STM32F042 based USB to MIDI or RS-232 Interface
(small PCB version)

03 December 2021

This is a DIY construction manual for a small USB to MIDI(or RS-232) interface based on the **STM32F042F6** microcontroller (20 pin TSSOP). Hard – and firmware fit for **STM32F070F6** (20 pin TSSOP) processors too.

Though a number of SMD parts is used, technology is held as simple as possible. The prototype board is bathroom made and freehand assembled.

The interface is "full speed USB2.0" grade. USB class selection between MIDI and CDC class (= virtual COM port for RS-232) is made by a jumper near the USB socket (may be connected to a switch). The 5V supply is scaled down by a linear low drop regulator to 3.3V which supplies the complete board except the serial output driver (total supply current about 26 - 30mA, depending on operation conditions).

The basic mode of operation is "transparent" bidirectional transfer of a byte stream. A simple combined MIDI and RS-232 port hardware is provided on the board with separately wired connectors. Depending on the activated USB class, the device works as a standard MIDI interface or as an external RS-232 port. A dual LED signals presence of power and data flow.

Two rows of pinheads (or socket arrays, as user prefers) are provided in the middle of the PCB, which break out all I/O pins of the microcontroller for additional tasks or "shields".

Up to 6 potentiometers or pushbuttons can be installed there to trigger user programmable messages merged over the "transparent" data transfer.

The board is preferably programmed with ST-LINK, which is connected to the upper 3 pins of the pinhead next to the USB connector. Wires for power and reset are not necessary. Alternatively, but less comfortably, the module can be programmed with the STM Flash Loader via the on board RS232 connector. Unfortunately the loader supports only USART1, which is not directly available because USB is mapped on these pins. A workaround is shown below.

In RS-232 operation, a special configuration mode can be activated. Here, for use as USB virtual COM port baud rates 9600, 19200, 38400 , 57600, 115200 plus MIDI (31250) are user configurable. **These baud rates are configurable independently for use as MIDI interface, too.** With the extended baudrate options, the device is useful as a data bridge between RS-232 and MIDI "worlds".

User specific USB Vid/Pid are configurable as well as MIDI messages to be triggered by pushbuttons or potentiometers.

The user setup can be stored nonvolatile in the microcontroller flash.

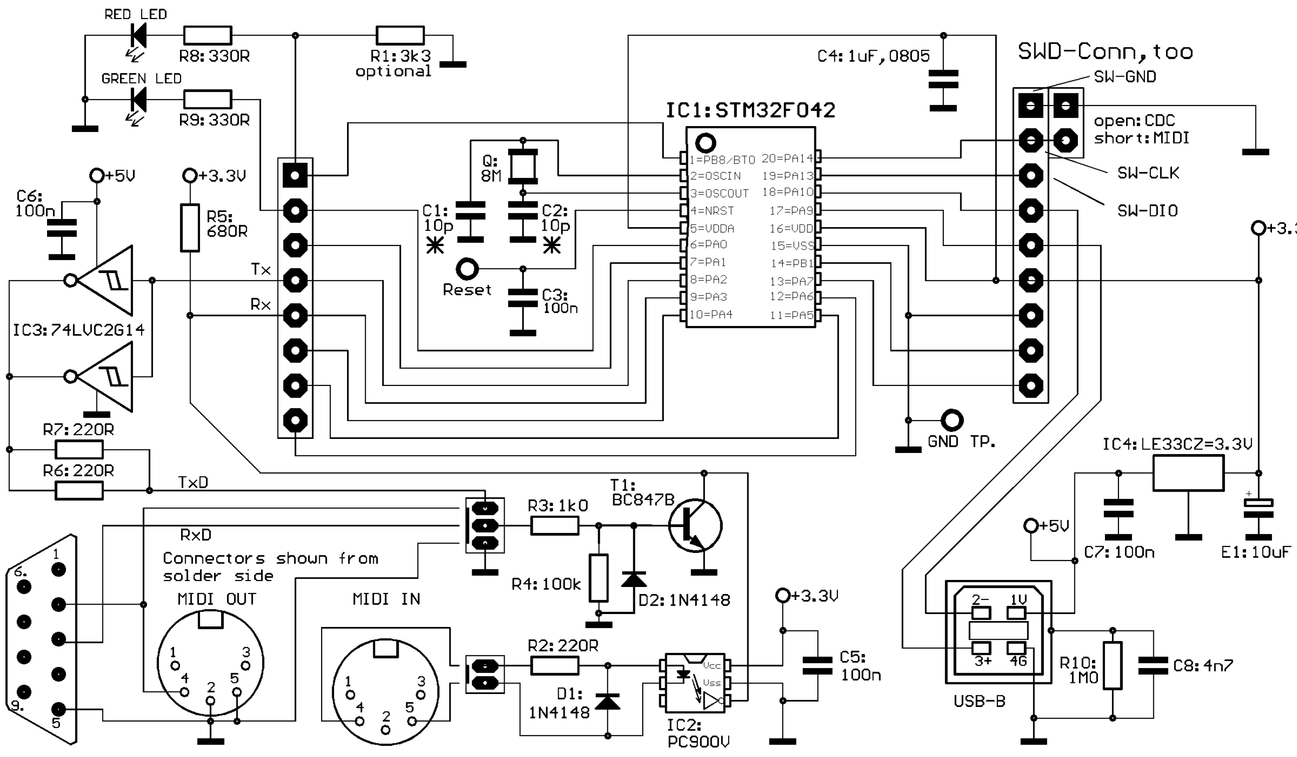
It is **NOT allowed to use this device together with any safety critical applications**, where malfunction could result in personal injury oder noticeable material damage !

All information about this project is provided 'as is' – without any warranty or responsbility

Hardware

For easy reproduction with simple tools, the hardware is built on a single layer PCB with a small number of jumper wires. Thickness and spacing of the printed wires are designed for hobbyist technologies. Assembly is made with a mixture of through hole and SMD parts.

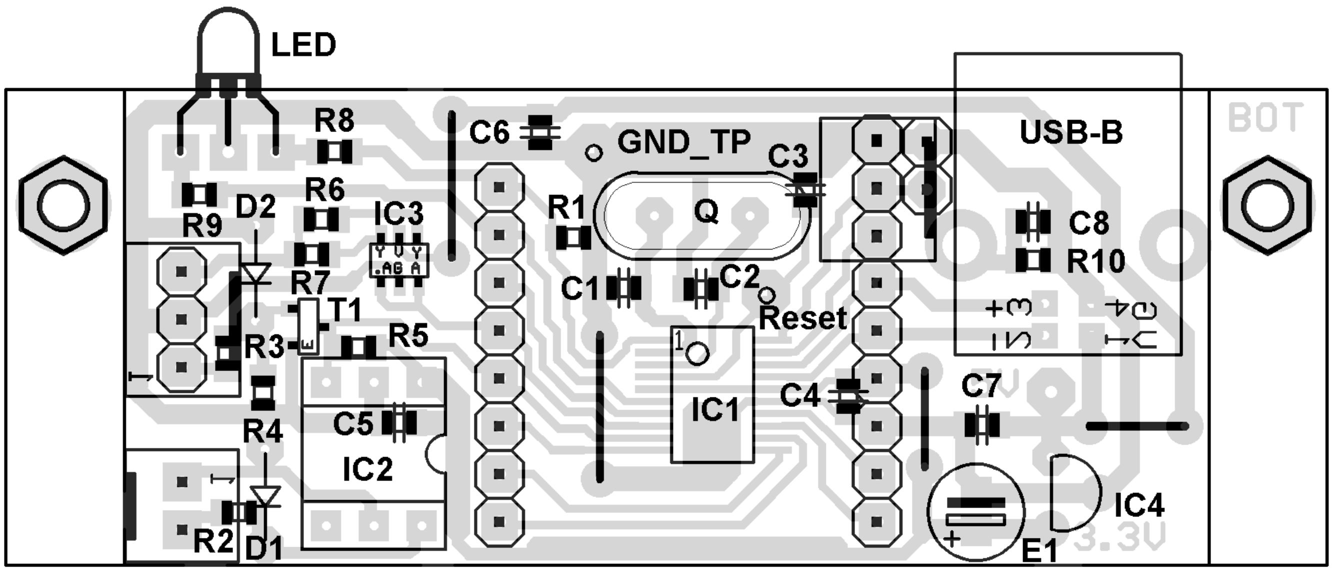
Schematic diagram:



The serial output driver of the actual hardware version is equiped with a 74VHC2G14 push-pull MOS driver. This improves noise immunity of long RS-232 cables.

Assembly:

This assembly drawing is shown from the **solder side**!
All SMD resistors and capacitors are size 0805.



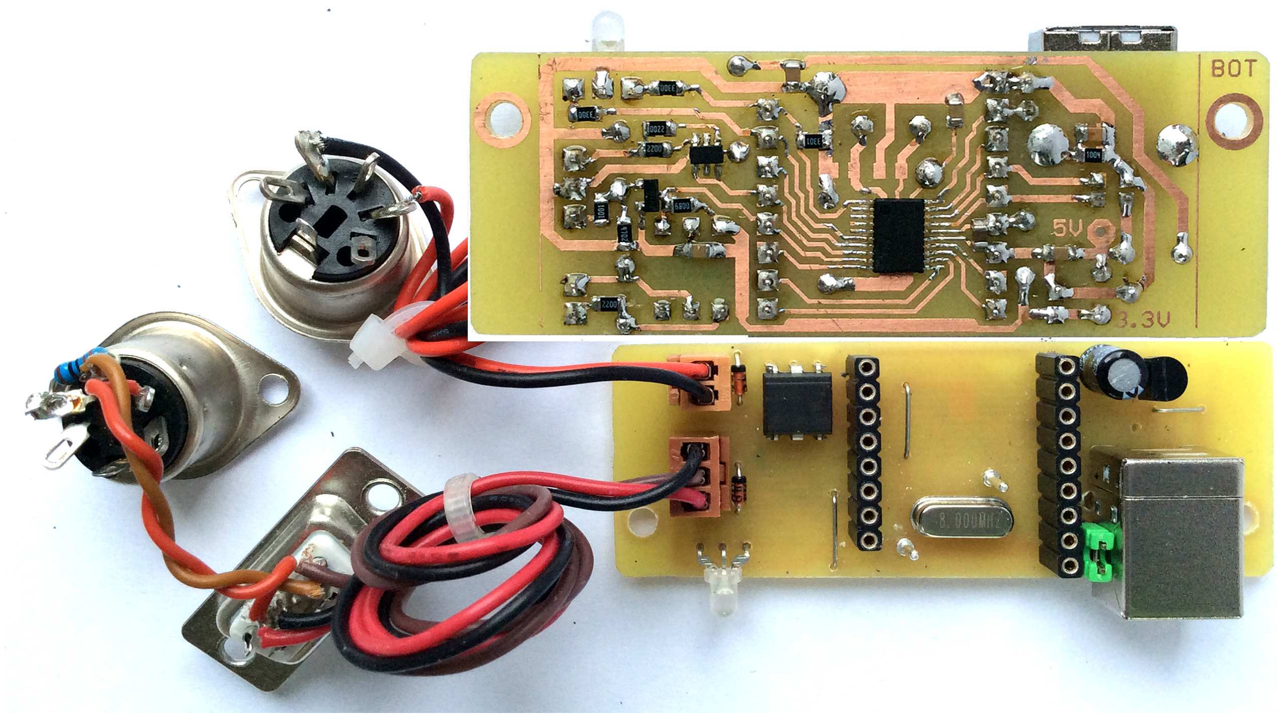
Many times the quartz oscillator works best, when C1 and C2 are not assembled. So it is recommended not to solder C1 and/or C2 and perform a first test with the assembled board. If the oscillator won't start reliably then, first try C1=C2= 10 pF or even 15pF, depending on the quartz type, too.

The LED has to be mounted with the flattened side towards the PCB to get the colors as described in the text.

R1 is only necessary to pull the BOOT0 (PB8) pin down if the leakage current of the LED is not sufficient.
The Reset pin is only necessary for programming.

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

IC1: STM32F042F6P6, STM32F070F6P6 source: www.tme.eu, Mouser, RS Components, Farnell
IC2: PC900V, source Conrad 184098
IC3: 74LVC2G14GV (SOT23-6 or SOT457 case pitch0.95mm), source Reichelt
IC4: 3.3 V regulator LE33CZ, source Conrad 1185305
Q:standard quartz HC49, 8.000 MHz.
LED: V-L-115-WEGW, source Conrad 187496. Flattened side directed to PCB for correct colors.
Other dual red/green LED types may be used, but then values of R8 and R9 have to be adjusted
Optocoupler: PC900V, source: Conrad 184098
C4: Murata High Cap, source: Reichelt X7R-G0805 1,0/25
connectors for MIDI/RS-232: source Reichelt PS 25/2G BR, PS 25/3G BR. Connectors may be soldered directly.



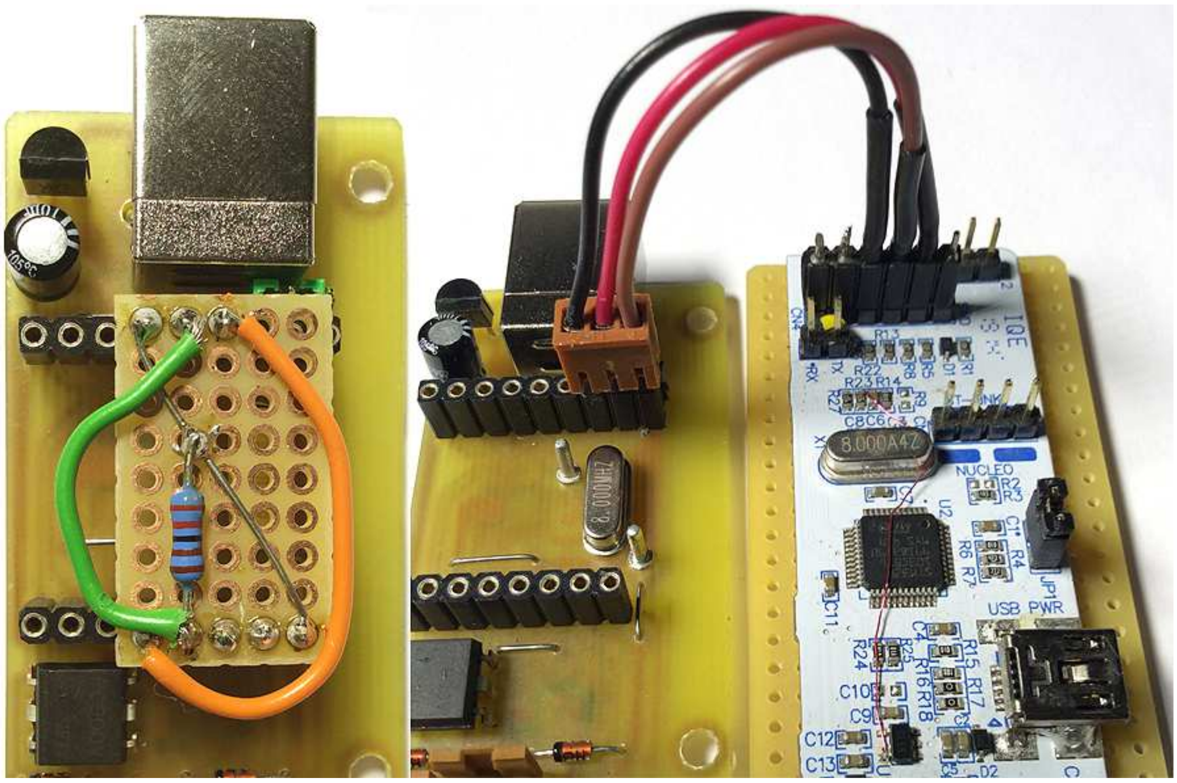
Programming

STM provides two quite useful but different programming tools.

Very comfortable programming is possible and recommended with the ST-LINK hardware and software. Instead of buying a dedicated ST-LINK module it is recommended to get a STM Nucleo-64 module, which can be configured as programmer. STM32 Nucleo-L476 (most flexible) or STM32 Nucleo-F446 (high speed) are most recommended. It is less expensive and additionally it may be used for other experiments with STM32 microcontrollers. How to program external parts with the Nucleo, read its manual. A special driver packet must be downloaded from the STM website and installed. When the programmer is used standalone and the programming software sends a problem like "no target voltage", 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.

Build a short programming adaptor (see picture below, wire length max 25 cm)**:**
--- connect the 2nd pin of the Nucleo SWD connector (counted from the side towards the Mini USB connector) with pin PA14 of our board.
--- connect the 3rd pin with Ground of our board.

--- connect the 4th pin with pin PA13 of our board.
These pins are broken out at the socket array near the USB connector, order see schematic.
--- start the ST-LINK software. Click item "Connect" of the "Target" menue while the Reset button is pressed. After some seconds, a screen with the connection report and a listing should appear. When you are connected, select "Program&Verify" from the "Target" menue and upload the hex code. After programming, remove the programming adaptor.



These photos are shot with the previous hardware version. But the procdure is the same

If no special programmer is available, the module can be programmed with the "STM FlashLoader" via the on board RS232 connector - but the ST-LINK works much better. Unfortunately the loader supports only USART1, which is not directly available because USB is mapped on these pins. The user RS-232 interface works with USART2. Following practical workaround is proposed:

First download and install the STM Flash Loader. Then build a small Veroboard as follows:
--- connect pin PB8 (red LED) with +3.3Volt (solid wire in the picture).
--- connect the TX of USART1 (= Pin PA9 = USB- pin) with TX of USART2 (PA2) (green wire in the picture). This pin has to be lifted to 3.3 Volt idle state with a pull-up resistor of about 2.2 kiloOhm (exact value is uncritical)
--- connect RX of USART1 (=pin PA10= USB+ pin) with RX of USART2 (PA3) (orange wire in the picture).

Now the device can be powered with 5 Volt via the USB connector, **but it is essential, that the USB cable is rather short and there is absolute no digital signal or other electric noise on the USB data wires!** If no USB power supply or USB power bank is available, any d.c. supply 4V up to max 5.5V can be used. Take care for correct polarity and connect it at the USB SIDE OF THE REGULATOR ! The LED on our board should light up red.

Connect the onboard RS-232 with a (real or virtual) COM port of your PC. Start the STM Flashloader software. There select the COM port number you are using and optimally 57600 Baud. Leave the other settings as proposed.

Immediately before you click "Next" at the STM Flashloader window, make a reset of our board: short the test pin in the middle of the PCB side with the Ground test pin near to the mPCB edge. When you have clicked "Next" a dialog with a green lamp symbol should appear. If so, follow the hints on the screen. Sometimes the Flashloader does not detect the microcontroller correctly, shows "STM8.." or so. Then disconnect power, restart the Flashloader and repeat the action.
If connection fails, don't resignate, check everything and repeat the action carefully. Sometimes the Flashloader is headstrong.

contact: wschemmert@t-online.de