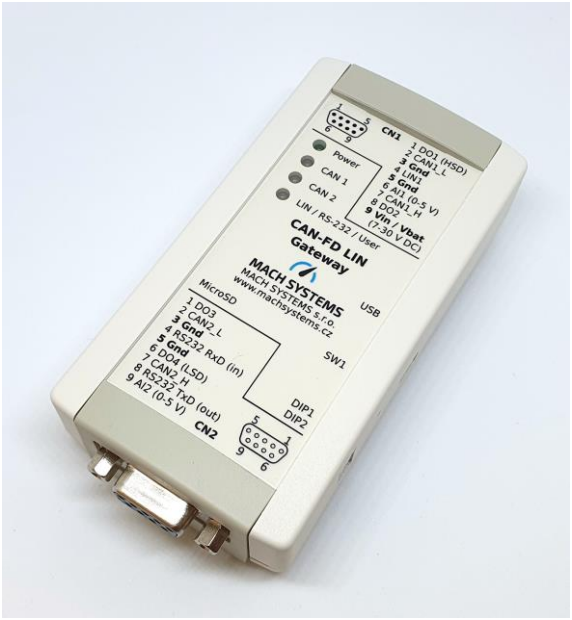
CAN FD LIN Gateway

User Manual



CHANGES

Date	Description	Created By	Review By
25.1.2021	Initial Release	VB, KH, MM	MM

Contents

1 About 4
2 Introduction 4
2.1 Features 5
3 Technical Specification 5
4 Device Description 7
4.1 Overview 7
4.2 Power 7
4.3 Pinout 8
4.3.1 DSUB connectors 8
4.3.2 Switches 8
4.4 CAN Bus Termination 9
4.5 Inputs and Outputs 9
4.6 USB 9
4.7 Galvanic Isolation 9
5 Usage 10
5.1 Example Project 10
5.2 System Bootloader 10
5.2.1 Bootloader Mode 10
5.2.2 Firmware Download 11
5.3 In-Circuit Serial Programming 13
6 Legal Information 15
6.1 Usage Warning 15
6.2 Disposal and Recycling Information 15
6.3 Declaration of Conformity 16
6.4 Patents, Copyrights and Trademarks 17
7 References 17
8 Ordering Information 18
9 Contact 18

List of Tables

Table 1 Technical Specification 6
Table 2 Connector 1 - Pin Assignment 8
Table 3 Connector 2 - Pin Assignment 8
Table 4 Switches 8
Table 5 Inputs and Outputs 9
Table 5 Ordering Numbers 18

List of Figures

Figure 1 CAN FD LIN Gateway 4
Figure 2 Block Diagram 4
Figure 3 - Product Photos 6
Figure 4 Device Overview 7
Figure 5 Power Options 7
Figure 7 CAN Bus Termination 9
Figure 7 J5 - ST-LINK SWD Connection and Pinout 14
Figure 8 STM32CubeIDE Debugger Configuration 14

1 About

This document describes the use of the **CAN FD LIN Gateway**.

P/N: CANFD-LIN-GW

Web: https://www.machsystems.cz/en/products/embedded-networking/gateways-and-bus-converters/canfd-lin-gateway

Figure 1 CAN FD LIN Gateway

2 Introduction

The **CAN FD LIN Gateway** is a freely programmable router/data-logger/simulator that features two CAN FD channels, a LIN channel, and a RS-232 port. The interface also offers a microSD card slot and multiple digital/analogue inputs and outputs, which makes it suitable for a broad range of use-cases such as protocol conversion, network bridging, data logging, rest-bus simulation, and external peripheral control and monitoring.

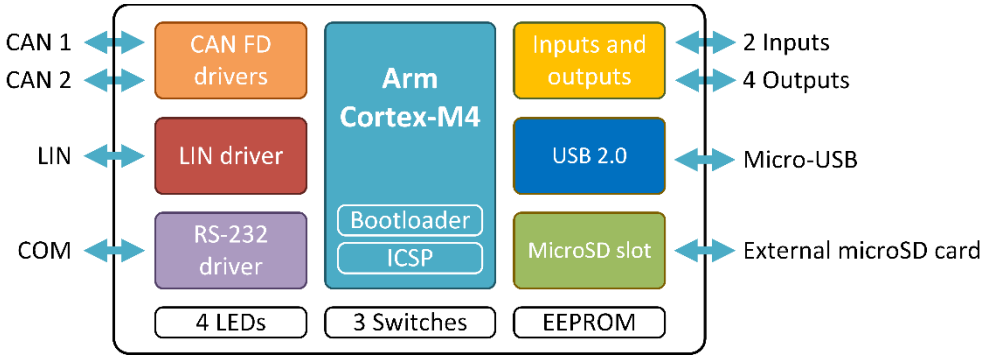


Figure 2 Block Diagram

Firmware can be developed in C/C++ and can be transferred into the device over USB, CAN, RS-232, or a standard ICSP SWD interface, which also offers code debugging. The device is based on a STM32G4 Arm Cortex-M4 MCU and comes with a free-of-charge IDE, GNU C/C++ compiler, and programming examples.

The on-board EEPROM memory can store user's application parameters, and the microSD card slot enables the user to load or save large data sets for simulations and data-logging purposes.

The four digital outputs (PWM capable) and the two analogue/digital inputs allow for both input and output triggering. The inputs can read 0-5 V analogue signals, and the outputs offer various output stages (push-pull, HSD, LSD) with currents up to 1.5 A enabling to easily control relays, valves, and other peripherals.

2.1 Features

- Two high-speed CAN channels with CAN FD support
- LIN channel
- RS-232 port
- MicroSD card slot
- 4 digital outputs
- 2 analogue/digital inputs
- 4 status LEDs
- 32-bit Arm Cortex-M4 MCU
- Freely programmable in C/C++ language
- Free-of-charge IDE and C/C++ compiler
- Programming examples available
- Firmware upload over USB, CAN, RS-232 or ICSP
- On-board 16 Kbit EEPROM
- Externally or USB-powered
- Table-top use or DIN-rail mount

3 Technical Specification

Communication and Peripherals	
Channels	2 CAN-HS (ISO 11898-2) with CAN FD support (ISO 11898-1:2015; CAN 2.0A/B, ISO CAN FD) 1 LIN bus (supports both master and slave; ISO 17987; LIN 2.2a) 1 RS-232 1 Virtual COM port (USB 2.0 CDC)
Inputs	2 Analogue/digital inputs (0-5 V)
Outputs	4 Digital outputs (PWM capable) DO1: HSD (5 V, max. 0.5 A) DO2, DO3: push-pull (5 V, max. 0.5 A) DO4: LSD (max. 40 V, 1.5 A)
Programming	Free-of-charge IDE and GNU C/C++ compiler (STM32CubeIDE) Programming examples available
Firmware update	over USB, CAN, RS-232, or ICSP (ST-LINK)
Debugging	ST-LINK SWD (a programming header needed)
Non-volatile memory	Internal 16 Kbit EEPROM External microSD card slot (a card is not part of delivery)
LEDs	3 Dual-color LED 1 Power LED

Electrical	
Power	External 7 - 30 V DC with polarity protection over DSUB connector USB-powered over Micro-USB (not for LIN bus)
Consumption	100 mA @ 12 V (approx. 1 W) *Note: When no digital output (DO1-DO4) is being driven.*
MCU	STM32G483 (Arm® 32-bit Cortex®-M4) with DSP and FPU; 170 MHz, 512 KB Flash, 128 KB SRAM
Transceivers	CAN-FD: MCP2562FD LIN: MCP2003B
Mechanical	
Connectors	1 D-SUB9M 1 D-SUB9F 1 MicroSD slot 1 Micro-USB
Buttons and switches	2 DIP switches 1 Tactile switch
Dimensions (L x W x H)	108 x 54 x 30 mm
Weight	85 g
Operating temperature	-20 to 70 °C
Protection	IP20
Placement	Table (adhesive pads included) DIN-rail mount (clip sold separately)

Table 1 Technical Specification

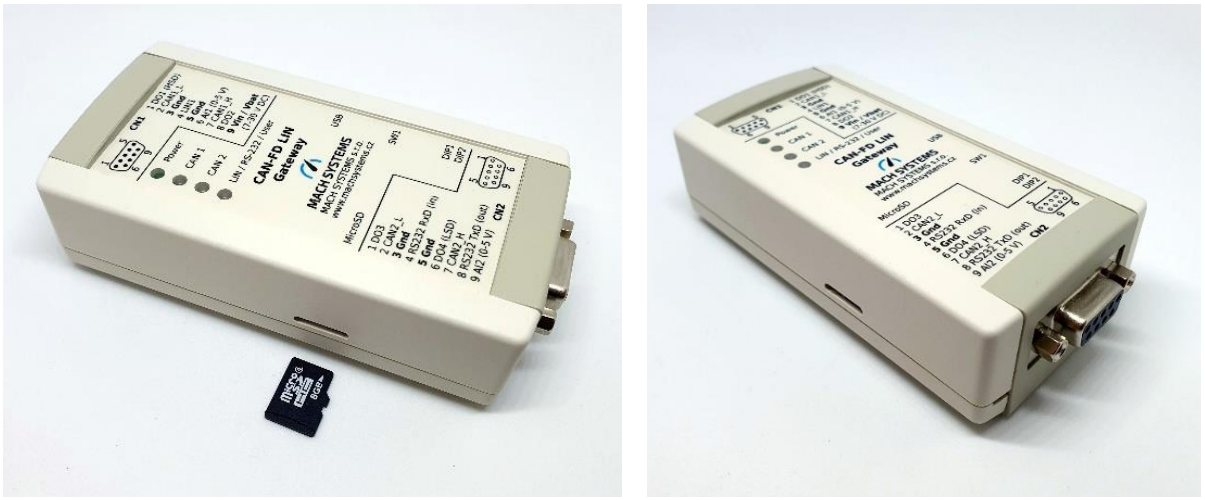


Figure 3 - Product Photos

4 Device Description

4.1 Overview

The converter has four connectors, four LEDs, two DIP switches, MicroSD card slot and a reset button.

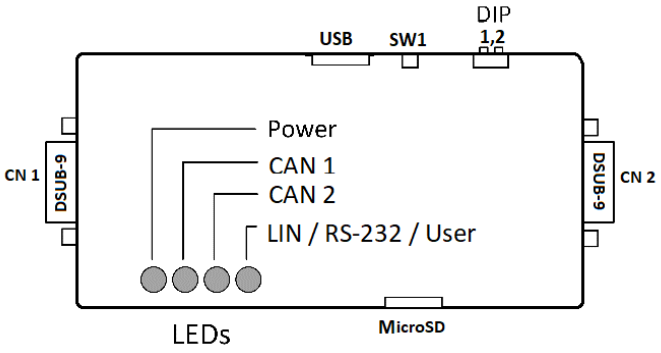


Figure 4 Device Overview

4.2 Power

The gateway can be powered externally via a DSUB connector or via a micro-USB connector. LIN bus needs the external power. All Ground signals are connected.

When the external power is connected, the power from USB gets internally disconnected by a MOSFET transistor. Thus, no power will be drawn from USB.

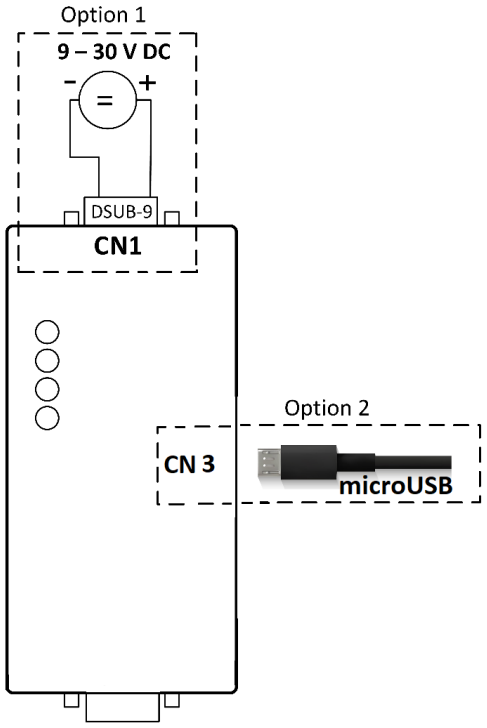


Figure 5 Power Options

4.3 Pinout

4.3.1 DSUB connectors

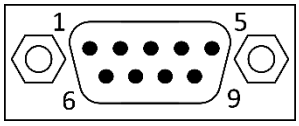
CN1 (DSUB9M)	Pin	Name	Note
 Front view	1	DO1	5 V HSD
	2	CAN1_L	
	3	Gnd	
	4	LIN1	
	5	Gnd	
	6	AI1 / DI1	0-5 V
	7	CAN1_H	
	8	DO2	5V push-pull
	9	Vin / Vbat	Power input, also used for LIN bus

Table 2 Connector 1 - Pin Assignment

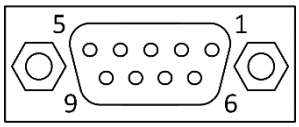
CN2 (DSUB9F)	Pin	Name	Note
 Front view	1	DO3	5V push-pull
	2	CAN2_L	
	3	Gnd	
	4	RS-232 RxD	In
	5	Gnd	
	6	DO4	LSD
	7	CAN2_H	
	8	RS-232 TxD	Out
	9	AI2 / DI2	0-5 V

Table 3 Connector 2 - Pin Assignment

4.3.2 Switches

DIP1, DIP2, and SW1 can be used in a user's application. The SW1 button can be pressed by tweezers or a pencil.

As of HW version 1.4, SW1 can be used to enter the system bootloader (see 5.2).

HW version prior up to 1.3 (incl.): SW1 is connected to PC13

HW version as of 1.4: SW1 is connected to PB8

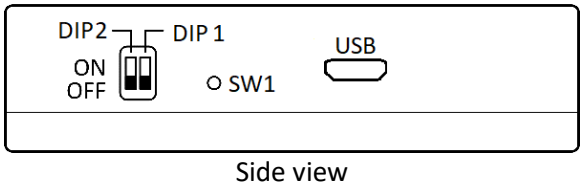
DIP switches and SW1
 Side view

Table 4 Switches

4.4 CAN Bus Termination

There are no internal termination resistors on either CAN channel inside the device. Therefore, a proper termination of the CAN bus is needed, and the user has to make sure the CAN bus is properly terminated at both ends.

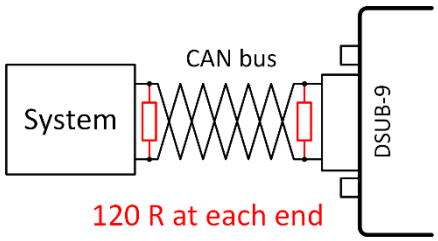


Figure 6 CAN Bus Termination

4.5 Inputs and Outputs

The device features two analogue/digital inputs and four digital outputs (PWM capable).

Signal	Direction	Function	Range	Note
AI1	Input	Analogue / Digital Input	0 - 5 V	
AI2	Input	Analogue / Digital Input	0 - 5 V	
DO1	Output	HSD Output	5 V, max. 0.5 A	PWM capable
DO2	Output	Push-pull Digital Output	5 V, max. 0.5 A	PWM capable
DO3	Output	Push-pull Digital Output	5 V, max. 0.5 A	PWM capable
DO4	Output	LSD Output	max. 40 V, 1.5 A	PWM capable

Table 5 Inputs and Outputs

A load connected to a digital output pins DO1-DO3 may draw significant current. In case digital outputs are used in a user application, it is strongly suggested to power the device from **an external power supply** and not over USB. Otherwise, USB power limits might be exceeded.

4.6 USB

Micro-USB connector uses the standard pinout, and can be used for firmware upload or in application as a virtual COM port.

4.7 Galvanic Isolation

The device does **not** have any galvanic isolation. The user has to make sure there are no ground loop in his setup.

5 Usage

The device's firmware can be fully developed by the user, and the user has a full control over the device's peripherals.

The device can be used as CAN FD to CAN bridge, CAN to LIN gateway, CAN/LIN to RS-232, data logger, ECU simulator, communication simulator or for remote monitoring inputs and remote control of outputs.

The user can make use of the system bootloader which allows for firmware programming over USB and RS-232. For flashing over CAN bus, please refer to [1].

5.1 Example Project

Example STM32CubeIDE project in C language is available free-of-charge. Please refer to [2].

The corresponding .ioc file for STM32CubeMx is also available from there.

This example project **enters the system bootloader** when this CAN message is received over either CAN channel (see 5.2.1):

Baud Rate = 500K
CAN Id = 0x1FFFFFFF (Ext. Id)
DLC=4
Data Bytes = 0x00 0x01 0x02 0x03

It is also strongly suggested to keep similar logic to enter the system bootloader programmatically.

5.2 System Bootloader

The STM32G4 MCU contains a system bootloader which is pre-programmed in ROM during manufacture. The system bootloader supports USB and RS-232, it does not support flashing over CAN bus. If the possibility to upload a firmware over CAN bus is needed, the OpenBLT bootloader described in [3] can be used.

It should be noted that when the device enters the system bootloader whilst USB is connected, the device can then be flashed over USB **only**. If the user wants to flash the device over RS-232, he has to either power the device over external power pins (see Option 1 in 4.2) or he has to make sure USB data lines are not connected. This is the limitation of the system bootloader.

5.2.1 Bootloader Mode

There are two ways to enter the system bootloader:

A. Programmatically from application

The user can call *jumpToBootloader*() function whenever he wants to enter the system bootloader. Do not to call the function directly from an interrupt routine. It is suggested to enable at least one way to programmatically enter the system bootloader.

An example how to call from CAN RX interrupt (taken from source code example in Chapter 5.1):

```
/* It is strongly suggested to keep the possibility to
jump to System Booloader from application */
```

```
    if (RxHeader.Identifier==0x1fffffff && datalen == 4
&& RxData[0] == 0 && RxData[1] == 1 && RxData[2] == 2 &&
RxData[3] == 3)
    {
      /* Cannot go to bootloader directly from ISR */
      bootloaderRequest = 1;
      return;
    }
```

In main while loop:

```
if (bootloaderRequest)
jumpToBootloader();
```

B. By hardware

For hardware versions **lower than v1.3 (incl.)**, it is necessary to open the enclosure in order to access the boot-enable pads located on the top layer of the PCB:

- Disconnect the USB and the external power supply
- Open the enclosure
- Short the boot pads together (a pair of tweezers can be used)
- Connect the power supply - either USB or external
- The device will enter the system bootloader
- Release the boot pads
- Firmware can be flashed as described in 5.2.2
- Close the enclosure

For hardware versions **as of v1.4**, SW1 switch can be used to enter the system bootloader on power-up:

- Disconnect the USB and the external power supply
- Press and hold SW1
- Connect the power supply - either USB or external
- The device will enter the system bootloader
- Release SW1

- Firmware can be flashed as described in 5.2.2

5.2.2 Firmware Download

The STM32CubeProgrammer application can be used for flashing firmware over the system bootloader. The application is available from [4].

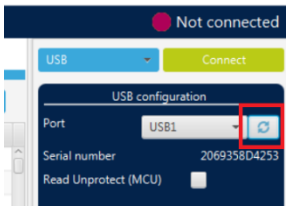
Note: If the device is powered over USB, the system bootloader allows flashing over USB only. In case it is desired to flash the device over, say, RS-232, the external power has to be supplied to the device and the USB shall remain disconnected.

Steps for uploading a firmware:

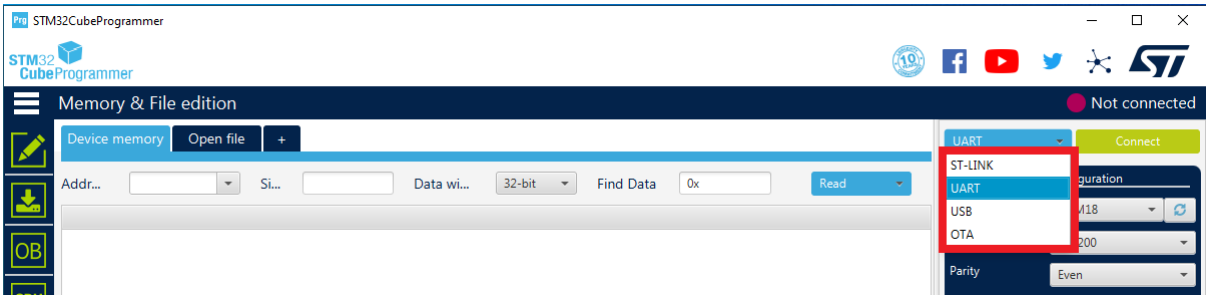
1. Open *STM32CubeProgrammer* application (see download link above)
2. Turn the device off
3. Enter the system bootloader as described in 5.2.1
4. Connect one of the following ports to the PC:
    a. Micro USB cable (this also powers the device)

b. UART (using USB to TTL UART converter) to the debug header X2
c. RS-232 over pins 4, 8 and 5 (RxD, TxD and Gnd, respectively)

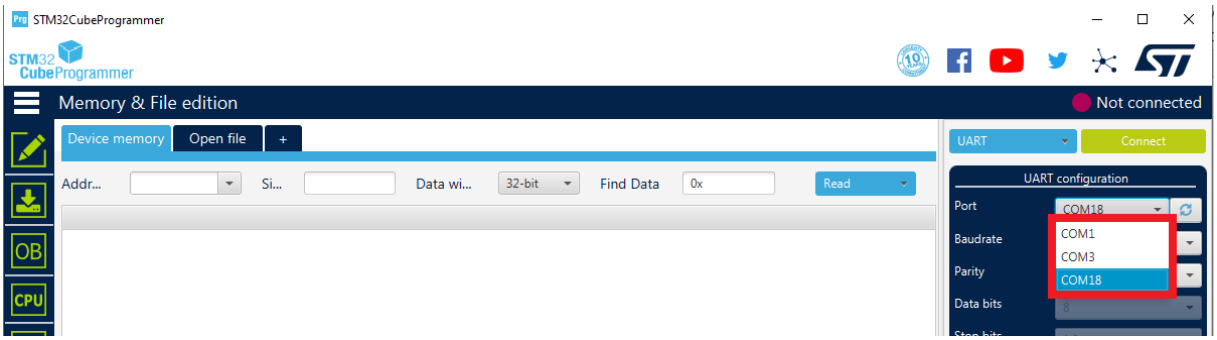
5. Further, the steps are very similar for all the ports. In the *STM32CubeProgrammer*:
   a. Click on the refresh arrows button to see available ports



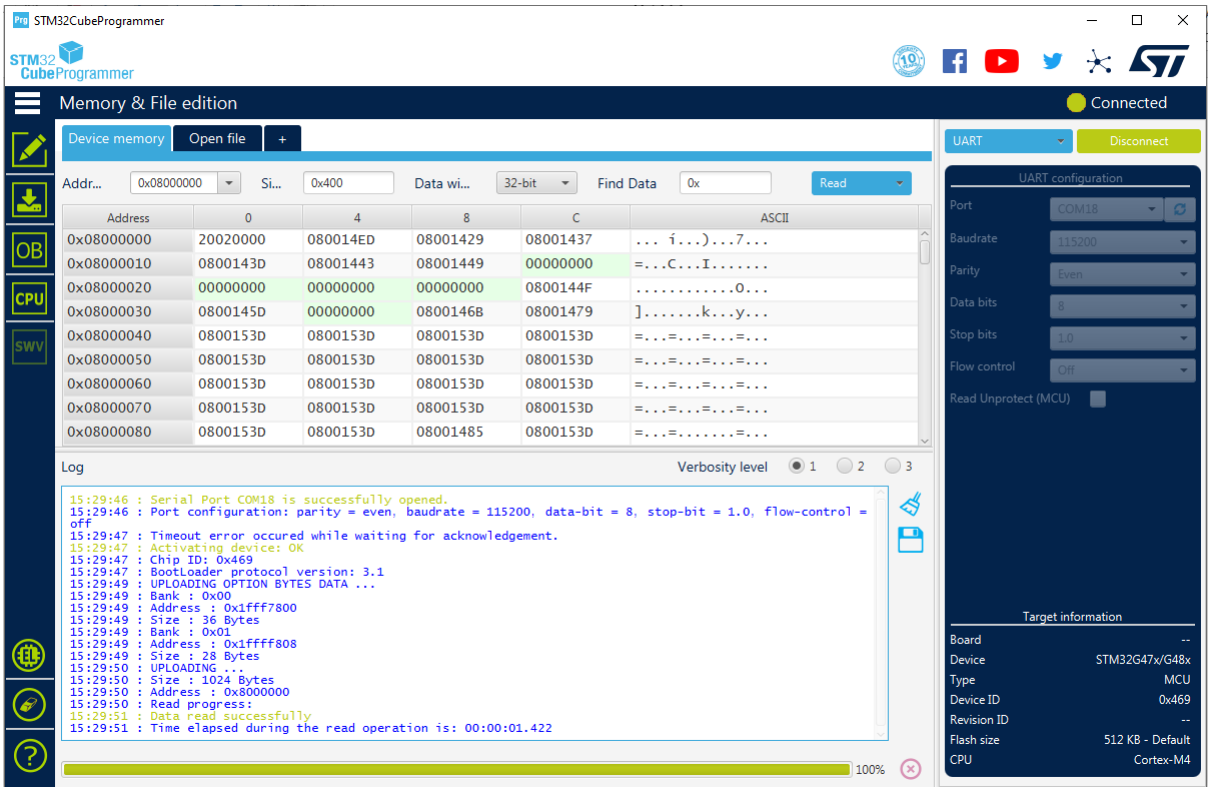
   b. Select the interface (USB or UART)



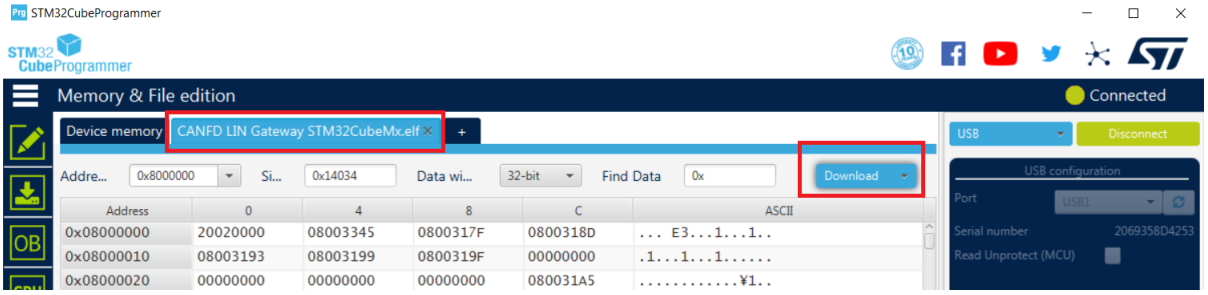
   c. Select the correct port



   d. You do not have to change the UART settings provided that the baud rate is lower or equal to 115200. Procedure of connecting to the USB is almost identical.
   e. Click connect. You will see screen similar to this one.



6. If everything went well, you are now connected to the bootloader. You can read/write memory, load hex files and edit Option Bytes. You cannot view MCU core window and Serial view window; this is possible only when debugging with connected ST-LINK.
7. In order to flash a new firmware, open an .elf file by „Open file“ button and press „Download“ button. A .hex file can be used also but the Address of 0x8000000 has to be chosen manually.



If you want to exit the bootloader, you must restart the device.

5.3 In-Circuit Serial Programming

The ST’s ST-LINK SWD connection can be used for both programming and debugging the code directly on the device. The device’s enclosure will have to be open in order to access ICSP pads.

The SWD signals are available over J5 as depicted in Figure 7.

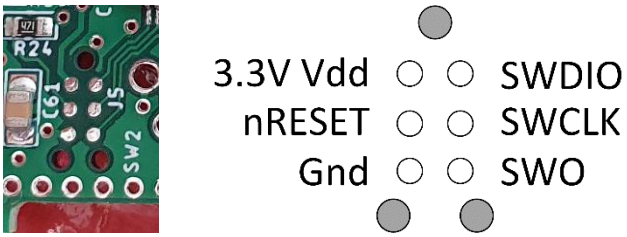


Figure 7 J5 - ST-LINK SWD Connection and Pinout

An ST-LINK v2 or v3 debugger and a TagConnect TC2030-NL header [1] is needed.
p/n: TC2030-CTX-NL-STDC14

Please make sure SWD Interface is selected in STM32CubeIDE project configuration as shown in Figure 8.

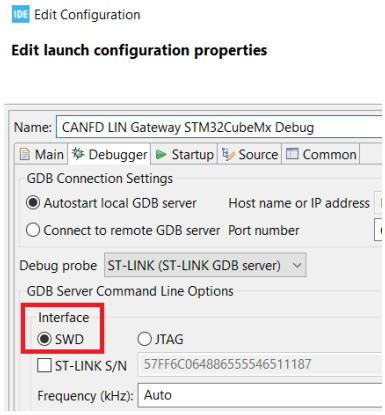


Figure 8 STM32CubeIDE Debugger Configuration

6 Legal Information

6.1 Usage Warning

WARNING FOR ALL USERS

WARNING! - YOUR USE OF THIS DEVICE MUST BE DONE WITH CAUTION AND A FULL UNDERSTANDING OF THE RISKS!

THIS WARNING IS PRESENTED TO INFORM YOU THAT THE OPERATION OF THIS DEVICE MAY BE DANGEROUS. YOUR ACTIONS CAN INFLUENCE THE BEHAVIOR OF A DISTRIBUTED EMBEDDED SYSTEM, AND DEPENDING ON THE APPLICATION, THE CONSEQUENCES OF YOUR IMPROPER ACTIONS COULD CAUSE SERIOUS OPERATIONAL MALFUNCTION, LOSS OF INFORMATION, DAMAGE TO EQUIPMENT, AND PHYSICAL INJURY TO YOURSELF AND OTHERS. A POTENTIALLY HAZARDOUS OPERATING CONDITION IS PRESENT WHEN THE FOLLOWING TWO CONDITIONS ARE CONCURRENTLY TRUE: THE PRODUCT IS PHYSICALLY INTERCONNECTED TO A REAL DISTRIBUTED EMBEDDED SYSTEM; AND THE FUNCTIONS AND OPERATIONS OF THE REAL DISTRIBUTED EMBEDDED SYSTEM ARE CONTROLLABLE OR INFLUENCED BY THE USE OF THE CAN NETWORK. A POTENTIALLY HAZARDOUS OPERATING CONDITION MAY RESULT FROM THE ACTIVITY OR NON-ACTIVITY OF SOME DISTRIBUTED EMBEDDED SYSTEM FUNCTIONS AND OPERATIONS, WHICH MAY RESULT IN SERIOUS PHYSICAL HARM OR DEATH OR CAUSE DAMAGE TO EQUIPMENT, DEVICES, OR THE SURROUNDING ENVIRONMENT.

WITH THIS DEVICE, YOU MAY POTENTIALLY:

- CAUSE A CHANGE IN THE OPERATION OF THE SYSTEM, MODULE, DEVICE, CIRCUIT, OR OUTPUT.
- TURN ON OR ACTIVATE A MODULE, DEVICE, CIRCUIT, OUTPUT, OR FUNCTION.
- TURN OFF OR DEACTIVATE A MODULE, DEVICE, CIRCUIT, OUTPUT, OR FUNCTION.
- INHIBIT, TURN OFF, OR DEACTIVATE NORMAL OPERATION.
- MODIFY THE BEHAVIOR OF A DISTRIBUTED PRODUCT.
- ACTIVATE AN UNINTENDED OPERATION.
- PLACE THE SYSTEM, MODULE, DEVICE, CIRCUIT, OR OUTPUT INTO AN UNINTENDED MODE.

ONLY THOSE PERSONS WHO:

(A) ARE PROPERLY TRAINED AND QUALIFIED WITH RESPECT TO THE USE OF THE DEVICE,

(B) UNDERSTAND THE WARNINGS ABOVE, AND

(C) UNDERSTAND HOW THIS DEVICE INTERACTS WITH AND IMPACTS THE FUNCTION

AND SAFETY OF OTHER PRODUCTS IN A DISTRIBUTED SYSTEM AND THE APPLICATION FOR WHICH THIS DEVICE WILL BE APPLIED, MAY USE THE DEVICE.

PLEASE NOTE THAT YOU CAN INTEGRATE THIS PRODUCT AS A SUBSYSTEM INTO HIGHER-LEVEL SYSTEMS. IN CASE YOU DO SO, MACH SYSTEMS s.r.o. HEREBY DECLARES THAT MACH SYSTEMS s.r.o.'s WARRANTY SHALL BE LIMITED TO THE CORRECTION OF DEFECTS, AND MACH SYSTEMS s.r.o. HEREBY EXPRESSLY DISCLAIMS ANY LIABILITY OVER AND ABOVE THE REFUNDING OF THE PRICE PAID FOR THIS DEVICE, SINCE MACH SYSTEMS s.r.o. DOES NOT HAVE ANY INFLUENCE ON THE IMPLEMENTATIONS OF THE HIGHER-LEVEL SYSTEM, WHICH MAY BE DEFECTIVE.

6.2 Disposal and Recycling Information

When this product reaches its end of life, please dispose of it according to your local environmental laws and guidelines.

6.3 Declaration of Conformity

MACH SYSTEMS

EU Declaration of Conformity (DoC)

We

Company Name	MACH SYSTEMS s.r.o.		
Postal Address	Pocernicka 272/96	**City**	Prague
Postcode	108 00	**Country**	Czech Republic

declare that the DoC is issued under our sole responsibility and belongs to the following products:

CAN-FD LIN Gateway

Objects of the declaration:

Product	Product Number
CAN-FD LIN Gateway	CANFD-LIN-GW

The objects of the declaration described above is in conformity with the relevant Union harmonisation legislation:

2014/30/EU - EMC Directive
2011/65/EU - RoHS (recast)

The following harmonised standards and technical specifications have been applied:

EN 55032	EN 61000-4-2
EN 63000	EN 61000-4-4

Signed for and on behalf of: MACH SYSTEMS s.r.o.

Place of issue: Prague, Czech Republic

Date of issue: December 14th 2020

Signature: 

Name, function: Miroslav Machacek, Managing Director

MACH SYSTEMS s.r.o.
www.machsystems.cz

6.4 Patents, Copyrights and Trademarks

7 References

[1] „TagConnect TC2030-IDC-NL,“ [Online]. Available: https://www.tag-connect.com/product/tc2030-ctx-nl-stdc14-for-use-with-stm32-processors-with-stlink-v3.

[2] „CAN FD LIN Gateway Github Repository,“ [Online]. Available: https://github.com/mach-systems/CANFD-LIN-Gateway.

[3] „CAN FD LIN Gateway - CAN Bootloader Application Note,“ [Online]. Available: https://www.machsystems.cz/wp-content/uploads/2021/01/CAN-FD-LIN-Gateway-CAN-Bootloader-AN.pdf.

[4] „STM32CubeProgrammer Web Site,“ [Online]. Available: https://www.st.com/en/development-tools/stm32cubeprog.html.

8 Ordering Information

Product Number	Description
CANFD-LIN-GW	CAN FD LIN Gateway
DIN-CLIP	Clip for mounting on a DIN rail

Table 6 Ordering Numbers

9 Contact

MACH SYSTEMS s.r.o.

www.machsystems.cz

info@machsystems.cz

Czech Republic

MACH SYSTEMS

Company registration: 29413893

EU VAT number: CZ29413893