H.-J. Schleißheimer
Soft- und Hardware Entwicklung GmbH

Am Kalkofen 10, 61206 Wöllstadt
Fon. : 06034 / 9148 - 0
Fax : 06034 / 9148 - 880
info@schleissheimer.de

Serial & CAN Gateway

Technical Manual

Authors: Michael Rück, Felix Dombrowsky
Company: Schleißheimer Soft- und Hardwareentwicklung GmbH
Department: Research and Development

Revision History

Revision	Date	Author(s)	Modifications List
0.1	16.01.2024	M.Rück	Initial creation
0.2	13.02.2024	F.Dombrowsky	Pinout added, format changes
0.3	14.02.2024	M.Rück	Review

H.-J. Schleißheimer
Soft- und Hardware Entwicklung GmbH

Am Kalkofen 10, 61206 Wöllstadt
Fon. : 06034 / 9148 - 0
Fax : 06034 / 9148 - 880
info@schleissheimer.de

Content

1 Overview

Serial to CAN Gateway (GW) purpose is protocol and interface conversion between UART and CAN. GW contains 1 CAN interface and 8 UART interfaces.
7 of UARTs are directly available to the user on the external connector, 1 is available on the PCB test points.
CAN interface works according to CAN 2.0 standard at 1 Mbps (1 000 000 bits per second) fixed baud rate.
UART interfaces are configurable individually. UART allowed baud rate values are from 1200 to 500kbit* per second. Parity control – no, odd, even. Data bits (not including parity bit) – 7 or 8. Stop bits – 0.5; 1; 1.5; 2.
Configuration is possible only via dedicated CAN messages. Changed settings applied immediately and stored to GW's flash memory automatically. This is also true for all other GW settings.
Every UART interface has internal 8 Kbytes FIFO buffer for RX data and 1 Kbytes FIFO buffer for TX data. The buffers are necessary to match CAN and UART data rates and provide space for received over UART data when CAN Data Logger is not ready to collect CAN traffic.

* UARTs 1,2,3,7 are reliable up to 1Mbit

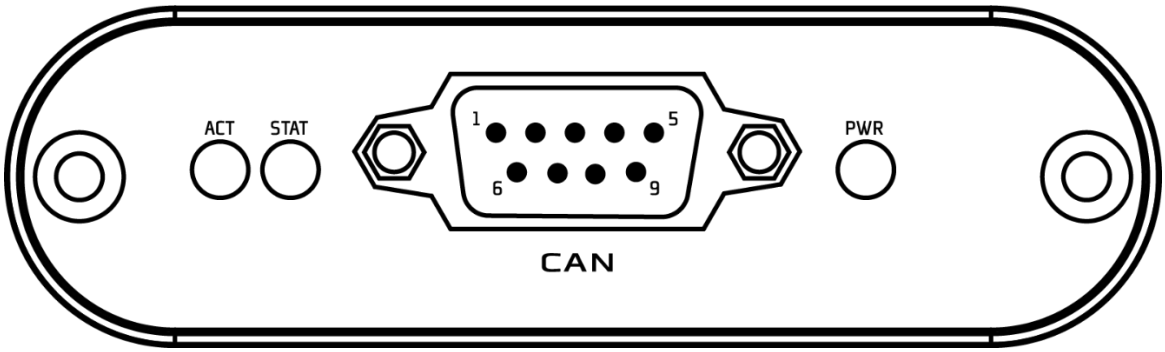
Am Kalkofen 10, 61206 Wöllstadt
Fon. : 06034 / 9148 - 0
Fax : 06034 / 9148 - 880
info@schleissheimer.de

2 Pinout

Pinout and LED identification.

CAN and power

Device connector: D-Sub 9pol male, UNC 4-40 threaded nut



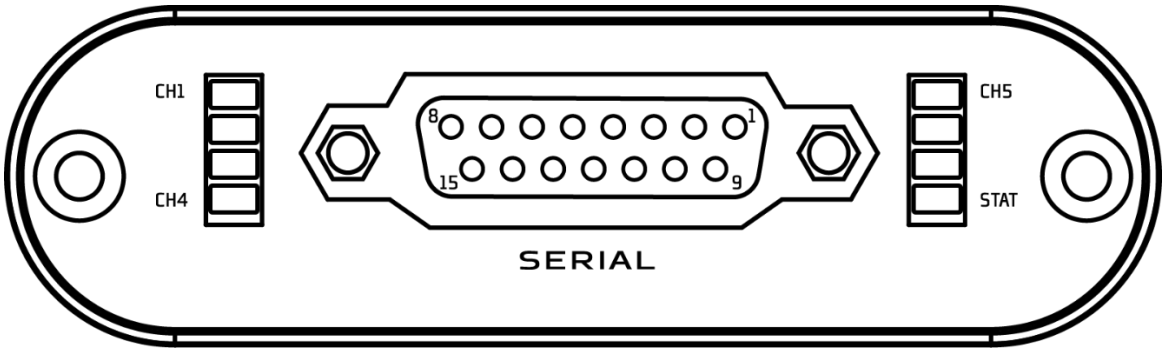
Connector J1:

#	Signal	Description
1	Wakeup	Wakeup, up to power supply voltage
2	CAN-L	CAN low
3	CAN-GND	CAN ground
4	ID1	CAN ID #1 selection
5	Shield	Shield
6	GND	Ground
7	CAN-H	CAN high
8	ID2	CAN ID #2 selection
9	5-24V	Power supply (input)

Am Kalkofen 10, 61206 Wöllstadt
Fon. : 06034 / 9148 - 0
Fax : 06034 / 9148 - 880
info@schleissheimer.de

Serial Interfaces

Device connector: D-Sub 15pol female, UNC 4-40 threaded nut



Connector J3:

#	Signal	Description
1	Rx #1	Serial input, 1.8V to 5V TTL
2	Rx #2	Serial input, 1.8V to 5V TTL
3	Rx #3	Serial input, 1.8V to 5V TTL
4	Rx #4	Serial input, 1.8V to 5V TTL
5	Rx #5	Serial input, 1.8V to 5V TTL
6	Rx #6	Serial input, 1.8V to 5V TTL
7	Rx #7	Serial input, 1.8V to 5V TTL
8	GND	Signal ground
9	Tx #1	Serial output, 3.3V TTL
10	Tx #2	Serial output, 3.3V TTL
11	Tx #3	Serial output, 3.3V TTL
12	Tx #4	Serial output, 3.3V TTL
13	Tx #5	Serial output, 3.3V TTL
14	Tx #6	Serial output, 3.3V TTL
15	Tx #7	Serial output, 3.3V TTL

H.-J. Schleißheimer
Soft- und Hardware Entwicklung GmbH

Am Kalkofen 10, 61206 Wöllstadt
Fon. : 06034 / 9148 - 0
Fax : 06034 / 9148 - 880
info@schleissheimer.de

3 CAN address setting

GW has 2 CAN ID input (J1:4 and J1:8) pins which allow the user to change address range used by the device for CAN communication. When the pin left floating it is pulled up by internal resistor, so has high logical level.
Change CAN ID offset by connecting necessary pins' combination to ground according to the table:

CanId1 (J1:4)	CanId2 (J1:8)	ID_offset
High (float)	High (float)	0x400
L(ground)	High (float)	0x300
High (float)	L(ground)	0x200
L(ground)	L(ground)	0x100

CAN ID offset applied immediately after change of pins configuration.

4 Wakeup input pin

This pin (J1:1) may be used to delay sending of CAN messages by the GW till external CAN Data Logger is not ready. When the Data Logger is ready to record CAN bus messages it should set the constant high or low logical level on this pin. Internally Wakeup pin is pulled down.
Configuration of Wakeup pin handling is possible via the dedicated CAN message.
Attention: GW will not send any CAN messages if this is prohibited by the current Wakeup pin configuration. If you don't see any CAN messages from the GW, please send the **Basic Configuration** message with **DL_ActivePinLevel** signal set to **0** to allow communications ignoring the pin level.

Am Kalkofen 10, 61206 Wöllstadt
Fon. : 06034 / 9148 - 0
Fax : 06034 / 9148 - 880
info@schleissheimer.de

5 Indication

GW has the following LEDs for status indication:

- **Power LED** - Lit constantly when the device is running.
- **Serial Activity LEDs** – one led dedicated to each of 7 externally available UART interfaces. Blinking when there is active RX or TX exchange on the corresponding port. Dark when no data exchange.
- **Serial Status LED** - Always lit green if all UART interfaces are configured successfully. Blinking while there is some low level UART configuration problem (check UART status report CAN messages for details).
- **CAN Activity LED** - Blinking on RX or TX exchange over CAN bus. OFF when no data exchange.
- **CAN Status LED** – constantly lit when no errors on the CAN bus. OFF - on fatal error, for example bus is not connected or no devices on the bus ACKing transmitted messages. Blinking when some amount of RX/TX errors detected, usually need to check the CAN bus wires or terminators.

Brightness of all LEDs can be adjusted in the range from 1 to 100% via the dedicated CAN message and signal. Changed brightness value applied immediately and automatically stored into the flash memory.

6 Firmware update

GW firmware is possible to update via the dedicated CAN messages.
For FW update need to have converted to .bin (by the normal STM32 build toolchain) firmware file.
Firmware build number can be read from this file at offset 512 (4 bytes, treat as uint32_t).
Implement and execute the next sequence:

1. Upload data from .bin file to the MCU's temporary flash memory location:
   1.1. Set write position to 0 - send message **Updater Write Position**. This will also initialize FW update sequence.
   1.2. Wait for the **Updater Status** message and verify that all error flags are not raised and write position is equal to expected.
   1.3. Write 8 bytes of firmware data using **Updater Write Data** message to the current write address. The current write position will be automatically moved forward. If firmware binary size is not proportional to 8, send last data bytes filled with 0-s.
   1.4. Go to step 1.2 and repeat until all FW data is written to the MCU.

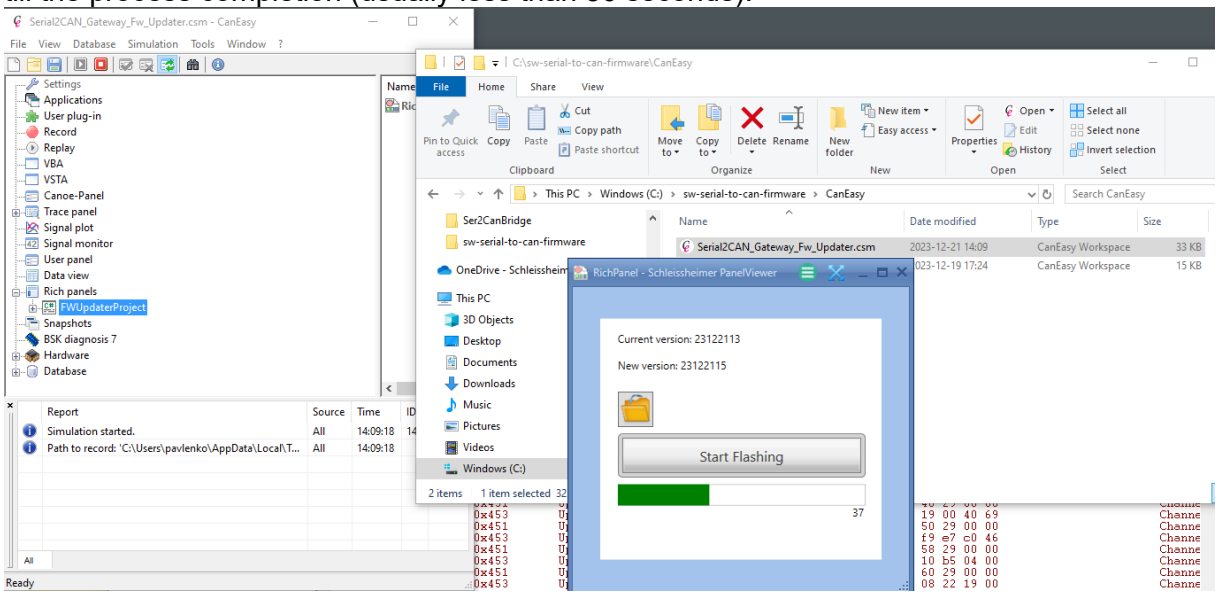
2. Finalize update - request the MCU to verify written FW data correctness, overwrite its executable code location area with the uploaded firmware data and reboot:
   2.1. Calculate CRC of the firmware data (only **FwImgSize** bytes) and send the **Updater Finalize** message.
   2.2. Wait for the next **Updater Status** message.
      2.2.1. In case of update failure check the error flags. You can try again (as many times as you want) - go to step 1.1.
      2.2.2. In case of success the device will reboot and run the new firmware, so this message will contain a new FW build number and no error flags active.

H.-J. Schleißheimer
Soft- und Hardware Entwicklung GmbH

Am Kalkofen 10, 61206 Wöllstadt
Fon. : 06034 / 9148 - 0
Fax : 06034 / 9148 - 880
info@schleissheimer.de

7 Firmware updater tool

The recommended way to update firmware it to use CanEasy Workspace and RichPanel application with already implemented before mentioned algorithm.
With this tool it is necessary to connect the GW device to the computer with installed CanEasy using one of many supported USB to CAN adapters, open the Firmware Updater Workspace, open the RichPanel, click the **Open** button and choose file to flash, then click the **Start Flashing** button and wait till the process completion (usually less than 30 seconds).



Am Kalkofen 10, 61206 Wöllstadt
Fon. : 06034 / 9148 0
Fax : 06034 / 9148- 880
info@schleissheimer.de

8 CAN messages map

CAN Speed: **1 000 000 bps**

Msg. ID	DLC	Cycle [ms]	Msg. Name	TX/RX by the Device	Description
ID_offset + 0x1	8	5000	**CAN Status**	TX	CAN transceiver status information, for diagnostics
Start Bit	Length [bit]	Signal	Unit	Valid Range	Description
0	32	HAL_ErrCode	bitset	HAL_FDCAN_Error_Code	0 if everuthing is fine. Set of error flags by HAL.
32	8	TxErrCnt	counter	0 … 255	Count of TX errors from CAN transceiver
40	8	RxErrCnt	counter	0 … 128	Count of RX errors from CAN receiver
48	8	Activity	bitset	FDCAN_communication_state	Look at FDCAN_communication_state from FDCAN_ProtocolStatusTypeDef
56	4	ProtocolErrCode	bitset	FDCAN_protocol_error_code	Look at FDCAN_protocol_error_code possible values
60	1	FDCAN_FailureFlag	flag	0: no error 1: failure	FDCAN device had failure with succesfull automatic recovery
61	1	ErrorPassiveFlag	flag	0: Error active state 1: Error passive state	FDCAN current work mode
62	1	WarningFlag	flag	0: no warnings 1: there are some issues with CAN bus	FDCAN warning status flag
63	1	RestrictedFlag	flag	0: normal work mode 1: restricted mode	FDCAN communication mode

Am Kalkofen 10, 61206 Wöllstadt
Fon. : 06034 / 9148 0
Fax : 06034 / 9148- 880
info@schleissheimer.de

Msg. ID	DLC	Cycle [ms]	Msg. Name	TX/RX by the Device	Description
ID_offset + 0x2	4	on change	**Basic Configuration**	RX	Request to change basic configuration settings
Start Bit	Length [bit]	Signal	Unit	Valid Range	Description
0	7	LedsBrightness	percent	0 … 100	Set the desired brightness of LEDs; 0 - LEDs off, 100 - maximum brightness
7	2	DL_ActivePinLevel	enum	0: always (default) 1: pin is high 2: pin is low	When the CAN Data Logger is ready (signal pin level), so the Gateway will send CAN messages on the bus.
9	23	Reserved		0	Unused area

Msg. ID	DLC	Cycle [ms]	Msg. Name	TX/RX by the Device	Description
ID_offset + 0x3	4	5000	**Basic Configuration Report**	TX	Report with current basic configuration settings
Start Bit	Length [bit]	Signal	Unit	Valid Range	Description
0	7	LedsBrightness	percent	0 … 100	Current brightness of LEDs; 0 - LEDs off, 100 - maximum brightness
7	2	DL_ActivePinLevel	enum	0: always (default) 1: pin is high 2: pin is low	When the CAN Data Logger is ready (signal pin level)
9	23	Reserved		0	Unused area

Am Kalkofen 10, 61206 Wöllstadt
Fon. : 06034 / 9148 0
Fax : 06034 / 9148- 880
info@schleissheimer.de

The device contains 8 UART transceivers, identified by the UART_index number [1 to 8]

Msg. ID	DLC	Cycle [ms]	Msg. Name	TX/RX by the Device	Description
ID_offset + 0x10 + UART_index	8	5000	**UART Status**	TX	UART status and configuration
Start Bit	Length [bit]	Signal	Unit	Valid Range	Description
0	16	RxQueueUsedBytes	counter	0 … 65535	Current fill level of UART RX data queue
16	16	TxQueueUsedBytes	counter	0 … 65535	Current fill level of UART TX data queue
32	21	SpeedBps	bits per second	UART baudrate	Configured speed of the UART transceiver, BPS
53	4	DataBits	bits count	7 or 8	Configured UART data word length in bits
57	2	StopBits	enum	0: 0.5 1: 1 2: 1.5 3: 2	Configured number of stop bits
59	2	Parity	enum	0: no parity 1: even parity 2: odd parity	Configured UART data parity control
61	1	ConfigErrFlag	flag	0: no error 1: error	Last configuration request result
62	1	RxQueueOvfFlag	flag	0: no overflow 1: overflow	is RX queue full (CAN bus is not fast enough to send all UART RX data)
63	1	TxQueueOvfFlag	flag	0: no overflow 1: overflow	is TX queue has no free space (CAN data arriving faster than UART able to send them)

Am Kalkofen 10, 61206 Wöllstadt
Fon. : 06034 / 9148 0
Fax : 06034 / 9148- 880
info@schleissheimer.de

Msg. ID	DLC	Cycle [ms]	Msg. Name	TX/RX by the Device	Description
ID_offset + 0x20 + UART_index	4	on config change request	**UART Configuration**	RX	UART configuration request
Start Bit	Length [bit]	Signal	Unit	Valid Range	Description
0	21	SpeedBps	bits per second	maximum is 2 Mbps	Required speed of the UART transceiver, BPS
21	4	DataBits	bits count	7 or 8	Required UART data word length in bits
25	2	StopBits	enum	0: 0.5 1: 1 2: 1.5 3: 2	Required number of stop bits
27	2	Parity	enum	0: no parity 1: even parity 2: odd parity	Required UART data parity control
29	3	Reserved		0	Unused area

Am Kalkofen 10, 61206 Wöllstadt
Fon. : 06034 / 9148 0
Fax : 06034 / 9148- 880
info@schleissheimer.de

Msg. ID	DLC	Cycle [ms]	Msg. Name	TX/RX by the Device	Description
ID_offset + 0x30 + UART_index	1 … 8	on UART data to send	**UART TX data**	RX	Data required to send via the UART interface
Start Bit	Length [bit]	Signal	Unit	Valid Range	Description
0	8 … 64	TX_Data	bytes	1 to 8 data bytes	Data bytes requested to be transmitted via the UART interface

Msg. ID	DLC	Cycle [ms]	Msg. Name	TX/RX by the Device	Description
ID_offset + 0x40 + UART_index	1 … 8	on UART data received	**UART RX data**	TX	Data received on the UART interface
Start Bit	Length [bit]	Signal	Unit	Valid Range	Description
0	8 … 64	RX_Data	bytes	1 to 8 data bytes	Data bytes received by the UART interface from the UART bus

Am Kalkofen 10, 61206 Wöllstadt
Fon. : 06034 / 9148 0
Fax : 06034 / 9148- 880
info@schleissheimer.de

Device firmware is possible to update via CAN

Msg. ID	DLC	Cycle [ms]	Msg. Name	TX/RX by the Device	Description
ID_offset + 0x51	8	5000 and on change	**Updater Status**	TX	FW Updater actual status
Start Bit	Length [bit]	Signal	Unit	Valid Range	Description
0	32	FwBuildNum	build number	unsigned 32-bit integer	Running firmware version information as decimal number: YYMMDDbb, YY - 2 digits of year, MM - month, DD - day of month, bb - daily build number, e.g. 23120802
32	17	FwWritePos	offset in bytes	0 … lengh of firmware binary	Current write position for firmware image data
49	1	FwUpdateFailedFlag	flag	0: no error 1: error	Last firmware update finalization result
50	1	FwCrcErrFlag	flag	0: no error 1: error	Last FW update was impossible due to FW image data CRC error
51	1	FwWriteErrFlag	flag	0: no error 1: error	Last FW update request failed because of flash memory writing error
52	12	Reserved		0	Unused area

Am Kalkofen 10, 61206 Wöllstadt
Fon. : 06034 / 9148 0
Fax : 06034 / 9148- 880
info@schleissheimer.de

Msg. ID	DLC	Cycle [ms]	Msg. Name	TX/RX by the Device	Description
ID_offset + 0x52	4	on write offset adjustment	**Updater Write Position**	RX	Request to set FW write offset pointer
Start Bit	Length [bit]	Signal	Unit	Valid Range	Description
0	17	FwWritePos	offset in bytes	0 … lengh of firmware binary	Change to this value current write position of firmware image data
17	15	Reserved		0	Unused area

Msg. ID	DLC	Cycle [ms]	Msg. Name	TX/RX by the Device	Description
ID_offset + 0x53	8	on next FW data to write	**Updater Write Data**	RX	Request to write FW image data
Start Bit	Length [bit]	Signal	Unit	Valid Range	Description
0	64	FwData	data bytes	8 bytes of binary data	Firmware image data bytes to write at the current write position

Msg. ID	DLC	Cycle [ms]	Msg. Name	TX/RX by the Device	Description
ID_offset + 0x54	8	on update completion	**Updater Finalize**	RX	Request to finalize update process (after all FW data has been sent)
Start Bit	Length [bit]	Signal	Unit	Valid Range	Description
0	32	FwImgCRC	FW data CRC	32-bit unsigned integer	Control check sum of the firmware data (simple arithmetic sum of values of all data bytes accumulated to uint32_t with overflow)
32	17	FwImgSize	size in bytes	lengh of firmware binary file	Size of firmware image data file (bytes)
49	15	Reserved		0	Unused area