

Bmbed*NucleoF042*

Embedded Software Framework

User Manual
Version 1.0



Contents

1	Introduction.....	2
1.1	Scope of this Document	2
1.2	Intended Audience	2
2	Before you begin	3
2.1	Free Bmbed Package for NUCLEO-F042.....	3
2.2	Prepare and Running Bmbed Board	4
2.2.1	Connections	4
2.2.2	Bootloader upload	5
2.3	Installing and running BmbedChat.....	7
3	Communication with BmbedChat	9
3.1	Overview.....	9
3.2	Events	9
3.3	Firmware Application Upload.....	10
3.3.1	Overview	10
3.3.2	Read Application Hex File	10
3.3.3	Erase Application FLASH Sectors	11
3.3.4	Send the Binary Data	11
3.3.5	End Upload.....	12
3.4	FLASH Data	13
3.5	FLASH Parameter	14
3.6	LED Blinky.....	15
3.7	Temperature.....	16
3.8	Debug Buffer	17
4	Manufacturer Information	18
5	Version History	19



1 Introduction

1.1 Scope of this Document

This document describes the using and the functions of the **Bmbed** Embedded Framework with the help of the evaluation board ST NUCLEO-F042 and the free software binaries

- **BmbedChat**
- STM32F042 Bootloader
- STM32F042 Application.

1.2 Intended Audience

Customer technical product management, customer technical responsible, development engineers.



2 Before you begin

2.1 Free Bmbed Package for NUCLEO-F042

Unzip the downloaded Bmbed_NUCLEO-F042_vvxxxyzz_yyyymmdd.zip file to a location of your choice on your PC.

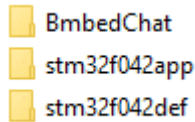


Figure 1 Free **Bmbed** NUCLEO-F042 Package

The free **Bmbed** Package for NUCLEO-F042 contains the following folder and files:

- **BmbedChat** **Bmbed** Communication Software for Windows
 - EXE and DLL files
- Stm32f042app **Bmbed** application software for STM32F042K6T6
 - Intel-hex file to upload with **BmbedChat**
- Stm32f042def **Bmbed** bootloader software and Keil project for STM32F042K6T6
 - Binary-axf file to upload with Keil μ Vision 5
 - Project files for Keil μ Vision 5 IDE
- BmbedNucleoF042_UserManual.pdf Manual to start with **Bmbed** and NUCLEO-F042



2.2 Prepare and Running Bmbed Board

2.2.1 Connections

The low-cost ST NUCLEO-F042 evaluation board is ready to use the **Bmbed** Embedded Framework.

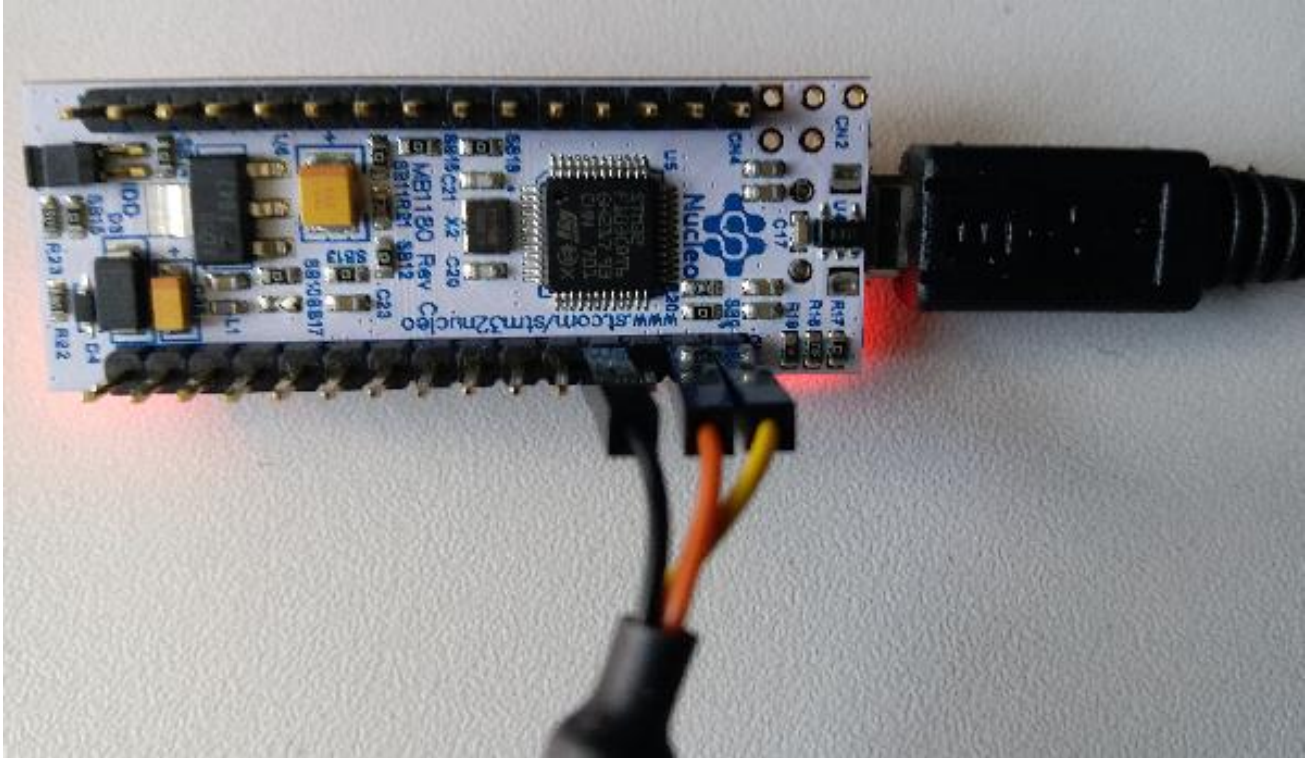


Figure 2 Free **Bmbed** NUCLEO-F042 Package

You need 2 cables to connect the board with your PC:

- USB 2.0 FTDI-UART cable (TTL-232R-RPi) for Bmbed communication.
 - Connect the Yellow wire to NUCLEO-F042 CN3/D1/Tx (Pin 1)
 - Connect the Orange wire to NUCLEO-F042 CN3/D0/Rx (Pin 2)
 - Connect the Black wire to NUCLEO-F042 CN3/Gnd (Pin 4)
 - Connect USB A to your PC
- USB 2.0 cable AB (Micro) for voltage supply and inbuild JTAG debugger.
 - Connect the micro plug to NUCLEO-F042 board
 - Connect USB A to your PC



2.2.2 Bootloader upload

First you can download and install the free (Evaluation version) Keil μ Vision IDE from the Keil homepage.

DoubleClick the default Keil project stm32f042def.uvprojx in the package stm32f042 folder.

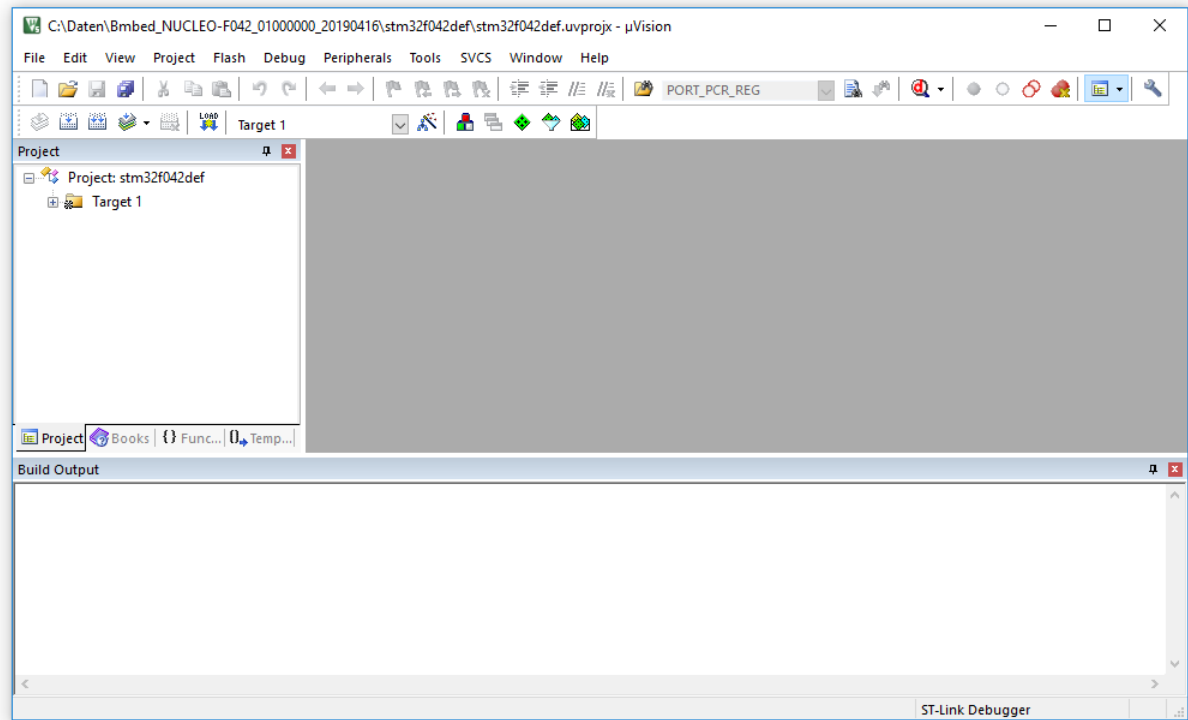


Figure 3 Keil uVision with NUCLOE-F042 default project

Maybe you're wondering because there is no source code. Sorry, source code is only to buy.

Now we load the **Bmbed** Bootloader to the virgin NUCLEO-F042 board.

- Be sure the board is connected to the PC with both USB cables.
- Click on the <Load> icon and the bootloader stm32f042def.axf file is downloading to the board.
- If no, check this:
 - USB cables
 - Keil Debugger: go to <Flash><Configure Flash Tools...><Debug>
 - Use: ST-Link Debugger
 - <Settings><Flash Download>: Erase Full Chip

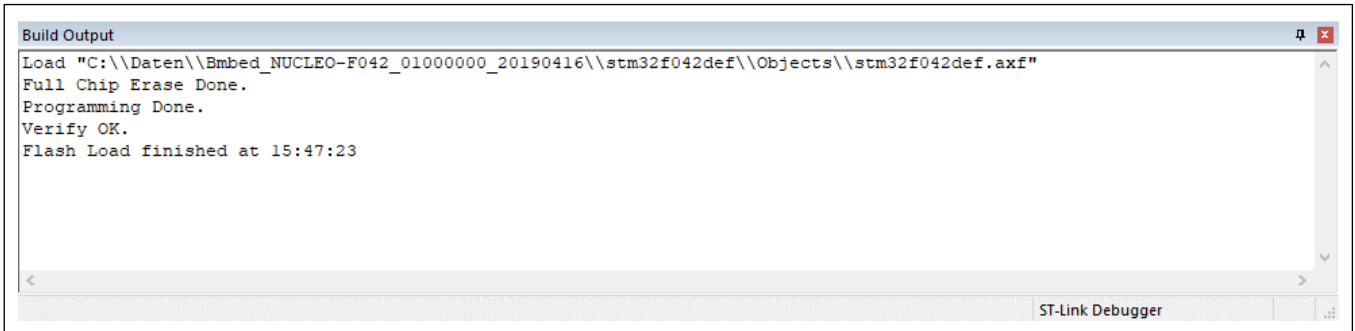
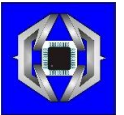


Figure 4 Keil uVision after successful bootloader upload

Now you can reset the board (off-on, reset button). Then the board starts with the **Bmbed** Bootloader. This is indicated by the blinking LED (about 1s interval).

Memory Map

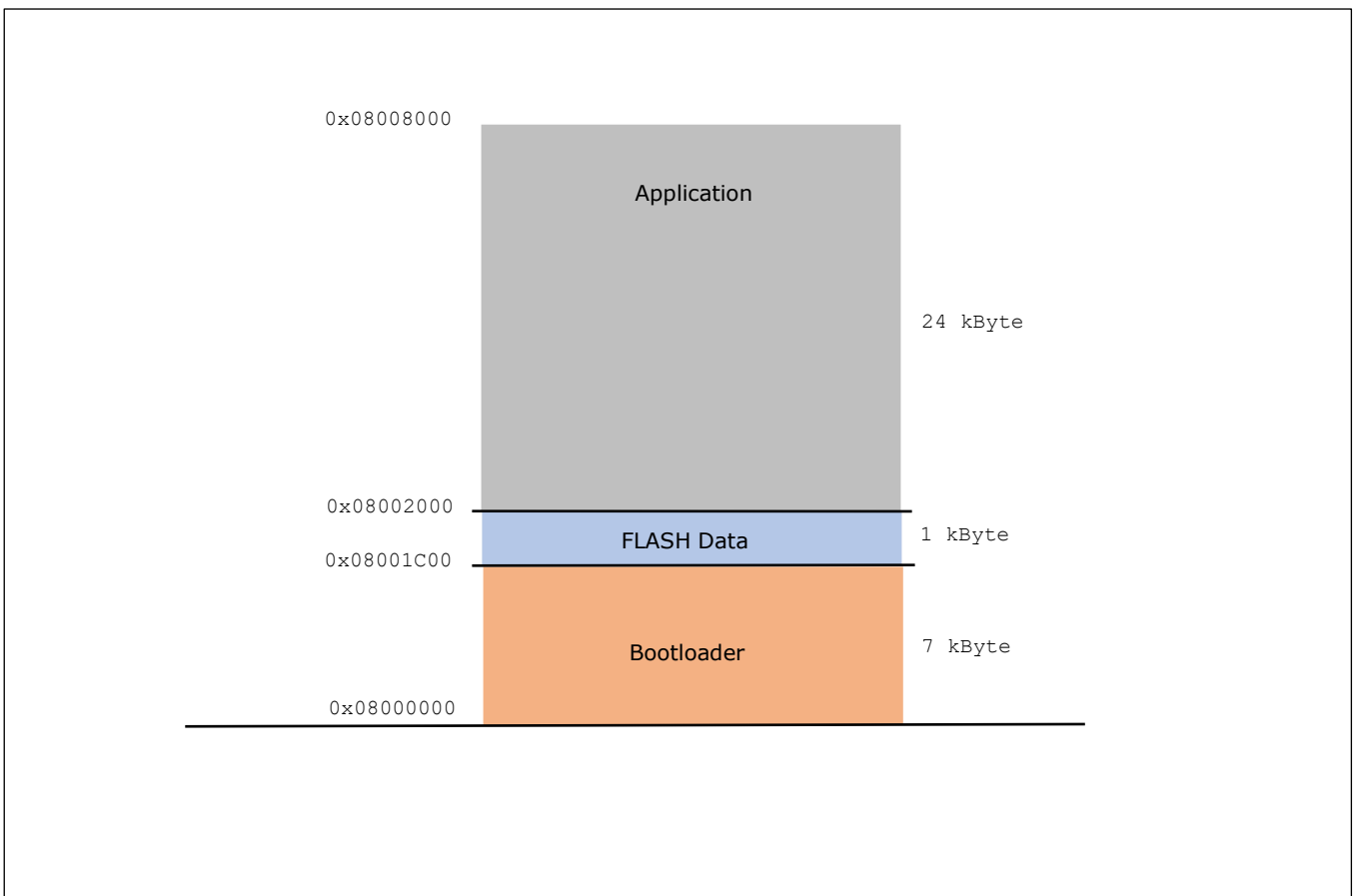


Figure 5 FLASH Memory Map

The board is now ready to communicate with **BmbedChat**.



2.3 Installing and running BmbedChat

BmbedChat software for Windows doesn't need a special installation.

- Double-click BmbedChat.exe.

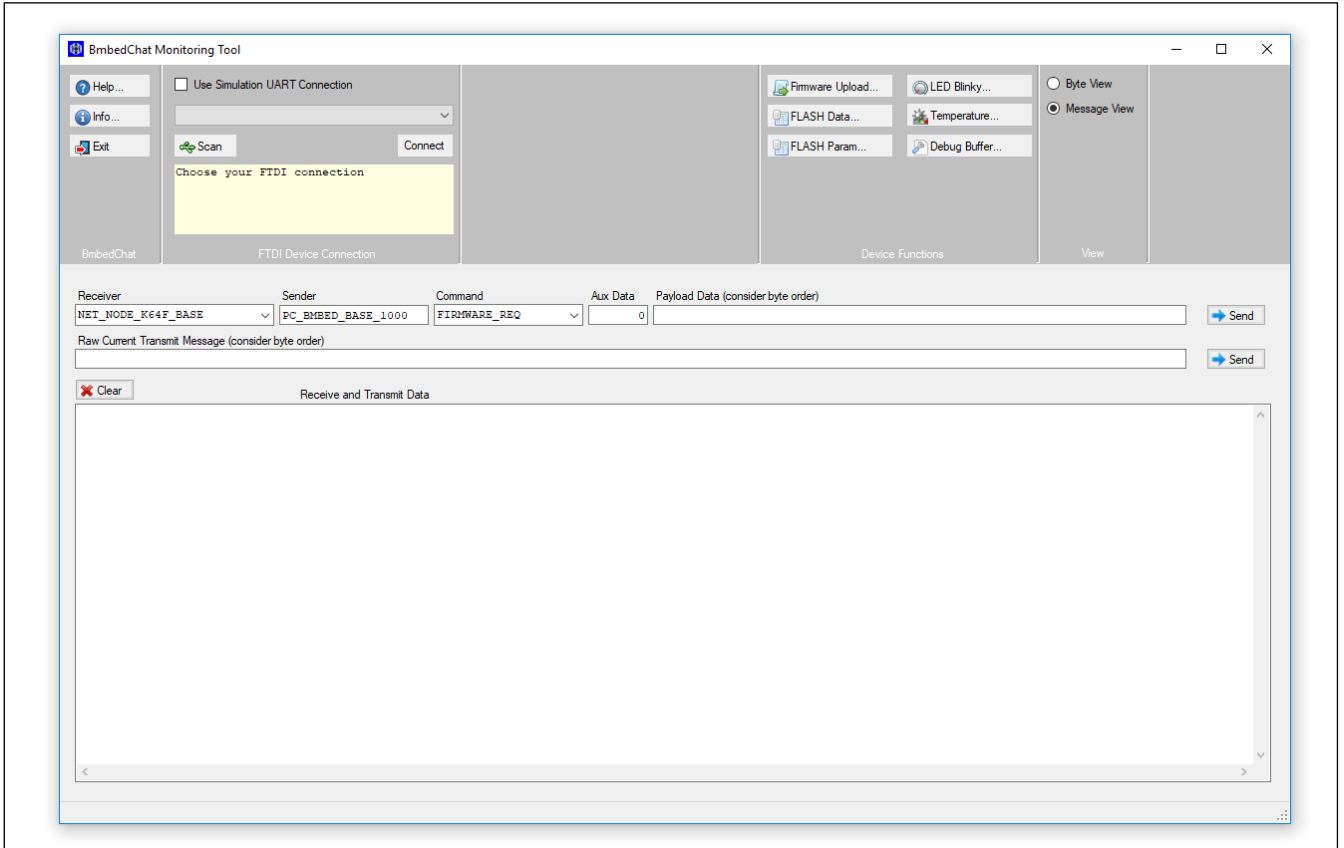


Figure 6 Started BmbedChat

Now we connect to the USB-FTDI

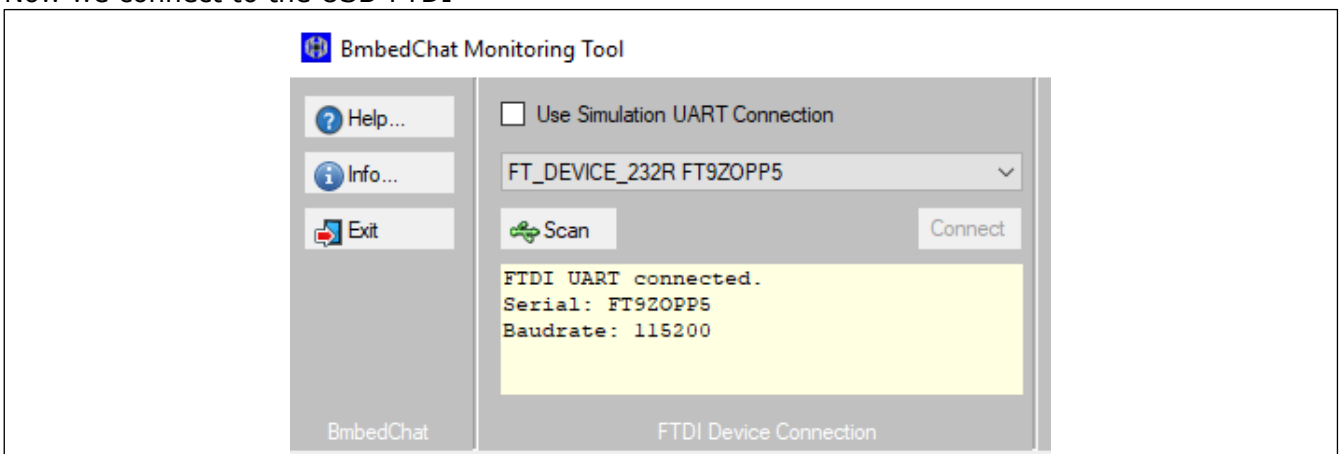
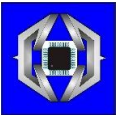


Figure 7 Connect with the USB-FTDI



And setting the correct receiver address for the NUCLEO-F042 board.

Receiver: **NET_NODE_STM32F0** (selected from dropdown)

Sender: PC_BMBED_BASE_1000

Command: FIRMWARE_REQ

byte order)

Receive and Transmit Data

Figure 8 Receiver address.

Click the upper <Send> and request the firmware data.

Receiver: NET_NODE_STM32F0

Sender: PC_BMBED_BASE_1000

Command: FIRMWARE_REQ

Aux Data: 0

Payload Data (consider byte order):

Raw Current Transmit Message (consider byte order): 90 00 00 10 00 02 00 00

Receive and Transmit Data

Time	Receiver	Sender	Command	Aux Data	Payload Data	Timestamp
07:25:11	NET_NODE_STM32F0	PC_BMBED_BASE_1000	FIRMWARE_REQ	0	Boot STM32F	1.0.0.0 Apr 16 2019 08:33:49
07:25:11	PC_BMBED_BASE_1000	NET_NODE_STM32F0	FIRMWARE_RES	0		

Figure 9 Firmware request with the response from the bootloader software.



3 Communication with BmbedChat

3.1 Overview

BmbedChat allows to communicate with the NUCLEO-F042 board. Follow commands, data and functions are available:

- Events
- Static firmware request
- FLASH data requests and settings
- FLASH parameter requests and settings
- LED request and setting
- Chip temperature request
- Debug buffer request
- Application firmware upload

3.2 Events

Events are one-way messages. Every **Bmbed** network node can send a message to any node. There are four kinds of events:

- Information
- Warning
- Error
- Alarm

Details are described in the **Bmbed** High-Level-Protocol documentation.

After a board reset, following information event is transmitted to **BmbedChat**:

```
07:33:38 PC_BMBED_BASE_1000 NET_NODE_STM32F0 EVENT_INFORMATION 0 43605 Bootloader started, no jump to application
```

Figure 10 Bootloader information event after board reset

There isn't any application on the board yet. Just the bootloader is running and is responding.



3.3 Firmware Application Upload

3.3.1 Overview

To use all features of the **Bmbed** Framework we must upload the embedded application software (firmware). This contains follow steps (Bootloader running):

- Read application hex file
- Erase application FLASH sectors on the board
- Send the binary data to the board
- End the upload process (board jumps from bootloader to the new application)

3.3.2 Read Application Hex File

Click <Firmware Upload...>. The Firmware upload dialog appears. Click on the open file icon.

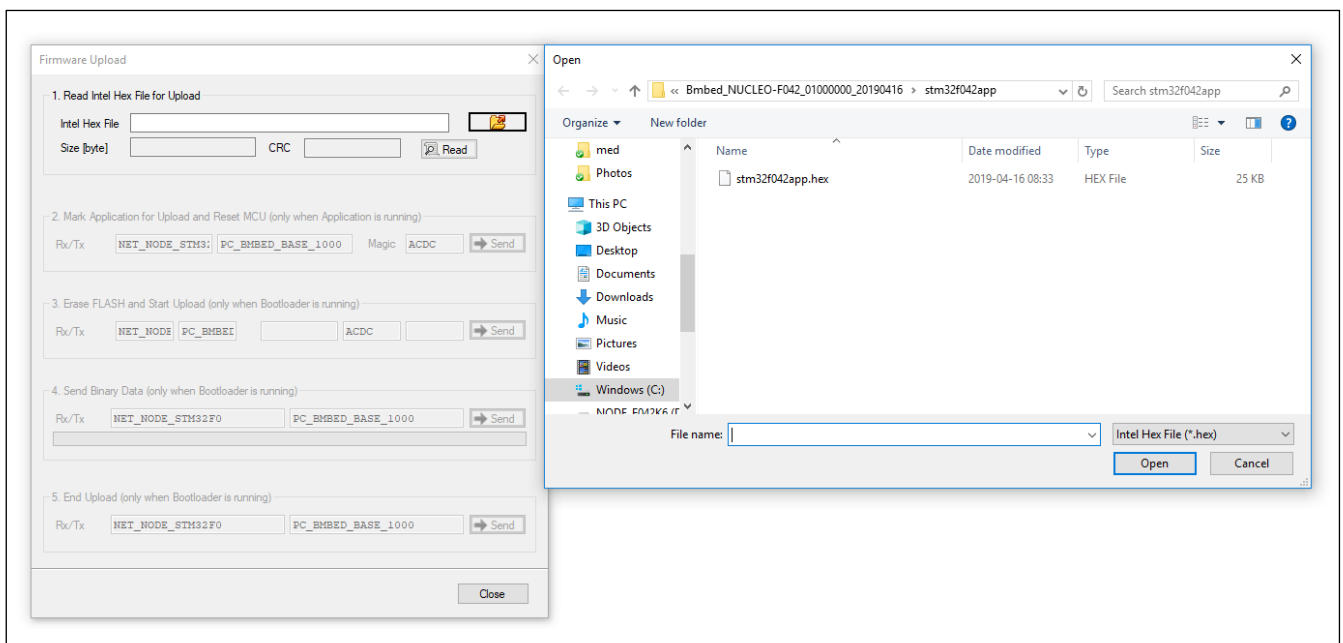


Figure 11 Read application hex file

Choose the stm32f042app.hex file. Click the <Read> button on the Firmware upload dialog to read all necessary information from the hex file.



3.3.3 Erase Application FLASH Sectors

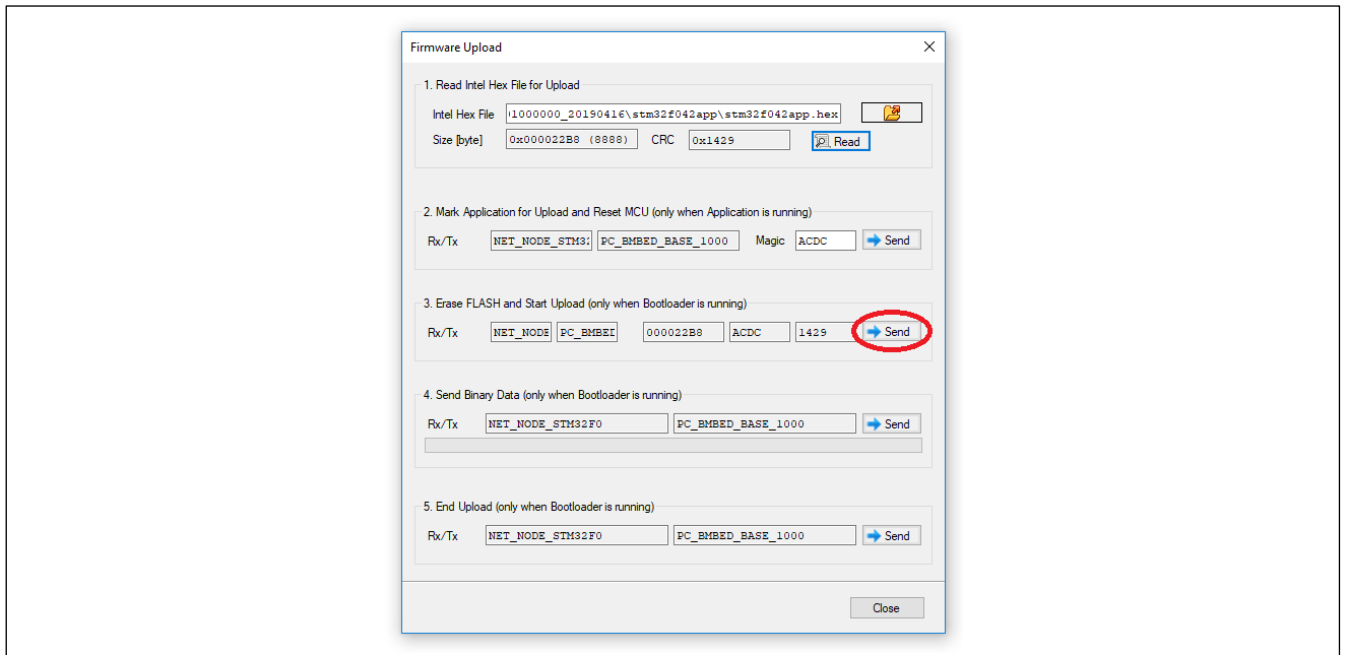


Figure 12 Erase application FLASH sectors

Only the bootloader can erase the application FLASH area. If you want to upload a new application while the old application is running, then you first must perform step two on the dialog to switch to the bootloader.

Look to the message field in the **BmbedChat** main-window. Every step needs a response from the board.

3.3.4 Send the Binary Data

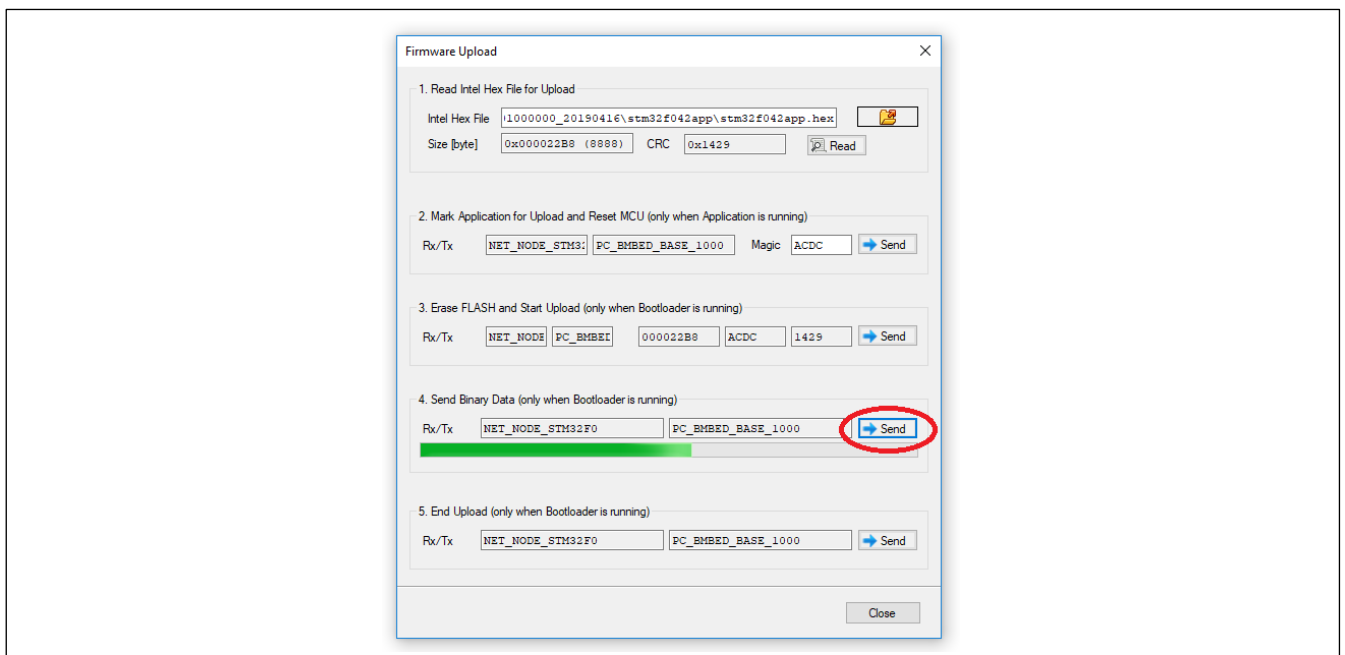


Figure 13 Send binary data from the hex file to the board



3.3.5 End Upload

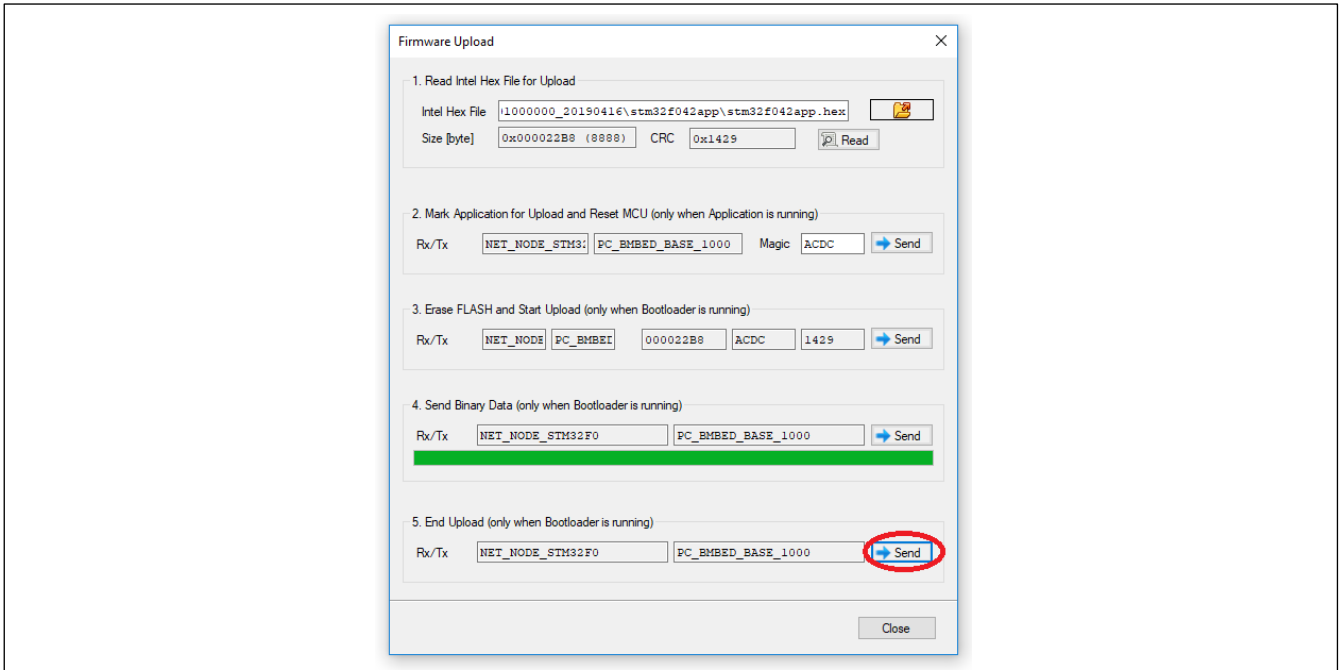


Figure 14 End Upload

At the end the bootloader marks the new application on the FLASH and jumps to it. The new application is starting. You can see it by means of the application event message.

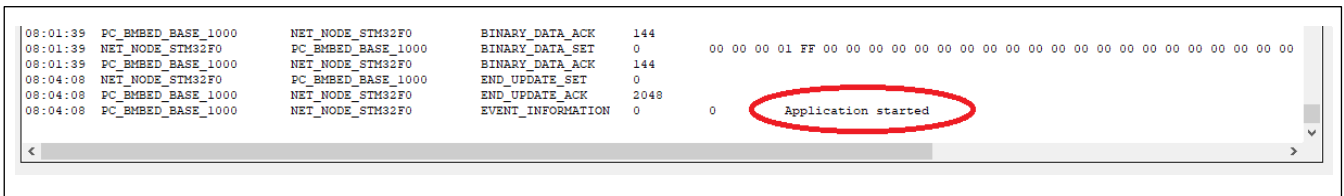


Figure 15 New application started after upload



3.4 FLASH Data

Bmbed supports different persistent data blocks. These data can be requested and mostly changed.

- Firmware (Name, Version, Data, Time)
 - not changeable (hardcoded in source code)
 - bootloader and application
- Hardware (Name, Version, Date, Time)
 - changeable by user
 - only application
- Company (Name)
 - changeable by user
 - only application
- Device (Name, Serie Nr.)
 - changeable by user
 - only application
- Production (Plant, Date, Time)
 - changeable by user
 - only application

All these data can be handled within the FLASH data dialog.

The FLASH Data dialog box is divided into five sections, each with a 'Get' button and a 'Set' button (where applicable). The 'Firmware' section is marked as 'Hardcoded'. The 'Hardware', 'Company', 'Device', and 'Production' sections allow for user modification.

Section	Field	Value	Action	
Firmware (Hardcoded)	Rx/Tx	NET_NODE_STM32F0	Get	
	PC_BMBED_BASE_1000			
	Name	App STM32F0		
	Version	1.0.0.0		
	Date	Apr 16 2019	Time: 08:33:22	
Hardware	Rx/Tx	NET_NODE_STM32F0	Get	
	PC_BMBED_BASE_1000			
	Name	STM32F042K6		
	Version	1.0.0.0		
	Date	2017-10-10	Time: 15:23	Set
Company	Rx/Tx	NET_NODE_STM32F0	Get	
	Name	Eicher Engineering, CH-8739 Rieden	Set	
Device	Rx/Tx	NET_NODE_STM32F0	Get	
	PC_BMBED_BASE_1000			
	Name	NUCLEO-F042 Bmbed Framework		
Production	Rx/Tx	NET_NODE_STM32F0	Get	
	PC_BMBED_BASE_1000			
	Name	Rieden SG		
Production	Date	2018-11-11	Time: 12:52	Set

Close

Figure 16 FLASH data dialog



3.5 FLASH Parameter

Bmbed supports different default persistent data parameter. These parameters can be requested and can be changed.

- Number
 - Parameter block number 1..4 integer format and 1..4 float format.
- Format
 - Integer (0) or float (1)
- Value
 - Current value
- Minimum
 - Minimum value
- Maximum
 - Maximum value
- Unit
 - Text field for any unit display
- Result
 - Text field for any test description or auxiliary description.

All these data can be handled within the FLASH parameter dialog:

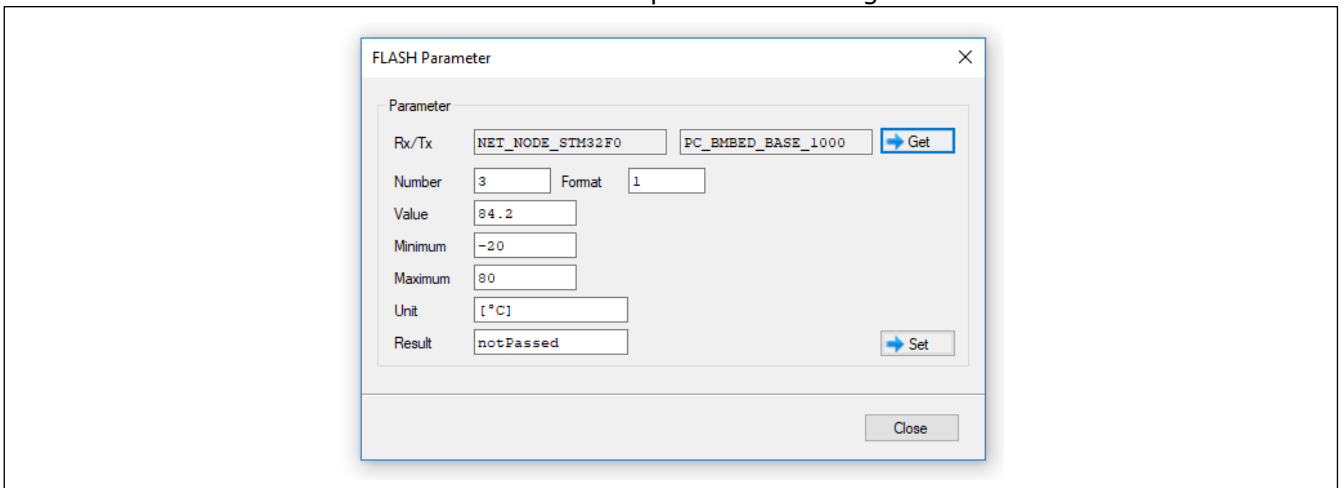
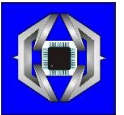


Figure 17 FLASH parameter dialog and values for number 3 and float format.



3.6 LED Blinky

The LED behaviour can be set

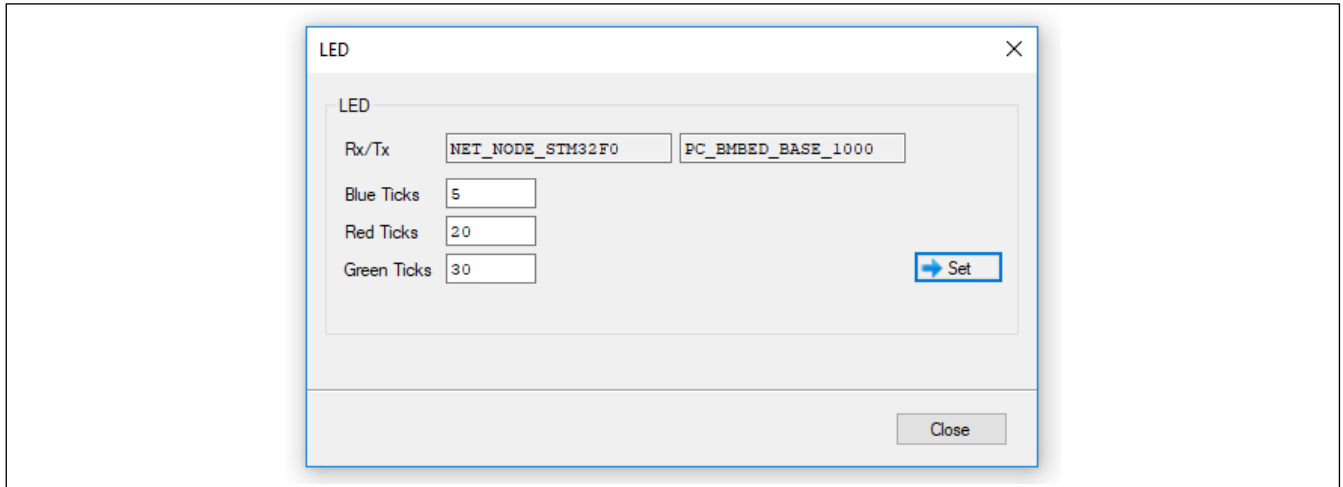


Figure 18 LED ticks for blinking.

BmbedChat supports different boards. The NUCLEO-F042 contains only one LED. Thereby only the ticks for the blue LED takes effect.

One tick time is about 20ms. 50 ticks leads to a blinking LED with 1s interval.



3.7 Temperature

The microcontroller temperature can be read.

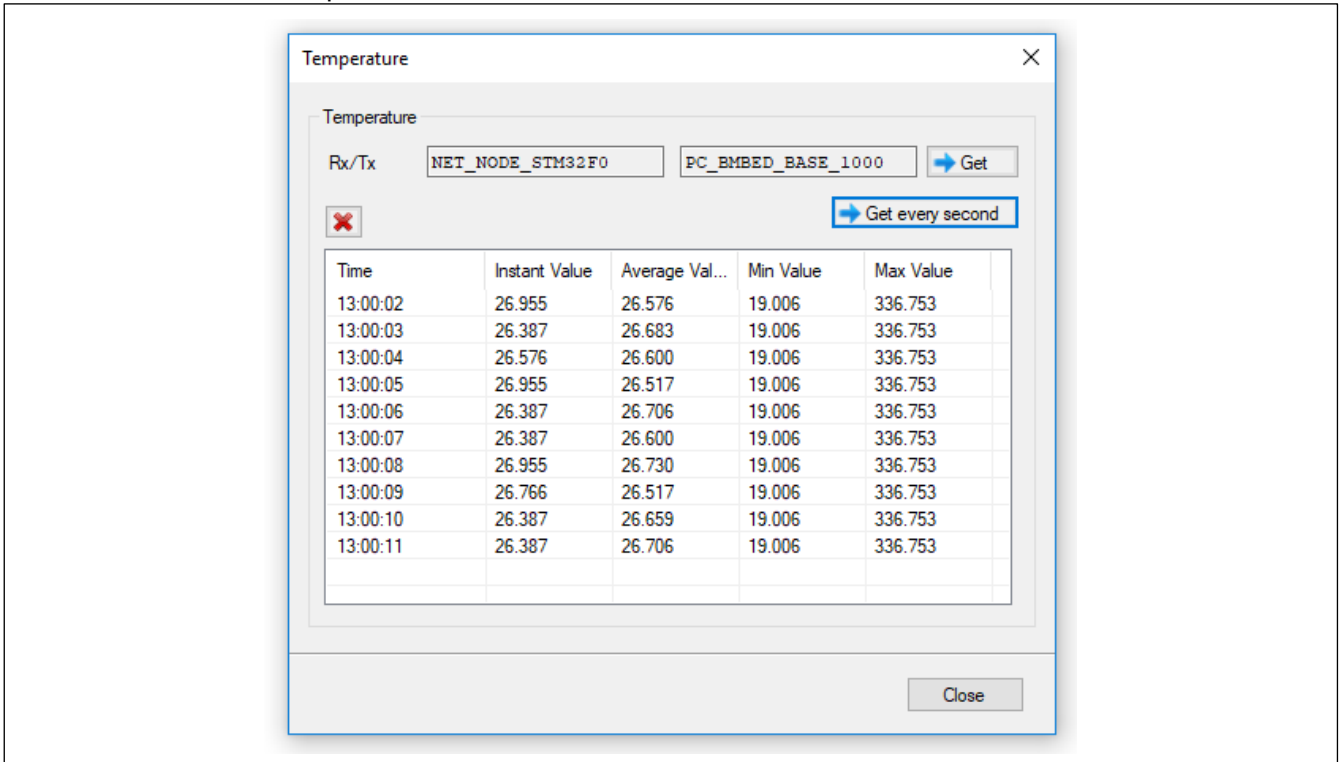


Figure 19 Microcontroller temperatures.

The STM32F042 has a build-in temperature sensor. Different temperature values can be read:

- Instant value Current value at the request
- Average value Moving average value over a buffer of 8 instant values.
- Minimum value Minimum value since the software was running.
- Maximum value Maximum value since the software was running.

All values in [°C]



3.8 Debug Buffer

The **Bmbed** embedded application fills in 20 debug values with two 16 Bit values each.

These values can be read with this dialog.

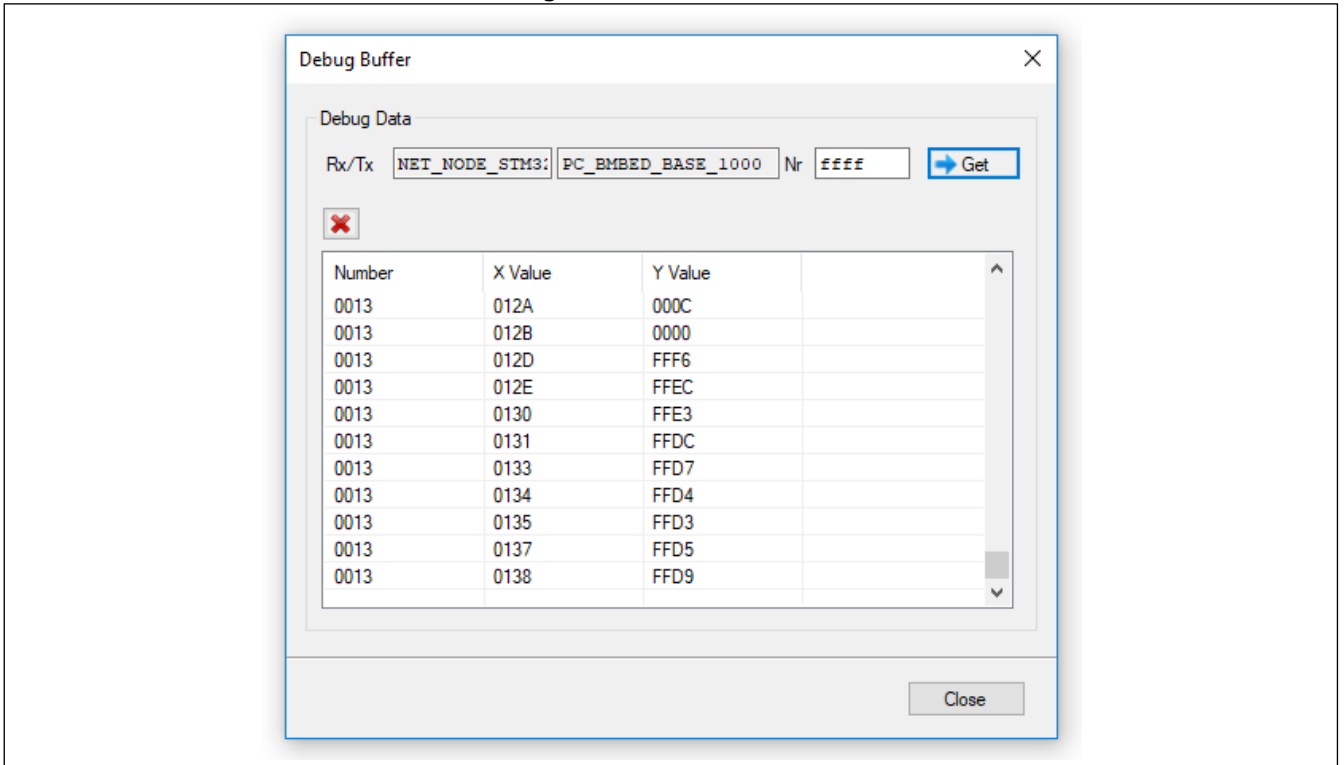


Figure 20 Debug buffer.

You can read one entry according their number (0...0x13) for 20 value pairs.

The entire debug buffer is transmitted if the number in the field is higher than the highest number (0x13).



4 Manufacturer Information

Customer support and technical support for **BmbedChat**



Eicher Engineering
Software Development
Frigadenstrasse 23
CH-8739 Rieden SG
Switzerland

www.bmbed.ch
bmbed@eichereng.ch

Bmbed NUCLEO-F042 User Manual Version 1.0

© 2016-2019 by Eicher Engineering

Bmbed NUCLEO-F042 described in this manual is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from Eicher Engineering.

Disclaimer of Warranty

THE SOFTWARE AND DOCUMENTATION HAS PROVIDED 'AS IS.' TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, EICHER ENGINEERING DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE. ANY LIABILITY OF EICHER ENGINEERING WILL BE LIMITED EXCLUSIVELY TO REFUND OF THE PURCHASE PRICE.



5 Version History

Version	Type	Date	Author	Comments
1.0	released	2019-04-17	ME	Initial writing