhre auch nicht. Es muss sich allerdings noch zeigen, ob dieser Headless-Betrieb sich auf Dauer bewährt oder nicht doch ein automatisches Herunterfahren sinnvoller ist (mit entsprechendem Goldcap als USV).

Also: super intelligente Lösung, lediglich ein Kabel muss man sich löten.

Aber: Der Preis für diese Kombination dürfte je nach Bastelkiste bei normalem Pi 60€+ Gehäuse liegen, mit dem Pi Zero etwa bei 40€. Gehäuse kann man bei Einbau ja sparen.

Mein Dank an Thomas für die intensive Fehlersuche.



Wilhelm Schütters

(Fernschreiber)

25060 TXVst Münster

25061117 ufs lingen T68d DW90

89899 schuett d T1000 Dw 94 Hauptstelle

826433b vdm d T100 Dw 91

841226 mizg d Lo15 Dw 95

84635 norman d T100 Dw 96

Fragen oder Anmerkungen /Interesse zu dieser Lösung bitte auch via i-Telex!!