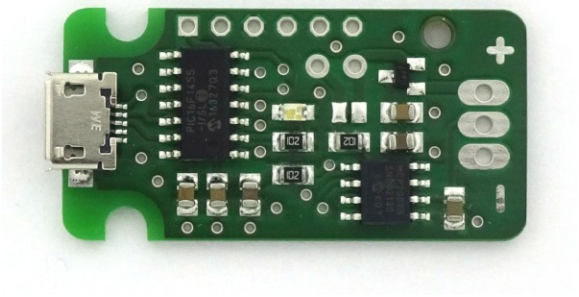
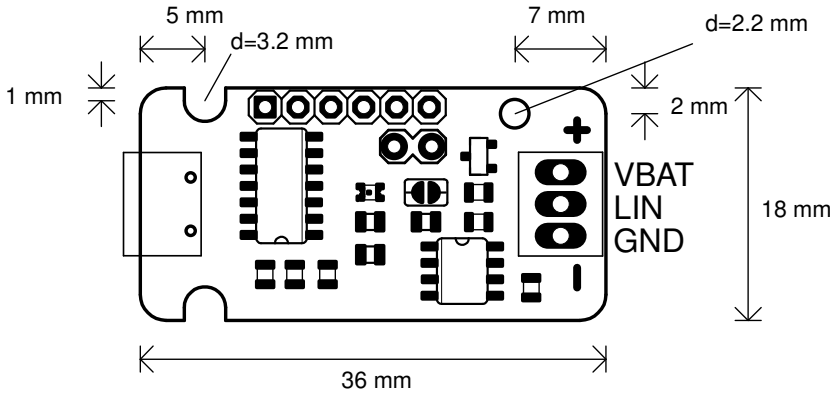
USBlini EB – LIN to USB Interface

Datasheet

USBlini is an interface to connect a LIN bus via USB to a host system. It provides easy access to LIN bus devices and can act as both master and slave. It features a logic analyzer function to sample the logic levels on the LIN RX line.





Features

- Open source Python library (Windows, MacOS, Linux)
- Master mode (generate frames) and slave mode (response to master requests)
- Logic analyzer function: sample logic levels on LIN-RX with 100ksps
- Integrated 1kOhm master pullup
- Flexible baudrate configuration via software 1200 – 20000 Baud (e.g. 9600, 10400, 19200)
- Firmware sources available; bootloader for firmware updates

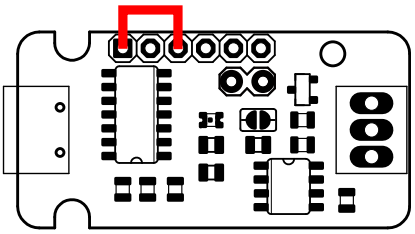
Parametrics

- VBAT - Transceiver supply voltage: 6V – 30V
- Transceiver LIN Bus Specifications: 1.3, 2.0, 2.1, SAE J2602
- Board supply voltage (via Micro USB-B): 5 V
- Board supply current @VUSB: 15 mA
- Operating temperature: -40 up to +85°C

Power supply
The microcontroller is powered via USB, the LIN transceiver via external supply VBAT.

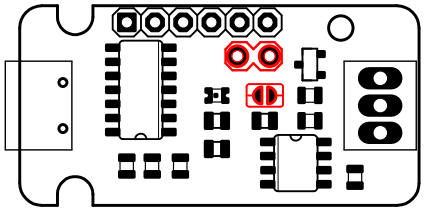
Connectivity
The board is connected to the LIN bus via three lines. Solder connections with a pitch of 2.54mm are available for this. These can be equipped with pin headers or screw terminals to meet the requirements of the application.

Bootloader
USBlini contains a bootloader that can be used to install firmware updates. The bootloader can be started by a command via USB or by bridging the following pins while plugging to USB:



To flash new firmware versions in bootloader mode, e.g. MPHidFlash can be used:
https://github.com/AdamLaurie/mphidflash

Master pullup resistor
The board is equipped with a 1kOhm pullup resistor which can be enabled by jumper.



Communication
The USBlini firmware enumerates as a USB vendor device. It uses endpoints EP0 (commands), EP1 (Reports to host: received frames and status reports) and EP2 (logic level bitstream).
An open-source Python library is available to easily access the functions of USBlini:
https://github.com/EmbedME/pyUSBlini

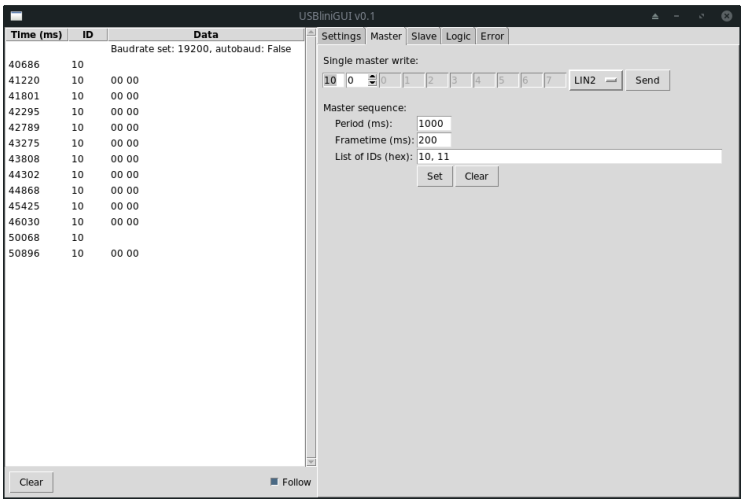
Example using python lib:

```
from usblini import USBlini
usblini = USBlini()
usblini.open()
data = usblini.master_write(0x10, USBlini.CHECKSUM_MODE_LIN2, [])
print(data) # print out response
usblini.close()
```

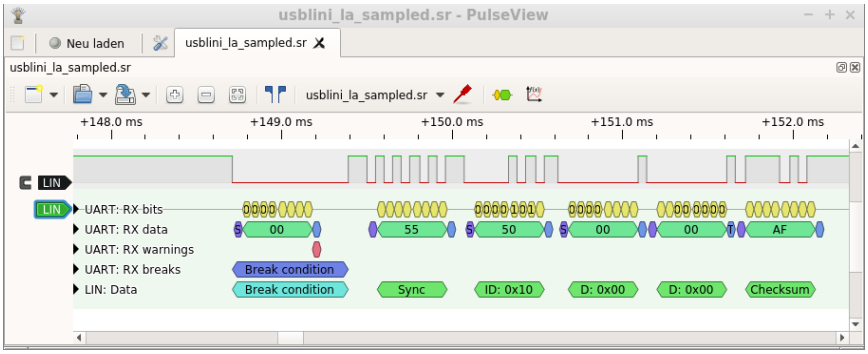
If you would like to implement the communication in another programming language instead, you can find further information about the USBlini protocol in the following document:

https://www.fischl.de/usblini/USBlini_EB_Protocol_v1.pdf

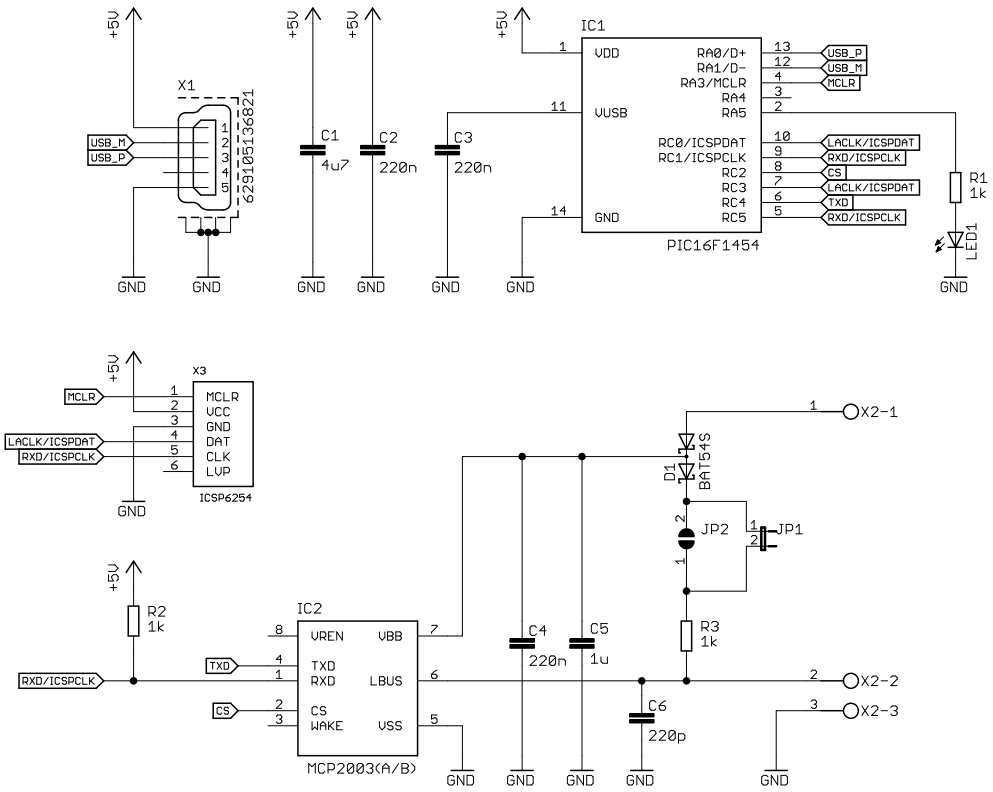
Software
Based on the python library there is a simple GUI application: USBliniGUI.
https://github.com/EmbedME/pyUSBlini/blob/main/tools/USBliniGUI.py



With the logic recording function, the logic level on the LIN RX line can be captured and then analyzed with e.g. PulseView.



Schematics diagram



Evaluation board notice

USBlini EB is an evaluation board intended for use for engineering development or evaluation purposes in laboratory enviroment only. It is not considered as an end user application. If you intend to use it in an end-product, you have to ensure in your own responsibility that the device meets the relevant regulations (e.g. CE, FCC).