

STM32 based USB to MIDI or RS-232 Interfaces

Operation Manual

©2021 Wolfgang Schemmert

03 December 2021

In default state (no connections at PA13,PA14, or Nucleo: both DIPSw OFF), **the board is connected to a virtual COM port at the PC.** This can be observed with the Windows Device Manager. When USB is connected the first time, the driver has to be installed. If you already have installed the "STM32 Virtual Com Port Driver" (is done together with the ST-LINK software), by default our board uses the same driver and will be listed under "COM and LPT ports" as "STMicroelectronics Virtual COM Port". Else download the "STM32 Virtual Com Port Driver" from the STM website and install it. 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.

Without further configuration, the local serial I/O works as a RS-232 interface, data format 8N1 (default 115200 baud, independent of the setting at your PC) or as a native MIDI interface (default MIDI baudrate= 31250, but may be set to any standard RS-232 baudrate).

Activate MIDI interface and input level messages:

Small PCB and Vero F042:

When a connection is made between pin PA14 and Ground (short the two pins near the USB connector), the board starts as native MIDI interface. When the USB cable is connected first, a driver gets installed automatically (Windows XP or later). In the Windows Device Manager it appears under "Audio-,Video & Game Controller" as "USB-MidiCom" or sometimes simply as "USB-Audio Device".

When this jumper is set or removed during operation, the board performs a new USB enumeration within 1-2 seconds. Unfortunately, a MIDI software or Virtual COM port on the PC has to be closed then (close it before mode change, else another reset of the interface is necessary) and restarted after.

Without a **connection between pin PA13 and Ground**, the board works as bidirectional USB to RS-232 or MIDI data interface (default). **When PA13 is connected to Ground (jumper or switch), port input change messages are merged into "transparent" data traffic.** If the RS-232 mode is active, the MIDI messages are sent as ASCII text.

No USB re-enumeration takes place when PA13 level is changed.

Replaces command "G" of earlier firmware versions.

STM32L476 Nucleo:

The DIP switch near the left board edge selects between RS-232 (open) and MIDI(closed). Else USB driver handling and enumeration is same as described above. **Trigger of port input change messages is switched ON and OFF with the other DIP switch.** This switch has the same function as PA13 described above.

To get into user setup mode (only possible during RS-232 operation), type three times a plus "+++" within max 1 second and one second no input before and after (as known from legacy modems). A prompt "CMD>" appears. Here you can enter some single character command codes which are prompted with ASCII text hints. Enter numeric input as described below. All input is case independent. Wrong input is refused with a question mark "?". Then simply repeat

the command. The interface is held quite simple, Backspace is not supported. You can **leave the setup mode with command 'E'**.

command 'V': enter your own USB Vid as a **4 digit hex number** (leading zeroes must be typed). Then a letter "P" appears and you enter your own USB Pid the same way as a 4 digit hex number. By default the STM Vid/Pid is used, **but this is only allowed for test and evaluation inside your shack! For any public use, your individual Vid/Pid must be activated!**

To install the device with your own Vid/Pid under Windows, for example you could copy and rename the file "stmcdc.inf" and change the respective entries with an ASCII text editor. Install it "manually". The new Vid/Pid **is not active before** the next USB enumeration and only if stored in Flash explicitly.

Vid/Pid can be **reset to default** by entry of Vid = 0000 and Pid= 0000.

command 'B': enter the local baud rate for **operation as virtual COM port** with its two leading characters: 96, 19, 31(=MIDI), 38, 57 or 11. The rest is inserted automatically. Default is 115200 Baud. The new baudrate **is not active before** the next USB enumeration and only if stored in Flash explicitly.

command 'M': same features as "B", but effective when **operated as USB-MIDI interface**. In this case, the default baud rate is MIDI. This baudrate is active immediately. Even standard RS-232 baudrates can be selected for cross-system communication.

command 'C': insert the general MIDI channel **1...16 in decimal format**. The MIDI channel which is inserted as low nibble into the MIDI **status byte - is always one less (0...hexF)**.

command 'H': sets the hysteresis for analog input messages. In most cases the default value=2 (Nucleo = 4) is optimum. Lower values trigger more garbage, higher values reduce precision. For 7-bit messages, the hysteresis is always 1 less (Nucleo: 2 less) (except H=0).

command 'X': sets the number base to **"hex"** for data input and message output. (default)

command 'D': sets the number base to **"decimal"** for data input and message output. Exception: because the MIDI data structure is essentially designed in hex, status bytes are always handled in hex format.

command '?': shows a list of command codes.

command '!: shows a list of actual setup (active and/or state stored in Flash)

command '~': saves the actual setup incl new changes in Flash

command '@': reloads the stored setup from Flash memory (inclusive new changes !)

command 'E': quits the setup mode back to "transparent" mode. You are asked to save new setup changes nonvolatile in the microcontroller flash memory.

Exit may be cancelled with <ESC> key

commands '1, 2, 3, 4, 5, 6' (small PCB)

or **'1, 2, 3, 4, 5, 6, 7' (F042 Vero)**

or **'1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, U' (L476 Nucleo)**

configure which MIDI message shall be sent when the corresponding microcontroller pin is connected to Ground (pushbutton) or - if configured as analog input - when the voltage changes (potentiometer between 3.3V and Ground).

Assignment of command number to port pin see below.

First, "Status" is prompted. Then enter the **MIDI status byte** you want to be sent in **2 digit hex format**. If you enter only the first nibble 9,A,B,C,D,E plus <return> or with leading zero, the preconfigured constant MIDI channel is inserted (see command 'C'). This way the MIDI channel can be changed easily for all messages with a single command. If you have entered F6,F8,FA,FB,FC,FE,FF, the input is complete now. Undefined MIDI messages, Timecode (F1,F4,F5,F9,FD) and SysEx messages (F0...F7) are not supported and prompted with "?". If you have entered 9x, Ax, Bx, Cx, Dx, Ex F2 or F3 you are asked if you want a digital or analog input state message.

If you configure a 3 byte message, next you are asked for the first MIDI data byte (**note pitch, controller number**) as a number "0"...**"7F"** (number base=hex) or "0"...**"127"** (number base=decimal).

Finally the letter "V" is prompted and you enter the last MIDI data byte of the MIDI message to be sent as a number:

If the corresponding input shall be driven by a pushbutton, enter "1" ... to "7F"(number base=hex) or **"127"**(number base=decimal). **as last data byte.** "0" is forbidden, this value is always sent when the pushbutton is released.

If the corresponding input shall be driven by a potentiometer, enter "80"(number base=hex) or **"128"**(number base=decimal). **as last data byte.** Usually the potentiometer is evaluated in 7 bit format. For MIDI **Pitch Wheel Change (status Ex)**, you can enter as special case **"81"**(number base=hex) or **"129"**(number base=decimal).. Then the potentiometer is evaluated in 12 bit format. The least 5 bits are messaged in the first data byte then as bits 2...6. Unfortunately the AD converter of this simple hardware - especially the Nucleo - is rather instable then.

Any **message can be deleted** (flash reset to empty state) if you enter the Status value 00. Then the default message will transmitted: Control Change, controller number as input no., 0x40 or 64 (sent when pushbutton pressed), 0 (sent when pushbutton released).

When the board **operates in RS-232 mode**, the triggered MIDI messages are sent in simplified, better human readable ASCII text format.

The command numbers are assigned as follows:

Analog messages (potentiometers) are supported by PA0....PA7 and PB0.

Small PCB:

1 corresponds with PA1, **2** with PA4, **3** with PA5, **4** with PA6, **5** with PA7, **6** with PB1

Vero F042:

1 corresponds with PA4, **2** with PA1, **3** with PA0, **4** with PA5, **5** with PA6,, **6** with PA7, **7** with PB1

L476 Nucleo:

1 corresponds with PA2, **2** with PA3, **3** with PB0, **4** with PA4, **5** with PA1, **6** with PA0, **7** with PA15, **8** with PB7, **9** with PB3, **10** with PB4, **11** with PB5, **12** with PA7,

13 with PA6, **14** with PA5, **15** with PB9, **16** with PB8, **'U'** with PC13 (user button)

If the unmodified Nucleo version is used, the serial communication signals are routed via PA2,PA3. In this case, messages#1 and 2 have to be deactivated.

These commands get active immediately, but are stored in Flash, too

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