

Ethernet-MIDI-USB Converter with STM32F042 or STM32F070

©2018 Wolfgang Schemmert Status 07 June 2018

This is a DIY construction manual for of a triple port communication interface between wired Ethernet (WIZ850io module), USB (MIDI Class or CDC Class) and a combined legacy MIDI or serial COM port.

By means of two jumpers or switches different modes of operation are selectable:

Jumper or switch no.1 selects between **general MIDI or serial COM port behaviour** of the device

Jumper or switch no.2 configures the priority of data flow:

no Jumper no.2 (default): **Ethernet bundles** (collects, merges and retransmits) bytes received from the legacy serial port or from USB.

Bytes received via Ethernet are retransmitted via serial port and USB simultaneously.

Jumper no.2 set or switch closed: the roles of Ethernet and USB are exchanged: **USB bundles** (collects, merges and retransmits) bytes received from the legacy serial port or from Ethernet.

Bytes received via USB are retransmitted via serial port and Ethernet simultaneously.

In contrast to a true MIDI Merger, received data are merged byte by byte instead of message by message. So only one of the merged communication ports may receive during a complete message.

To make DIY construction as easy as possible, an **STM32F042F6P6 (20 pin TSSOP) microcontroller** was chosen, which is mounted on a commonly available TSSOP to DIL (600 mil, 15.24mm) adaptor PCB. An **STM32F070F6 microcontroller** is useable as well. The adaptor board may be soldered directly on the base PCB, but it is recommended to solder a 20 pin DIL socket (32 pin socket shortened to 20 pins for example) on the base board and to plug the adaptor PCB in there.

The additional circuitry for the conventional MIDI/serial interface, the WIZ850io, USB connector, jumpers and LED **is located on a 8.5x7.5 cm single layer PCB**. Its design is less elegant, but held strictly in 2.54mm raster (except USB connector). **So it is possible to rebuild the complete project without PCB etching on a Veroboard** with round copper dots. Mounting holes are provided for a low-cost "Euro-Box" enclosure available from Reichelt and Conrad for example. If the PCB is cut 10x7.5cm wide (see PCB layout), it fits into a rugged aluminium box ("Fischer Elektronik FRAME").

The **USB interface** is "full speed USB2.0" grade. During setup it operates in USB CDC Class, i.e. provides a virtual COM port at the host. During MIDI transfer it is configured as USB MIDI Class Audio device.

The **wired Ethernet** interface is realized with a compact **Wiznet WIZ850io** module. If general **MIDI behaviour** is selected, by default it operates in UDP Multicast mode, which provides a very good MIDI interface on the host together with the ipMIDI driver (available from www.nerds.de). As an alternative, standard UDP can be configured by setup.

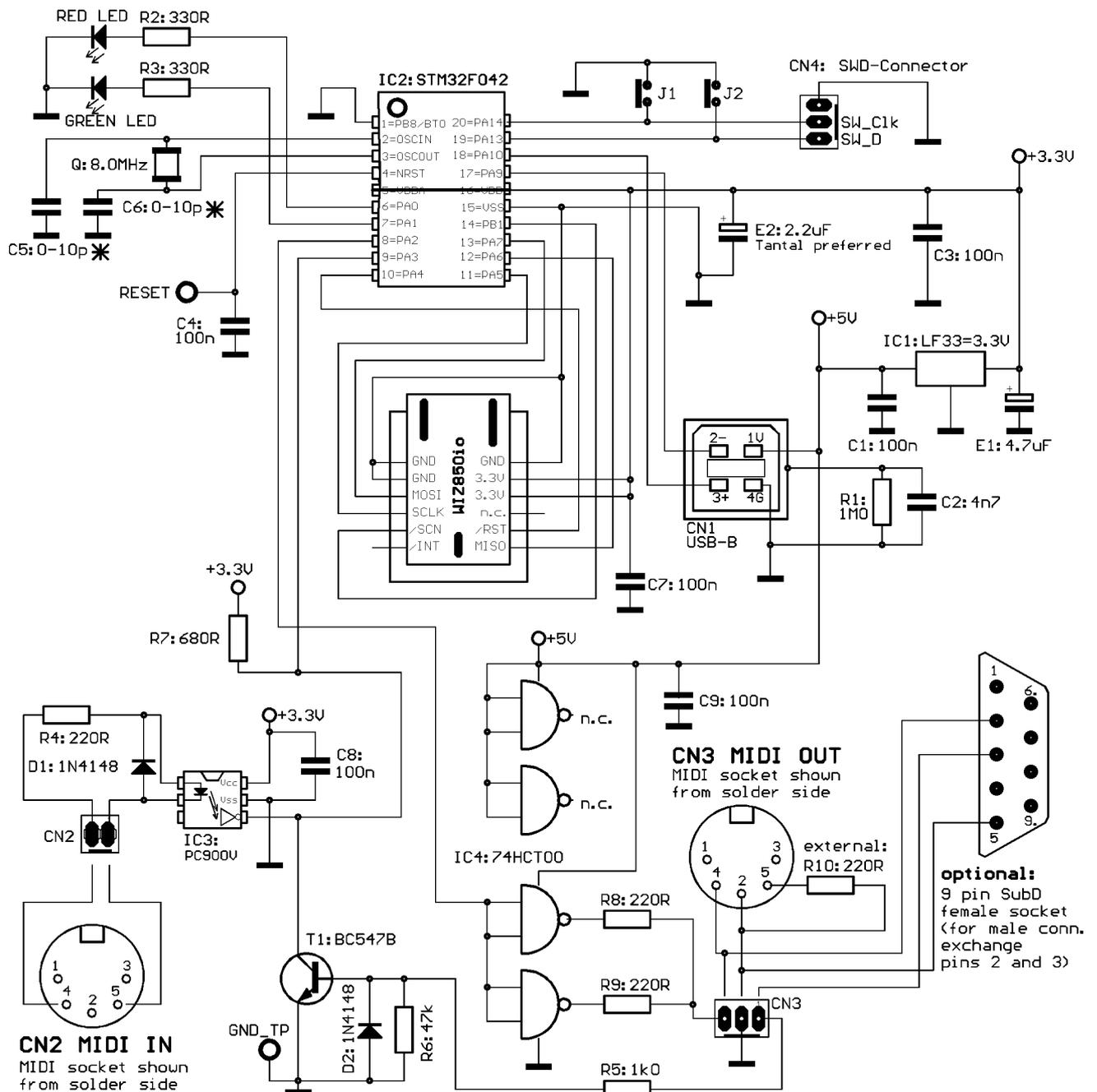
If general **serial COM Port behaviour** is selected, by default it operates in TCP mode (simplified Telnet), but may be set to UDP by setup.

5 Volt power is supplied through the USB connector, even possible by a USB power bank. USB connectivity is not necessary when Ethernet works as main communication interface (Jumper 2 not set).

It is **NOT** allowed to use this device together with any safety critical applications, where malfunction could result in personal injury oder noticeable material damage !
 All information about this project is provided 'as is' – without any warranty or responsibility

Hardware of the base PCB

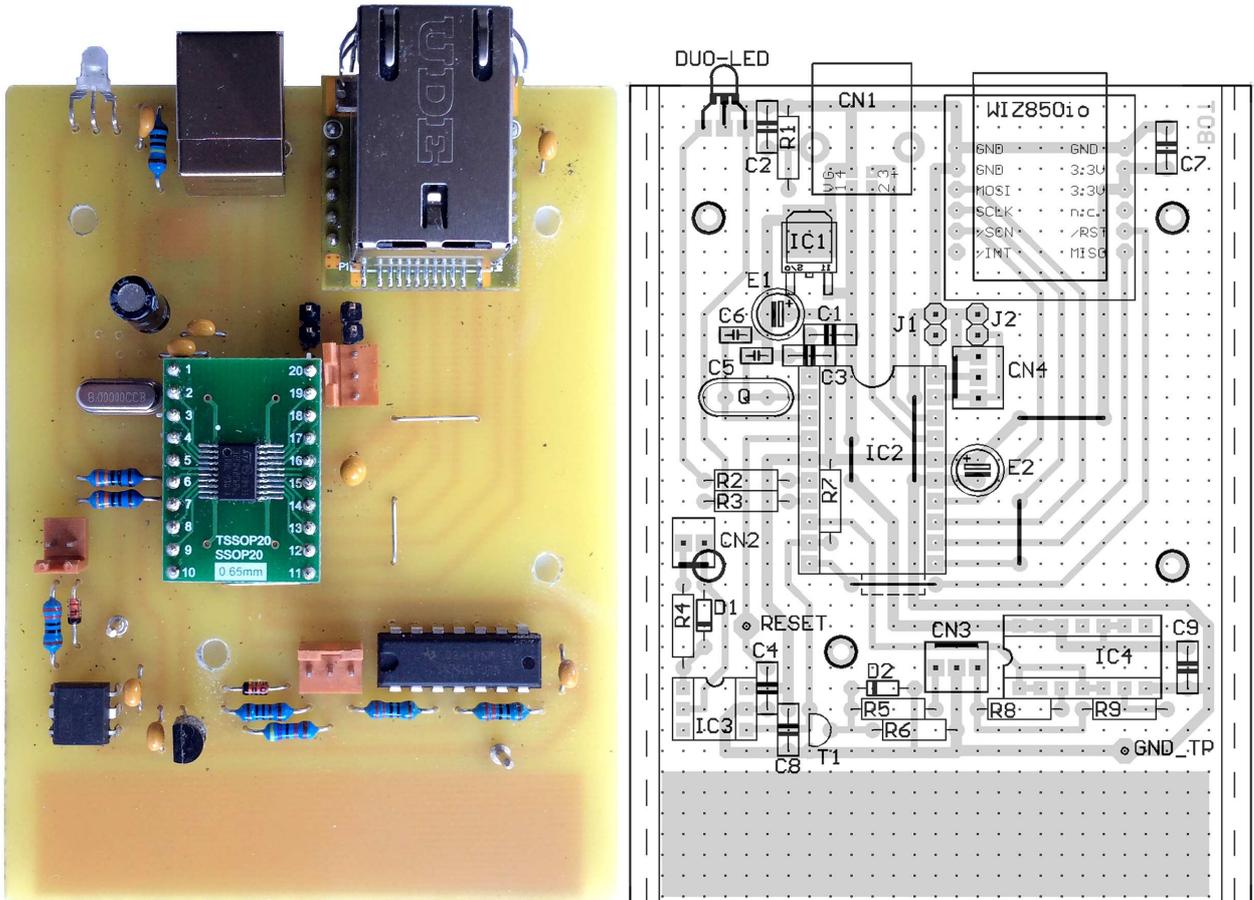
Schematic diagram:



Assembly:

Most times the quartz oscillator works best, when C5 and C6 are not assembled. So it is recommended not to solder C5 and/or C6 and perform a first test with the assembled board. If the oscillator won't start reliably then, first try C5=C6=4.7pF or one capacitor with 10 pF.

Be aware of the **2 jumper wires** and **one resistor** below the **20 pin DIL socket**. Must be soldered before the socket !



Recommended drill diameters:

resistors, capacitors, diodes, ICs, jumper wires, USB socket = 0.8mm, else 1.0mm

The USB socket does not fit into the 2.54 mm raster. Recommended drill for case shielding 2.0mm

A 1:1 TIF file with the PCB bottom layer and Veroboard assembly is available for download at my webpage: www.midi-and-more.de/ethernet-midi-usb.htm

Special parts: (suppliers are my sources for example. Components are available from others, too)

IC1: LF33CDT (D-PAK case): source e.g. Reichelt, Conrad

IC2: STM32F042F6P6 or STM32F070F6P6: sources www.tme.eu, Mouser, RS-Components, Farnell

LED: V-L-115-WEGW: source Conrad 187496. Flattened side directed to PCB for correct colours.

Other dual red/green LED types may be used, but then values of R20 and R21 have to be adjusted
Optocoupler: PC900V; source Conrad 184098

20p TSSOP (0.65mm pitch) to DIL (600mil) adaptor PCB: e.g.: www.gme.cz, Adafruit, Naltronic_de at eBay
pinhead array to be soldered at the TSSOP to DIL adaptor: e.g. Reichelt AW122/20

connectors CN2,3,4: source Reichelt PS 25/2G BR, PS 25/3G BR

Programming

Prototype development and programming is made with the ST-LINK part of a STM32 Nucleo module. Instead of buying a dedicated ST-LINK programmer, the Nucleo module is recommended, because it is less expensive and it is a good base for other programming projects. Installation of the ST-LINK software see below.

On the Ethernet-MIDI-USB PCB a 3 pin socket is provided for programming by ST-LINK.

How to program external parts with the Nucleo ST-LINK module, read its manual. When the programmer is cut off and used standalone, the programming software possibly sends a problem like "no target voltage". Then connect the 3.3V output (pin next to text U1) of the regulator (5 pins) on the ST-LINK part with R23 (4.7 kOhm, pad directed towards the SWD connector) to pretend a supply voltage of the programmed device.

How to build a programmer adaptor:

- connect the 2nd pin of the Nucleo SWD connector (counted from the side towards the Mini USB connector) with pin PA14 of the microcontroller on the remaining Nucleo board.
- connect the 3rd pin with Ground.
- connect the 4th pin with pin PA13 of the microcontroller

These pins are broken out on the PCB as **connector CN4**. (orientation see assembly drawing).

While ST-LINK is connected, jumpers J1 and J2 MUST be removed / switches OFF !

- start the ST-LINK software. Click item "Connect" of the "Target" menu. After some seconds, a screen with the connection report and a listing should appear. In very rare cases it is necessary to reset our board for a moment while you connect it with ST-LINK. When you are connected, select "Program&Verify" from the "Target" menu and upload the hex code.
- The ST-LINK may be directly operated from the Embitz IDE

User Setup:

For proper setup and operation, **the ST-LINK connector should be removed !**

The User Setup mode is available when USB is configured as virtual COM port (jumper J1 NOT set) via the legacy RS-232 interface and additionally via the prioritized digital bus interface (Ethernet or USB, depending on J2).

In first-use default state (no jumpers set and no previous User Setup), **Ethernet is started as TCP "server" (listen state) with IP 192.168.0.240 and port 23 and the serial interface is configured for 115200 baud**. On the host PC, Ethernet can be opened with a Telnet capable terminal software, e.g. "Teraterm".

Before USB can be used, an appropriate driver has to be installed on the host PC first. This happens automatically when the ST-LINK driver package is installed:

If you already have installed the "STM32 Virtual Com Port Driver", by default our board uses the same driver and will be listed in Windows Device manager under "COM and LPT ports" as "STMicroelectronics Virtual COM Port". Else download the "ST-LINK utility" software including the "STM32 Virtual Com Port Driver" from the STM website and install it. Possibly the driver is not loaded automatically when you connect the converter. Then select the location of the driver "manually". You can change the COM port number under "advanced settings" in the Device Manager. Else keep the default settings. The baud rate is not relevant in this case, any works.

In MIDI mode the PC installs another driver, but usually this is handled automatically. Will be listed by MIDI software and Windows Device Manager as "Ether-MIDI-USB Conv" or simply as "USB Audio Device".

To get into the User Setup mode when the device is operating in one of the serial COM modes as described above, type three times a plus "+++" within max 1 second and one second no input before and after (as known from legacy modems). The message "Ethernet-MIDI-USB Converter" shall be prompted. Then you are in the setup mode and can enter single byte "command codes" followed by some numeric parameters as described below. All input is case independent. Wrong input is refused with a question mark '?'. Then simply repeat the command. Else the interface is held quite simple, Backspace is not supported. You can leave the setup mode with command 'E' back to transparent data transfer.

When you enter command code '!', a list of the actual settings is displayed (i.e. when called first = default settings). The following list can be requested with command ? :

```

!: list Actual Setup
V: set USB Vid/Pid of this device (hex)
A: set Ethernet MAC ID (hex)m5.m4.m3.m2.m1.m0
G: set Gateway IP (ip3.ip2.ip1.ip0)
I: set IP of this device (ip3.ip2.ip1.ip0)
S: set Subnet Mask of this device (m3.m2.m1.m0)
L: set Lan Mode for MIDI: 0=UDP Multicast, 1=std.UDP, 2=TCP
C: set Lan Mode for Serial COM: 0=UDP Multicast, 1=std.UDP, 2=TCP
M: set Multicast UDP IP(ip3.ip2.ip1.ip0) must be MCAST compatible!
H: set Multicast UDP Local(Rx) Port number
D: set Multicast UDP Destination(Tx) Port number
U: set Std.UDP Destination IP incl.Broadcast(ip3.ip2.ip1.ip0)
R: set UDP Local(Rx) Port number (std.UDP, not Multicast)
T: set UDP Destination(Tx) Port number (std.UDP, not Multicast)
P: set TCP (Telnet) Port number
X: set TCP Disconnect Time (0 ... 65535 seconds, 0=no auto disconnect)
F: set Bit0:block TimingClk, Bit1:block ActSens ,Bit2:block TuneReq
B: set Serial COM Baudrate: first 2 digits,MIDI=31 (active in RS-232 Mode)
N: set MIDI Baudrate: first 2 digits,MIDI=31 (active in MIDI mode)
@: reload setup from Flash (changes since last Power On are cancelled)
|: save new setup permanently (will become active at next power up)
E: exit SETUP, return to bidirectional communication

```

Please note that the default values for USB Vid, Pid and Ethernet MAC ID are allowed for device test and evaluation only! For any use else, you have to enter your own Vid, Pid and unique MAC ID !

The default Vid/Pid is the same as installed with ST-LINK, no additional Windows driver installation is necessary. The default unique MAC ID is provided by courtesy of Cinetix.

To avoid unexpected communication breakdown, **the new setup is not activated before the next reset or power up. So new entries are lost, when you leave Setup mode without saving the new setup permanently with command '|' !!** Exception: commands N, F and X are active immediately but are stored permanently, too.

Factory Reset: Use the ST-LINK utility and "Erase Sectors" page 31 (address 0x08007C00)

MIDI System Messages like "Active Sensing" are annoying sometimes without having a real effect. **You can block them with command "F"**.setting Bit0 means: add 1, setting Bit1 means add 2, setting Bit 2 means add 4. For example: F=7 blocks all listed ones.

LED blinking is blocked in any case when MIDI Timing Clock (0xF8) or Active Sensing (0xF6) is received.

Operation

After setup the Ethernet-MIDI-USB converter essentially works quietly in the background.

About latency: regardless of the logic content, every byte is retransmitted as fast as possible. Corresponding with MIDI baudrate, the systemic latency is about 1/3 millisecond per byte. The internal processing time is negligible compared with this. Ethernet messages are bundled and transmitted asynchronously (ARM SysTick timer) in packets of 1 ms raster to get a good compromise between single MIDI channel messages and bigger data blocks like System Exclusive messages. USB data are transmitted per millisecond in packets up to 64 bytes.

Independent of the device internal latency - additional latency will be added by the PC operating system, which depends on PC specific details and cannot be estimated in general.

contact: wschemmert@t-online.de

* Right of technical modifications reserved. Provided 'as is' - without any warranty. Any responsibility is excluded.

* This description is for information only, no product specifications are assured in juridical sense.

* Trademarks and product names cited in this text are property of their respective owners