GTM-IP Application note

AN026 - DTM Shut-off feature

Contents

1 Introduction

This application note focuses on the DTM Shut-off feature, which is implemented in the Phase Shift Unit (PSU) of the DTM module. The shut-off feature can be used to force the GTM outputs to a defined level on behalf of an external or internal event.

This application note refers to features of GTM-IP v3.1.5 and above.

In section 2 the architecture of the DTM module with special focus on the shut-off feature functionality is described. Section 3 contains two application examples for DTM Shut-off functionality.

2 DTM Architecture

2.1 Overview

Figure 1 shows the signal data path of one DTM channel. A DTM consists of four such channels. The channel architecture can be divided into three parts. The part marked with green is used to control the signal path of the incoming signals. This signal path is controlled by registers DTM_CH_CTRL1 and DTM_CH_CTRL3. The blue area of the channel is used for the dead time generation, while the orange part is used for routing the signals to the DTM output. This orange part can be configured with the DTM_CH_CTRL2 register.

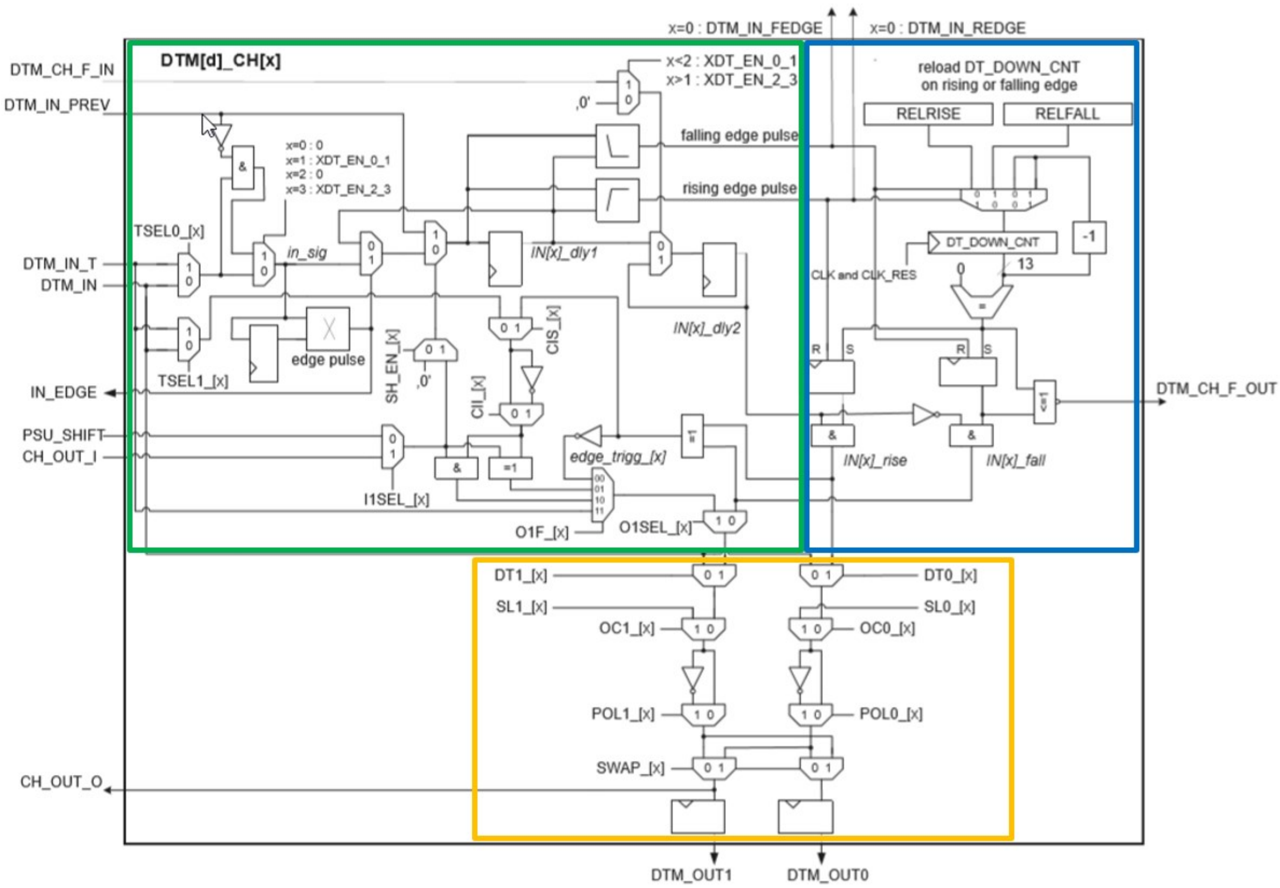

Fig. 1: DTM Channel architecture.

Especially, the orange part of the DTM channel needs to be controlled by the shut-off feature, to define, which signals are generated at the output, when the shut-off condition is met. Therefore, the DTM_CH_CTRL2 register, which controls the DTM output signal path, has dedicated shadow registers DTM_CH_CTRL2_SR and DTM_CH_SR, which are put in place, when the shut-off feature should be applied or released again.

On the other hand, the shut-off request has to be feed into the DTM. Figure 2 depicts the signal path of the shut-off signal. The shut-off signal can come from the TIM module, a dedicated DTM_AUX_IN input of the DTM, a PSU subunit of another DTM or from additional inputs. These additional inputs are marked with ① in the figure and are in place in GTM-IP starting from version 4.1. They are not used within this application note.

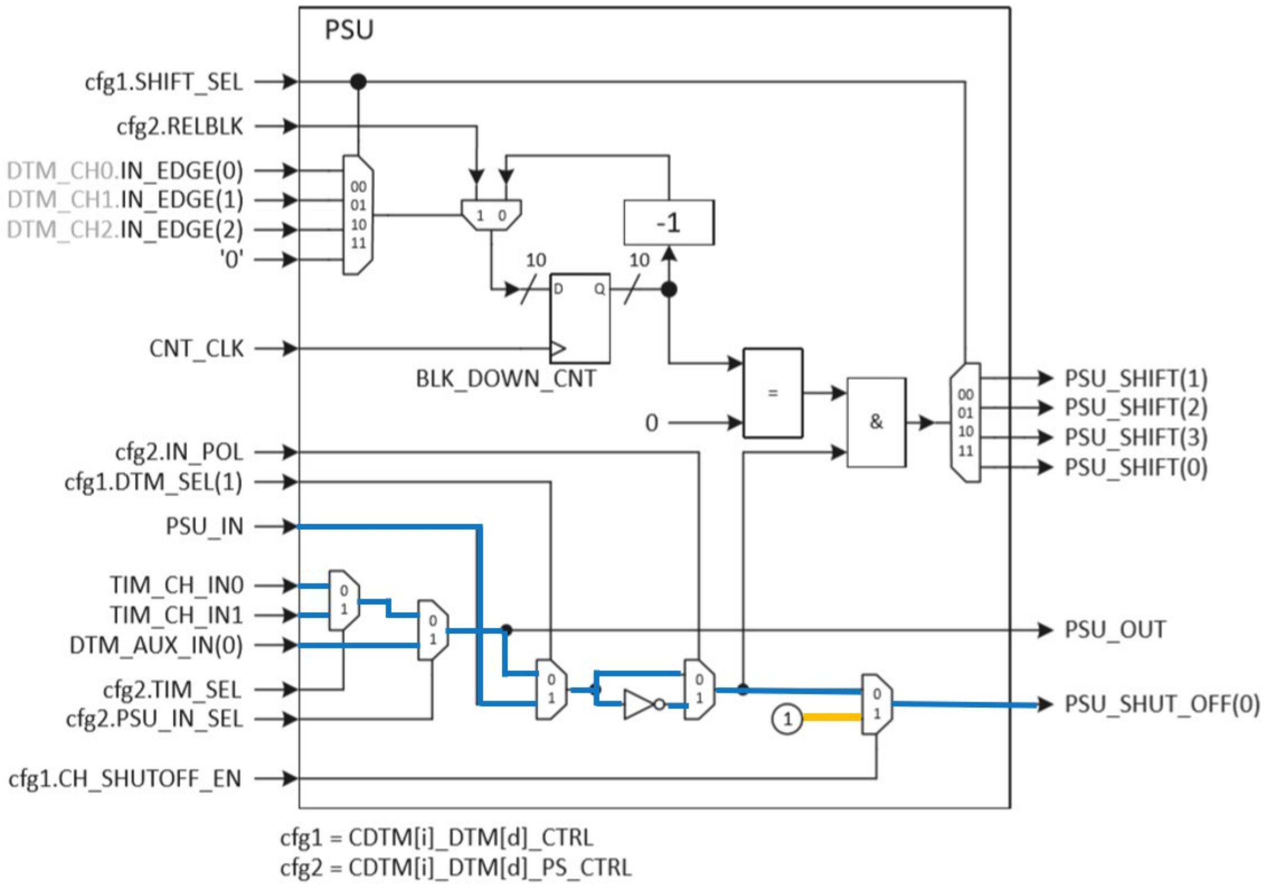

Fig. 2: Sources and routing of shut-off signal.

The resulting signal PSU_SHUT_OFF(0) is one of the signals that can control the update mechanism of register DTM_CH_CTRL2. As it can be seen from figure 3, the shut-off can be reset either by a DTM input rising or falling edge, by SHUT_OFF_RST bit of configuration register DTM_CTRL or by PSU_SHUT_OFF(0) register changing signal state.

The following chapter shows two application examples how the shut-off feature can be used in detail.

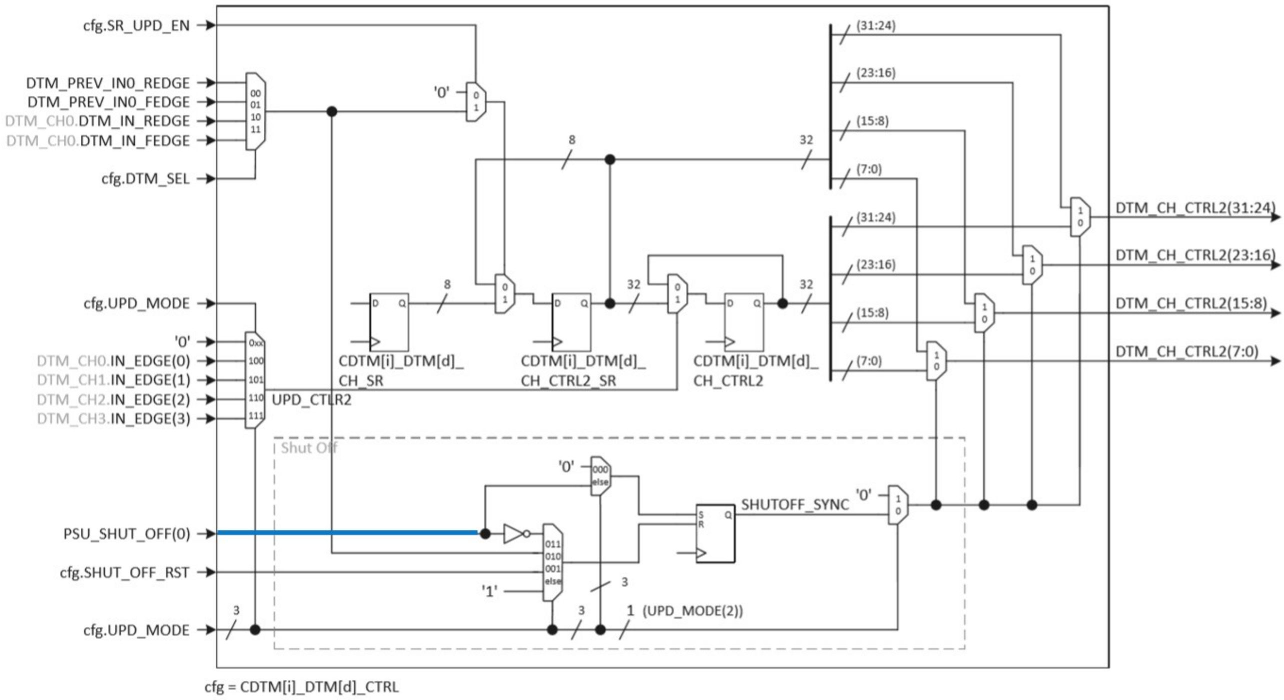

Fig. 3: Synchronous update mechanism of register DTM_CH_CTRL2.

3 Application examples

3.1 Application setup

This section discusses two application examples how to use the shut-off feature of the DTM submodule. The shut-off feature should be applied on three ATOM channel outputs. The DTM generates inverted signals from those three ATOM outputs and applies a dead time. This results in six output signals in total.

The shut-off request comes from an external circuit which is connected to a TIM input. When the shut-off is requested, the outputs of the DTM should be set to a defined state and should stay there for a specified time. Afterwards, the DTM outputs should switch back to another defined state (Use case 1), or should switch back to the six output signals generated out of the three ATOM inputs (Use case 2). All these steps should be done without CPU interaction.

The following resources are needed for the two application examples:

- One TIM input channel, receiving the shut-of request from an external source
- Three ATOM channels generating the output signals on which the shut-off should be applied
- One ATOM channel to determine the time when the shut-off should be released
- Three DTM channels to generate six output signals out of three input signals coming from three ATOM channels

The system architecture of the application example is shown in figure 4. The shut-off request is coming from an external source and is indicated there by a rising edge. The signal is connected to TIM channel 0 where it is split into two parts. The original signal is routed directly to the PSU subunit of the DTM to drive the shut-off event. In addition a start trigger for a time out counter is generated and feed to ATOM channel 0. When the time out counter of ATOM channel 0 expires, the ATOM CH0 generates a shut-off release signal to the DTM.

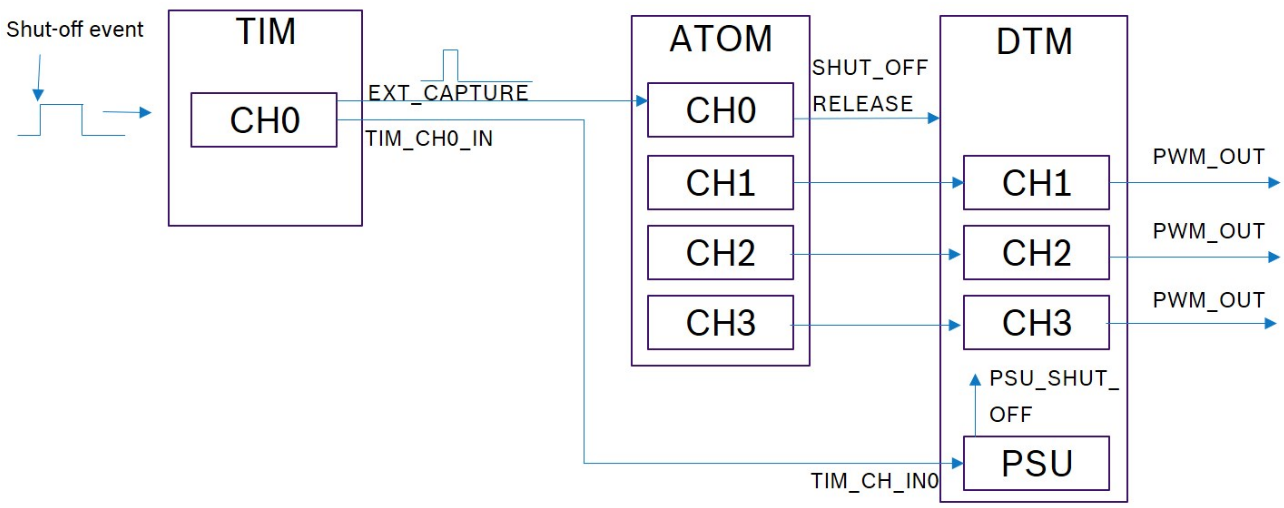

Fig. 4: Application example for DTM Shut-off feature.

3.2 Use case 1

3.2.1 Description

In use case 1, the six DTM outputs should be set to a specified signal level after the shut-off time expired. Therefore, two signal level registers have to exist for each of the six DTM outputs. One signal level register for the signal level output during shut-off and a second one after the shut-off time exired.

3.2.2 Configuration of ATOM and DTM submodule for output signal generation

The ATOM channels one, two and three are used to generate three PWM signals which are routed to the DTM. The duty cycle signal level should be high and as counter clock the CMU_CLK1 should be used. Listing 1 shows the code for configuration of the three channels.

```
/* Enable update of compare registers (SR --> CM) for channels 1, 2, 3*/
GTM.ATOM[0].AGC.GLB_CTRL = 0x00A80000;
/* Configure ATOM CH1, CH2, CH3 */
regContent_u32 = (0x2 << GTM_ATOM_CH_CTRL_MODE_POS) |
                 (0x1 << GTM_ATOM_CH_CTRL_SL_POS)   |
                 (0x1 << GTM_ATOM_CH_CTRL_CLK_SRC_SR_POS);
GTM.ATOM[0].CH[1].CTRL = regContent_u32;
GTM.ATOM[0].CH[2].CTRL = regContent_u32;
GTM.ATOM[0].CH[3].CTRL = regContent_u32;

GTM.ATOM[0].CH[1].SR1 = 250;       /* SR1=250 ticks duty cycle */
GTM.ATOM[0].CH[1].SR0 = 1000;      /* SR0=1000 ticks period */
GTM.ATOM[0].CH[2].SR1 = 500;
GTM.ATOM[0].CH[2].SR0 = 1000;
GTM.ATOM[0].CH[3].SR1 = 750;
GTM.ATOM[0].CH[3].SR0 = 1000;
```

Listing 1: Configuration of ATOM_CH1, _CH2, _CH3 for PWM output generation.

3.2.3 Configuration of ATOM submodule for shut-off release signal generation

The ATOM channel 0 should be used to determine the point in time after which the DTM outputs should come out of their shut-off state. As it can be seen from figure 3, the shut-off state can be left on behalf of several DTM input signal rising and falling edges. For the two example applications a rising edge on ATOM channel 0 output should be used as shut-off release event.

To generate the rising edge for the shut-off release signal, the ATOM One-shot mode with external trigger should be used. The operation principle of the One-shot mode is shown in figure 5.

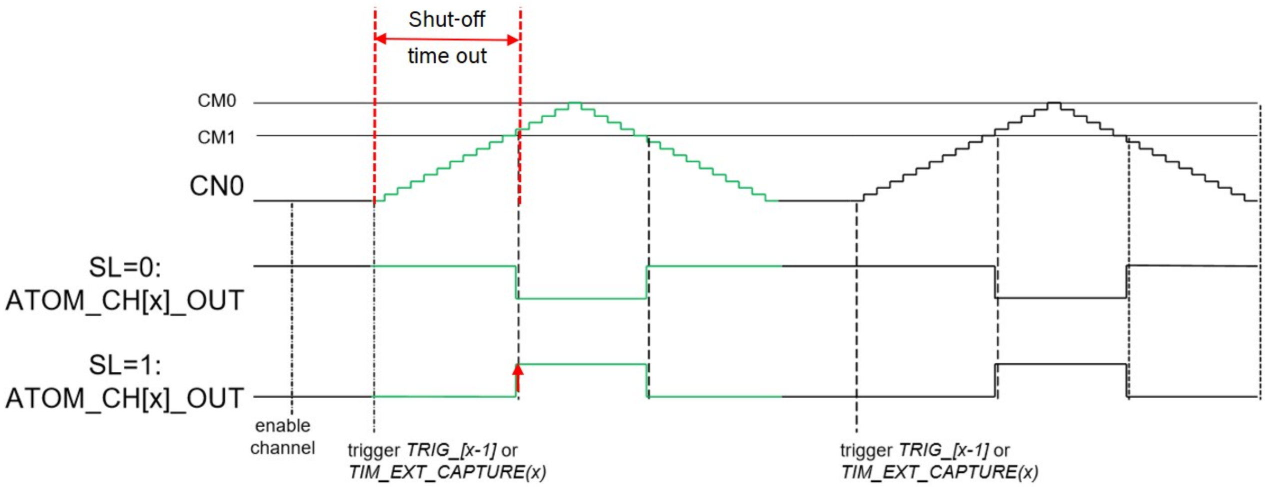

Fig. 5: PWM Output in one-shot counting up-down mode triggered by TRIG_[x-1] or TIM_EXT_CAPTURE(x).

When the external trigger is received by the ATOM channel, the counter ATOM_CH0_CN0 starts counting. A positive edge is generated, when the counter reaches CM1 and the signal level bit SL is set to one. The ATOM SOMP One-shot mode with the up-down counter is choosen, to avoid problems with the Errata GTM-IP-298. In the Errata a problem with the One-shot mode using an up-counter is described, when the start of the one-shot is triggered by TIM_EXT_CAPTURE(x). Due to this Errata, the shut-off would be driven immediately after the TIM_EXT_CAPTURE(x) signal arrives at the ATOM channel.

The configuration of the ATOM channel 0 is shown in listing 2. As time out time 5000 clock ticks of CMU_CLK1 is configured.

```
/* Configure ATOM CH0 */
regContent_u32 = (0x2 << GTM_ATOM_CH_CTRL_MODE_POS)  |
                 (0x1 << GTM_ATOM_CH_CTRL_SL_POS)     |
                 (0x1 << GTM_ATOM_CH_CTRL_CLK_SRC_SR_POS)  |
                 (0x1 << GTM_ATOM_CH_CTRL_UDMODE_POS)  |
                 (0x1 << GTM_ATOM_CH_CTRL_OSM_TRIG_POS)  |
                 (0x1 << GTM_ATOM_CH_CTRL_EXT_TRIG_POS)  |
                 (0x1 << GTM_ATOM_CH_CTRL_OSM_POS);
GTM.ATOM[0].CH[0].CTRL = regContent_u32;

GTM.ATOM[0].CH[0].CM1 = 5000;        /* SR1=5000 ticks till time out expires */
GTM.ATOM[0].CH[0].CM0 = 5500;
```

Listing 2: Configuration of ATOM_CH0_CTRL for shut-off release signal generation.

3.2.4 Configuration of TIM submodule

To trigger the SOMP One-shot mode up-down counter of ATOM channel 0 a TIM_EXT_CAPTURE event has to be generated out of an incomming signal edge. This event can be generated with the EXTCAPSRC subunit of a TIM channel. This subunit is depicted in figure 6.

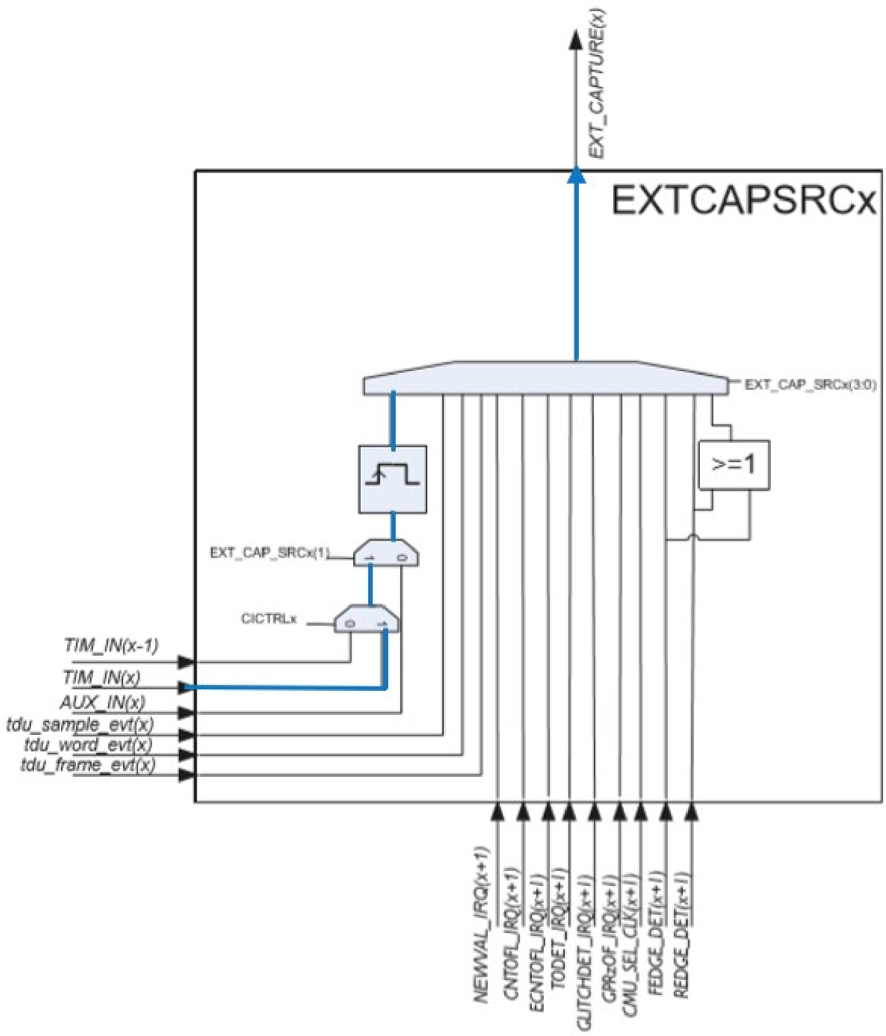

Fig. 6: TIM EXTCAPSRC subunit to generate TIM_EXT_CAPTURE(x) signal.

Since the TIM input signal is also needed to trigger the shut-off start event in the DTM via the TIM_CH_IN0 signal line it has to be feed through the TIM channel too. To do this, the INPUTSRC subunit is used, which is shown in figure 2. Since the multiplexer controlled by bit CICTRL of register TIM_CH_CTRL has to be set to ’1’, to generate the EXT_CAPTURE event in the EXTCAPSRC subunit, the signal TIM_IN(x-1) is feed through the TIM channel by default. Therefore, an alternative path has to be used, so that TIM_IN(x) is routed through the remaining TIM channel. This path is drawn in blue in figure 7.

As it can be seen, the LUT input line 2 is routed through the LUT and to the F_IN(x) signal. This signal is than visible as TIM_CH_IN0 at the DTM. The LUT has to be configured according to table 1 to route through the EXT_CAPTURE signal. The configuration of the LUT is shown in line one of listing 3.

Overall, the configuration of the TIM channel 0 is shown in listing 3.

```
1 GTM.TIM[0].CH[0].TDUC   = 0x00F00000;      /* Configure INPUTSRC LUT */
```

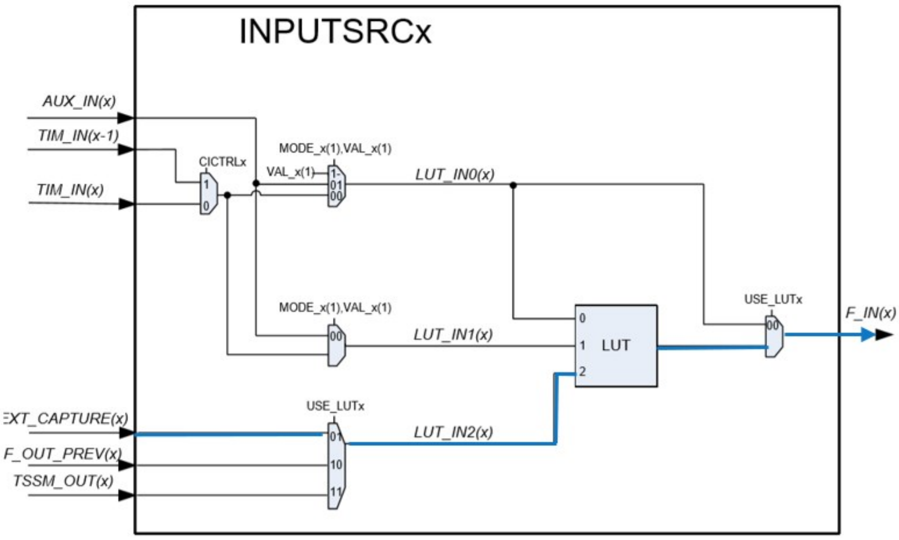

Fig. 7: TIM INPUTSRC subunit and the routing of the EXT_CAPTURE(x) signal to TIM output signal.

LUT_IN2(x) (EXT_CAPTURE(x))	LUT_IN1(x) (don't care)	LUT_IN0(x) (don't care)	$F\_IN = f(LUT\_IN2(x))$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Table 1: INPUTSRC LUT configuration to route through the EXT_CAPTURE signal.

```
GTM.TIM[0].CH[0].ECTRL = (0x3 << GTM_TIM_CH_ECTRL_EXT_CAP_SRC_POS) |
                         (0x1 << GTM_TIM_CH_ECTRL_USE_LUT_POS);
GTM.TIM[0].CH[0].CTRL  = (0x1 << GTM_TIM_CH_CTRL_CICTRL_POS) |
                         (0x1 << GTM_TIM_CH_CTRL_TIM_EN_POS);
```

Listing 3: Configuration of TIM_CH0 for shut-off time out detection.

3.2.5 Configuration of DTM submodule

Last not least, the DTM submodule has to be configured, to generate the PWM output signals according to the ATOM inputs and to incorporate the shut-off feature if the shut-off request receives at the TIM input. Listing 4 shows the configuration of the DTM module.

```
GTM.CDTM[0].DTM[4].CTRL = (0x1 << GTM_CDTM_DTM_CTRL_CLK_SEL_POS)  |
                          (0x1 << GTM_CDTM_DTM_CTRL_DTM_SEL_POS)  |
                          (0x1 << GTM_CDTM_DTM_CTRL_UPD_MODE_POS)  |
                          (0x1 << GTM_CDTM_DTM_CTRL_SR_UPD_EN_POS);

GTM.CDTM[0].DTM[4].CH_CTRL2 = (0x1 << GTM_CDTM_DTM_CH_CTRL2_DT0_1_POS)  |
                              (0x1 << GTM_CDTM_DTM_CH_CTRL2_DT1_1_POS)  |
                              (0x1 << GTM_CDTM_DTM_CH_CTRL2_DT0_2_POS)  |
                              (0x1 << GTM_CDTM_DTM_CH_CTRL2_DT1_2_POS)  |
                              (0x1 << GTM_CDTM_DTM_CH_CTRL2_DT0_3_POS)  |
                              (0x1 << GTM_CDTM_DTM_CH_CTRL2_DT1_3_POS);

GTM.CDTM[0].DTM[4].CH_CTRL2_SR = (0x1 << GTM_CDTM_DTM_CH_CTRL2_DT0_1_POS)  |
                                 (0x1 << GTM_CDTM_DTM_CH_CTRL2_OC0_1_POS)  |
                                 (0x1 << GTM_CDTM_DTM_CH_CTRL2_DT1_1_POS)  |
                                 (0x1 << GTM_CDTM_DTM_CH_CTRL2_OC1_1_POS)  |
                                 (0x1 << GTM_CDTM_DTM_CH_CTRL2_DT0_2_POS)  |
                                 (0x1 << GTM_CDTM_DTM_CH_CTRL2_OC0_2_POS)  |
                                 (0x1 << GTM_CDTM_DTM_CH_CTRL2_DT1_2_POS)  |
                                 (0x1 << GTM_CDTM_DTM_CH_CTRL2_OC1_2_POS)  |
                                 (0x1 << GTM_CDTM_DTM_CH_CTRL2_DT0_3_POS)  |
                                 (0x1 << GTM_CDTM_DTM_CH_CTRL2_OC0_3_POS)  |
                                 (0x1 << GTM_CDTM_DTM_CH_CTRL2_DT1_3_POS)  |
                                 (0x1 << GTM_CDTM_DTM_CH_CTRL2_OC1_3_POS);

GTM.CDTM[0].DTM[4].CH_SR = 0x000000FF;    /* DTM output signal level after
                                           * shut-off time expired
                                           */

regContent_u32 = 0x001E001E; /* RELFALL=30ticks, RELRISE=30ticks */
GTM.CDTM[0].DTM[4].CH_DTV[1] = regContent_u32;
GTM.CDTM[0].DTM[4].CH_DTV[2] = regContent_u32;
GTM.CDTM[0].DTM[4].CH_DTV[3] = regContent_u32;
```

Listing 4: Configuration of DTM module for use case 1.

3.2.6 Simulation result

Figure 8 shows the simulation result for use case 1.

The figure shows the TIM input, where the shut-off signal is received, the ATOM channel 0, where the shut-off time out is measured and the three DTM channels, where the shut-off request is applied. The shut-off value at the six outputs DTM_CHx_OUTy is driven, when the rising edge at the TIM0_CH0_IN arrives. This is marked with the first cursor. Starting from this point in time, the six DTM Output signals are driven with the signal level defined in DTM_CH_CTRL2_SR.SLx_y_SR register bits.

In parallel, the counter ATOM_CH0_CN0 starts counting. After the counter reaches the ATOM_CH0_CM1 value, the ATOM output is driven high, which leads to the trigger DTM_IN0_REDGE at the DTM (cursor two). With this trigger, the six output signals DTM_CHx_OUTy are driven with the signal level specified in register DTM_CH_SR.

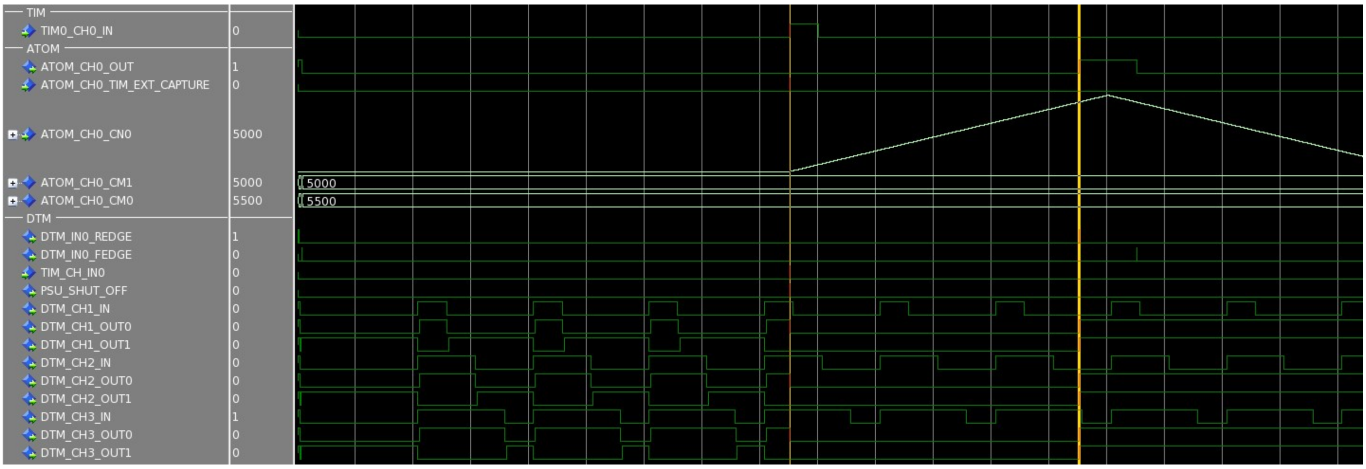

Fig. 8: Simulation result for use case 1, DTM outputs at defined state after shut-off time expired.

3.3 Use case 2

3.3.1 Description

In use case two, the behaviour after the shut-off expires, is slightly different. There, the six output signals DTM_CHx_OUTy should not be driven by a value specified in register DTM_CH_SR, but with the generated signals from the three incoming DTM signals DTM_IN, as it was before the shut-off request. This behaviour can be reached by configuring the shut-off reset behaviour of the multiplexer shown in figure 9.

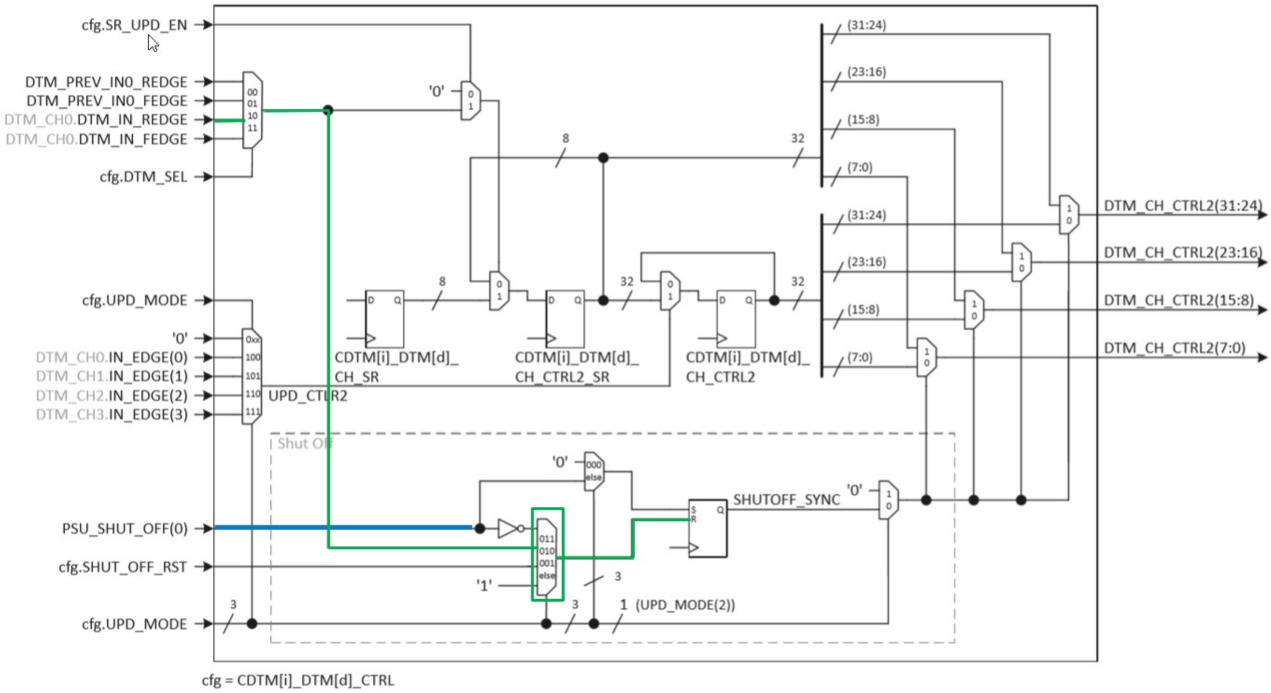

Fig. 9: Configuration of shut-off reset request.

Listing 5 shows the configuration for use case 2. Please note, that the UPD_MODE configuration in line three and the SR_UPD_EN in line four changed. The configuration of register DTM_CH_SR is skipped, since this is not needed for this use case.

```
GTM.CDTM[0].DTM[4].CTRL = (0x1 << GTM_CDTM_DTM_CTRL_CLK_SEL_POS) |
(0x1 << GTM_CDTM_DTM_CTRL_DTM_SEL_POS) |
(0x2 << GTM_CDTM_DTM_CTRL_UPD_MODE_POS) |
(0x0 << GTM_CDTM_DTM_CTRL_SR_UPD_EN_POS);

GTM.CDTM[0].DTM[4].CH_CTRL2 = (0x1 << GTM_CDTM_DTM_CH_CTRL2_DT0_1_POS) |
(0x1 << GTM_CDTM_DTM_CH_CTRL2_DT1_1_POS) |
(0x1 << GTM_CDTM_DTM_CH_CTRL2_DT0_2_POS) |
(0x1 << GTM_CDTM_DTM_CH_CTRL2_DT1_2_POS) |
(0x1 << GTM_CDTM_DTM_CH_CTRL2_DT0_3_POS) |
(0x1 << GTM_CDTM_DTM_CH_CTRL2_DT1_3_POS);

GTM.CDTM[0].DTM[4].CH_CTRL2_SR = (0x1 << GTM_CDTM_DTM_CH_CTRL2_DT0_1_POS) |
(0x1 << GTM_CDTM_DTM_CH_CTRL2_OC0_1_POS) |
(0x1 << GTM_CDTM_DTM_CH_CTRL2_DT1_1_POS) |
(0x1 << GTM_CDTM_DTM_CH_CTRL2_OC1_1_POS) |
(0x1 << GTM_CDTM_DTM_CH_CTRL2_DT0_2_POS) |
(0x1 << GTM_CDTM_DTM_CH_CTRL2_OC0_2_POS) |
(0x1 << GTM_CDTM_DTM_CH_CTRL2_DT1_2_POS) |
(0x1 << GTM_CDTM_DTM_CH_CTRL2_OC1_2_POS) |
(0x1 << GTM_CDTM_DTM_CH_CTRL2_DT0_3_POS) |
(0x1 << GTM_CDTM_DTM_CH_CTRL2_OC0_3_POS) |
(0x1 << GTM_CDTM_DTM_CH_CTRL2_DT1_3_POS) |
(0x1 << GTM_CDTM_DTM_CH_CTRL2_OC1_3_POS);

GTM.CDTM[0].DTM[4].CH_SR = 0x000000FF;    /* DTM output signal level after
* shut-off time expired
*/

regContent_u32 = 0x001E001E; /* RELFALL=30ticks, RELRISE=30ticks */
GTM.CDTM[0].DTM[4].CH_DTV[1] = regContent_u32;
GTM.CDTM[0].DTM[4].CH_DTV[2] = regContent_u32;
GTM.CDTM[0].DTM[4].CH_DTV[3] = regContent_u32;
```

Listing 5: Configuration of DTM module for use case 2.

3.3.2 Simulation result

Figure 10 shows the simulation result for use case two. As it can be seen from the figure, the six DTM outputs DTM_CHx_OUTy remain low during shut-off time, and switch back to the six signals generated by the DTM dead time generation circuit (blue part of figure 1).

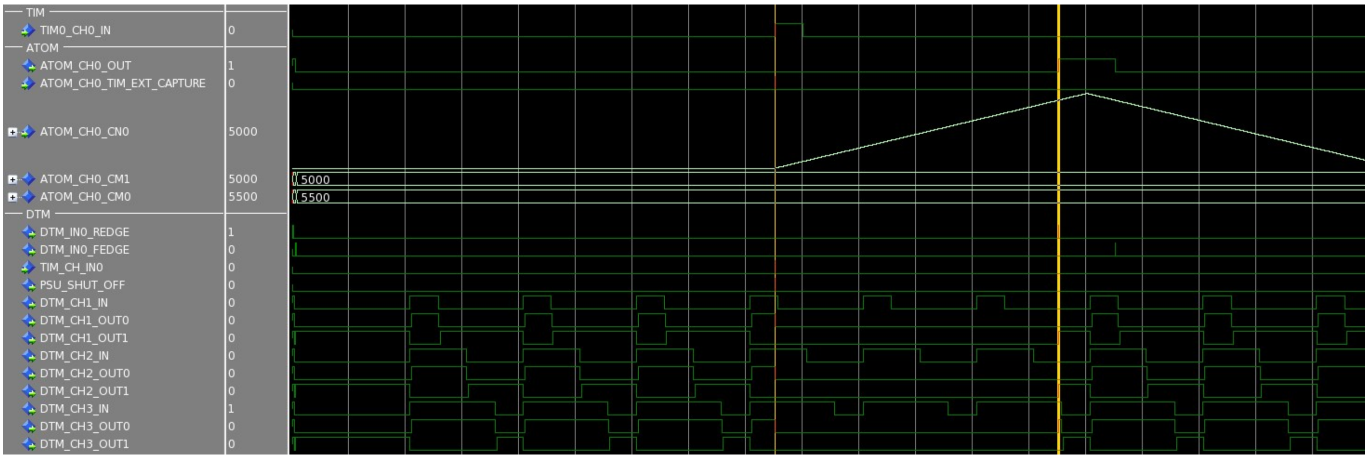

Fig. 10: Simulation result for use case 2, DTM outputs return to DTM_IN after shut-off time expired.

4 General information about this document

4.1 Legal notice

The content of this document is subject to continuous developments and improvements. All particulars and its use contained in this document are given by BOSCH in good faith.

NO WARRANTIES: TO THE MAXIMUM EXTENT PERMITTED BY LAW, NEITHER THE INTELLECTUAL PROPERTY OWNERS, COPYRIGHT HOLDERS AND CONTRIBUTORS, NOR ANY PERSON, EITHER EXPRESSLY OR IMPLICITLY, WARRANTS ANY ASPECT OF THIS SPECIFICATION, SOFTWARE RELATED THERETO, CODE AND/OR PROGRAM RELATED THERETO, INCLUDING ANY OUTPUT OR RESULTS OF THIS SPECIFICATION, SOFTWARE RELATED THERETO, CODE AND/OR PROGRAM RELATED THERETO UNLESS AGREED TO IN WRITING. THIS SPECIFICATION, SOFTWARE RELATED THERETO, CODE AND/OR PROGRAM RELATED THERETO IS BEING PROVIDED "AS IS", WITHOUT ANY WARRANTY OF ANY TYPE OR NATURE, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, AND ANY WARRANTY THAT THIS SPECIFICATION, SOFTWARE RELATED THERETO, CODE AND/OR PROGRAM RELATED THERETO IS FREE FROM DEFECTS.

ASSUMPTION OF RISK: THE RISK OF ANY AND ALL LOSS, DAMAGE, OR UNSATISFACTORY PERFORMANCE OF THIS SPECIFICATION (RESPECTIVELY THE PRODUCTS MAKING USE OF IT IN PART OR AS A WHOLE), SOFTWARE RELATED THERETO, CODE AND/OR PROGRAM RELATED THERETO RESTS WITH YOU AS THE USER. TO THE MAXIMUM EXTENT PERMITTED BY LAW, NEITHER THE INTELLECTUAL PROPERTY OWNERS, COPYRIGHT HOLDERS AND CONTRIBUTORS, NOR ANY PERSON EITHER EXPRESSLY OR IMPLICITLY, MAKES ANY REPRESENTATION OR WARRANTY REGARDING THE APPROPRIATENESS OF THE USE, OUTPUT, OR RESULTS OF THE USE OF THIS SPECIFICATION, SOFTWARE RELATED THERETO, CODE AND/OR PROGRAM RELATED THERETO IN TERMS OF ITS CORRECTNESS, ACCURACY, RELIABILITY, BEING CURRENT OR OTHERWISE. NOR DO THEY HAVE ANY OBLIGATION TO CORRECT ERRORS, MAKE CHANGES, SUPPORT THIS SPECIFICATION, SOFTWARE RELATED THERETO, CODE AND/OR PROGRAM RELATED THERETO, DISTRIBUTE UPDATES, OR PROVIDE NOTIFICATION OF ANY ERROR OR DEFECT, KNOWN OR UNKNOWN. IF YOU RELY UPON THIS SPECIFICATION, SOFTWARE RELATED THERETO, CODE AND/OR PROGRAM RELATED THERETO, YOU DO SO AT YOUR OWN RISK, AND YOU ASSUME THE RESPONSIBILITY FOR THE RESULTS. SHOULD THIS SPECIFICATION, SOFTWARE RELATED THERETO, CODE AND/OR PROGRAM RELATED THERETO PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL LOSSES, INCLUDING, BUT NOT LIMITED TO, ANY NECESSARY SERVICING, REPAIR OR CORRECTION OF ANY PROPERTY INVOLVED TO THE MAXIMUM EXTEND PERMITTED BY LAW.

DISCLAIMER: IN NO EVENT, UNLESS REQUIRED BY LAW OR AGREED TO IN WRITING, SHALL THE INTELLECTUAL PROPERTY OWNERS, COPYRIGHT HOLDERS OR ANY PERSON BE LIABLE FOR ANY LOSS, EXPENSE OR DAMAGE, OF ANY TYPE OR NATURE ARISING OUT OF THE USE OF, OR INABILITY TO USE THIS SPECIFICATION, SOFTWARE RELATED THERETO, CODE AND/OR PROGRAM RELATED THERETO, INCLUDING, BUT NOT LIMITED TO, CLAIMS, SUITS OR CAUSES OF ACTION INVOLVING ALLEGED INFRINGEMENT OF COPYRIGHTS, PATENTS, TRADEMARKS, TRADE SECRETS, OR UNFAIR COMPETITION.

INDEMNIFICATION: TO THE MAXIMUM EXTEND PERMITTED BY LAW YOU AGREE TO INDEMNIFY AND HOLD HARMLESS THE INTELLECTUAL PROPERTY OWNERS, COPYRIGHT HOLDERS AND CONTRIBUTORS, AND EMPLOYEES, AND ANY PERSON FROM AND AGAINST ALL CLAIMS, LIABILITIES, LOSSES, CAUSES OF ACTION, DAMAGES, JUDGMENTS, AND EXPENSES, INCLUDING THE REASONABLE COST OF ATTORNEYS' FEES AND COURT COSTS, FOR INJURIES OR DAMAGES TO THE PERSON OR PROPERTY OF THIRD PARTIES, INCLUDING, WITHOUT LIMITATIONS, CONSEQUENTIAL, DIRECT AND INDIRECT DAMAGES AND

ANY ECONOMIC LOSSES, THAT ARISE OUT OF OR IN CONNECTION WITH YOUR USE, MODIFICATION, OR DISTRIBUTION OF THIS SPECIFICATION, SOFTWARE RELATED THERETO, CODE AND/OR PROGRAM RELATED THERETO, ITS OUTPUT, OR ANY ACCOMPANYING DOCUMENTATION.

GOVERNING LAW: THE RELATIONSHIP BETWEEN YOU AND ROBERT BOSCH GMBH SHALL BE GOVERNED SOLELY BY THE LAWS OF THE FEDERAL REPUBLIC OF GERMANY. THE STIPULATIONS OF INTERNATIONAL CONVENTIONS REGARDING THE INTERNATIONAL SALE OF GOODS SHALL NOT BE APPLICABLE. THE EXCLUSIVE LEGAL VENUE SHALL BE DUESSELDORF, GERMANY.

MANDATORY LAW SHALL BE UNAFFECTED BY THE FOREGOING PARAGRAPHS.

INTELLECTUAL PROPERTY OWNERS/COPYRIGHT OWNERS/CONTRIBUTORS: ROBERT BOSCH GMBH, ROBERT BOSCH PLATZ 1, 70839 GERLINGEN, GERMANY AND ITS LICENSORS.

4.2 Revision history

Version	Date	Description
0.1	10.02.2020	Initial version
1.0	24.02.2020	Released