TLE985x Bridge Driver Getting Started

About this document

Scope and purpose

The „TLE985x Bridge Driver Getting Started“ Application Note gives a short introduction into the usage of the Bridge Driver module of the TLE985x. It bases on the example code provided by the SDK package. This document is intended to be used with the product's User Manual and Datasheet. It gives additional specific information about the Bridge Driver Module configuration in the Application.

Intended audience

All TLE985x Bridge Driver users which are familiar with the usage of the Embedded Power Software Development Kit (SDK).

Table of contents

1 Introduction

This Application Note bases on the software examples provided by the SDK package (in particular „BDRV_CCU6_ADC1_MOTOR_EXAMPLE_TLE9855“ for the TLE9855 EVALKIT from the SDK V1.1.8). It discusses the relevant settings in the Config Wizard and the respective lines in the source code.

The software examples run a DC motor with speed and direction control:

- Switch on/off the motor by the on-board push button
- Configure the direction of the motor by a software variable
- Change the speed of the motor by the on-board potentiometer (i.e. by an analog input value)

This Application Note focuses on the basic configuration of the Bridge Driver and the PWM generator CCU6. It doesn't go into details about the application control logic concerning direction and speed control which involves further modules like the ADC1 and the MON button.

2 Basic concept

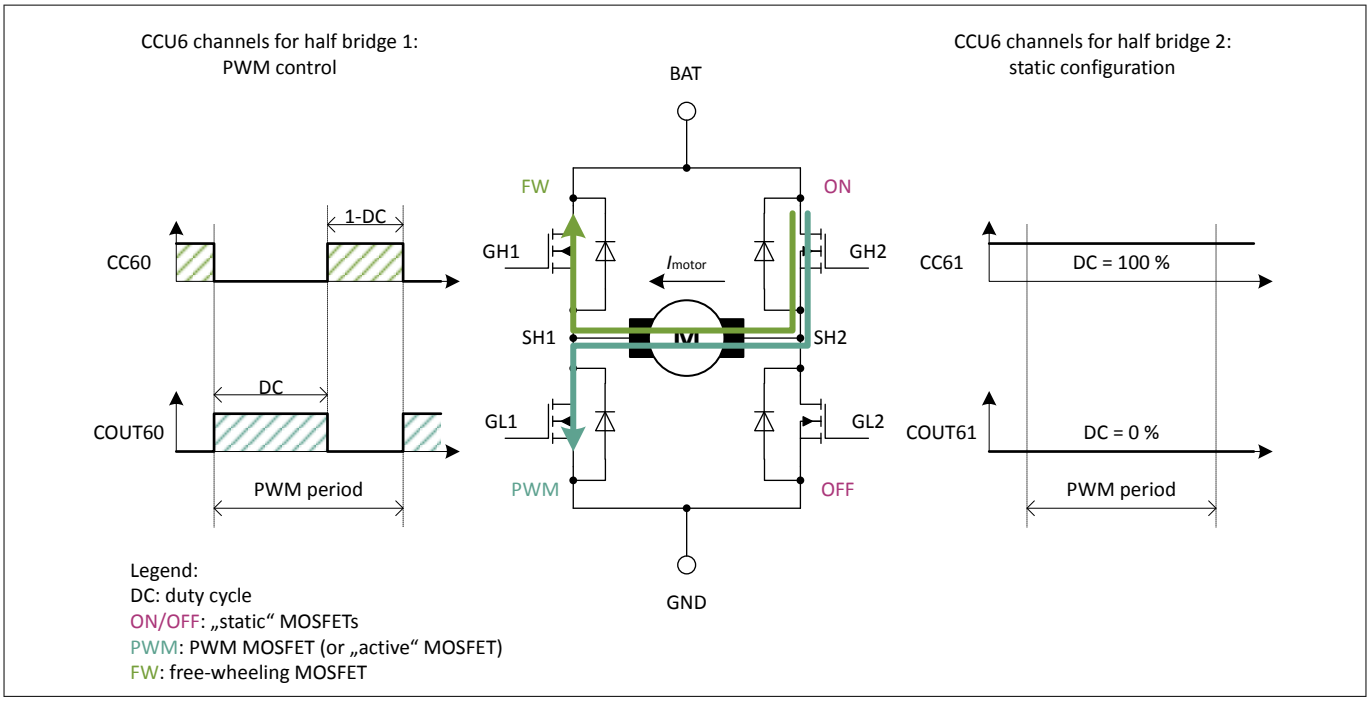

Figure 1 H-Bridge

The figure above shows the basic concept of PWM controlling a DC motor as it is implemented in the software examples in the SDK of the TLE985x:

- A typical H-Bridge configuration with the assigned gate driver outputs (see GHx/GLx).
- The role of each MOSFET and the paths of the motor current (see colored annotations).
- The mapping of the CCU6 channels and their PWM configuration (see CC6x/COUT6x).

Note: The static control of half bridge 2 (GH2 always on) is realized by setting the duty cycle of the respective CCU6 channels to 0 and 100%.

2 Basic concept

The resulting waveforms are expected to look like this:

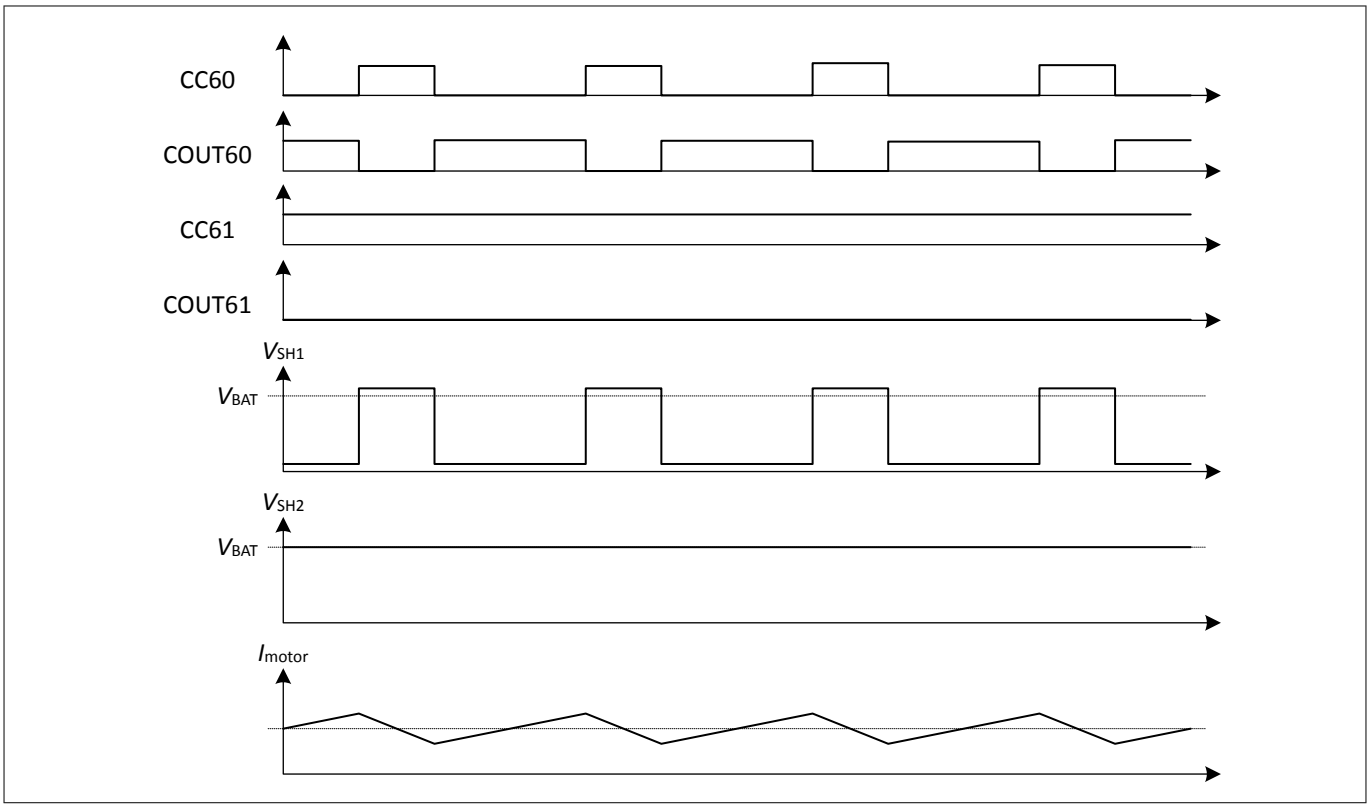

Figure 2 Waveforms

3 ConfigWizard settings

This chapter describes the small number of settings which have to be done in the ConfigWizard compared to the default values of an empty software project to be able to control a DC motor like demonstrated by the software examples of the SDK.

3.1 CCU6 Timer12

Figure 3 CCU6 Timer12 settings

CCU6 Timer12 settings:

- Period value: 50 µs (20 kHz)
- Dead time: 2 µs
- Dead time control enabled for channels 0 and 1

3.2 CCU6 Channels

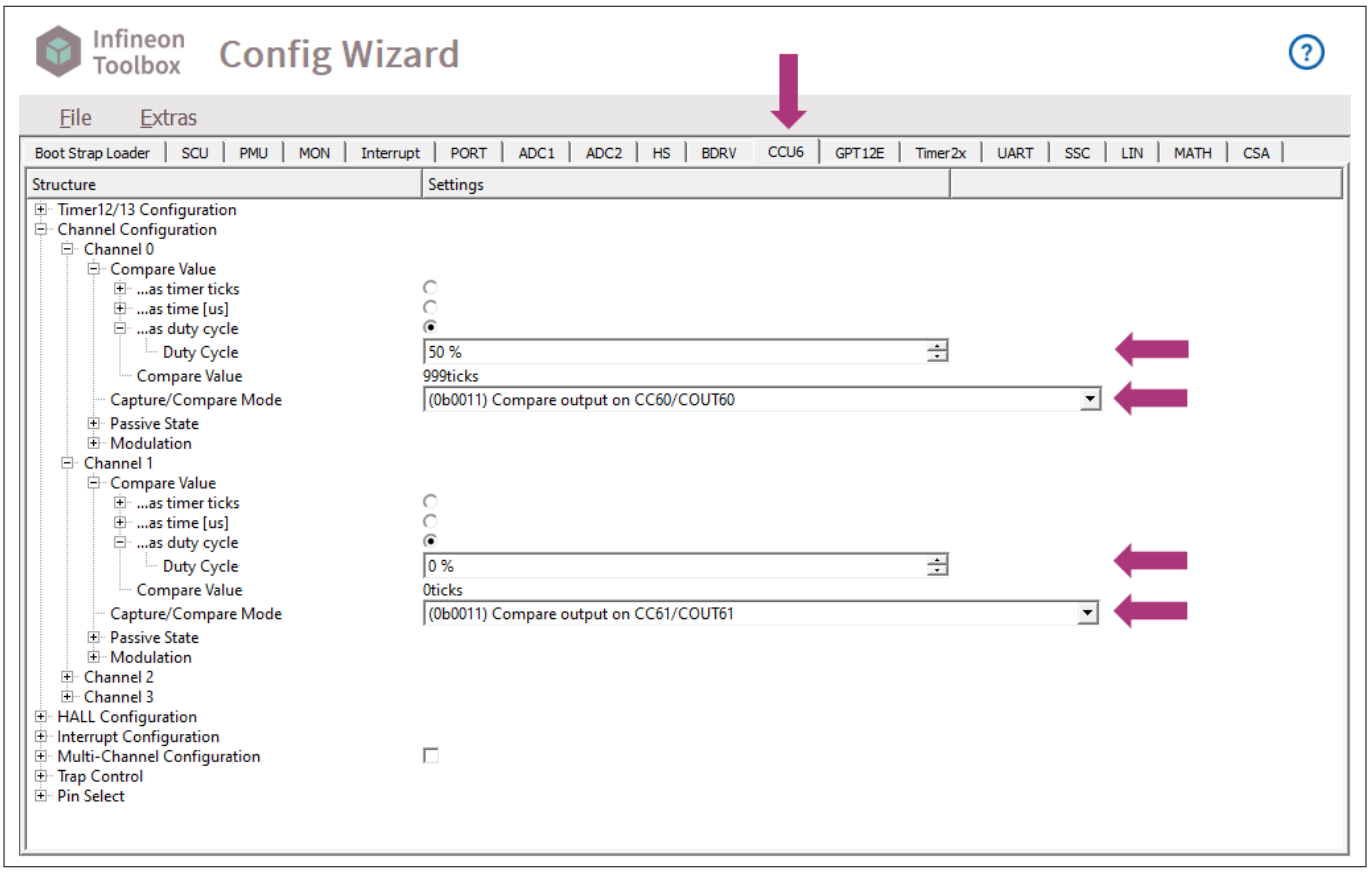

Figure 4 CCU6 channel settings

CCU6 channel settings:

- Channel 0 compare value as duty cycle: 50 % (serves only as a starting value and will be overwritten by software during runtime)
- Channel 0 capture/compare mode: enable compare output on CC60 and COUT60
- Channel 1 compare value as duty cycle: 0 % (i.e. 0 % at GL2 and 100 % at GH2 to switch on the high-side MOSFET of half bridge 2)
- Channel 1 capture/compare mode: enable compare output on CC61 and COUT61

3.3 BDrv

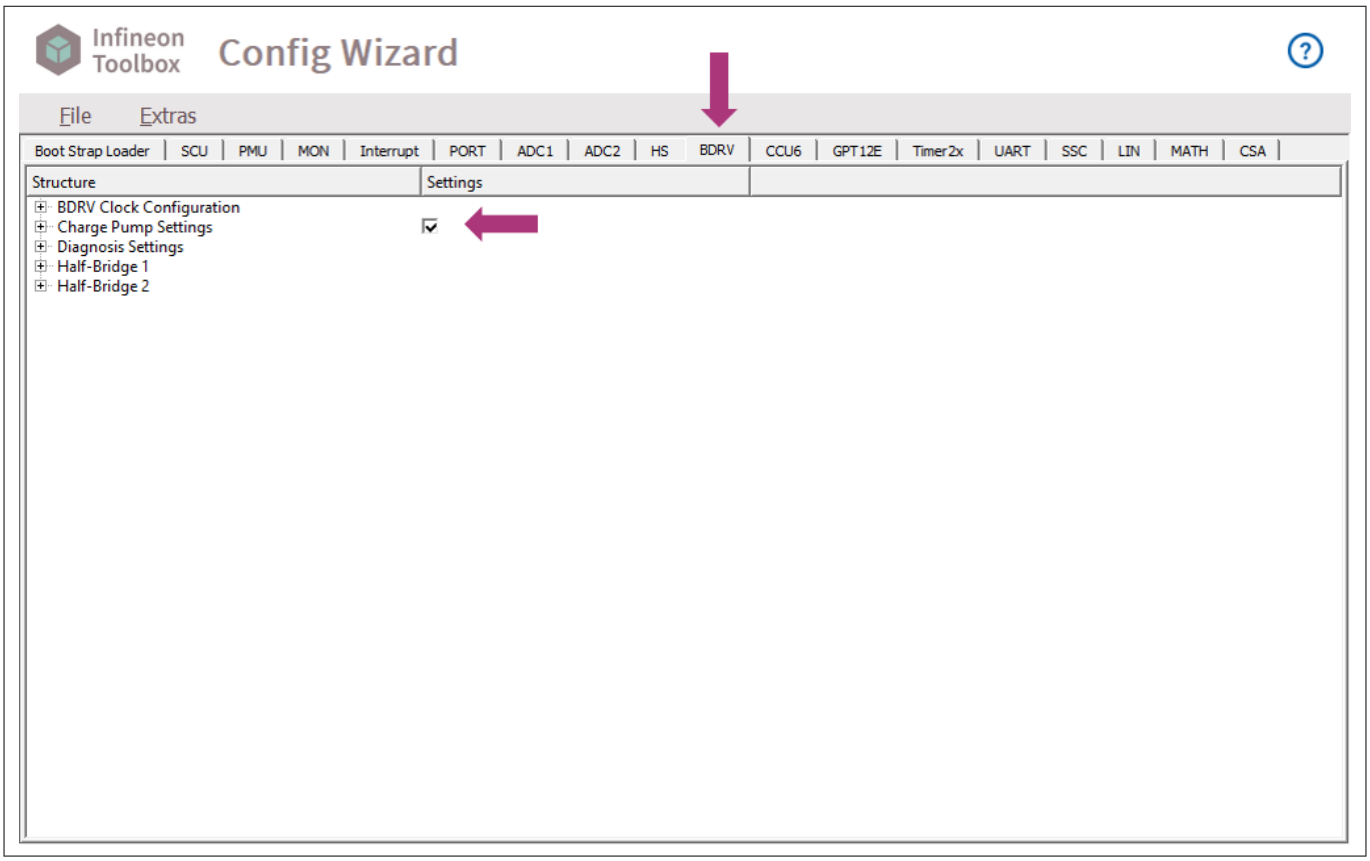

Figure 5 BDrv settings

BDrv settings:

- Enable the charge pump

4 Source code

This chapter describes the small number of code lines in the software example "BDRV_CCU6_ADC1_MOTOR_EXAMPLE_TLE9855" of the SDK which are necessary to run a DC motor.

4.1 Initialization

```
int main(void)
{
  uint16_t adc_poti;
  bool motor_on = false;
  /* Set motor_dir to direction_clockwise */
  /* motor_dir can also be set to direction_counterclockwise here */
  enum motor_directions_t motor_dir = direction_clockwise;


  /*******************************************************************************
  ** initialization of the hardware modules based on the configuration done     **
  ** by using the IFXConfigWizard                                               **
  *******************************************************************************/
  TLE_Init();

  /* Initiate T12 shadow transfer */
1 CCU6_T12_Str_En();
  /* Start T12 */
2 CCU6_StartTmr_T12();

  /* set PWM signals to LS1, HS1, LS2 and HS2 */
3 BDRV_Set_Bridge(Ch_LS_PWM, Ch_PWM, Ch_LS_PWM, Ch_PWM);
```

Figure 6 Initialization code

The three steps to be done during initialization are:

1. Initiate the shadow transfer for the CCU6 Timer12 registers to run the CCU6 according to the definitions done in the ConfigWizard
2. Start the CCU6 Timer12
3. Configure the Bridge Driver to be PWM controlled by the CCU6 output signals

4.2 Main loop

```
for (;;)
{
  /* service watchdog timer */
  (void)WDT1_Service();

  /* if MON1 button has been pressed */
  if (MONx_Get_Status(1u) == 0)
  {
    /* wait until MON1 buttom is released */
    MONx_Debounce(1u,1);

    if(motor_on == false)
    {
      /* turn motor on, T12 PWM pattern modulates CCU6 output signal */
      CCU6_T12_Modulation_En(CCU6_MASK_COUT60 | CCU6_MASK_CC60 | CCU6_MASK_COUT61 | CCU6_MASK_CC61);
      motor_on = true;
    }
    else
    {
      /* turn motor off */
      CCU6_T12_Modulation_En(0);
      motor_on = false;
    }
  }
```

1

2

Figure 7 Main endless loop

In the main loop the motor can be started or stopped according to:

1. Enable the CCU6 Timer12 modulation on all used channels: COUT60, CC60, COUT61, and CC61.
2. Disable the CCU6 Timer12 modulation on all channels.

5 Measurement results

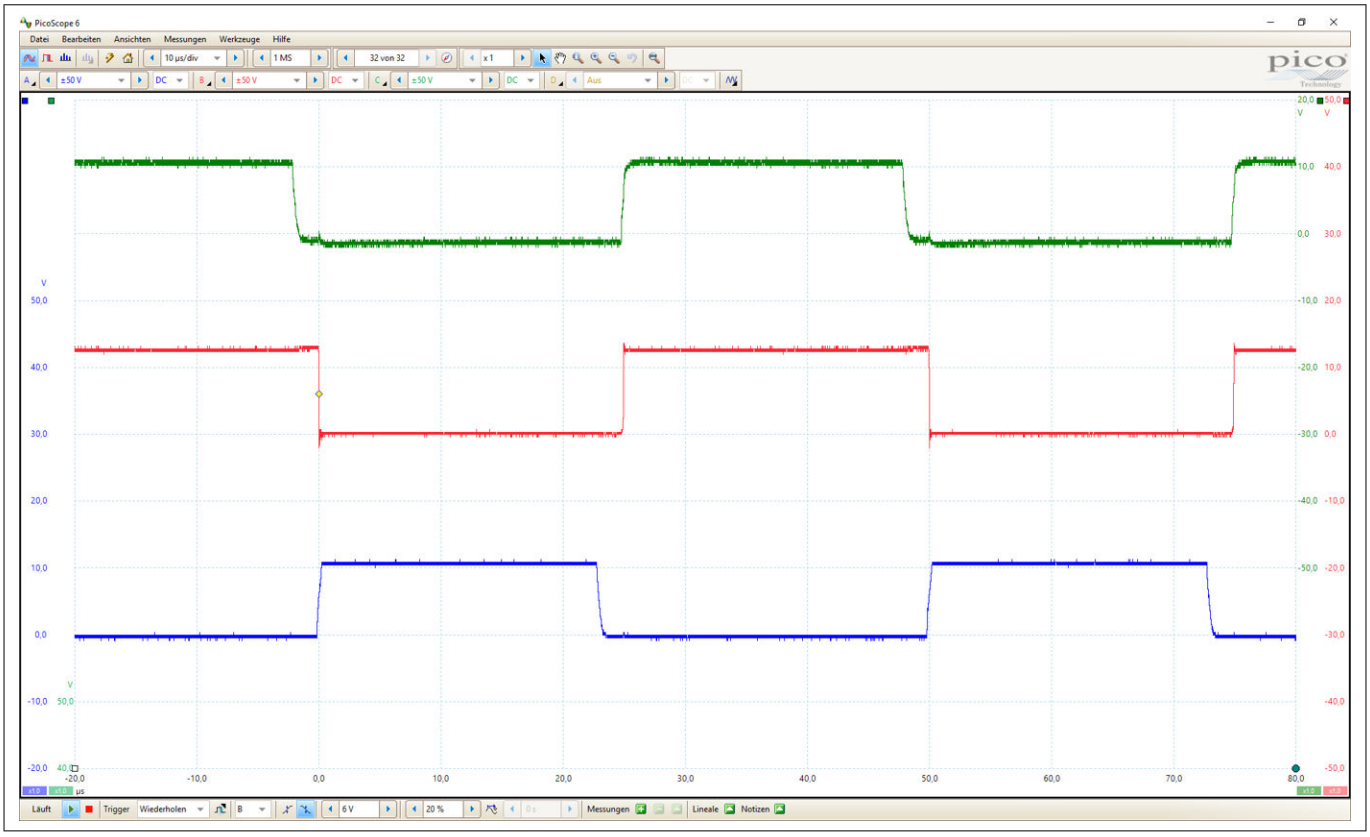

Figure 8 Scope plot

The scope plot shows the following signals of the half bridge 1:

- green: V_{GS} of the high-side MOSFET (free-wheeling MOSFET)
- red: resulting PWM voltage at the motor pin (V_{SH1})
- blue: V_{GS} of the low-side MOSFET (PWM or "active" MOSFET)

6 Appendix

6.1 Pitfalls

The software examples from the SDK run out of the box on the evaluation boards and kits. But, these default configurations are only starting points for further evaluation. If some parameters are changed these might cause problems. This chapter lists the most common pitfalls during Bridge Driver evaluation.

Charge pump voltage too low

The charge pump voltage must reach a certain level before the MOSFETs can be switched on. The status of the charge pump can be read by the Charge Pump Status Register BDRV_CP_IRQS.

Drain-source monitoring

By default, the drain-source monitoring comparators are configured to a low threshold voltage and short blanking and filter times. If, for example, the gate current settings are changed the default drain-source monitoring configuration may no more fit and can be adjusted by the ConfigWizard:

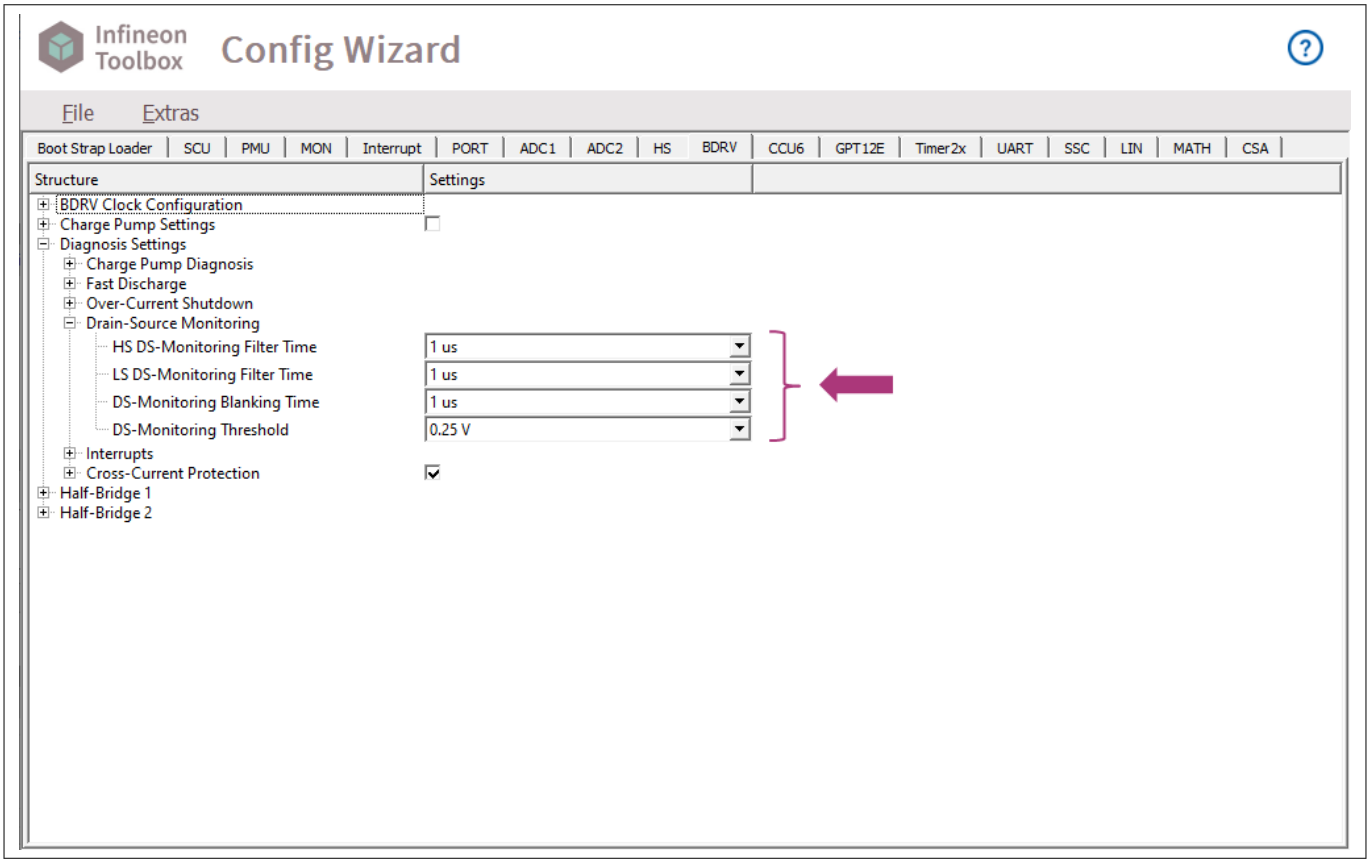

Figure 9 Drain-source monitoring settings

Status flags

If the Bridge Driver doesn't behave as expected it might have been switched off automatically due to an error. In this case the reason for switching off is stored in the respective status flags in the BDRV_IRQS or BDRV_CP_IRQS registers. After removing the root cause for the error the status flags have to be cleared using the respective clear bits in the BDRV_IRQCLR or BDRV_CP_IRQCLR registers before the Bridge Driver can be operated again.

Trademarks

All referenced product or service names and trademarks are the property of their respective owners.

Edition 2020-01-30

Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about any aspect of this document?

Email: erratum@infineon.com

Document reference
IFX-Z8F68092562

IMPORTANT NOTICE

The information contained in this application note is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this application note must verify any function and other technical information given herein in the real application. Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of non-infringement of intellectual property rights of any third party) with respect to any and all information given in this application note.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

WARNINGS

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury