MICRO CRYSTAL SWITZERLAND

User Manual

Graphical User Interface GUI for Micro Crystal's Real-Time-Clocks

using USB -- I2C-Bus Dongle

Markus Hintermann
Global Technical Marketing /
Sales Manager
at Micro Crystal

A COMPANY OF THE SWATCH GROUP

Headquarters:	Micro Crystal AG Muehlestrasse 14 CH-2540 Grenchen Switzerland	Tel. Fax Internet Email	+41 32 655 82 82 +41 32 655 82 83 www.microcrystal.com sales@microcrystal.com

Content

1. Summary

The growing number of features of Real-time Clocks can be optimally evaluated through an intuitive Graphical User Interface (GUI). To simplify the hardware setup Micro Crystal offers a kit with the USB port to I²C-Bus dongle and a type specific RTC demo board. The GUI it is the straight forward approach for communicating with the RTC and monitoring all functions including register and flag settings. The string of applied I²C-Bus commands is reflected in real time as reference for the firmware development. A dedicated mask for communication with other I²C-Bus circuits is provided in addition.

Features:

- Ready to run dongle establishing bidirectional USB to I²C-Bus translation.
- Three parallel wired I²C-Bus ports include also +5 V for supplying power to the demo board.
- USB and customized cables for the I²C-Bus are included

2. Key Functions

2.1. Hardware: USB-I²C interface dongle

The dongle is a ready to run module. It creates a virtual COM-port via the USB connection. It provides three I²C-Bus connections with optional output of 5 V for powering up the application (max 450 mA).

Power consumption:	Module / total: <50 mA / max 500 mA
I²C-Bus clock frequency:	245 Hz – 400 kHz
USB driver for Windows:	Windows XP, Windows 7, Windows 8
Size:	50 mm × 40 mm × 15 mm

2.2. Software

The software control via a GUI allows a fast setup for communicating with the Real-Time-Clock module. Aside from the detailed GUI pages for the specific Real-Time Clocks, allows communicating with any I²C-Bus device by entering directly the data hex codes.

Example: `s 64 2A p`

Nomenclature: All terms are in hexadecimal format!

`s` stands for the I²C START condition
`64` for the slave address
`2A` for the first data byte to be sent
`p` for the I²C STOP condition

3. Bridge circuit USB to I²C-Bus

3.1. Circuit diagram of the dongle

The block diagram illustrates both connections: to the PC and to the application under evaluation.

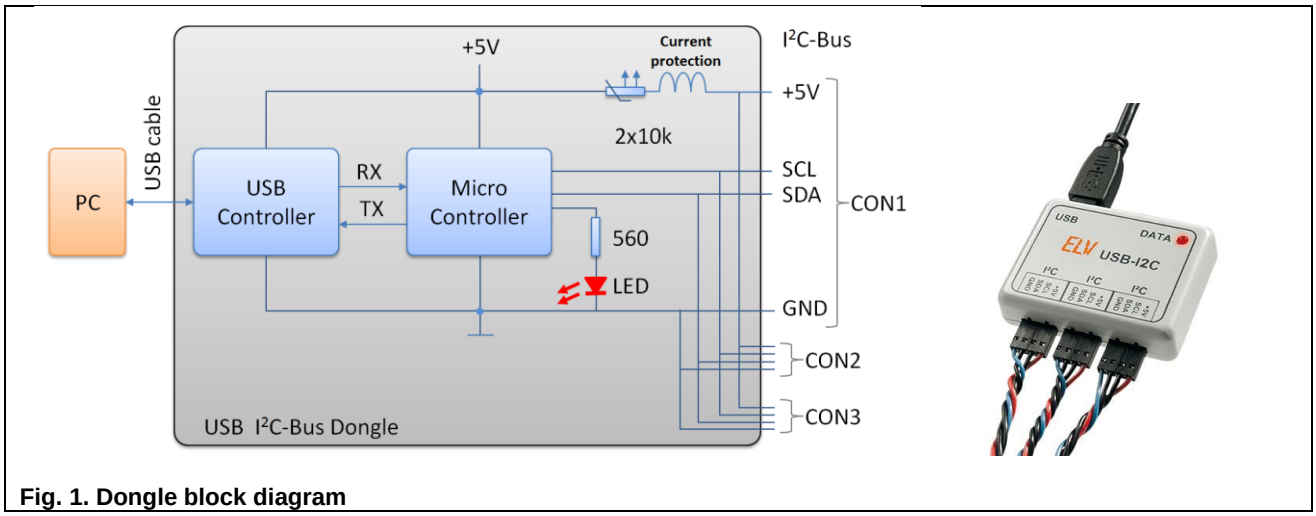

Fig. 1. Dongle block diagram

The three connectors CON1, CON2, and CON3 are wired in parallel and carry the four required signals; their order is imprinted on the package:

+5 V: Optional 5 V supply can be used. Total consumption of all 3 outputs must be kept below 450 mA.
SCL: Serial Clock line *
SDA: Serial Data line *
GND: Ground
*) : The pull-up resistors are located on the demo board

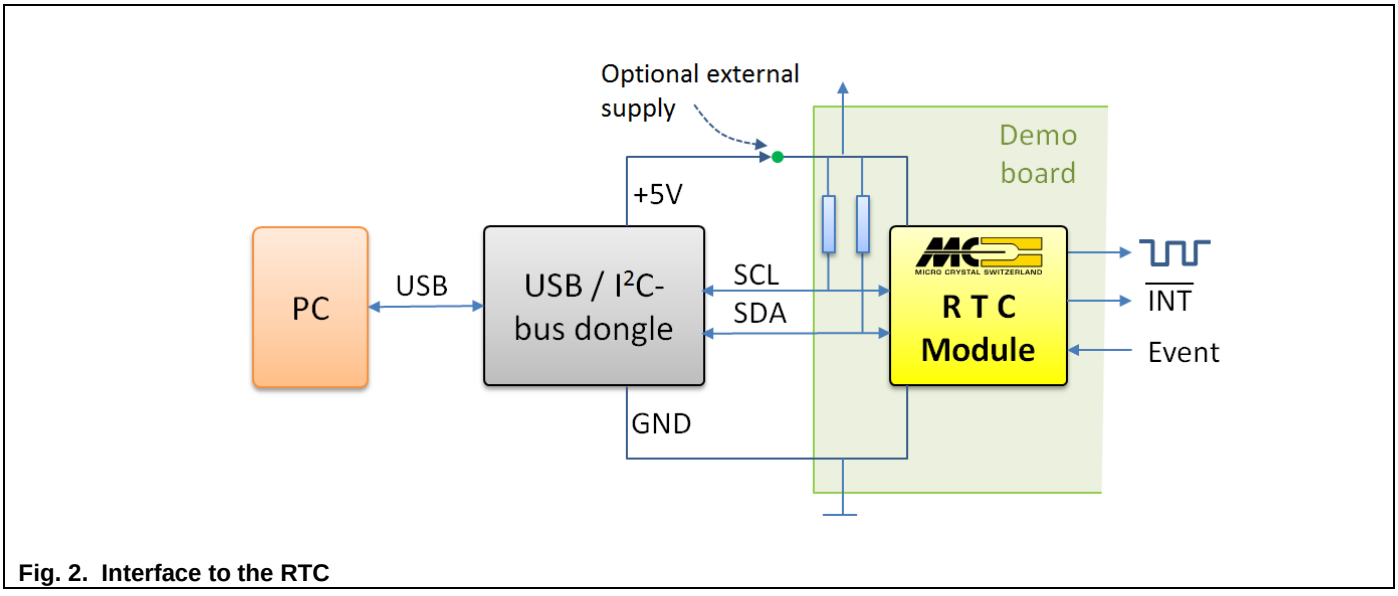

Fig. 2. Interface to the RTC

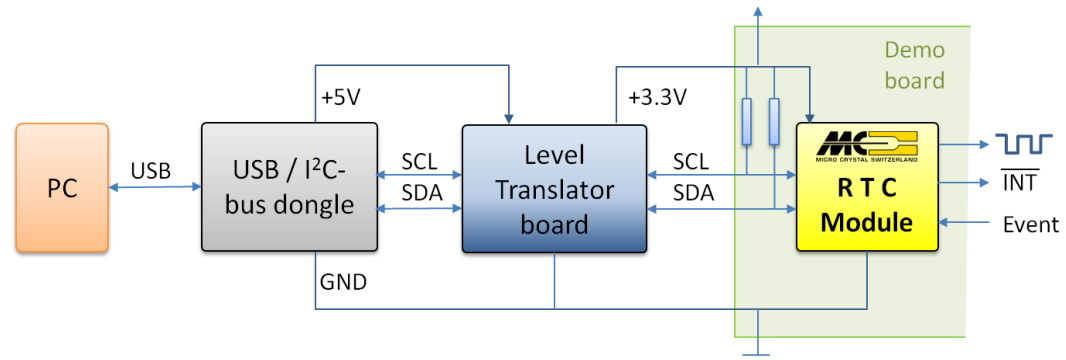

Fig. 3. Interfacing low voltage RTC

4. Installation

Use the following sequence:

4.1. Software Driver

- Install the USB software driver before connecting the interface module!
- Download the **CP210x USB to UART Bridge VCP Drivers** from Silicon Labs website: https://www.silabs.com/
- Unpack the **CP210x** Drivers zip file
- Install the driver in administrator mode, by double click.

4.2. Hardware

Connect the USB cable between the Dongle and your PC as well the as the I²C-Bus cable to your target RTC board. In case an external supply voltage is used make sure it is now turned on.
The red LED will light up indicating the dongle successfully started up. The system is now ready for starting the GUI software.

4.3. Software GUI

Installation of the GUI

a) Run the exe file: USB-I2C-RV*8803*.exe (program name *8803* is RTC specific)
b) The start window will pop up (Fig. 4)
c) Choose the appropriate com-port from the drop-down list (usually the last one on the list)
d) Press Connect: Status changes to *Connected to COMn* and turns green

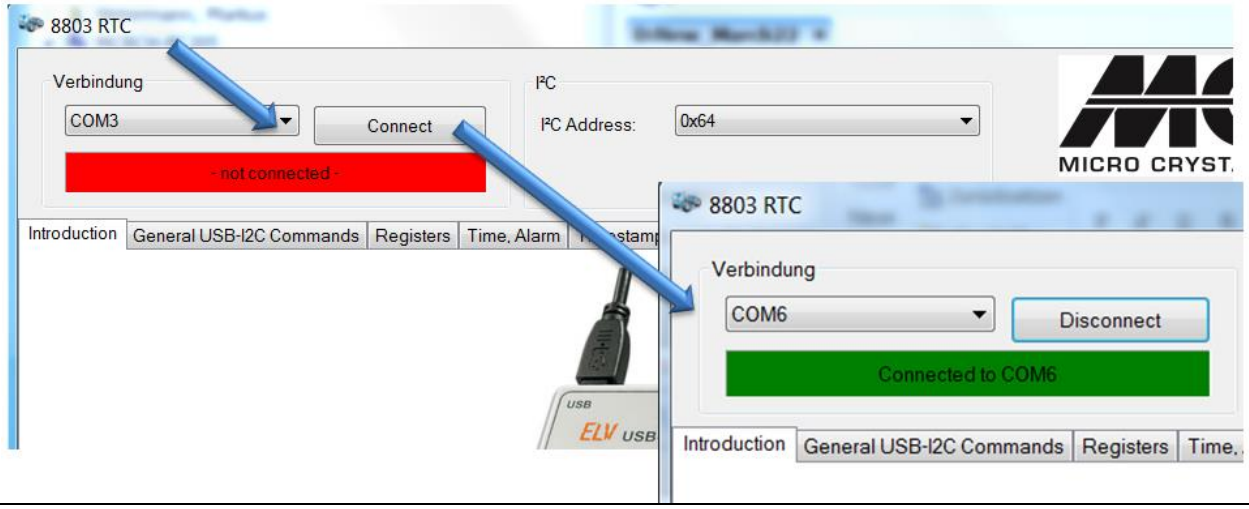

Fig. 4. Start window of the GUI

5. Operation

The Graphical User Interface is organized by a number of drop-down menus. Specific windows visualize the actual status and allow controlling the specific Micro Crystal RTC-module. In a separate menu the GUI can be used as a universal I²C-Bus interface for controlling any peripheral circuit. Details under 5.2

5.1. Procedure

The drop-down windows follow a general structure (Fig. 5 below)

1. Input fields to set e.g. the current time, to start transmission to the RTC press EXECUTE.
2. Output fields are updated after the READ button is pressed
3. Interactive knobs allow activating different functions or selecting specific parameters
4. Monitoring bus transmission: After every executed bus transmission the bus protocol is listed in the field *Transmitted Data.* Any data received by the controller is listed in the field *Received Data.*
   This is ideal for verifying the code needed for proper execution.

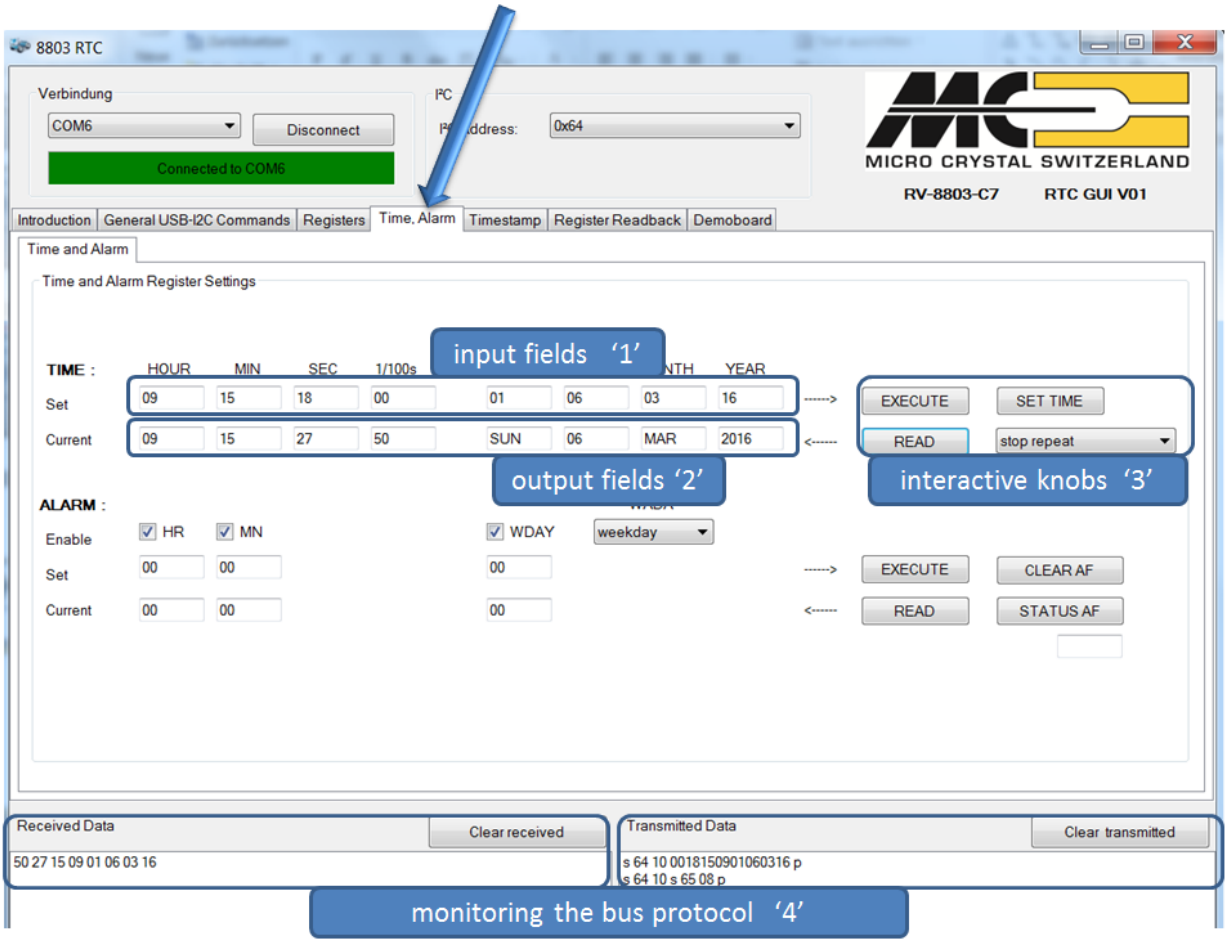

Fig. 5. Structure of the GUI, example menu for 'Time, Alarm'

5.2. General USB I²C-Bus Commands

The Universal I²C interface allows controlling any circuit by entering directly the I²C-Bus instructions.

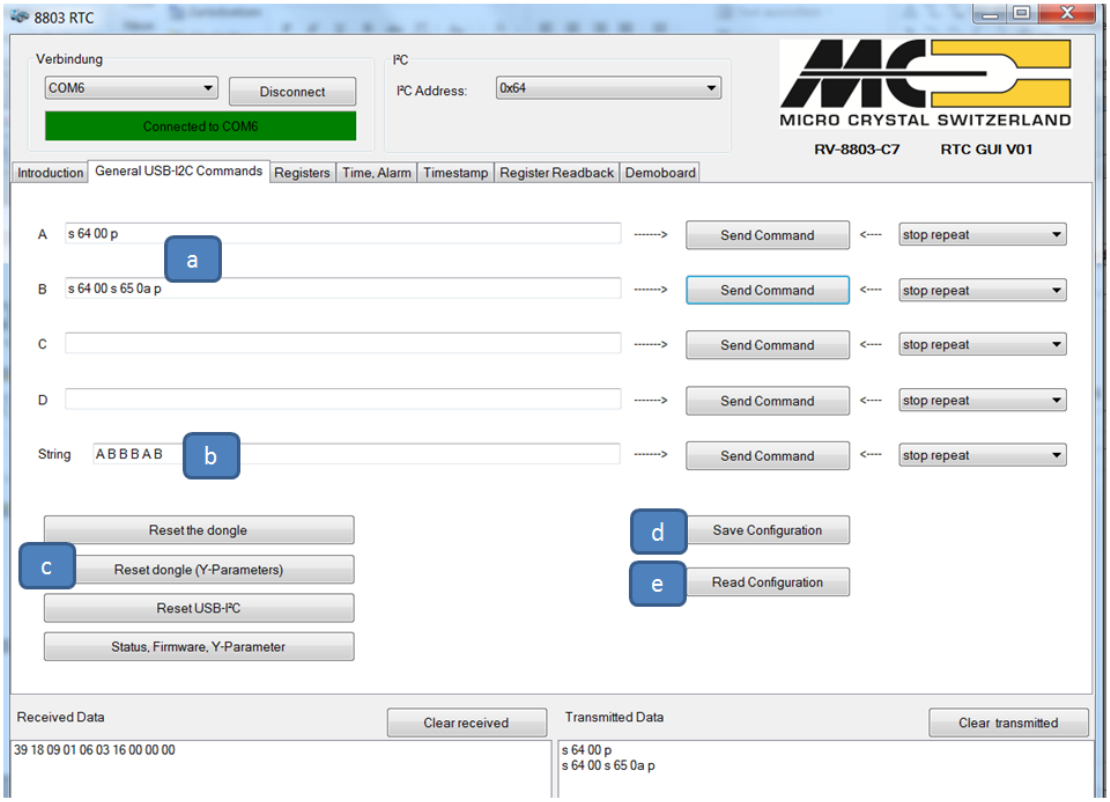

Fig. 6. General USB-I²C-Bus interface

a) There are 4 command lines A to D for preparing communication. The data format is straight forward I²C-Bus protocol:
e.g.: for sending data: `s 64 02 04 p` press *Send Command*
e.g.: for reading data: `s 64 02 s 65 0A p`: press *Send Command* setting address pointer, then reading 10 bytes (0Ah)
(`s` = START, `64` = slave address, `02 04` = data, `p` = STOP)
The command lines can be fired off once per click or repeatedly every 100ms, 1s, 2s, 10s.

b) The string allows to sequentially firing the above command lines: e.g. A B B D C. When using fast repetition rate it is of good practice to increase the I²C-Bus speed. (Use the T400000 instruction for setting the bus speed to 400kHz, details are shown below under 'Additional instructions')

c) Reset functions can be executed

d) Saving the actual configuration on the PC, including all settings on the RTC. Assign a specific name and select a directory.

e) Reloading of stored settings to continue former tasks.

Additional instructions:

^ Space is ignored, letters can be in capital or small.
Sequences like **`S6400P`** or **`S 64 00 p`** are interpreted identically

`S` I²C-Bus Start condition, also for repetitive start condition

`P` I²C-Bus-Bus STOP condition

`L n` Inserts a delay of n ms: 1 to 65535ms (n= 0001…FFFF)

`T m` Setting the I²C-Bus frequency: minimum 226Hz m= `000226`, maximum 409.6kHz m= `409600`

`?` System status and settings are read from the dongle