FRIWO | SDK
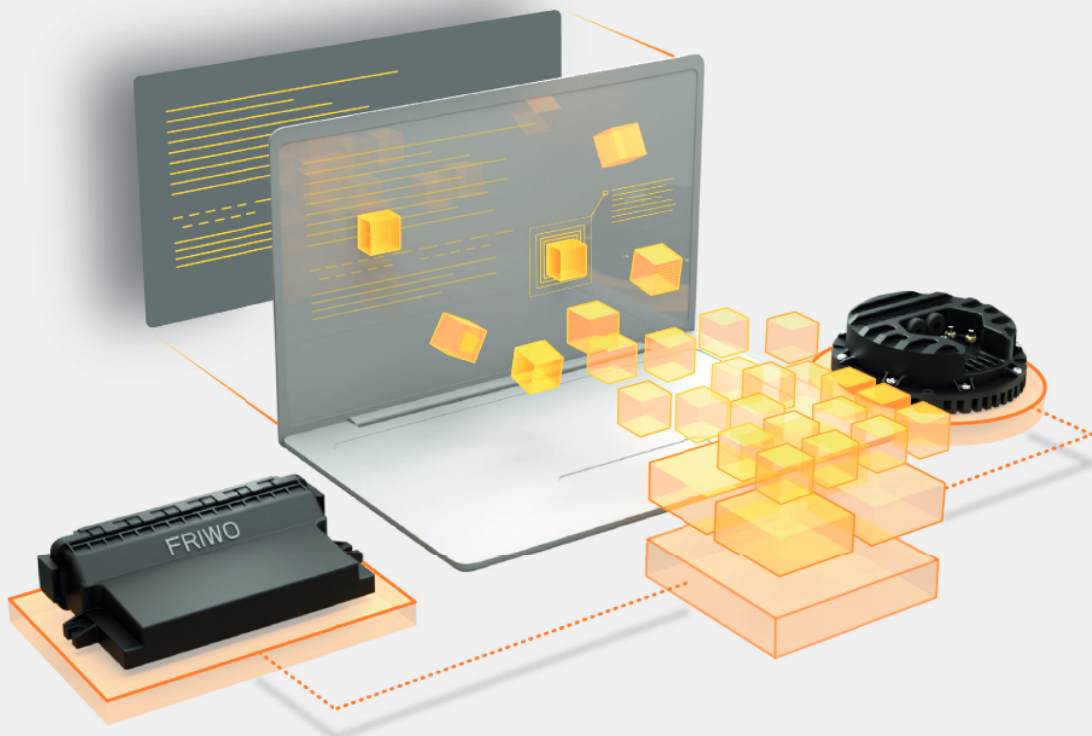
SOFTWARE DEVELOPMENT KIT

End-to-end Development Environment Setup Solution

MANUAL

Version 2.0

1 Table of Contents

2   List of Figures

3   List of Tables

4   Copyright

The content of this document is protected and may not be used, not even in part, for own publications and implementations. Exceptions require written approval.

5 Safety Information, Terms of Use and Liability Waiver

Warning!

Using the motor control unit and battery management systems referenced in this document can be dangerous. Please follow instructions it with extreme caution. Stop your application work if you encounter unexpected behavior.

Terms of use

1. Scope of the Terms of Use

1. FRIWO Gerätebau GmbH, von-Liebig-Str. 11, 48346 Ostbevern, Germany, („**FRIWO**") offers a platform that allows Customers, among other things, to get access to the Developer Platform (means access to all software or parts of the software relating to the Battery Management System „**BMS**", Motor Control Unit „**MCU**" etc.) of FRIWO to obtain a license to use the Software Development Kit („**SDK**"), provided by FRIWO. These Terms of Use ("**Terms of Use**") govern exclusively the legal relationship between FRIWO and the Customer, or the entity the Customer is representing, regarding the access to the Developer Platform and the use of .

2. FRIWO operates the Developer Platform for businesses only and does not accept consumers within the meaning of § 13 of the German Civil Code as Customers.

3. Customer, in case of an individual, warrants that he is lawfully able and has the capacity to enter into contracts (e.g. he is not a minor). If a person is entering into this agreement as a representative for an entity, such as the company for which the Customer is working for, such person warrants to FRIWO that their company is duly organized, validly existing and in good standing under the laws of the country in which it is organized or incorporated and that he has legal authority, permission, resolution or power of attorney to bind that company.

4. Terms and conditions of Customer or of third parties that deviate from or conflict with these Terms of Use, shall not apply, even if FRIWO does not expressly contradict these terms and conditions of the Customer.

5. FRIWO is entitled to commission any service from third parties (including Affiliates of FRIWO) acting as subcontractors.

6. The Listing of a Product by a Customer via the Developer Platform is not in the scope within these Terms of Use and requires a separate agreement between FRIWO and the Customer.

2. Subject Matter

1. FRIWO provides the access to the Developer Platform and the use of the on the basis of these Terms of Use.

2. The Developer Platform is a platform with various via which a Customer can, inter-alia, influence or control a connected electronic control system such as motor-controllers, battery-managament-units or other general electronic control devices.

3. As far as the Developer Platform and the are provided free of charge, there are no warranties and there is no right to access or use the Developer Platform and the without interruption. FRIWO does not warrant that the access to the Developer Platform will not be impaired by downtime, maintenance activities, further developments, updates and upgrades or malfunctions. FRIWO will use

reasonable efforts to ensure that the Developer Platform are as user-friendly as possible. Technical malfunctions (e.g. interruption in the electricity supply, hardware and software errors, and technical problems in the data lines) also can, however, give rise to temporary restrictions or interruptions.

3. Account, Users and Notices

1. The access to the Developer Platform requires an account. Customer must provide certain information required by FRIWO to open an account ("**Registration**") including the entire term of the access to the account.

2. The data requested by FRIWO at the time of the Registration must be stated completely and correctly, e.g. company name, current address, VAT identification number (if applicable), telephone number, valid e-mail address and a representative's name ("**Registration Data**"). The Registration of a legal person may only be carried out by an authorized natural person, who must be mentioned by name.

3. By submitting Registration Data, Customer is making FRIWO an offer to conclude a contract based on these Terms of Use. After submitting the Registration Data, these are validated by FRIWO for completeness and plausibility. If the Registration Data is correct, FRIWO shall decide on whether to accept Customer's offer at FRIWO's own discretion. If Customer's Registration is not confirmed by FRIWO by e-mail to the e-mail address provided by Customer within a reasonable time, Customer shall no longer be bound by the offer. If an e-mail confirmation is received, an limited user relationship shall be established,according to the lenght of the term choosen by the Customer, on the basis of these Terms of Use and Customer shall be entitled to use the Developer Portal in accordance with these Terms of Use.

4. In particular, Customer is responsible for Customer's Users' compliance with the obligations set out in Section 4.

5. All notices will be sent in electronic form to the email address associated with the Account.

6. Customer is liable for all activities performed under Customer's Account, to the extent that he is responsible for them.

4. Customer's Duties

1. Customer is obliged to provide in due time reasonable cooperation required from him. Therefore Customer is

   - obliged to immediately change all initial passwords which are assigned to him into passwords known only to the Customer. Customer is required to carefully store Customer's Access Credentials and protect them from unauthorized access. If a Customer becomes aware that a third party has used Customer's Access Credentials, he is required to notify FRIWO immediately;

   - obliged to correct the information in Customer's Account immediately, if, after the Registration, the indicated data changes.

2. Customer shall not (and shall procure that any User under Customer's Account shall not):

   - Intentionally access, tamper with, or use non-public portions of the Developer Platform or the technical delivery systems of FRIWO;

   - Use any robot, spider, scraper, or other similar automated data collection or extraction tools, program, algorithm or methodology to search, access, acquire, copy or monitor any portion of the Developer Portal other than via documented API endpoints;

- Post or transmit any file which contains viruses, worms, Trojan horses or any other contaminating or destructive features, or that otherwise interfere with the proper working of the Developer Platform;

- Attempt to decipher, decompile, disassemble, or reverse-engineer or otherwise attempt to discover or determine the source code of any software or any proprietary algorithm used, comprising or in any way making up a part of the Developer Portal;

- Attempt to probe, scan or test the vulnerability of the Developer Portal or breach or impair or circumvent any security or authentication measures protecting the Developer Platfrom;

- Frame or mirror the Developer Platform; or

- Use any device, software, or routine that interferes with any application, function, or use of the Developer Platform or is intended to damage, create undue load, detrimentally interfere with, surreptitiously intercept, or expropriate any system, data, or communication stored or transmitted therewith.

3. Customers' access to the Developer Platform and the use of it shall comply with all applicable laws, including copyright or trademark laws, antitrust and competition laws, export control laws, data protection laws, or other laws in any applicable jurisdiction or be in contravention of any agreement that the Customer has signed with a third party. Customer is responsible for making sure that its access to the Developer Portal and the use of is in compliance with the applicable laws and regulations.

5.   Usage Rights, Intellectual Property, Open Source

1. FRIWO owns all right, title, and interest in and to the Developer Platfrom and all related technology, software and intellectual property rights. It is not permitted to remove notices and notations on the Developer Platform that refer to confidentiality, copyrights, trademark rights, patent rights and other intellectual property rights. FRIWO hereby grants to Customer the timely limited, non-exclusive, non-sub-licensable, revocable and non-transferable right to access the Developer portal solely in accordance with this Terms of Use.

2. By using the Developer Platform Customer agrees that FRIWO is allowed to implement an error reporting to make statics of the error feedbacks. Personalised data will not be evaluated or saved. The provisons of section 9 hereto shall apply.

3. The Customer agrees that FRIWO is allowed to save a copy of the code which was programmed by the Customer. For further explanation please refer to the Quick Start Guide.

4. Provided that FRIWO makes new versions, updates, upgrades, modifications or extensions of the Developer Platform available or carries out other changes with respect to the Developer Platform , the provisions of Section 5 shall also apply thereto.

5. The Customer is only entitled to download Content and to print out Content, if there is a possibility to download or print Content as a functionality on the Developer Platform (e.g. by means of a download or print button).

6. Customer shall have no rights which are not explicitly granted to the Customer under these Terms of Use. Customer shall not be entitled to use the Developer Platform beyond the scope of use granted in these Terms of Use or make the Developer Platform available to third parties. In particular, it is not permitted to duplicate the Developer Platform and the or to provide it for use for a limited period of time, in particular not to lease it or loan it.

7. If Customer violates any of the provisions of this Section 5, FRIWO shall be entitled to and reserves the right to terminate Customer's access to the Developer Platform in accordance with Section 8, provided that such suspension remedies the violation. Customer shall be informed prior to such suspension. In case of repeated violations by the Customer, FRIWO shall be entitled to terminate the Account for cause, unless Customer is not responsible for the violation.

8. Free and Open Source Software (FOSS) components including the applicable FOSS license conditions used within the Developer Platform shall be illustrated on the Developer Platform, if providing via software as a service requires explicitly the fulfilment of FOSS license obligations.

6. Customer Content/Customized Software/ License Agreement

1. The Customer is allowed to modify the software provided by FRIWO on the Developer Platform for his intended purposes during the term of Registration (as defined in section 3) provided that the modification complies with all applicable statutory provisions („Customized Software").

2. In case of selling or other ways of distributing or using the Customized Software by the Customer, FRIWO and Customer shall reach a seperate agreement in regards of a license fee („License Agreement") for the parts of the initial software components developed by FRIWO.

3. The license fee will be invoiced automatically on a monthly basis. Therefor the selling or other ways of distribution of the Customized Software will be tracked On- and Offline by FRIWO.

4. All other provisions including but not limited to provision 5 shall remain unaffected.

7. Liability

1. FRIWO shall be liable in accordance with the statutory provisions

    -   in the event of intent or gross negligence;

    -   in accordance with the provisions of the German Product Liability Act or any applicable corresponding mandatory laws;

    -   within the scope of a guarantee given by FRIWO; and

    -   for loss of life or bodily injury/harm.

2. In the event of any property and financial damage caused negligently in any other way, FRIWO and persons engaged by it for the performance of its obligations shall be liable only in the event of a breach of a material contractual obligation, the amount being limited, however, to the damages foreseeable at the time of acceptance of these Terms of Use by the Customer and typical of the type of contract; material contractual obligations are those obligations which are necessary to the fulfillment of these Terms of Use, upon which the Customer may therefore rely ("**Material Obligation"**).

3. Notwithstanding the provision in Section 6.1, in the event of a negligent breach of a Material Obligation evidenced by Customer, the amount of FRIWO's liability for all damaging events occurring in the same contract year is limited to 100% of the fees incurred under the Account of the Customer in the year of the damaging event, however, at least Euro 50,000 per contract year. A contract year shall be the 12-month period starting on the date of Registration and every subsequent twelve-month period.

4. No-fault liability for damages per § 536a BGB shall be excluded for defects that existed at the time of conclusion of these Terms of Use.

5. FRIWO shall not be liable for loss of data personal injury or damage of property caused by misuse of the program or by use of improper settings.

6. Furthermore, FRIWO's liability is excluded.

7. The foregoing limitations of liability shall also apply in the event of fault by a person engaged by FRIWO in the performance of its obligations and to the personal liability of employees, representatives and corporate bodies of FRIWO.

8.   Indemnification

1. The Customer is obligated to indemnify, defend and hold FRIWO harmless for and from any third party claims related to

   - any breach of this Terms of Use, the Use Policy or any violation of any applicable law or regulation by the Customer;

   - the incorrect use of the Developmer Platform or software in particular but not limited to the misuse of the software program or by use of improper settings

   - the usage of the Developer Platform or the by the Customer in a manner in which it infringes or violates any third-party rights;

   - the use of Open Source Software by the Customer;

   - any infringement of data protection law respectively applicable or other data protection law relevant stipulations as agreed between the Parties herein or in another context by the Customer or through Customer Content;

   - any violation of applicable export and re-export control laws and sanctions regulations by the Customer; unless Customer is not responsible for the claim.

2. FRIWO shall notify the Customer without delay and leave to the Customer the defense of these claims. FRIWO will provide the Customer with all reasonable support. In particular, FRIWO will, as far as possible, provide the Customer with all the necessary information of the use and the possible processing of the Developer Portal , which are subject to this Terms of Use.

9.   Term and Termination

1. The term of this Terms of Use commences with the date of Registration and will remain in effect until terminated by FRIWO or by the Customer.

2. Termination for Convenience

   - The Customer may terminate this Terms of Use for any reason and close the Account for all , for which FRIWO provides an account closing mechanism by providing three (3) months' notice effective at the calendar month-end in a case of an annual runtime. In a case of a monthly runtime the Customer shall provide a fourteen (14) days notice effective at the calendar month-end.

- FRIWO may terminate this Terms of Use for any or no reason, at any time, by providing one (1) month notice effective at the calendar month-end.

3. The right of the Parties to terminate these Terms of Use for cause and without notice shall remain unaffected. Good cause shall be in evidence when one Party grossly breaches express obligations under these Terms of Use, in particular if the Customer utilizes the Developer Portal outside the scope of these Terms of Use and remains in breach after the end of a reasonable cure period following a warning notification by FRIWO.

4. Effect of Termination

    Upon the effective date of termination

    - the Developer Portal will no longer be accessible;

    - the Customer remains responsible for all fees and charges which have incurred through the date of termination, including fees and charges for in-process tasks completed after the date of termination; and

    - all authorizations of the Customer according to these Terms of Use end.

10.   Data Privacy

1. The Parties shall comply with the provisions of data protection law respectively applicable and bind their employees engaged in connection with this contractual relationship and the execution thereof to data privacy and confidentiality in accordance with the applicable laws, except to the extent that they are already under a general obligation to act accordingly.

2. FRIWO processes personal data in accordance with the Developer Portal.

11.   Collection of General Data and Information

1. The online-network infrastructure and software program collects a series of general data, electronic-control-unit-specific data and information when a data subject or automated system uses the program. This general data, electronic-control-unit -specific data and information are stored in the server log files. Collected may be (1) the browser types and versions used, (2) the operating system used by the accessing system, (3) the website from which an accessing system reaches our website (so-called referrers), (4) the data of the connected electronic-control-unit, (5) the date and time of access to the online site, (6) an internet protocol address (IP address), (7) the Internet service provider of the accessing system, and (8) any other similar data and information that may be used in the event of attacks on our information technology systems.

2. When using these general data and information, FRIWO does not draw any conclusions about the data subject. Rather, this information is needed to (1) deliver the content of the software program properly, (2) grant or deny access to the data oft the electronic-control-unit, (3) ensure the long-term viability of our information technology systems and website technology, and (4) provide law enforcement authorities with the information necessary for criminal prosecution in case of a cyber-attack or violation of street-legal regulation. Therefore, FRIWO analyzes anonymously collected data and information statistically, with the aim of increasing the data protection and data security of our enterprise, and to ensure an optimal level of protection for the personal data we process. The anonymous data of the server log files are stored separately from all personal data provided by a data subject.

12. Confidentiality

1. The Parties agree to keep all Confidential Information that becomes known to them while accessing Developer Portal and using the strictly confidential and only use such information for the contractually agreed purposes. "**Confidential Information**" for the purpose of this use shall mean information, documents, details and data, which is marked as such or that reasonable should be understood to be confidential given the nature of the information.

2. If Confidential Information within the above definition is requested by a public authority, then the other Party – to the extent that it is legally allowed - shall be informed without delay and before the Confidential Information is supplied to the public authority.

3. The obligations under Section 16.1 shall lapse for such information or parts thereof with respect to which the Party receiving the information evidences that it

   - was known to that Party or generally accessible prior to the date of receipt or became known from a third Party after the date of receipt in a lawful manner and without any confidentiality obligation;

   - was already known to the general public or was generally accessible prior to the date of receipt;

   - became known to the general public or became generally accessible after the date of receipt without the party receiving the information being responsible for this,

   - has waived its right to confidentiality in respect of which the notifying Party has waived its right to confidentiality by means of a written declaration to the receiving Party.

4. Each Party may disclose Confidential Information to its employees and Affiliates if

   - this is necessary for the purpose of the fulfillment of these Terms of Use,

   - Confidential Information will only be disclosed to the extent that it is necessary for integration in these Terms of Use („need to know"), and

   - the recipient of Confidential Information is obliged to maintain confidentiality in accordance with Section 16.5.

5. Each Party is obliged to commit employees and Affiliates within the meaning of Section 16.4 to maintain confidentiality in accordance with the provisions of Section 16.1 to 16.3 insofar as they are not already otherwise obliged to do so by virtue of contractual and/or statutory provisions to an extent that is at least equivalent to the level of protection resulting from Section 16.1 to 16.3.

6. The rights and obligations under Section 16 shall not be affected by the termination of an Account, and for as long as there is no evidence of an exception under Section 16.3.

13. Changes

a. At any time, FRIWO shall have the right to rebrand the Developer Platform, modify the online-network infrastructure, make new available either free of charge and/or for a fee and as well as to change these Terms of Use and the Use Policy. FRIWO reserves the right to modify these Terms of Use and the Use Policy. Customer shall be notified of changes by email no later than thirty (30) calendar days before the planned effective date of the changes. If Customer does not object within thirty (30) days of receipt of the notification, then the changes shall be deemed to have been effectively agreed as from the expiry date of the time limit. In

the event of an objection, the contractual relationship shall be continued subject to the conditions applying hitherto. If an objection is raised, FRIWO is entitled to terminate these Terms of Use subject to a one (1) month notice period. Customer shall be advised of its right to object and of the consequences in the change notification.

b. Editorial changes to these Terms of Use, i.e. changes, which do not affect the contractual relationship, such as correcting typing errors, will be made without notifying the Customer.

14. Applicable law and place of jurisdiction

a. The contractual relationships between the Parties shall be governed by the laws of the Federal Republic of Germany. Application of the UN Convention on Contracts for the International Sale of Goods (CISG) is excluded.

b. The exclusive place of jurisdiction for any legal disputes arising from or in connection with these Terms of Use shall be Münster, Germany.

15. Miscellaneous

a. Force Majeure: Operational disruptions caused by force majeure or other unavoidable events beyond FRIWO's control,

    i. which could not be averted with reasonable effort; and

    ii. which could not have been foreseen even when exercising with extreme care, and

    iii. which make FRIWO's obligations under these Terms of Use considerably more difficult or completely or partially impossible,

such as strikes, lockouts, exceptional weather conditions, operational or traffic disruptions and transport obstructions, discharge FRIWO from its obligations under these Terms of Use for the duration of such an event. Any statements or notifications to be made by Customer to FRIWO after Customer's acceptance of these Terms of Use (e.g. setting of time limits, notification of defects) must be provided in text form in order to be effective.

b. Customer may not assign any or all portion of its obligations without the prior written consent of FRIWO, not to be unreasonably withheld.

c. Should any provision of these Terms of Use be or become invalid or unenforceable, this shall, however, not affect the remaining provisions.

FRIWO Gerätebau GmbH

6 General Information

6.1 Usage

The FRIWO Software Development Kit (FRIWO SDK) is an application for developing and generating software based on the standard firmware for FRIWO E-Mobility products. The FRIWO firmware is modular with a defined interface of inputs and outputs. This allows individual modules to be replaced by adapted variants of the module or completely new developments of the module.

The adapted modules are compiled together with the basic firmware and released as signed software via a connection to FRIWO servers. Thus, the software can be flashed to released FRIWO hardware. The flash process is not part of the SDK, but takes place in the **FRIWO Enable Tool**.

https://friwo.link/qs/enable-tool

The SDK is typically used for the following purposes.

6.1.1 Simple Modification

- FRIWO offers a standard version for each customizable module, as it is also included in the series software
- On this basis, minor adjustments of values and parameters are possible
- Only existing parameters are edited, there are no new variables created

6.1.2 New Feature Development

- New parameters are created, which can also be made available in the Enable Tool
- New functions are introduced, which consist of a combination of inports and change the behavior of the outports compared to the standard firmware.
- On request, additional inports can be enabled to develop software that enables features tailored to new use cases.

6.2 Nomenclature of parameters and measured values

This document describes parameters and measured values that influence or make measurable the behavior of the software at runtime.

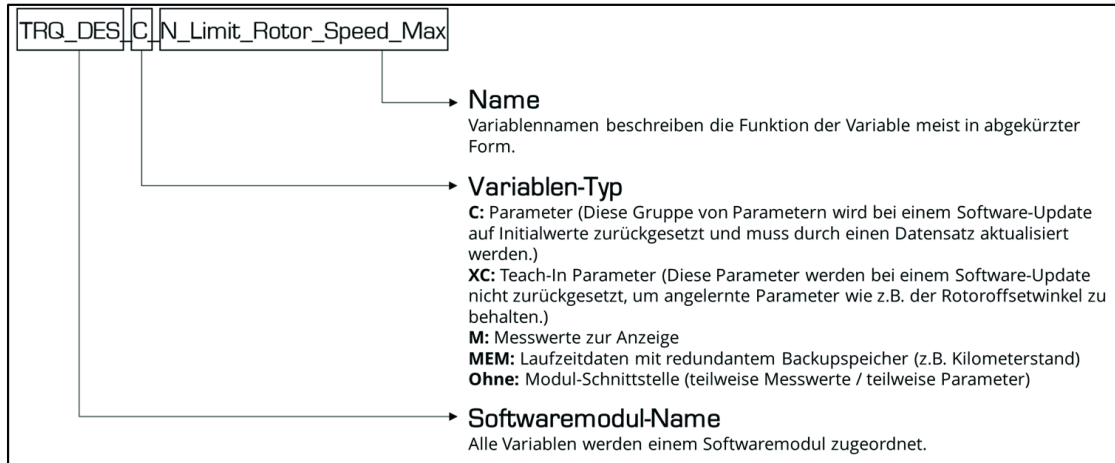The nomenclature is based on the following principle:

TRQ_DES_C_N_Limit_Rotor_Speed_Max

Name
Variablennamen beschreiben die Funktion der Variable meist in abgekürzter Form.

Variablen-Typ
C: Parameter (Diese Gruppe von Parametern wird bei einem Software-Update auf Initialwerte zurückgesetzt und muss durch einen Datensatz aktualisiert werden.)
XC: Teach-In Parameter (Diese Parameter werden bei einem Software-Update nicht zurückgesetzt, um angelernte Parameter wie z.B. der Rotoroffsetwinkel zu behalten.)
M: Messwerte zur Anzeige
MEM: Laufzeitdaten mit redundantem Backupspeicher (z.B. Kilometerstand)
Ohne: Modul-Schnittstelle (teilweise Messwerte / teilweise Parameter)

Softwaremodul-Name
Alle Variablen werden einem Softwaremodul zugeordnet.

Figure 1: Variable education.

Parameters and measured values can be used, modified, or generated with the SDK. Before using the SDKs, read the corresponding module description of the modules used carefully.

6.3 Important notes on changing parameters

Changing parameters can have a negative effect on the behaviour of the controller. Please observe the following notes when making adjustments:

- Start your application work only after you have read this document and the module descriptions completely and understood the function of the parameters.

- Change parameters in as small steps as possible and avoid unconrolled programming loops. Note that you are working on a real-time system!

- Great importance was attached to your safety during the development of the motor control software. Incorrect inputs are mostly intercepted and parameter changes are limited to safe values. This is usually done outside your module.

- Some parameters are monitored by superimposed protective functions, so that only limited changes are possible during motor operation.

- In order to protect you and your devices from damage, we limit your working space and the output variables inside the SDK. The warning and error handlers inside our FRIWO Firmware will always monitor the whole system. Due to security reasons the following ANSI C commands or structures are not allowed:

  - External Variables

  - External Functions

  - Any code that tries to access Registers

- Please be aware, that we cannot guarantee full protection when using custom code.

7 Installing the program on a Windows PC

7.1 System and Software Requirements

The SDK requires an x86-based PC with the following minimum requirements:

- Screen resolution: at least 1024 x 720 pixels.

- CPU: 1.8 GHz Intel Core Duo or better.

- Working memory: min. 2GB

- Hard disk capacity: Approx. 250MB

- Windows Operating Systems: Windows 7 or Windows 10 - You need local Administrator access!

- Keil Compiler v4.7 incl. License, **please visit keil.com**
  https://friwo.link/m/keil

7.2 Installation Guide

Please see our SDK Quickstart Guide to install the software on your machine.
https://friwo.link/m/quickstart

7.3 The first compilation of your own software

Please see our SDK Quickstart Guide to see how to compile a software for the first time on your machine.
https://friwo.link/m/quickstart

8   Features

8.1 Overview Graphical User Interface

This chapter describes the different views of the SDK. Depending on the available modules and the selected frameworks, it may look different on your PC.

Every element of the View is described in the table underneath it.
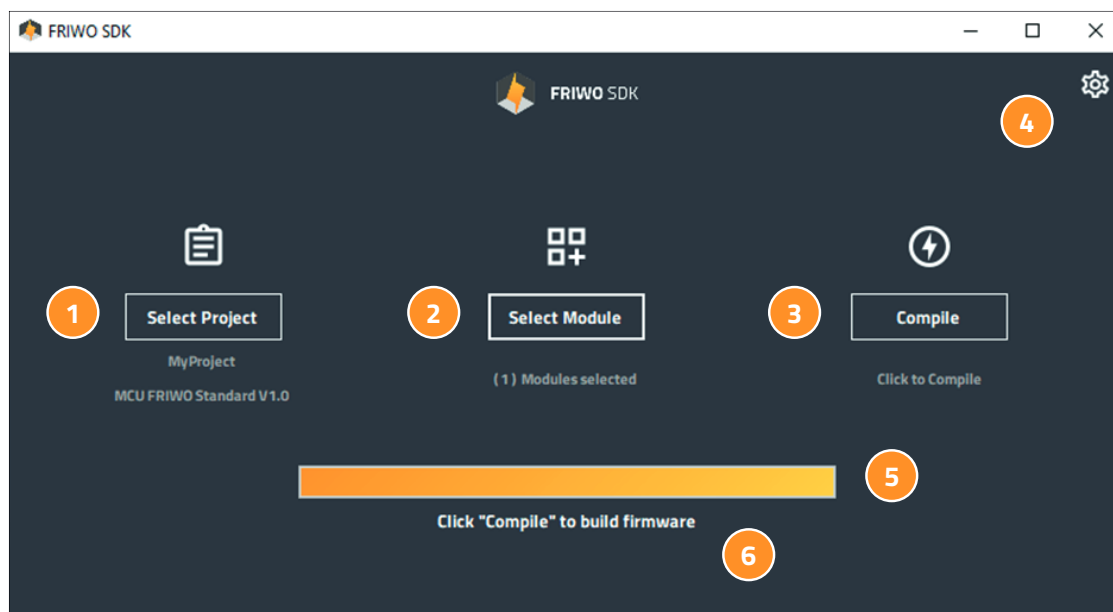
Figure 1: Main View graphical user interface

Features of the Main View of the graphical user interface	
element	**description**
1	Select Project: Here you load an existing project or create a new one, with different Frameworks.
2	Select Module: Choose the modules that you want to customize
3	Compile: Start the compilation process, if your project and modules are selected and you verified your code
4	Settings: Change different settings, like Keil Compiler location or error reporting.
5	Progress Bar: During the compilation process, it indicates the progress
6	Message Box: Shows what to do next, progress information during compilation and error codes if something went wrong

Table 1: Features of the Main View of the graphical user interface

When you click on "Select Project" in Figure 1, you will see the following view:
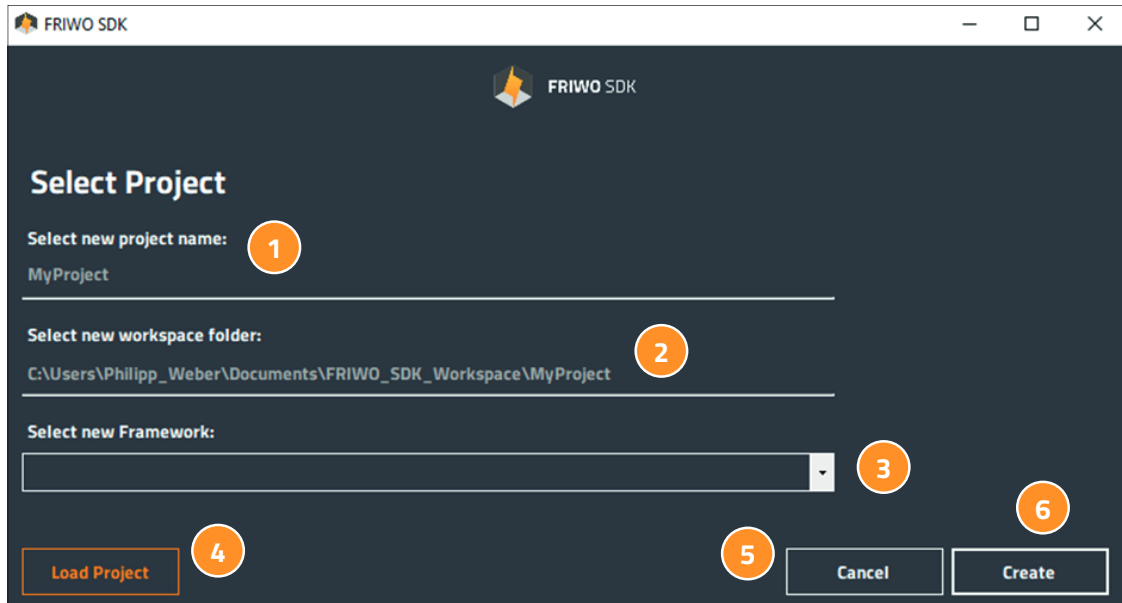
Figure 2: -Select Project- View

Features of the -Select Project- View	
element	**description**
1	Give your new project a name
2	Select the workspace folder, where you want to store all your project data including the generated software after compilation.
3	Select the Framework that you want to work with, in your project. Please see chapter 8.2
4	Load an existing project. Your framework, selected modules and code will be automatically loaded.
5	Cancel the select project process and go back to main view. Your last used project will be selected by default.
6	Create the new project & the view will automatically switch back to main view.

Table 2: Features of the -Select Project- View

When you click on "Select Module" in Figure 1, you will see the following view:

Figure 3: -Select Module- View

Features of the -Select Module- View	
element	**description**
1	When you select the module, it will use your custom code. If you select it for the first time, the default code will be copied to your workspace folder, see chapter 8.3. If you deselect the module, the default FRIWO firmware will be used for compilation.
2	Indication how many modules you are able to select. Equals the number of elements in the module list.
3	Indication how many modules you have selected.
4	Click on this button to overwrite your custom code with it's FRIWO Default. **ATTENTION:** This can't be undone! Your customized code will be lost!!
5	Click on this button to open the module folder inside your selected project
6	Press OK, to go back to the Main View.

Table 3: Features of the -Select Module- View

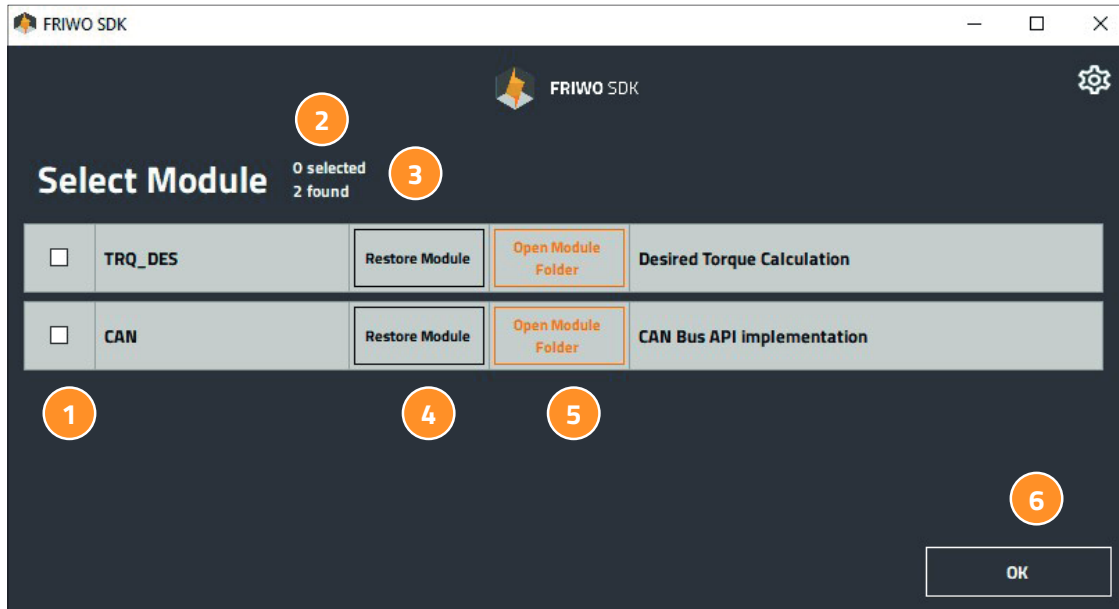When you click on "Settings" in Figure 1, you will see the following view:
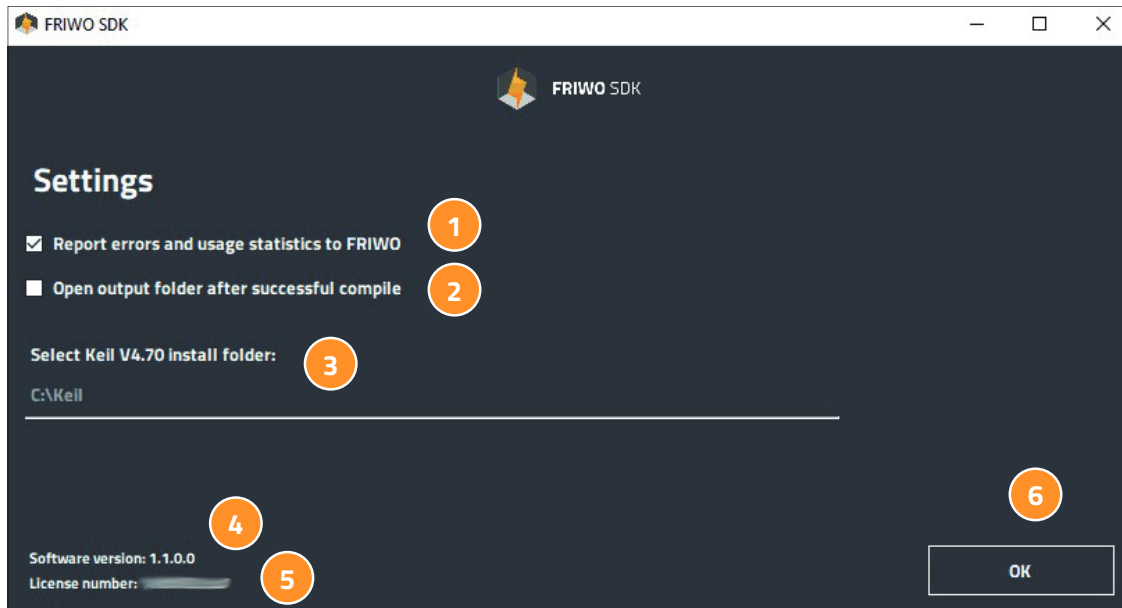
Figure 4: -Settings- View

element	description
	Features of the -Settings- View
1	You can select whether you want to report errors and statistic data to FRIWO or not. Per default it is activated. **ATTENTION: We highly recommend to activate this option.**
2	If checked, after the successfull compilation process, the resulting firmware folder will be opened automatically
3	Select the path to your Keil Compiler. Per default it is located under: C:\Keil
4	Indication of the SDKs software version that you are using. Updates will be provided automatically, when available
5	Indicates your SDKs license number
6	Press OK, to go back to the Main View.

Table 4: Features of the -Settings- View

8.2 SDK Framework

An SDK Framework describes the setup, for which you want to write your software. It defines for which hardware you want to write your own software, and which basic series software version you want to build on. Every project can only have one framework.

8.3 SDK Workspace Folder

When you create a new project, a new project folder is created inside your SDK workspace. When you open the project folder, you should see something like this:

📁 FirmwareRelease	25.03.2021 15:12	Dateiordner	
📁 module_TRQ_DES	24.03.2021 14:14	Dateiordner	
⬛ MyProject.sdkproj	25.03.2021 17:59	Friwo SDK Project ...	1 KB

Figure 5: Overview Project Folder

In the first folder „FirmwareRelease" is your customized firmware stored, after the first successful compilation. In the "module_TRQ_DES" folder, your custom code of the module "Desired Torque – Trq_Des" will be stored. When you select the module for the first time in your project, this folder will be created automatically and filled with the standard series software. The file "MyProject.sdkproj" contains all project information and is named after the project. With a double-click on it, FRIWO SDK will launch automatically, when installed and load all project data from the last time you closed it. When you modify project data inside the SDK, it will be stored automatically in the project file. There can be multiple "module_XXX" folders, depending how many modules you have selected to be customized.

When you enter the FirmwareRelease folder, you should see something like this:

📁 100999054	24.03.2021 14:24	Dateiordner
📁 100999065	25.03.2021 15:09	Dateiordner
☑ 📁 100999066	25.03.2021 15:12	Dateiordner

Figure 6: Firmware Release Folder

During the compilation process, your firmware will get signed by FRIWO License Servers and will obtain a software release number. Each folder stands for a successful compilation and is named after it's specific release numbers. Please do not change the folder name.

When opening the blue marked folder from Figure 6, the folder should look like this:

📄 Emerge_MC_V100999066.eef	25.03.2021 15:12	EEF-Datei
📄 Emerge_MC_V100999066.xml	25.03.2021 15:12	XML-Dokument
📄 VARIABLES_CONFIG_MC_DEV_V100999066.x...	25.03.2021 15:12	XML-Dokument
📄 VARIABLES_CONFIG_MC_SDK_default_V1009...	25.03.2021 15:12	XML-Dokument

Figure 7: Firmware Release Files

https://friwo.link/m/enable-tool

Here you can see, the flash file .eef, which you can use together with the **FRIWO Enable Tool** to flash your firmware onto your device. Using the XML files, you can load your custom view into the Enable Tool. Your custom variables that you previously declared inside the **SDK Application Guide** can be monitored or modified live.

https://friwo.link/m/application-guide

When you open the "module_Trq_Des" folder, you should see this:

TRQ_DES_custom.c	24.03.2021 14:14	C Source
TRQ_DES_custom.h	24.03.2021 14:14	C/C++ Header
TRQ_DES_variables.xml	24.03.2021 14:14	XML-Dokument

Figure 8: Module Trq_Des Files

The folder in Figure 8 shows your customized ANSI C code in form of an .c and .h files. Furthermore, it contains an XML file, where the global variables are stored, that you can read and write. These variables are part of the FRIWO basic firmware and are used as interface to your module. They are specific for every module, for further information have a look at the **SDK Module Description**.

https://friwo.link/m/module-description

8.4 Error Reporting

To constantly improve the features and user experience of our products, we use an advanced error reporting system. To provide you quick support or help you in case of errors, we upload per default log files to FRIWO servers. These log files contain usage statistics and detailed information about the compilation process. During the compilation process the log output of the Keil Compiler itself is merged into the SDK log files. Please always keep the error reporting activated. When contacting FRIWO, please provide the error code, so that we can help you as quick as possible.

8.5 SDK Update

When a new version of the SDK is available, you will be notified on the start of the SDK Application.
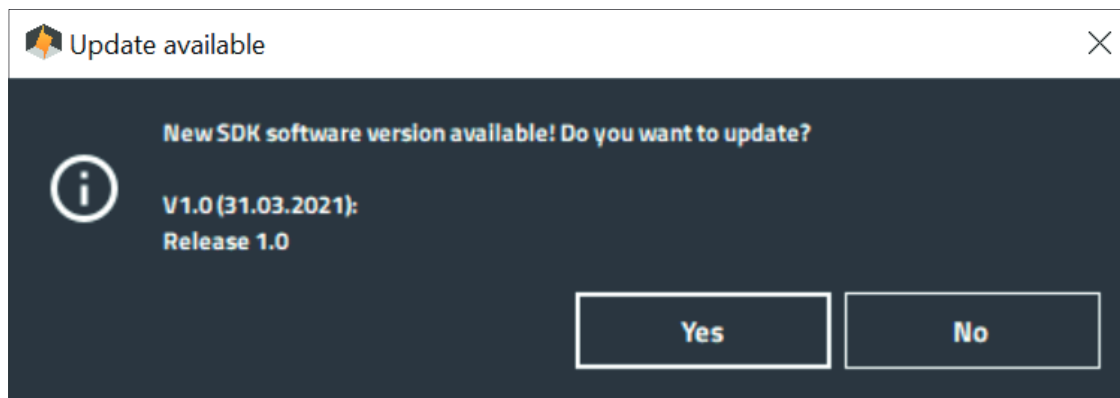
Figure 9: Confirm Update of SDK

Press "Yes" to update to the latest Applikation version. You don't have do anything afterwards. All projects and configurations will be retrieved from the old version. The SDK will now download all necessary data from the FRIWO Servers. During this time, please do NOT shut down your computer or disconnect it from the internet!

After the successful download, you will see the following window:
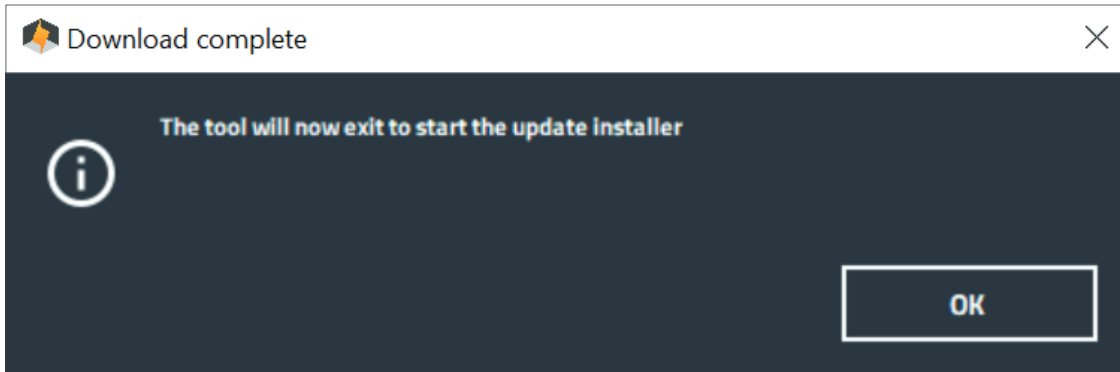
Figure 10: Download complete confirmation

Now press „OK" to restart your SDK. The known FRIWO SDK Installer from the **SDK Quickstart Guide** will now start automatically and guide you as usual through the installation process. After the successful installation process, the latest SDK version can be used.

9   API

To write your own custom code, you need access to internal variables from protected software modules. The access to these modules is restricted, but with API functions, you can access the signals. For a detailed instruction on how to access a signal via API, please refer to the corresponding module description that you want to customize.

This chapter lists all signals accessible with FRIWO SDK. The column "API" denotes, through which API the specific signal can be read or written. In the description column, you can find the defined values for each signal.

9.1 API VARIABLE DESCRIPTION: MOTOR CONTROL UNIT (MCU)

Applicable Frameworks:
- MCU FRIWO Standard V1.0
- MCU FRIWO Standard V1.1

Signal	API	Datatype	Min	Max	Unit	Description
APP_Throttle_Signal_Channel	trqdesApi_Get	Float32	0	5	–	Selection of input channel which will be used as throttle input: 0 = not selected / off 1 = AIN1 2 = AIN2 3 = PWM@DIN2 4 = CAN-Bus 5 = USB (default)
APP_Brake_Signal_Channel	trqdesApi_Get	Float32	0	7	–	Selection of input channel which will be used as brake input: 0 = not selected / off 1 = AIN1 2 = AIN2 3 = PWM@DIN2 4 = CAN-Bus 5 = USB (default) 6 = DIN1 7 = DIN2
APP_Reverse_Gear_Signal_Channel	trqdesApi_Get	Float32	0	4	–	Selection of input channel which will be used as reverse gear input: 0 = not selected / off 1 = DIN1 2 = DIN2 3 = CAN-Bus 4 = USB (default)
APP_Disp_Ride_Mode	trqdesApi_Get canApi_Get	Float32	0	3	–	Selected ride-mode
APP_Boost_Info	canApi_Get	UInt32	0	$2^{32}-1$	–	Bitcoded configuration of Boost feature: 1 = Boost function generally enabled 2 = Ready to use 4 = Fully charged 8 = Ext. boost limits active 16 = Canceled 32 = Cool down 64 = Collecting 128 = Depleting 256 = Depleted
APP_Boost_Avail_Rel	canApi_Get	Float32	0	100	%	Available boost relative to the maximum collectable amount
APP_Boost_Avail_As	canApi_Get	Float32	0	65535	As	Available boost in ampere-seconds

Signal	API	Datatype	Min	Max	Unit	Description
AIN1_Throttle	trqdesApi_Get	Float32	0	100	%	Mapped relative torque request from Analog Interface 1
AIN2_Throttle	trqdesApi_Get	Float32	0	100	%	Mapped relative torque request from Analog Interface 2
BSW_IO_F_CAN_BSW_BusOff	canApi_Get	Int16	0	1	-	BusOff status flag of CAN peripheral: 0 = BusOn 1 = BusOff
BSW_IO_F_CAN_BSW_Passive	canApi_Get	Int16	0	1	-	BusPassive status flag of CAN peripheral: 0 = BusActive 1 = BusPassive
BSW_IO_F_CAN_BSW_Warning	canApi_Get	Int16	0	1	-	BusWarning status flag of CAN peripheral: 0 = BusReady 1 = BusWarning
CAN_EXT_Reverse_Gear	trqdesApi_Get canApi_Set	Float32	0	1	-	Reverse gear selection from CAN-Bus
CAN_EXT_Torque_Request	trqdesApi_Get canApi_Set	Float32	-100	100	%	Relative torque request from CAN-Bus
CAN_EXT_Alive_Counter	canApi_Set	UInt8	0	255	-	CAN message counter should increase with every message received; Missing messages can be handled by the errorhandler
CAN_EXT_State_Request	canApi_Set	Float32	0	2	-	State request via CAN: 0 = Init/Standby 1 = Request active torque control
CAN_EXT_Ride_Mode	canApi_Set	Float32	0	2	-	Ride-mode selection from CAN-Bus
CAN_EXT_ROC_Start	canApi_Set	Float32	0	1	-	Start the automatic rotor offset calibration
CAN_EXT_Boost_Enable	canApi_Set	Float32	0	1	-	Enable extended boost limits
CAN_EXT_Rotor_Speed_Max	canApi_Set	Float32	0	2400	1/s	Mechanical rotor speed limit for positive and negative speeds throttle
CAN_EXT_Skip_Signal_Checks	canApi_Set	Float32	0	1	-	Skip the signal checks for thortle and brake for direct initialisation
CAN_Immo_Unlock_Request_Lower	canApi_Set	UInt32	0	$2^{32}-1$	-	Immobilizer key; Used to control the powerstage if the activation mode is setup accordingly
CAN_Immo_Unlock_Request_Higher	canApi_Set	UInt32	0	$2^{32}-1$	-	Immobilizer key; Used to activate the powerstage if the activation mode is setup accordingly
CAN_BMS_Pack_Voltage	canApi_Set	Float32	0	500	V	Actual battery voltage received from BMS
CAN_BMS_Pack_Current	canApi_Set	Float32	-500	500	A (DC)	Battery current; negative values indicate discharge current; positive values indicate charge current

Signal	API	Datatype	Min	Max	Unit	Description
CAN_BMS_Errorcode	canApi_Set	UInt32	0	$2^{32}-1$	-	Errorcode of battery, received via CAN-Bus: 1 = Overtemperature (MOSFET1) 2 = Overtemperature (MOSFET2) 4 = Charge Current 8 = Discharge Current 16 = Overvoltage (Pack) 32 = Overvoltage (dynamic) 64 = Current-sensor offset 128 = Dataset missing/invalid 256 = Cell-monitoring (CRC) 512 = External Enable 1024 = Cell-monitoring (Alert) 2048 = Cell-monitoring (Fault) 4096 = MOSFET-Failure 8192 = Overload during pre charge 16384 = Output voltage too high 32768 = Undervoltage (Pack) 65536 = Discharge Voltage 131072 = Undervoltage (Cell) 262144 = Overvoltage (Cell) 524288 = Dynamic overcurrent 1048576 = Overtemperature in charge-mode 2097152 = Overtemperature in discharge-mode 4194304 = Undertemperature in charge-mode 8388608 = Undertemperature in discharge-mode 16777216 = Current-flow in passive state 33554432 = CAN timeout 67108864 = Cell-connection check failed 134217728 = Selftest required 268435456 = Permanent shutdown 536870912 = external current-sensor monitoring 1073741824 = external voltage-sensor monitoring 2147483648 = Keylock
CAN_BMS_Charge_Plug_Detection	canApi_Set	Float32	0	1	-	Status of chargeplug detection: 0 = chargeplug not detected 1 = chargeplug detected
CAN_BMS_State	canApi_Set	Float32	0	65535	-	BMS State: 0 = Init 1 = Init 2 = Init 3 = Init 4 = Init 5 = Standby-mode 6 = Discharge-mode 7 = Charge-mode 10 = Error 11 = Critical Error 50 = Alarm Mode 98 = Announce Shutdown 99 = Shutdown 100=Off

Signal	API	Datatype	Min	Max	Unit	Description
CAN_BMS_SOC	canApi_Set	Float32	0	100	%	State of charge of the battery calculated by BMS and received via CAN-Bus
CAN_BMS_State_of_Health	canApi_Set	Float32	0	100	%	State of Health of the battery; 0 = 80% of nominal full charge capacity left
CAN_BMS_Remaining_Capacity	canApi_Set	Float32	0	$4*10^9$	mAh	Remaining capacity of the battery
CAN_BMS_Fullcharge_Capacity	canApi_Set	Float32	0	$4*10^9$	mAh	Capacity of fully charged battery
CAN_BMS_TEMP_Powerstage1	canApi_Set	Float32	-50	300	°C	MOSFET temperature of the battery measured at sensor 1
CAN_BMS_TEMP_Powerstage2	canApi_Set	Float32	-50	300	°C	MOSFET temperature of the battery measured at sensor 2
CAN_BMS_TEMP_MCU	canApi_Set	Float32	-50	300	°C	Temperature of the microcontroller of the BMS
CAN_BMS_TEMP_Cell1	canApi_Set	Float32	-50	300	°C	Cell-temperature of the battery measured at sensor 1
CAN_BMS_TEMP_Cell2	canApi_Set	Float32	-50	300	°C	Cell-temperature of the battery measured at sensor 2
CAN_BMS_Max_Charge	canApi_Set	Float32	-500	500	A (DC)	Max charge current of the battery
CAN_BMS_Max_Discharge	canApi_Set	Float32	-500	500	A (DC)	Max discharge current of the battery
CAN_BMS_Max_Voltage	canApi_Set	Float32	0	255	V	Upper charge voltage threshold of the battery
CAN_BMS_Min_Voltage	canApi_Set	Float32	0	255	V	Lower discharge voltage threshold of the battery
CAN_BMS_Warning_Status	canApi_Set	Float32	0	1	-	Shows if battery management system encountered a warning: 0 = No warning 1 = Warning active
CAN_BMS_Pending_HV_Shutdown	canApi_Set	Float32	0	1	-	Battery voltage-output shutdown imminent; Sent very shortly before the battery deactives the output voltage
CAN_BMS_Pending_Bordnet_Shutdown	canApi_Set	Float32	0	1	-	Bordnet shutdown pending; Sent shortly before the battery enters deepsleep
CAN_BMS_PushButton_ShortPress_Detected	canApi_Set	Float32	0	1	-	User Push Button was pressed short
CAN_BMS_PushButton_LongPress_Detected	canApi_Set	Float32	0	1	-	User Push Button was pressed long
CAN_BMS_PushButton_SuperLongPress_Detected	canApi_Set	Float32	0	1	-	User Push Button was pressed superlong
CAN_BMS_PushButton_SuperLongPress_Ongoing	canApi_Set	Float32	0	1	-	User Push Button is pressed constantly
CAN_Custom_Timeout_Bit27	canApi_Set	UInt8	0	1	-	Bit 27 of timeout codeword available for individual use

Signal	API	Datatype	Min	Max	Unit	Description
CAN_Custom_Timeout_Bit28	canApi_Set	UInt8	0	1	-	Bit 28 of timeout codeword available for individual use
CAN_Custom_Timeout_Bit29	canApi_Set	UInt8	0	1	-	Bit 29 of timeout codeword available for individual use
CAN_Custom_Timeout_Bit30	canApi_Set	UInt8	0	1	-	Bit 30 of timeout codeword available for individual use
CAN_Custom_Timeout_Bit31	canApi_Set	UInt8	0	1	-	Bit 31 of timeout codeword available for individual use
CAN_Disp_Reset_Trip	canApi_Set	Float32	0	1	-	Trip reset via CAN-display (long button press in trip display): 0 = No action 1 = Reset
CAN_Dyno_Torque	canApi_Set	Float32	-1000	1000	Nm	Measured torque from dynamo-meter
CAN_Dyno_DC_Current	canApi_Set	Float32	-1000	1000	A (DC)	Battery current from additional external sensor
CAN_Dyno_DC_Voltage	canApi_Set	Float32	-1000	1000	V	Battery voltage from additional external sensor
CAN_Dyno_Elec_Power_Input	canApi_Set	Float32	-1000	1000	kW	Electrical power from additional external sensor
DIN_DIN1_Signal	trqdesApi_Get	Float32	0	1	-	Relative torque request from DIN1
DIN_DIN2_Signal	trqdesApi_Get	Float32	0	1	-	Relative torque request from DIN2
ERR_Errorcode	canApi_Get	UInt32	0	$2^{32}-1$	-	System Errorcode: 1 = Hardware init pending 2 = MOSFET Error 4 = Over current (battery current) 8 = DC-Link voltage too low 16 = DC-Link voltage too high 32 = Overtemperature (MOSFET) 64 = Accelerator-pedal signal range 128 = Motor blocking detection 256 = DC-Link voltage (dynamic over-/undervoltage) 512 = Motor sensor error 1024 = Current-sensor offset 2048 = Motor overspeed 4096 = Current-setpoint monitoring 8192 = Rotor-offset calibration failed 16384 = Dataset missing/is invalid 32768 = CAN-timeout 65536 = BMS-announced shutdown 131072 = Over current (Motor-current) 262144 = Hardware-variant coding 524288 = Flux-angle monitoring 1048576 = Overtemperature micro-controller 2097152 = CAN BZ counter 4194304 = Brake Monitor 8388608 = Motor Phase Connection 16777216 = Internal LV Voltage 33554432 = Overtemperature (Motor) 67108864 = Overtemperature (Aux-input) 134217728 = Immobilizer

Signal	API	Datatype	Min	Max	Unit	Description
ERR_MEM_Trace_0_Errorcode	canApi_Get	UInt32	0	$2^{32}-1$	-	System Errorcode trace 0 stored in Memory: 1 = Hardware init pending 2 = MOSFET Error 4 = Over current (battery current) 8 = DC-Link voltage too low 16 = DC-Link voltage too high 32 = Overtemperature (MOSFET) 64 = Accelerator-pedal signal range 128 = Motor blocking detection 256 = DC-Link voltage (dynamic over-/undervoltage) 512 = Motor sensor error 1024 = Current-sensor offset 2048 = Motor overspeed 4096 = Current-setpoint monitoring 8192 = Rotor-offset calibration failed 16384 = Dataset missing/is invalid 32768 = CAN-timeout 65536 = BMS-announced shutdown 131072 = Over current (Motor-current) 262144 = Hardware-variant coding 524288 = Flux-angle monitoring 1048576 = Overtemperature micro-controller 2097152 = CAN BZ counter 4194304 = Brake Monitor 8388608 = Motor Phase Connection 16777216 = Internal LV Voltage 33554432 = Overtemperature (Motor) 67108864 = Overtemperature (Aux-input) 134217728 = Immobilizer
IHS_Vibration_Detected	trqdesApi_Get	Float32	0	1	-	Vibration detected on hall-sensors; StateList;1=Vibration detected
INFO_ODO_Total_Kilometers	canApi_Get	Float32	0	$4*10^9$	km	Total mileage during the entire controller lifetime
INFO_ODO_Trip_Kilometers	canApi_Get	Float32	0	$4*10^9$	km	Mileage since last trip-reset [km]
INFO_Motor_Current	canApi_Get	Float32	-1000	1000	A (AC)	Motor-current
INFO_Motor_Current_Iq	canApi_Get	Float32	-1000	1000	A (AC)	Current on torque-axis
INFO_Motor_Current_Id	canApi_Get	Float32	-1000	1000	A (AC)	Current on flux-axis
INFO_DC_Current	canApi_Get	Float32	-1000	1000	A (DC)	DC-Link current
INFO_Voltage_DC_Link	canApi_Get	Float32	0	1000	V	DC-Link voltage
INFO_Rotor_Speed	canApi_Get	Float32	-2400	2400	1/s	Actual mechanical rotor speed
INFO_Vehicle_Speed	canApi_Get	Float32	$-4*10^9$	$4*10^9$	km/h	Vehicle speed
INFO_Remaining_Distance	canApi_Get	Float32	0	32767	km	Remaining distance in current ride-mode

Signal	API	Datatype	Min	Max	Unit	Description
INFO_Consumption_Ave_Trip	canApi_Get	Float32	-32767	32767	Wh/km	Average consumption since last trip-reset
INFO_Ah_Pos	canApi_Get	Float32	0	$2*10^9$	Ah	Current-meter for the consumed current during the entire controller lifetime
INFO_Ah_Neg	canApi_Get	Float32	0	$2*10^9$	Ah	Current-meter for the recovered current during the entire controller lifetime
INFO_Rel_Torque_Setpoint	canApi_Get	Float32	0	100	%	Relative torque related to setpoint of the torque controller
INFO_Rel_Torque_Max	canApi_Get	Float32	0	100	%	Relative torque related to the maximum possible value in the actual ride mode
INFO_Rel_Torque_Mapping	canApi_Get	Float32	0	100	%	Relative torque related to the actual value of the motor curve mapping
PWMI_Throttle	trqdesApi_Get	Float32	0	100	%	Relative torque request from PWM-Input
SOC_State_of_Charge	canApi_Get	Float32	0	100	%	State of Charge (SOC) of the battery
SM_OUT_SYS_Trq_Control	trqdesApi_Get canApi_Get	float32	0	1	-	Status of torque-control: 0 = Torque-control deactivated (Powerstage disabled) 1 = Torque-control activated (Powerstage and Current Controller enabled)
SM_PE_Mode_Req_Int	canApi_Get	Float32	0	65535	-	Operational mode of the PE as requested by APP_Activation_Mode: 0 = SET_PS_Mode_Req 1 = CAN_EXT_State_Reqest 2 = CAN_Immo_Unlock_Request 3 = BLE_Secure_Access_State
SM_BMS_Control_State	canApi_Get	Float32	0	1	-	Remote control command to the BMS: 0 = Turn 48V off 1 = Turn 48V on
TEMP_FET_Max	canApi_Get	Float32	-300	300	°C	Temperature of the hottest MOSFET
TEMP_Motor	canApi_Get	Float32	-300	300	°C	Motor temperature
TEMP_MCU	canApi_Get	Float32	-300	300	°C	Temperature of the microcontroller
TEMP_Combined_Max_Rel	canApi_Get	Float32	0	100	%	Combined relative temperature of Motor and MOSFET

Signal	API	Datatype	Min	Max	Unit	Description
TRQ_DES_Driver_Throttle	trqdesApi_Set	Float32	0	100	%	Throttle signal used for desired driver torque calculation after channel selection through SDK_APP_Throttle_Signal_Channel. Cross connections: This signal is compared against the SM_C_TRQ_Enable_Treshold parameter to control initial system start-up as well as system restart when an error occurs. In both cases the following condition must hold to enable system startup/restart: SDK_TRQ_DES_Driver_Throttle <= SM_C_TRQ_Enable_Treshold This signal is also used for Throttle Preactivation Check if SM_C_Check_Throttle_Enable is set to true.
TRQ_DES_Driver_Brake	trqdesApi_Set	Float32	0	100	%	Brake signal used for desired driver torque calculation after channel selection through SDK_APP_Brake_Signal_Channel. Cross connections: This signal is also used for Brake Preactivation Check if SM_C_Check_Brake_Enable is set to true.
TRQ_DES_Driver_Reverse_Gear	trqdesApi_Set canApi_Get	Float32	0	1	-	Signal which shows if reverse gear is selected: 0 = Forward gear selected 1 = Reverse gear selected Cross connections: This signal is also used in the modules Analog Interface AIN1 and Torque Strategy TRQ_STR to select between different throttle/current mappings.
TRQ_DES_Trq_Req_Rel	trqdesApi_Set	Float32	-100	100	%	Final desired driver torque. This signal serves as input for the following torque limitation (TRQ_LIM) module.
TRQ_LIM_Derating_Active	canApi_Get	Float32	0	1	-	Torque setpoint limitation flag: 0 = No limitation 1 = Derating active
TRQ_LIM_Derating_Temp_MCU	canApi_Get	Float32	0	1	-	The accelerating torque-setpoint is limited to keep the microcontroller temperature: 0 = No limitation 1 = Maximum microcontroller temperature reached

Signal	API	Datatype	Min	Max	Unit	Description
TRQ_LIM_Derating_Max_Positive_Current	canApi_Get	Float32	0	1	-	The accelerating torque-setpoint is limited by AC or DC current limit; StateList; 0=-;1=AC or DC current limit
TRQ_LIM_Derating_Max_Negative_Current	canApi_Get	Float32	0	1	-	The regenerative torque-setpoint is limited by AC or DC current limit: 0 = No limitation 1 = AC or DC current limit
TRQ_LIM_Derating_DC_Link_Voltage_Max	canApi_Get	Float32	0	1	-	The regenerative torque-setpoint is limited to stay below the maximum DC-Link-voltage: 0 = No limitation 1 = Upper DC-Link voltage threshold reached
TRQ_LIM_Derating_DC_Link_Voltage_Min	canApi_Get	Float32	0	1	-	The accelerating torque-setpoint is limited to stay above the minimum DC-Link-voltage: 0 = No limitation 1 = Lower DC-Link voltage thres hold reached
TRQ_LIM_Derating_Rotor_Speed	canApi_Get	Float32	0	1	-	The accelerating torque-setpoint is limited to stay below maximum rotor-speed 0 = No limitation 1 = Maximum rotor-speed reached
TRQ_LIM_Derating_Temp_FET	canApi_Get	Float32	0	1	-	The accelerating torque-setpoint is limited to keep the MOSFET temperature 0 = No limitation 1 = Maximum MOSFET temperature reached
TRQ_LIM_Derating_Temp_Motor	canApi_Get	Float32	0	1	-	The accelerating torque-setpoint is limited to keep the motor temperature: 0 = No limitation 1 = Maximum motor temperature reached

Table 5: API Variable Description MCU

9.2 API VARIABLE DESCRIPTION: BATTERY MANAGEMENT SYSTEM (BMS)

Applicable Frameworks:
- BMS FRIWO Standard V1.0

Signal	API	Datatype	Min	Max	Unit	Description
BCF_DC_Current	canApi_Get	Int32 LSB: 2^{-16}	-500	500	A	Battery current; filtered value discharge = negative current
BCF_Output_Voltage	canApi_Get	Int32 LSB: 2^{-16}	0	100	V	Output voltage measured at the BMS output terminals
BCF_Output12V_Voltage	canApi_Get	Int32 LSB: 2^{-16}	0	100	V	Output 12V voltage measured at the BMS aux terminals
BSW_IO_F_CAN_BSW_BusOff	canApi_Get	Int16 LSB: 2^{-16}	0	1	-	BusOff status flag of CAN peripheral: 0 = BusOn 1 = BusOff
BSW_IO_F_CAN_BSW_Passive	canApi_Get	Int16 LSB: 2^{-16}	0	1	-	BusPassive status flag of CAN peripheral: 0 = BusActive 1 = BusPassive
BSW_IO_F_CAN_BSW_Warning	canApi_Get	Int16 LSB: 2^{-16}	0	1	-	BusWarning status flag of CAN peripheral: 0 = BusReady 1 = BusWarning
CANIN_RX_Timeout	canApi_Set	UInt16 LSB: 2^{-16}	0	1	-	Global CAN RX timeout flag 1 = CAN-timeout detected
CM_PackVoltage	canApi_Get	UInt32 LSB: 2^{-16}	0	500	V	Corepack voltage based on the sum of the single cell voltages
CM_Cell_Temp1	canApi_Get	Int32 LSB: 2^{-16}	-55	200	[°C]	Cell temperature 1
CM_Cell_Temp2	canApi_Get	Int32 LSB: 2^{-16}	-55	200	[°C]	Cell temperature 2
CM_Cell_Temp3	canApi_Get	Int32 LSB: 2^{-16}	-55	200	[°C]	Cell temperature 3
CM_Cell_Temp4	canApi_Get	Int32 LSB: 2^{-16}	-55	200	[°C]	Cell temperature 4
CM_Cell_Temp5	canApi_Get	Int32 LSB: 2^{-16}	-55	200	[°C]	Cell temperature 5
CM_Cell_Temp6	canApi_Get	Int32 LSB: 2^{-16}	-55	200	[°C]	Cell temperature 6
CM_M_Cell_Voltage_Device1_1	canApi_Get	Int32 LSB: 2^{-16}	2.9	4.2	V	1st cell voltage, monitoring device 1
CM_M_Cell_Voltage_Device1_2	canApi_Get	Int32 LSB: 2^{-16}	2.9	4.2	V	2nd cell voltage, monitoring device 1
CM_M_Cell_Voltage_Device1_3	canApi_Get	Int32 LSB: 2^{-16}	2.9	4.2	V	3rd cell voltage, monitoring device 1

Signal	API	Datatype	Min	Max	Unit	Description
CM_M_Cell_Voltage_Device2_1	canApi_Get	Int32 LSB: 2^{-16}	2.9	4.2	V	1st cell voltage, monitoring device 2
CM_M_Cell_Voltage_Device2_2	canApi_Get	Int32 LSB: 2^{-16}	2.9	4.2	V	2nd cell voltage, monitoring device 2
CM_M_Cell_Voltage_Device2_3	canApi_Get	Int32 LSB: 2^{-16}	2.9	4.2	V	3rd cell voltage, monitoring device 2
CM_M_Cell_Voltage_Device2_4	canApi_Get	Int32 LSB: 2^{-16}	2.9	4.2	V	4th cell voltage, monitoring device 2
CM_M_Cell_Voltage_Device2_5	canApi_Get	Int32 LSB: 2^{-16}	2.9	4.2	V	5th cell voltage, monitoring device 2
CM_M_Cell_Voltage_Device2_6	canApi_Get	Int32 LSB: 2^{-16}	2.9	4.2	V	6th cell voltage, monitoring device 2
CM_M_Cell_Voltage_Device3_1	canApi_Get	Int32 LSB: 2^{-16}	2.9	4.2	V	1st cell voltage, monitoring device 3
CM_M_Cell_Voltage_Device3_2	canApi_Get	Int32 LSB: 2^{-16}	2.9	4.2	V	2nd cell voltage, monitoring device 3
CM_M_Cell_Voltage_Device3_3	canApi_Get	Int32 LSB: 2^{-16}	2.9	4.2	V	3rd cell voltage, monitoring device 3
CM_Highest_Cell_Voltage	canApi_Get	Int32 LSB: 2^{-16}	2.9	4.2	V	Highest cell voltage in stack
CM_Lowest_Cell_Voltage	canApi_Get	Int32 LSB: 2^{-16}	2.9	4.2	V	Lowest cell voltage in stack
CM_Level_of_Inbalance_V	canApi_Get	Int32 LSB: 2^{-16}	0	2	V	Level of inbalance
CM_PackVoltage_Curr_Comp	canApi_Get	UInt32 LSB: 2^{-16}	0	500	V	Current compensated corepack voltage
ERR_Errorcode	canApi_Get	UInt32 LSB: 2^{0}	0	$2^{32}-1$	-	System Errorcode 1 = Overtemperature powerstage sensor 1 2 = Overtemperature powerstage sensor 2 4 = Charge-current 8 = Discharge-current 16 = Pack-voltage too high 32 = Analog overvoltage 64 = Current-sensor offset 128 = EEPROM 256 = Cell-monitoring CRC 512 = External enable 1024 = Cell-monitoring Alert 2048 = Cell-monitoring Fault 4096 = Powerstage 8192 = Pre-charge error 16384 = Output-voltage high 32768 = Pack-voltage low 65535 = Cell-voltage discharge limit 131072 = Cell-voltage critical 262144 = cell-voltage charge limit 524288 = Analog overcurrent 1048576 = Overtemperature in charge-mode

Signal	API	Datatype	Min	Max	Unit	Description
						2097152 = Overtemperature in discharge-mode 4194304 = Undertemperature in charge-mode 8388608 = Undertemperature in discharge-mode 16777216 = Current-flow in passive state" 33554432 = CAN-Bus timeout 67108864 = Cell-connection 134217728 = Selftest required 268435456 = Permanent safety shutdown 536870912 = ICS-current monitoring 1073741824 = ICS-voltage monitoring 2147483648 = Keylock
ERR_Warningcode	canApi_Get	UInt32 LSB: 2^0	0	$2^{32}-1$	-	Warningcode Description see ERR_Errorcode
ERR_MEM_Trace1_a	canApi_Get	UInt32 LSB: 2^0	0	$2^{32}-1$	-	System Errorcode trace 0 stored in Memory Description see ERR_Errorcode
ERR_MEM_Trace1_a_timestamp	canApi_Get	UInt32 LSB: 2^0	0	$2^{32}-1$	-	Errorcode tracer timestamp
INFO_Discharge_Voltage_Limit	canApi_Get	UInt16 LSB: 2^0	0	255	V	Lower voltage threshold to prevent cut-off
INFO_Charge_Voltage_Limit	canApi_Get	UInt16 LSB: 2^0	0	255	V	Upper voltage threshold to prevent cut-off
INFO_Max_Discharge_Current	canApi_Get	Int16 LSB: 2^{-6}	0	500	A	Maximum dynamic discharge current for the next 3s to prevent cutoff
INFO_Max_Charge_Current	canApi_Get	Int16 LSB: 2^{-6}	0	500	A	Maximum charge current for the next 30s to prevent cutoff
INFO_Rem_Discharge_Time	canApi_Get	UInt16 LSB: 2^0	0	65535	s	Remaining time for discharge at current rate
INFO_Rem_Charge_Time	canApi_Get	UInt16 LSB: 2^0	0	65535	s	Remaining time to full charge
MEDKit_IO_CM_Control_Device1	canApi_Get	UInt16 LSB: 2^0	0	255	-	Cell balancing control output
MEDKit_IO_CM_Control_Device2	canApi_Get	UInt16 LSB: 2^0	0	255	-	Cell balancing control output
MEDKit_IO_CM_Control_Device3	canApi_Get	UInt16 LSB: 2^0	0	255	-	Cell balancing control output
MEDKit_IO_Charger_Detected	canApi_Get	UInt16 LSB: 2^0	0	1	-	Charger Detection Input 1 = Charger detected
PROD_M_HW_ID1	canApi_Get	UInt32 LSB: 2^0	0	$2^{32}-1$	-	HW ID

Signal	API	Datatype	Min	Max	Unit	Description
SM_State_Req_Ext	canApi_Set	UInt16 LSB: 2^0	0	6	–	Manual state request 0 = manual control / idle 1 = active mode 2 = charge mode 3 = 12V mode (48V off) 6 = Deep sleep
SM_Bordnet_Shutdown_Imminent	canApi_Get	UInt16 LSB: 2^0	0	1	–	Status of the bordnet shutdown info 1 = 48V cutoff imminent
SM_HV_Shutdown_Imminent	canApi_Get	UInt16 LSB: 2^0	0	1	–	48V cut-off indication 1 = 48V will be cut off in 100ms
SOC_Num_Charge_Cycles	canApi_Get	UInt16 LSB: 2^0	0	65535	–	Number of charge cycles
SOC_Remaining_Capacity	canApi_Get	Int32 LSB: 2^{-10}	0	2m	mAh	Remaining capacity
SOC_Full_Charge_Capacity	canApi_Get	UInt16 LSB: 2^0	0	65535	mAh	Full charge capacity
SOC_State_of_Charge	canApi_Get	UInt16 LSB: 2^0	0	100	%	State of charge relative to the full charge capacity
SOC_State_of_Health	canApi_Get	UInt16 LSB: 2^{-1}	0	120	%	State of health; relation of full charge capacity to the original design capacity of the cells
SOC_R_Cell_Estim	canApi_Get	Int32 LSB: 2^{-16}	0	10	Ohm	Estimated cell resistance
SDC_Time_PowerOn_Total	canApi_Get	UInt32 LSB: 2^0	0	$2^{32}-1$	s	Time power on total
SET_Pack_Num_Serial_Cells	canApi_Get	UInt16 LSB: 2^0	0	14	–	Active setting: Number of serial cells in pack
SET_C_Prod_Info_1	canApi_Get	UInt32 LSB: 2^0	0	$2^{32}-1$	–	custom number
TEMP_FET1	canApi_Get	Int32 LSB: 2^{-16}	-50	150	°C	Temperature at the MOSFET [°C]
TEMP_FET2	canApi_Get	Int32 LSB: 2^{-16}	-50	150	°C	Temperature for the BMSV1 at the MOSFET and for the BMSV2 at the SHUNT
TEMP_MCU	canApi_Get	Int16 LSB: 2^0	-50	150	°C	Temperature of micro controller
TEMP_Cell_Max	canApi_Get	Int32 LSB: 2^{-16}	-50	150	°C	Highest cell temperature, showing the highest cell temperature of all measured temperatures
TEMP_Cell_Min	canApi_Get	Int32 LSB: 2^{-16}	-50	150	°C	Lowest cell temperature, showing the lowest cell temperature of all measured temperatures

Table 6: API Variable Description BMS

10   Troubleshooting

The FRIWO SDK distinguishes between a variety of possible errors to be able to provide information on the cause of the error and to help you quickly correct the error.

10.1   Error codes

Description of all error codes		
Error Code	**Type**	**Description**
1000	Server Error	–
1001	Server Error	NoOnlineConnection
2000	Client Error	–
2100	Client Error	FileSystemError
2101	Client Error	DirectoryNotFoundToCopy
2102	Client Error	FileNotFoundToLoadFramework
2103	Client Error	FailedToLoadFrameworkFile
2104	Client Error	FileNotFoundToLoadProject
2105	Client Error	FailedToLoadProjectFile
3000	Process Error	–
3100	Process Error	PreBuildError
3101	Process Error	ReleaseVersionError
3102	Process Error	ParameterVersionError
3103	Process Error	FirmwareVersionFileNotFoundError
3104	Process Error	FirmwareVersionFileFormatError
3105	Process Error	ModuleUserFileValidationFailed
3106	Process Error	ModuleUserDeafultFileNotExist
3200	Build Error	–
3201	Build Error	KeilPathError
3202	Build Error	PathMissing
3203	Build Error	CleanFailed
3204	Build Error	CopyFailed
3205	Build Error	CompilerInstructionFailed
3206	Build Error	LinkerInstructionFailed
3207	Build Error	CompilerRunFailed
3208	Build Error	LinkerExecutionFailed
3209	Build Error	HexBuilderRunFailed
3210	Build Error	PostBuildSrecordToolFailed
3211	Build Error	PostBuildToolsMissing
3212	Build Error	PostBuildCrcToolFailed

3213	Build Error	CompilerArgumentFileNotFound
3214	Build Error	CompilerPathNotSet
3215	Build Error	LinkerArgumentFileNotFound
3216	Build Error	LinkerPathNotSet
3217	Build Error	HexBuilderInputFileNotFound
3218	Build Error	HexBuilderPathNotSet
3300	Post Build Error	–
3301	Post Build Error	DeleteBuildFolderFailed
3302	Post Build Error	CreateBuildFolderFailed
3303	Post Build Error	CreateOutputFolderFailed
3304	Post Build Error	CopyHexFileFailed
3305	Post Build Error	CopyVersionFileFailed
3306	Post Build Error	LoadBasisVariableFileFailed
3307	Post Build Error	LoadAdditionalVariableFileFailed
3308	Post Build Error	VariableDuplicateError
3309	Post Build Error	LoadMapFileFailed
3310	Post Build Error	DefaultVariableConfigCreationError
3311	Post Build Error	ExtendVariableConfigError
3312	Post Build Error	FlashFileEefCreationError
3313	Post Build Error	FlashFileEefxCreationError
3314	Post Build Error	UNKNOWN_Exception
3315	Post Build Error	CopyBinFileFailed
3316	Post Build Error	VariablesNotFoundInCode
3400	Roll Out Error	–
3401	Roll Out Error	ReleaseVersionAlreadyExists
3402	Roll Out Error	UploadBuildInfoToDbFailed
3403	Roll Out Error	UploadBuildFilesFailed
3404	Roll Out Error	UploadSoftwareDescriptionFileFailed
3405	Roll Out Error	CopyBuildToWorkspaceFailed
4000	License Error	–
4001	License Error	NoSdkLicenseFoundBeforeBuild
4002	License Error	NoSdkLicenseFoundAfterBuild
4003	License Error	SdkLicServerErrorAfterBuild
4004	License Error	UsageDenied
4005	License Error	RegistrationSkipped
5000	Settings Error	–
5001	Settings Error	ProjectToFrameworkMismatch
5002	Settings Error	NoFrameworkForProject
5003	Settings Error	NoProjectFileLoaded
5004	Settings Error	SetupNotValid

Table 7: Description of all error codes

11 Versioning

Versions of this document		
version	**date**	**comment**
V1.0	20210331	initial Document
V1.1	20210611	added CAN module
V2.0	20210802	added support for BMS hardware

Table 8: Versions of this document.

Feedback

We are working very hard to improve our products and therefore **feedback** is indispensable! Please send us your valuable feedback as contact form or via Mail to feedback@friwo.com

https://friwo.link/m/feedback