

# AN038: Using TRINAMIC's IC Software API and Examples

Document Revision V1.00 • 2017-May -16

**This application note describes TRINAMIC's software API for TRINAMIC chip solutions and evaluation boards and how it can be used to develop your own firmware.**

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Software API Project Structure . . . . .	2
1.2	Evaluation Board Project Structure . . . . .	3
<b>2</b>	<b>Requirements and Installation Guide</b>	<b>4</b>
2.1	Tools . . . . .	4
2.1.1	Eclipse . . . . .	4
2.1.2	GNU ARM Embedded Toolchain . . . . .	5
2.1.3	TMCL-IDE . . . . .	5
2.2	How to create and update a new firmware image . . . . .	5
2.2.1	Download TMC-API and TMC-EvalSystem from TRINAMIC website . . . . .	5
2.2.2	Configure GNU Toolchain . . . . .	6
2.2.3	Import TMC-API and TMC-Evalsystem Eclipse projects . . . . .	6
2.2.4	Build firmware . . . . .	8
2.2.5	Update a module . . . . .	8
<b>3</b>	<b>References</b>	<b>10</b>
<b>4</b>	<b>Revision History</b>	<b>11</b>



# 1 Introduction

This application note describes the structure of the TRINAMIC software API and how it can be used to simplify and speed up your firmware development based on hardware with TRINAMIC motion control chips. The software API is part of the Technology Access Package (TTAP) [1][2], a comprehensive collection of technical support information and tools for TRINAMIC's chip and module solutions. This application note explains, which compiler and tools can be used to compile the TMC-EvalSystem project and how to upload the firmware to a module using a third party programmer or the standard bootloader with the TMCL-IDE.

## 1.1 Software API Project Structure

The API is designed for C applications and can be used for different CPU types. For example the API is used for TRINAMIC's modular evaluation system boards called Landungbrücke (Freescale MK20 cortex M4 processor) [5] and Startrampe (STM-F205 cortex M3 processor) [6].

The API is structured as shown in figure 1:

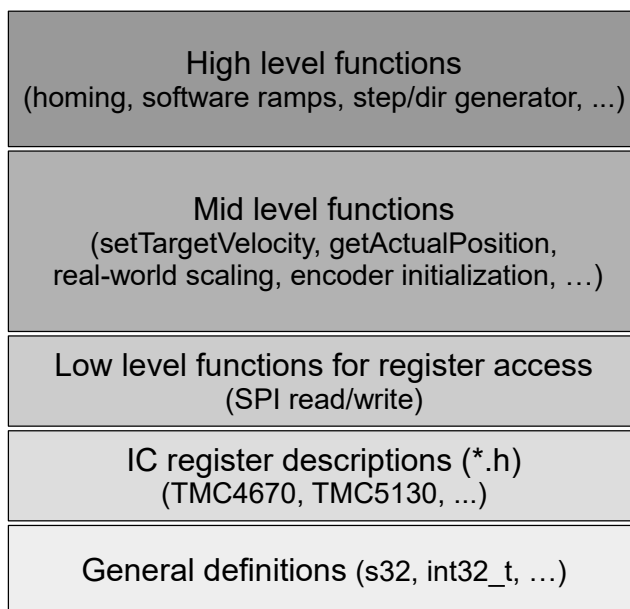


Figure 1: TMC-API structure

The folder tmc/helpers contains general definitions for data types and bit handling, independent of the processor used. The folder tmc/ic contains a folder for each TRINAMIC chip, e.g. tmc/ic/TMC4670. Each of these folders contain a header file with the chip's register address list (e.g. TMC4670\_Register.h). Another .h and .c file (e.g. TMC4670.h/.c) provide low level, mid level, and high level functions for the specific chip. Low level functions are necessary for SPI read and write access. Mid level functions like setTargetVelocity(...) or getActualPosition() demonstrate the typical register access for different application scenarios, e.g. for velocity or position mode. For some chips, there are also functions for encoder initialization and real-world current scaling available.

The next chapter shows the use of the API in a firmware project on the example of the TRINAMIC evaluation board firmware project for Landungbrücke and Startrampe.



## 1.2 Evaluation Board Project Structure

The evaluation board project demonstrates the usage of the TMC-API and can be used as starting point for your own firmware development. It should be an example for you to simplify and speed up firmware development based on hardware with TRINAMIC motion control IC's. The structure of the evaluation board project is shown in figure 2:

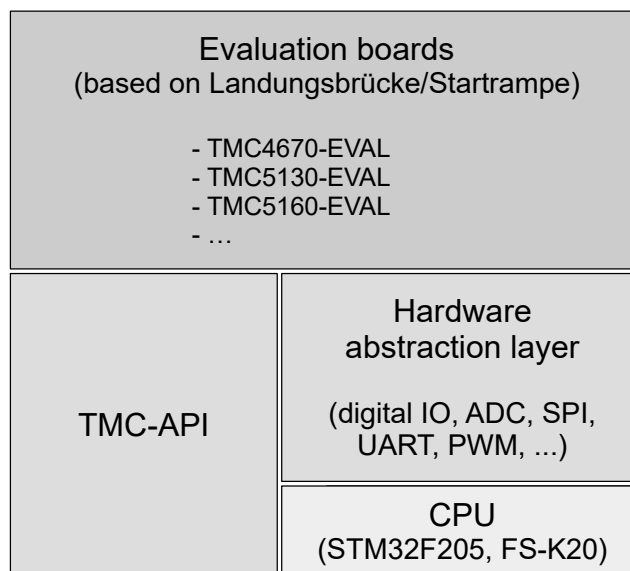


Figure 2: TMC Eval System structure

The evaluation board project is a makefile project and supports two processor types: a Freescale MK20 (cortex M4 processor used in the Landungbruecke) [5] and a STM-F205 (cortex M3 processor) [6]. The processor can be selected according to a build-command for the makefile. The complete firmware project is provided as is available for free use and adaption [2]. The project uses the Eclipse IDE for C/C++ developers and can be extended for further processor types and evaluation boards.

On top of the CPU specific firmware part, a hardware abstraction layer has been implemented, to abstract the access to digital IO, the ADC measurement, SPI, communication, PWM, and eval board detection from the CPU specific firmware part. The eval board firmware detects connected eval boards automatically and loads a board specific firmware part dynamically corresponding to the actual connected eval board. This firmware part uses the TMC-API for the header file and the \*.h/\*.c file for the specific functionality of the actual used TRINAMIC motion control IC. Using the hardware abstraction layer, each eval board has access to a wide range of CPU functions in a CPU independent way.



## 2 Requirements and Installation Guide

This section describes the list of tools needed to compile the firmware project and upload it to the Landungsbrücke or Startrampe. For the installation and configuration, a step-by-step guide with screenshots is presented.

### 2.1 Tools

These are the software packages you need in order to use the TMC-API in your project:

#### Eclipse

An open-source integrated development environment (IDE)



#### GNU ARM Embedded Toolchain

Pre-built GNU toolchain for ARM Cortex-M processors



#### TMCL-IDE

The visual software to control and update TRINAMIC modules and evaluation boards



#### 2.1.1 Eclipse

Eclipse is used to organize, program and compile your software projects. Other IDE's can be used instead, but the TMC-API project was developed with it and the Eclipse project files are provided along with the code.

Please download the latest version of Eclipse from the website <https://eclipse.org/downloads/>, choose the package "Eclipse IDE for C/C++ Developers" as shown on figure 3, and follow the installation wizard.

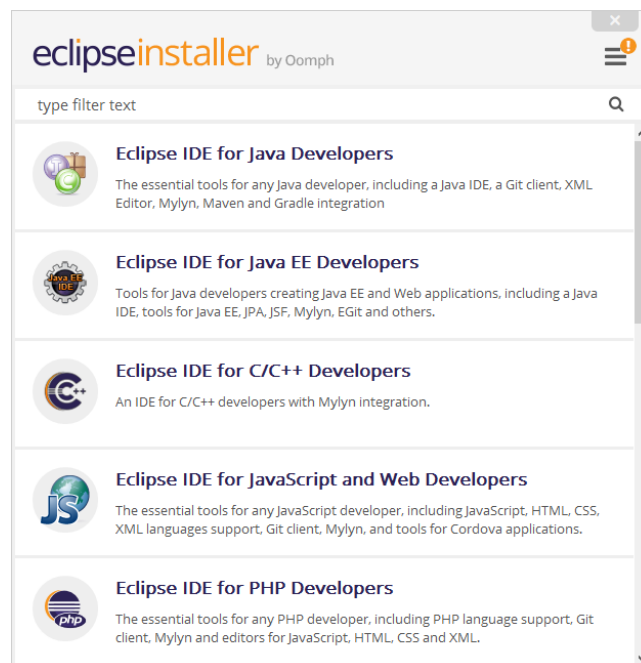


Figure 3: Eclipse package selection



For older versions of Eclipse or other installation methods, it is necessary to either download the “Eclipse IDE for C/C++ Developers” package directly or install the Eclipse CDT plugin available at: <https://eclipse.org/cdt/>.

### 2.1.2 GNU ARM Embedded Toolchain

The GNU ARM Embedded GCC can be used to compile the firmware. You can download the most recent release here: <https://launchpad.net/gcc-arm-embedded/+download>. Please follow the installation wizard to set up the toolchain in your system.

### 2.1.3 TMCL-IDE

The TMCL-IDE software tool is developed by TRINAMIC and offers many tools and wizards to control various TRINAMIC modules and evaluation board products. You can use the TMCL-IDE to update the firmware of a TRINAMIC evaluation board. Please download the latest TMCL-IDE release here: <http://www.trinamic.com/support/software/> and install it in your system.

If you want to update the firmware of a module without bootloader feature (e.g. self-designed boards), you will need a programmer device. The following are two common options for some of the most popular embedded processor families:

- ST-LINK: In-circuit debugger and programmer for STM8 and STM32 MCUs  
<http://www.st.com/en/development-tools/st-link.html>
- Segger J-Link: USB powered JTAG debug probe supporting a large number of CPU cores  
<https://www.segger.com/jlink-debug-probes.html>

## 2.2 How to create and update a new firmware image

This section explains the steps you need to follow in order to set-up the programming environment to use the TRINAMIC TMC-API project. Afterwards, you can create your own firmware image and update it to a TRINAMIC evaluation board.

- Download TMC-API [3] and TMC-EvalSystem [4] projects
- Download TTAP from the TRINAMIC website [2]
- Configure GNU Toolchain
- Import TMC-API and TMC-EvalSystem Eclipse projects
- Build the firmware
- Update your evaluation board

### 2.2.1 Download TMC-API and TMC-EvalSystem from TRINAMIC website

You can download the TRINAMIC TMC-API [3] and TMC-EvalSystem [4] source code projects from the TRINAMIC website for free use and adaption. Extract the \*.zip files “TMC-API.zip” and “TMC-EvalSystem.zip” to a folder of your system.



### 2.2.2 Configure GNU Toolchain

The makefile of the TMC-EvalSystem project uses “cs-make.exe” and “cs-clean.exe”. You can find these files in the folder “Firmware/Toolchain” of the TTAP package [3]. Copy these files and browse to the installation folder of the GNU ARM Toolchain (which you have installed in Section 2.1.2). The path of the folder should be like “C:\Program Files (x86)\GNU Tools ARM Embedded\5.4 2016q2” (where “5.4” is the installed version and “2016q2” the release date). Open sub-folder “bin” and paste the files “cs-make.exe” and “cs-clean.exe” there.

Afterwards, the “bin” folder of the compiler (the folder where the files were copied) must be added to the Windows PATH variable. The process may differ between Windows versions. In Windows 10, search for and select “System (Control Panel)”. Click the advanced system settings link and then click on Environment Variables. Select the PATH environment variable and select it. Then click Edit as shown in Figure 4. If the PATH environment variable does not exist, click New. Otherwise, add the path to the “bin” folder as shown in the figure.

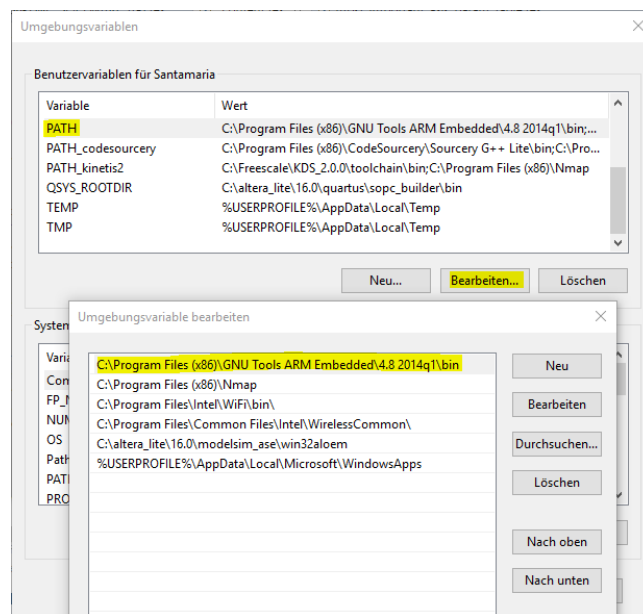


Figure 4: Add folder to PATH variable

### 2.2.3 Import TMC-API and TMC-Evalsystem Eclipse projects

Open Eclipse, choose a workspace path and click on “Workbench” to close the welcome screen. As next, click on File - Import and choose “Existing Projects into Workspace” under “General” as shown in figure 5. Then click on “Next”.

Click on “Browse” and choose the path where you have downloaded the TMC-API and TMC-EvalSystem projects. Select “TMC-API” and “TMC-EvalSystem” and check “Copy projects into workspace”. Proceed with “Finish” as shown in figure 6.



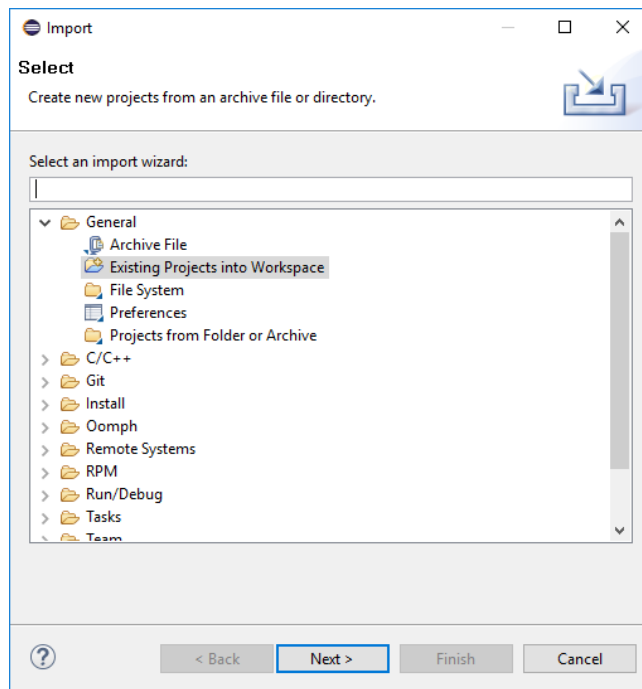


Figure 5: Import TMC-API

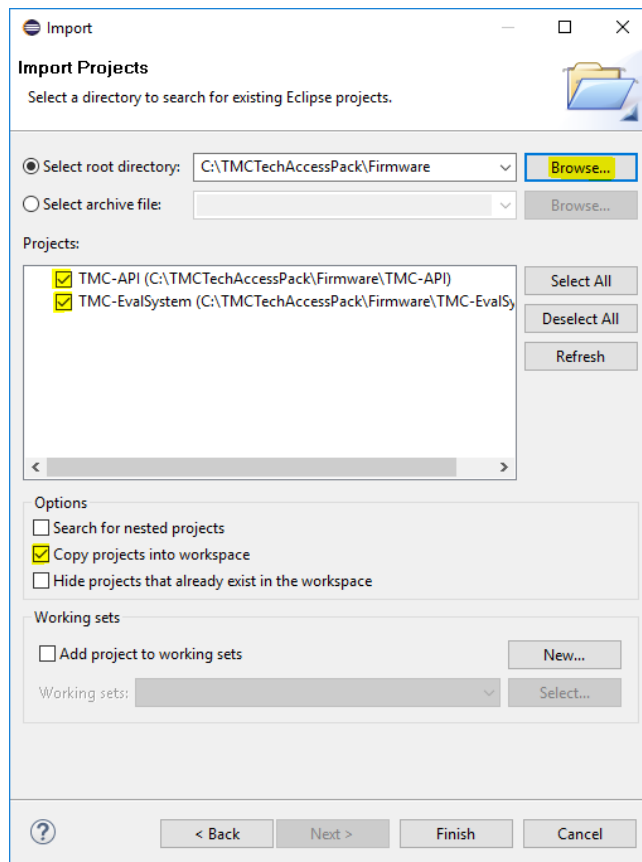


Figure 6: Copy TMC-API to workspace



Now you have imported both projects containing the source code into Eclipse. You can choose a file and start to edit the file. For example, on the left panel, open the TMC-EvalSystem project and double click on *main.c* to open the file. The source code will be shown on the central panel as shown in figure 7.

## 2.2.4 Build firmware

At this point, you are ready to build the firmware image. Click on the project "TMC-EvalSystem" to select it. To start the compilation, you have to find "Build Targets" on the top right panel. In new Eclipse releases, you can also find it on the file tree view under the project name shown in figure 7.

Select the appropriate target for your board. If you have a Landungsbrücken evaluation board, double-click on "Landungsbruecke\_BL" (as highlighted in the image). If your board is a Startrampe evaluation board, double-click on "Startrampe\_BL". In the console (bottom panel) you will see the compile output finished with a "Build finished" message when the process finishes successfully.

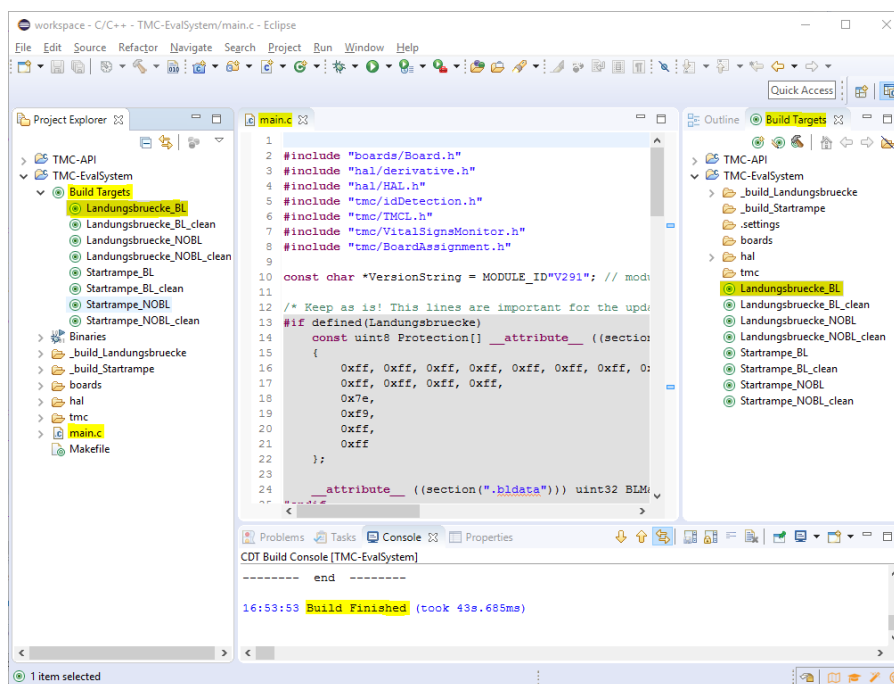


Figure 7: Build Firmware Image

Targets ending with "\_NOBL" are for boards without a bootloader. The created firmware images will have to be flashed with a programmer device instead of using the TMCL-IDE firmware update tool.

## 2.2.5 Update a module

Open the TMCL-IDE software. Power and connect your module. At the first start, the TMCL-IDE will ask you for permission to install the TRINAMIC USB driver.

After installing the driver, your module will appear on the left panel. Click on Tools - Firmware Update or on its icon on the top-right toolbar (an arrow pointing down, highlighted in figure 8). Check that the selected module is correct and click on "Browse" to find your firmware image.





Move to your Eclipse workspace path and open the build folder of your module inside the TMC-EvalSystem project. There you will find an `.hex` file whose name indicates your module and firmware version.

Figure 8 shows an example with a Landungsbrücke evaluation board. You can see that the selected module matches the one shown on the left panel. The “TMC-EvalSystem” path appears in the smaller window. There, the “Landungsbruecke\_2.91\_BL.hex” file from the “\_build\_Landungsbruecke” folder is selected.

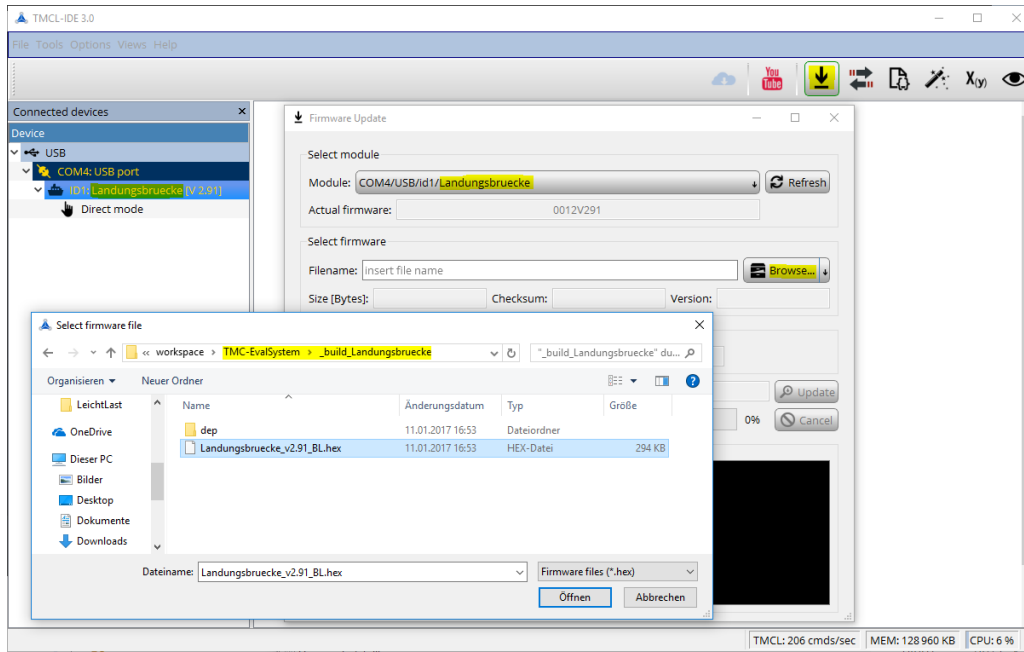


Figure 8: TMCL-IDE Browse image file

If the firmware image is compatible with the module, you can click on “Update” as shown in figure 9.



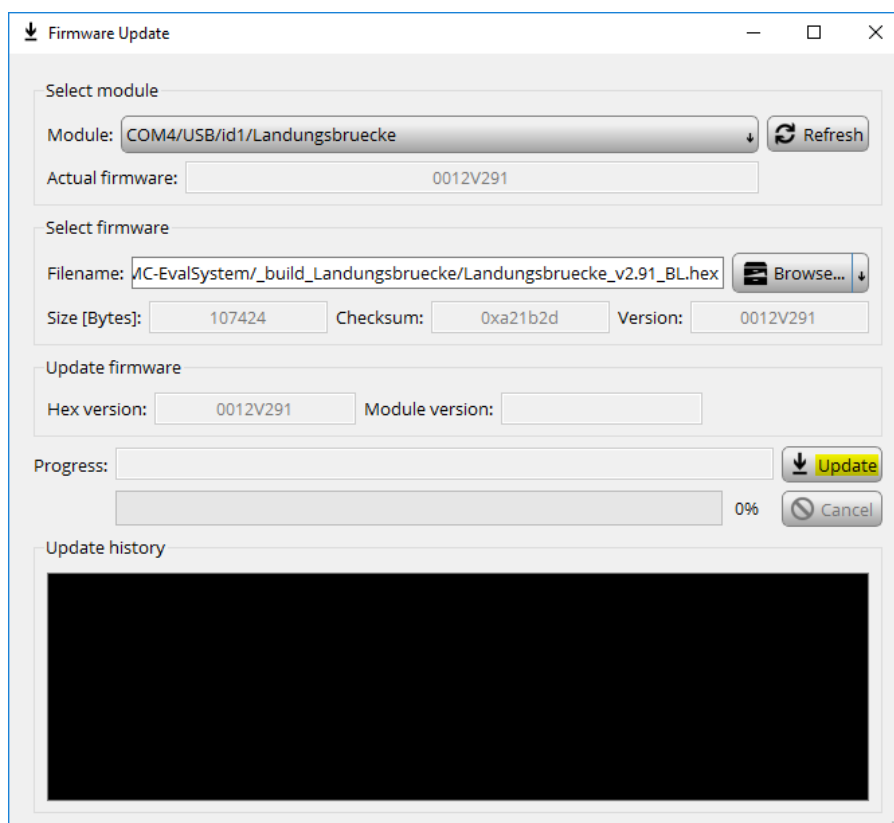


Figure 9: TMCL-IDE Update image file

For a non-bootloader firmware image, connect your programmer device to your PC and your module (e.g. via USB and JTAG respectively). Configure the programmer software for your module's processor and open the "\_NOBL".hex. Connect to the processor and proceed to flash the image.

Afterwards, please reboot your module and have fun with the evaluation board or your own designed module.

### 3 References

- [1] TRINAMIC, "AN37: TRINAMIC Technology Access Package" [www.trinamic.com/support/help-center/downloads/AN037-TRINAMIC\\_Technology\\_Access\\_Package](http://www.trinamic.com/support/help-center/downloads/AN037-TRINAMIC_Technology_Access_Package)
- [2] "TRINAMIC Technology Access Package (TTAP)" [www.trinamic.com/support/software/](http://www.trinamic.com/support/software/)
- [3] TRINAMIC, "TMC-API eclipse project" [www.trinamic.com/support/software/](http://www.trinamic.com/support/software/)
- [4] TRINAMIC, "TMC-EvalSystem eclipse project" [www.trinamic.com/support/software/](http://www.trinamic.com/support/software/)
- [5] TRINAMIC, "Landungsbruecke evaluation board" [www.trinamic.com/support/eval-kits/details/landungsbruecke/](http://www.trinamic.com/support/eval-kits/details/landungsbruecke/)
- [6] TRINAMIC, "Startrampe evaluation board" [www.trinamic.com/support/eval-kits/details/startrampe/](http://www.trinamic.com/support/eval-kits/details/startrampe/)



## 4 Revision History

Version	Date	Author	Description
V0.10	22.12.2016	SK, ED	Added first content and structure
V0.20	23.01.2017	BS	Added Section 2: how-to guide.
V0.90	12.05.2017	ED	Added Section 1 and prepared final version.
V1.00	16.05.2017	ED	Final version.

*Table 2: Document Revision*

