FAQ Application Note for TLE986xQX, TLE987xQX

Frequently Asked Questions and Application Hints

About this document

Scope and purpose

This Application Note is intended to provide helpful suggestions and hints how to set up and handle specific modules and functionalities which are not subject of the Users Manual or Data Sheet and might be interesting for end users. It is organized in a frequently asked question style and doesn't follow any specific order.

Note: *The following information is given as a hint for the implementation of the device only and shall not be regarded as a description or warranty of a certain functionality, condition or quality of the device.*

Intended audience

This template is intended for Customer and FAE to document frequently asked question and answers for the embedded Power IC, TLE986xQX and TLE987xQX device family.

Table of Contents

1 Introduction

This Application Note lists topics emerged from frequently asked questions of users or from changed user requirements. Each topic is organized in three sections:

• **Topic:**

– Short description of the issue.

• **Description:**

– More details about the topic

• **Implementation hint (optional):**

– Instruction how to handle this topic

2 Collection of Questions and Topics

This chapter gives an overview of the collected Questions and Topics.

Table 1   Table of Questions

Topic	Chapter	Page
What PIN is connected to which peripheral? **GPIO Port Map and Alternate Functions**	**Chapter 3**	**Page 3**
How to connect the Debug Interface? **SWD (Serial Wire Debug) Interface Circuitry**	**Chapter 4**	**Page 5**
Why does the chip not start up, after reset? **Bootup Configuration**	**Chapter 5**	**Page 6**
Why does the LIN slope change after a NAC? **Bootup Configuration**	**Chapter 5**	**Page 7**
Why does the chip do resets every second? **Watchdog Handling WDT1**	**Chapter 6**	**Page 10**
Does the NAC affect the LIN slope? **LIN slope after NAC**	**Chapter 5**	**Page 7**
What state does the device enter after system overtemperature? **Device state after system overtemperature**	**Chapter 7**	**Page 12**
What is the demagnetization filter and how does it work? **BEMF comparator demag-pulse filter**	**Chapter 8**	**Page 14**
What is the difference between the VQFN and TQFP package variants? **VQFN vs TQFP package**	**Chapter 9**	**Page 16**
What is the difference between the on and off-state diagnosis in the bridge-driver module? **BDRV on-state vs off-state diagnosis**	**Chapter 10**	**Page 17**
What can the internal voltage regulators be used for? **Internal voltage regulators**	**Chapter 11**	**Page 18**
Which ASIL rating does the TLE9876x/7x have? **BEMF comparator demag-pulse filter**	**Chapter 12**	**Page 18**

Table 1   Table of Questions

Topic	Chapter	Page
How can the power dissipation inside the TLE986x/7x be calculated? **Power dissipation inside TLE986x/7**	**Chapter 13**	**Page 18**
How can very high/low PWM duty cycles be acchieved? **Very high/low duty cycle PWM**	**Chapter 14**	**Page 19**

3 GPIO Port Map and Alternate Functions

What PIN is connected to which peripheral?

The TLE986xQX and TLE987xQX have 15 port pins organized in three parallel ports: Port 0 (P0), Port 1 (P1) and Port 2 (P2). Each port pin has a pair of internal pull-up and pull-down devices that can be individually enabled or disabled. Either pull-up or pull-down devices can be enabled at a time, for a single port pin. P0 and P1 are bidirectional and can be used as general purpose input/output (GPIO) or to perform alternate input/output functions for the on-chip peripherals. When configured as an output, the open drain mode can be selected. On Port 2 (P2) analog inputs are shared with general purpose input.

3.1 Description: GPIO Register description

Each port consists of 8-bit control and data registers. The registers are defined in **Table 2**.

Table 2   Port Register

Register Short Name	Register Long Name	Description
Px_DATA	Port x Data Register	x = {0,1,2}
Px_DIR	Port x Direction Register	x = {0,1,2}
Px_OD	Port x Open Drain Control Register	x = {0,1}
Px_PUDSEL	Port x Pull-Up/Pull-Down Select Register	x = {0,1,2}
Px_PUDEN	Port x Pull-Up/Pull-Down Enable Register	x = {0,1,2}
Px_ALTSEL0	Port x Alternate Select Register 0	x = {0,1}
Px_ALTSEL1	Port x Alternate Select Register 1	x = {0,1}

3.2 Implementation: Alternate Function configuration example

The ports P0 and P1 can be configured to four different output functions. The default configuration is the GPIO function. The three remaining functions are alternate output functions.

The alternate output function selection is splitted in two bitfields (e.g. **P1_ALTSEL0** and **P1_ALTSEL1**). **ALTSEL1** contains the most significant bit. **ALTSEL0** contains the least significant bit. The given example code shows how to configure these bitfields to connect UART2 module (TXD, RXD) with the GPIOs (P1.0, P1.1).

```
/* connect UART2 to GPIO                              */
/* set P1.1 to UART2_TXD:                             */
PORT->P1_DIR.bit.P1 = 1u;          /* PORT P1.1 output configuration */
PORT->P1_ALTSEL0.bit.P1 = 1u; /* UART2_TXD alternate function 3  */
PORT->P1_ALTSEL1.bit.P1 = 1u; /* UART2_TXD alternate function 3 */
/* Set P1.2 to UART2_RXD:                             */
PORT->P1_DIR.bit.P2 = 0u;          /* PORT P1.2 input configuration     */
```

3.3 Implementation: Port Map of Alternate Functions

Graphical Portmap of Alternate Functions

Each pin is able to handle multiple purposes. **Figure 1** shows the internal signals mapped to GPIOs. The arrow boxes contain the signal names and indicate the data flow direction.
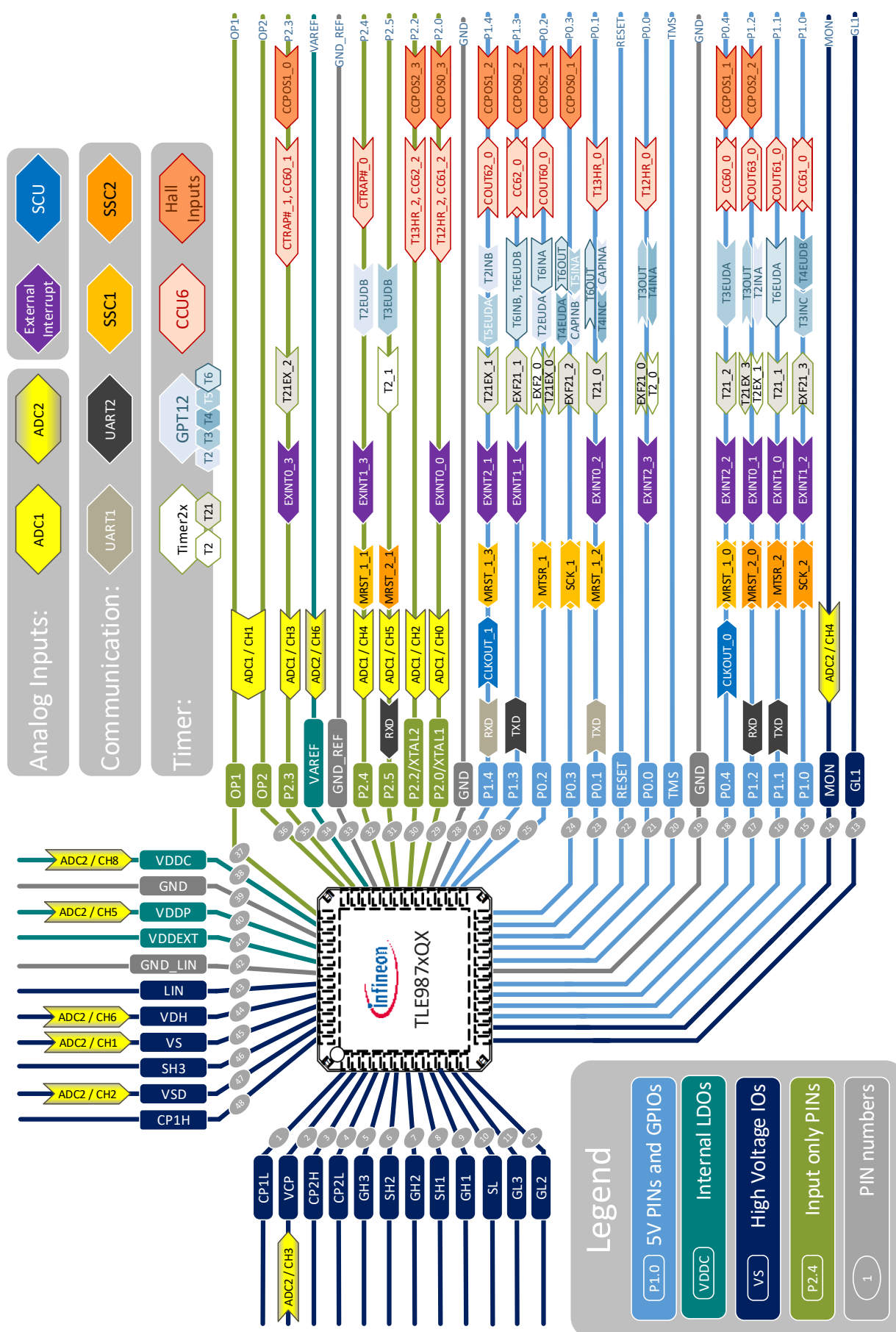
GPIO Port Map and Alternate Functions

Figure 1   Port Map of Alternate Functions

4 SWD (Serial Wire Debug) Interface Circuitry

How to connect the Debug Interface?

The Serial Wire Debug interface is used to download code to the embedded Power IC or to debug the chip. This Topic explains how to implement the circuitry around the chip to achieve a successfull SWD connection.

4.1 Description: SWD (Serial Wire Debug) Interface

Serial Wire Debug (SWD) provides a debug port for severely pin limited packages, often the case for small package microcontrollers but also complex ASICs where limiting pin-count is critical and can be the controlling factor in device costs.

For SWD the TLE9879 uses the pins TMS (data) and P0.0 (clock). On the Evaluation boards, the signals are routed through a 5x2 pinheader (SWD connector). The following Implementation explains the connection between embedded Power IC and SWD Interface.

4.2 Implementation: SWD Interface connection to TLE986x/ TLE987x

The SWD Interface can be directly connected to the TLE987x and TLE986x family. The use of external pull up or pull down resistors is not needed, due to internal pull down resistors. **Figure 2** shows the interconnections between TLE Device and and SWD Connector.
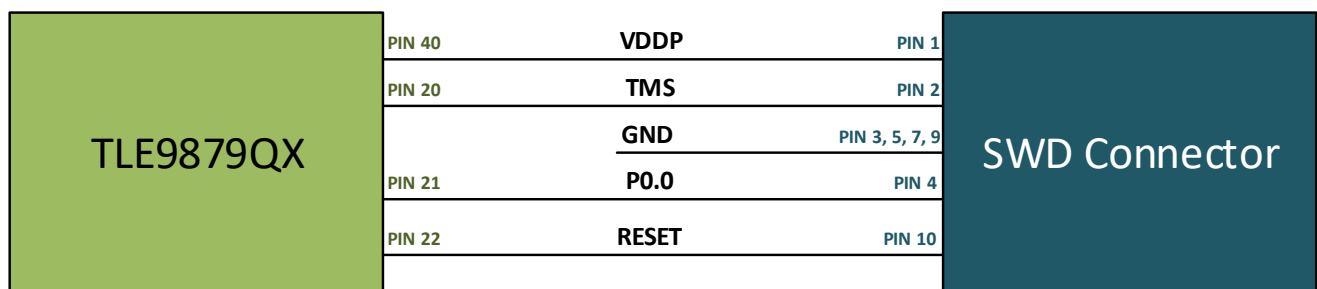

Figure 2   SWD Connection to the TLE987x and TLE986x Device

On TLE9879 and TLE9869 Evalkit SWD Interface PIN 9 is used to deactivate the onboard debugging circuit. For a typical implementation this PIN is used as GND. The Pinout is shown in **Figure 3**.
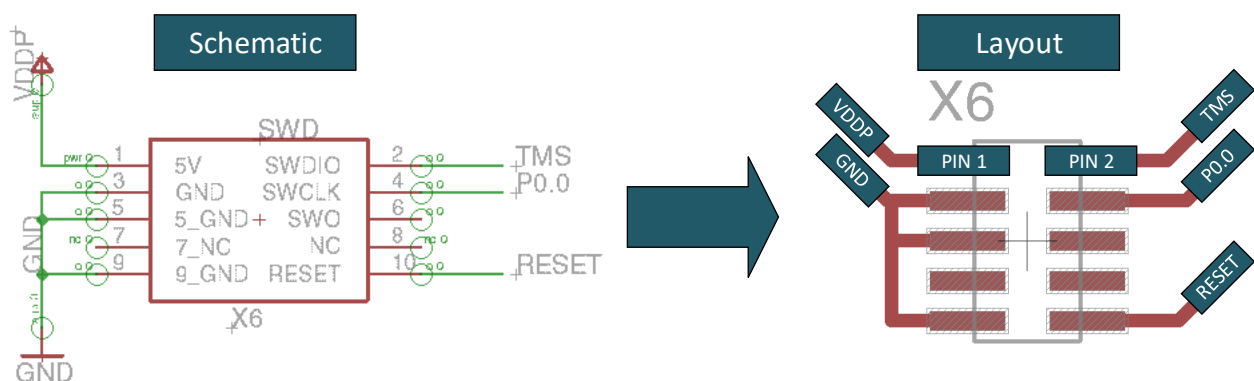

Figure 3   SWD Interface implementation for application

5 Bootup Configuration

Why does the chip not start up, after reset?

Using a new device can cause issues, due to the device is not executing user code. This chapter explains how to configure the chip to boot up and enter Users Code as expected.

5.1 Description: BSL Connection Window

After the reset PIN releases and some fundamental initializations of the device has been done, the BSL connection acception window starts. The duration of this timing window is defined by the None-Activity-Counter (**NAC**). In this time period, the device can be programmed via LIN. After the NAC expires, the chip will enter the user mode as shown in **Figure 4**.The following chapter provides some more inside view into the NAC value.
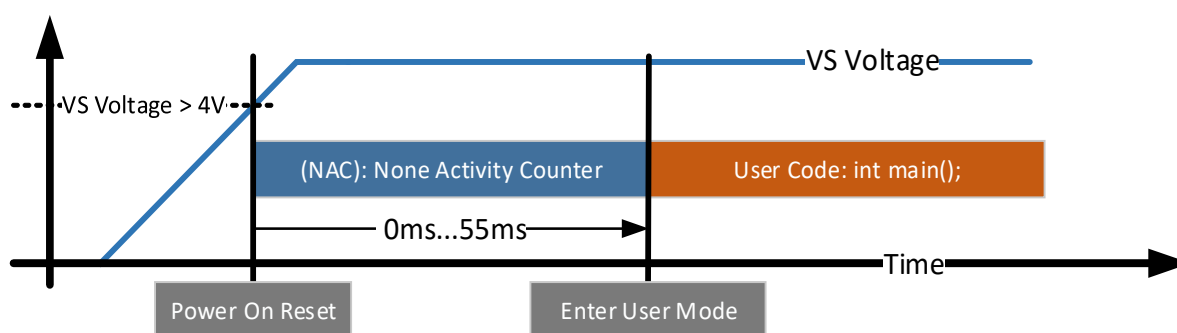

Figure 4   Startup Procedure

5.1.1 None-Activity-Counter - NAC

The NAC timer is started during startup before the BSL mode, inside the firmware, starts. Once the NAC timer has expired the boot-up process will be continued and the control of the device will be given to the user application. The NAC is a value which will be defined by the user and stored inside the code flash. Usually the NAC value is part of the user application which was downloaded before.

The NAC value is stored inside one byte. Only six bits of the NAC byte are defining the timeout value of the NAC. The NAC value inside the NVM is secured by storeing it as a 1s-complement value in the following byte . Only if the true NAC value and the complement NAC do match (means: not (true NAC) == complement NAC) the NAC value is valid and the NAC timer will be activated during start-up. In case of an invalid value, i.e. like for an erased flash where both bytes **containing 0xFF**, the **NAC timer never expires**, means the device will stay and wait in BSL mode. This behavior is especially usefull for fresh devices, where no user application has been downloaded to the device yet. Here the firmware will not branch into user mode, but instead it will stay in BSL mode and keeps waiting for any BSL communication to download any user application into the flash.

The NAC value and its complement is stored at the following addresses inside the user accessible flash:

Table 3   Address of the NAC values inside flash

Address	Usage
0x11000000 + (Total_Flash_Size - 0x1004)	true NAC value
0x11000000 + (Total_Flash_Size - 0x1003)	complement NAC value

The meaning of the NAC bits are listed in **Table 4**.

Table 4   NAC bit meaning

NAC Bit	Usage
5..0	NAC expire value n, (n - 1) * 5ms, 0ms..55ms 0x00, 0x0D..0x3F: NAC never expires, device stays in BSL Mode, BSL mode is active 0x01: BSL mode is skipped, BSL mode deactivated 0x02..0x0C: NAC timeouts between 5ms..55ms, BSL mode is active
6	Reserved
7	0b0: BSL interface selection Fast-LIN 0b1: BSL interface selection UART

A fresh device, or a completly ereased device always selects Fast-LIN as BSL interface and stays in BSL mode during start-up.

5.1.2   Node Address - NAD

The NAD is used for the BSL communication to select an individual node. The NAD is a 8 bit value, where only the values 0x01 to 0xFF are valid, 0x00 is an invalid value. The value 0xFF acts as a broadcast, this means all devices connected to the LIN line are addressed no matter which NAD value is programmed inside the device. The broadcast can be used to establish a BSL connection to devices where the programmed NAD value is unknown.

The NAD value is stored inside one byte inside the code flash area. The NAD value inside the NVM is secured by storeing it as a 1s-complement value in the byte following the NAD value. Only if the true NAD value and the complement NAD do match (means: not (true NAD) == complement NAD) the NAD value is valid otherwise the device will only react on the default NAD value, which is 0x7F.

The NAD value and its complement is stored at the following addresses inside the user accessible flash:

Table 5   Address of the NAD values inside flash

Address	Usage
0x11000000 + (Total_Flash_Size - 0x1002)	true NAD value
0x11000000 + (Total_Flash_Size - 0x1001)	complement NAD value

Table 6   NAD meaning

NAD Bit	Usage
0x00	invalid NAD value, device will not react on these value, default NAD = 0x7F is used
0x01..0xFE	valid NAD values
0xFF	broadcast NAD, all devices will react on this value

5.1.3   LIN slope after NAC

If a NAC value is configured between 0x02 and 0x0C the device will automatically configure the LIN module for fastLIN communication, a proprietary Infineon protocol passed on the LIN standard with a fixed baud rate of 115.2 kBaud. Due to the high baud rate the LIN slope will be automatically configured as well.

Table 7   LIN slope after NAC

Field	Bit	Value	Describtion
LIN->CTRL_STS.SM	12:11	0b11	Flash Mode

If no BSL connection is performed during the duration of the NAC, the device will leave the BootROM and jump directly into the user-code. If the user does not reconfigure the LIN in th user-code, the LIN slope will stay configured to Flash Mode, resulting in higher EMI.

5.2   Implementation: Write NAC NAD values to the correct position in Flash

According to **Chapter 5.1.1** and **Chapter 5.1.2** the NAC and NAD Value have to be written to the correct position in the Code Flash. **Figure 5** shows the correct position for every Device of the TLE987x and TLE986x embedded Power family.

The Implementation of the NAC-NAD value setting can be found in the Software Developement Kit based on Keil µVision5.

Bootup Configuration

Sector No.	Sector Start Addr.	TLE987xQX, TLE986xQX			
		128K Device	64K Device	48K Device	36K Device
31	1101F000	Data Flash			
30	1101E000	NAC / NAD			
29	1101D000				
28	1101C000				
27	1101B000				
26	1101A000				
25	11019000				
24	11018000				
23	11017000				
22	11016000				
21	11015000				
20	11014000				
19	11013000				
18	11012000				
17	11011000				
16	11010000				
15	1100F000		Data Flash		
14	1100E000		NAC / NAD		
13	1100D000				
12	1100C000				
11	1100B000			Data Flash	
10	1100A000			NAC / NAD	
9	11009000				
8	11008000				Data Flash
7	11007000				NAC / NAD
6	11006000				
5	11005000				
4	11004000				
3	11003000				
2	11002000				
1	11001000				
0	11000000				

Detail (Normal Code Flash): 1101EFFF NAD, 1101EFFE NAD, 1101EFFD NAC, 1101EFFC NAC, 1101EFFB, 1101EFFA, ·, ·, ·, 1101E003, 1101E002, 1101E001, 1101E000

Detail (Code Flash): 1100EFFF NAD, 1100EFFE NAD, 1100EFFD NAC, 1100EFFC NAC, · **Code Flash**

Detail (Code Flash): 1100AFFF NAD, 1100AFFE NAD, 1100AFFD NAC, 1100AFFC NAC, · **Code Flash**

Detail (Code Flash): 11007FFF NAD, 11007FFE NAD, 11007FFD NAC, 11007FFC NAC, · **Code Flash**
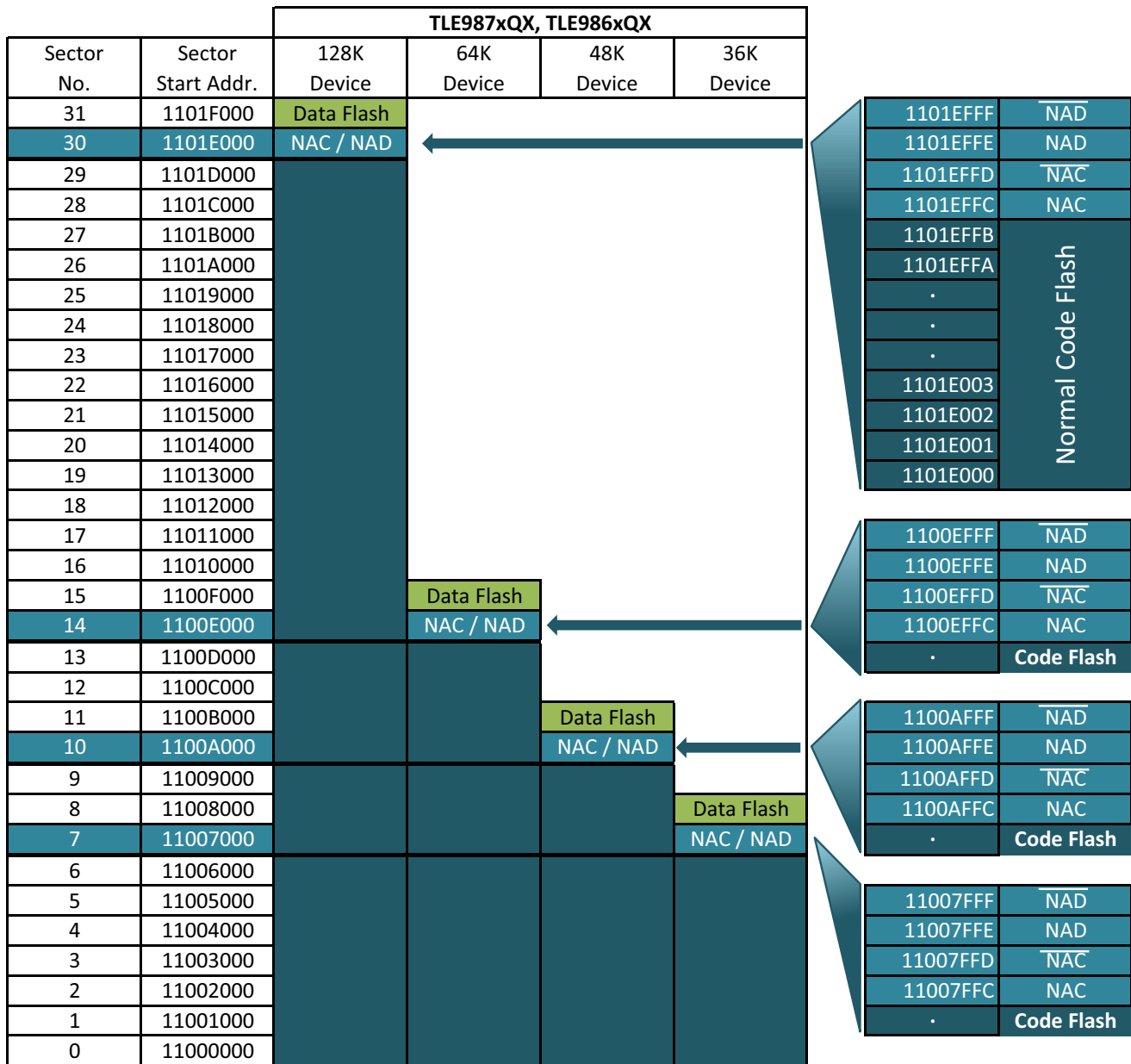
Figure 5   Memory Map

The following lines of code are part of the files "system_TLE987x.c/...6x.c". The defines can be found in "TLE987x.h/...6x.h" and "tle_device.h". The Code will also work "standalone"

#define ProgFlashSize   (0x8000U) /*Flashsize for TLE9871/...61   */
#define ProgFlashSize   (0xB000U) /*Flashsize for TLE9873QXW40   */
#define ProgFlashSize   (0xF000U) /*Flashsize for TLE9877/...67   */
#define ProgFlashSize   (0x1F000U) /*Flashsize for TLE9879/...69   */

Please, use only one of the definitions above, at a time. For example TLE9879(0x1F000U) for TLE9879 Evalkit.

#define NAD_NAC (0xFE01BA45u) /*Example: 4 bytes for NAC = 0x45u and NAD = 0x01u */
#define ProgFlashStart 0x11000000U)   /*start address code flash*/
#define DataFlashStart(ProgFlashStart + ProgFlashSize) /*start address data flash*/
#define NACStart   (DataFlashStart - 4U)   /*start NAC value   */
/* Set NAC NAD values as attribute: */
const uint32 p_NACNAD __attribute__((at(NACStart),used)) = (uint32)NAD_NAC;

6 Watchdog Handling WDT1

Why does the chip do resets every second?

The WDT1 Watchdog will perform frequently resets, if it is not serviced within the open window.

6.1 Description: Window Watchdog Timer WDT1

The WDT1 provides a reliable and secure way to detect and recover from SW or HW failures. It has an independent clocking source and power supply. If the WDT1 is not serviced (refreshed) within the allowed window a system malfunction is assumed and an internal RESET is performed.

A reset occurs with each missed service, or servicing in the wrong window. If WDT1 servicing failed 5 times, the device enters SLEEP MODE. The window can be freely programmed. The WDT1 cannot be switched off in Active Mode (exception in Debug Mode).

Figure 6 shows the relation between closed and open window. The safe trigger area expects the clock accuracy.
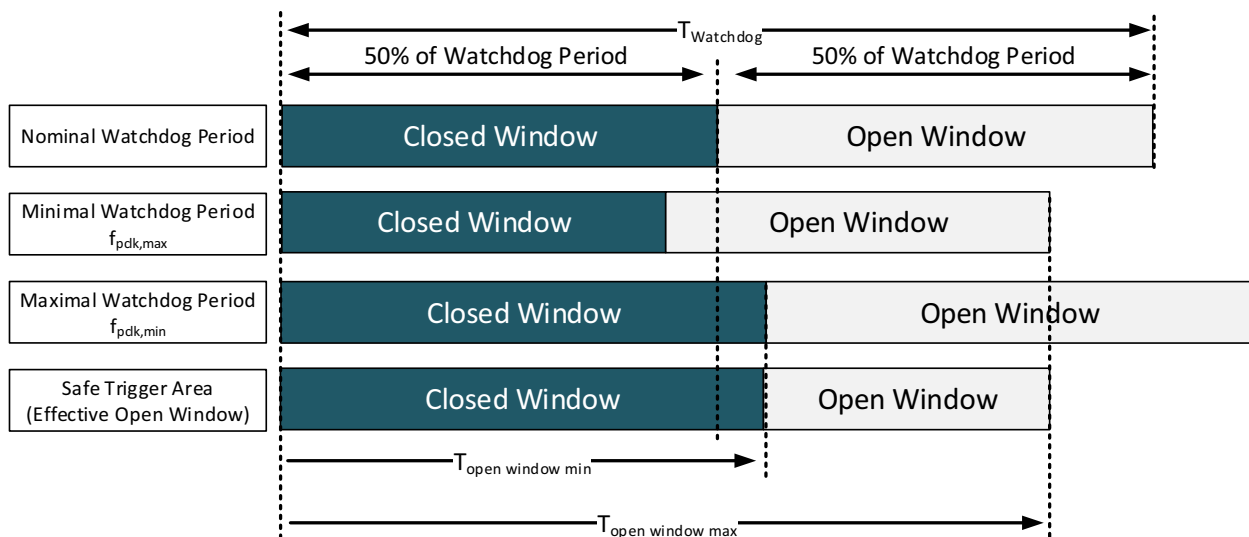
Figure 6   Watchdow Window Structure

6.2 Implementation: Watchdog Handling µVision 5

The following Code can be found in "wdt1.c". The "WDT1_Init()" function is called in "SystemInit()"

```
extern uint32 WD_Counter; /*Part of "Wdt1.h"                    */
void WDT1_Init(void)     /*Part of "Wdt1.c"                  */
{
 uint32 ui;
 /*calc SysTick reload based on SystemFrequency              */
 ui = (uint32)SCU_FSYS / SysTickFreq;
 CPU->SYSTICK_RL.reg = ui;  /* program SysTick timer             */
 CPU->SYSTICK_CUR.reg = 0u; /* reset SysTick timer              */
 CPU->SYSTICK_CS.bit.CLKSOURCE = 1u;/* CLKSRC=CPU clk             */
 CPU->SYSTICK_CS.bit.TICKINT = 1u;  /* TICK Interrupt = enabled       */
 CPU->SYSTICK_CS.bit.ENABLE = 1u;  /* ENABLE SysTick Timer          */
 SCUPM->WDT1_TRIG.reg = (uint8) SCUPM_WDT1_TRIG;/* trigger inital WDT1 service  */
 WD_Counter = 0u;        /* reset window counter             */
 bSOWactive = false;     /* reset SOW active signal           */
}
```

Watchdog Handling WDT1

In the code examples, included in the pack file, the Watchdog is serviced with "WDT1_Service(void)". The function checks if the window is already open. In this case, the Watchdog is triggered and a "1" is returned. If the Window Counter is less than 70% of the WDT1 period, the default value "0" is returned.

```
int WDT1_Service(void) /* This function is part of the file "wdt1.c"         */
{
 int result;
 result = 0;
 /* check if Window Counter is beyond 70% of WDT1 period            */
 /* or if a SOW service has been done before                */
 if ((WD_Counter > SCUPM_WDT1_TRIGGER) || (bSOWactive == true))
 {
  SCUPM->WDT1_TRIG.reg = (uint8) SCUPM_WDT1_TRIG; /* service WDT1         */
  WD_Counter = 0u;   /* reset window counter                */
  bSOWactive = false;/* reset "short open window" active flag         */
  result = 1;
 }
 return (result);
}
```

6.3 Implementation: WDT1 Hints

In this Chapter some further Informations are given. They can be used to find the reason of a potential issue.

6.3.1 Potential WDT1 Traps

- System seems not to run at all, but in Debugger it works
  - WDT1 gets serviced without the "Long Open" Window after reset
  - WDT1 would get serviced after the "Long Open" Window has expired
  - WDT1 is not getting serviced at all
- System runs to a certain point but then performs a reset
  - WDT1 gets serviced within the "Closed" Window part of the WatchDog period
  - WDT1 would get serviced after the WatchDog period has expired

6.3.2 Consequence

- Flashing of new "fixed" user code might not be possible anymore

6.3.3 Root Cause

- The device enters SLEEP Mode after five WatchDog fails, debugger connection is not possible

6.3.4 Solution:

- The device needs to keep awake, using MON1 as a wake-up source, and RESET as wake-up trigger
- For this purpose connect the pin RESET with pin MON1
- Set VS below 8V (the MONx threshold is defined as VS/2, RESET drives 5V max.)
- The output of RESET is fed into the MON1 and is recognized as wake-up event
- By this the device stays alive and can be reflashed
- after successfull flash update, the connection between RESET and MON1 can be removed, VS can be risen again

7 Device state after system overtemperature

What state does the device enter after system overtemperature?

After a system overtemperature detection the device will enter Sleep Mode. If cyclic wake is enabled the device will go into Sleep Mode with cyclic wake.

7.1 Description: System overtemperature detection

In case of overtemperature ($T_j > T_{j,max}$) the system will be sent to Sleep Mode. This functionality is intended to protect the system from thermal overstress. One possibility to avoid this thermal shutdown is to actively reduce the power dissipation of the system, by clocking down the microcontroller subsystem, or reducing the PWM frequency of motor control, which helps to reduce the power dissipation in the system. This procedure has to be implemented in user software, and should be triggered by the overtemperature prewarning (ADC2.Ch9 lower-threshold).

During Sleep Mode, the supply to the whole MCU subsystem including ADC2 is shut down. The cyclic Wake Mode triggers a synchronus wake-up after a predefined period in Sleep Mode. Once the period has elapsed, the PMU enters the Start-up Mode and proceeds to active Mode, where the software takes over the system control.

If system overtemperature is still present, the device will go back to Sleep Mode with cyclic wake. The state transition can be found in **Figure 7**.
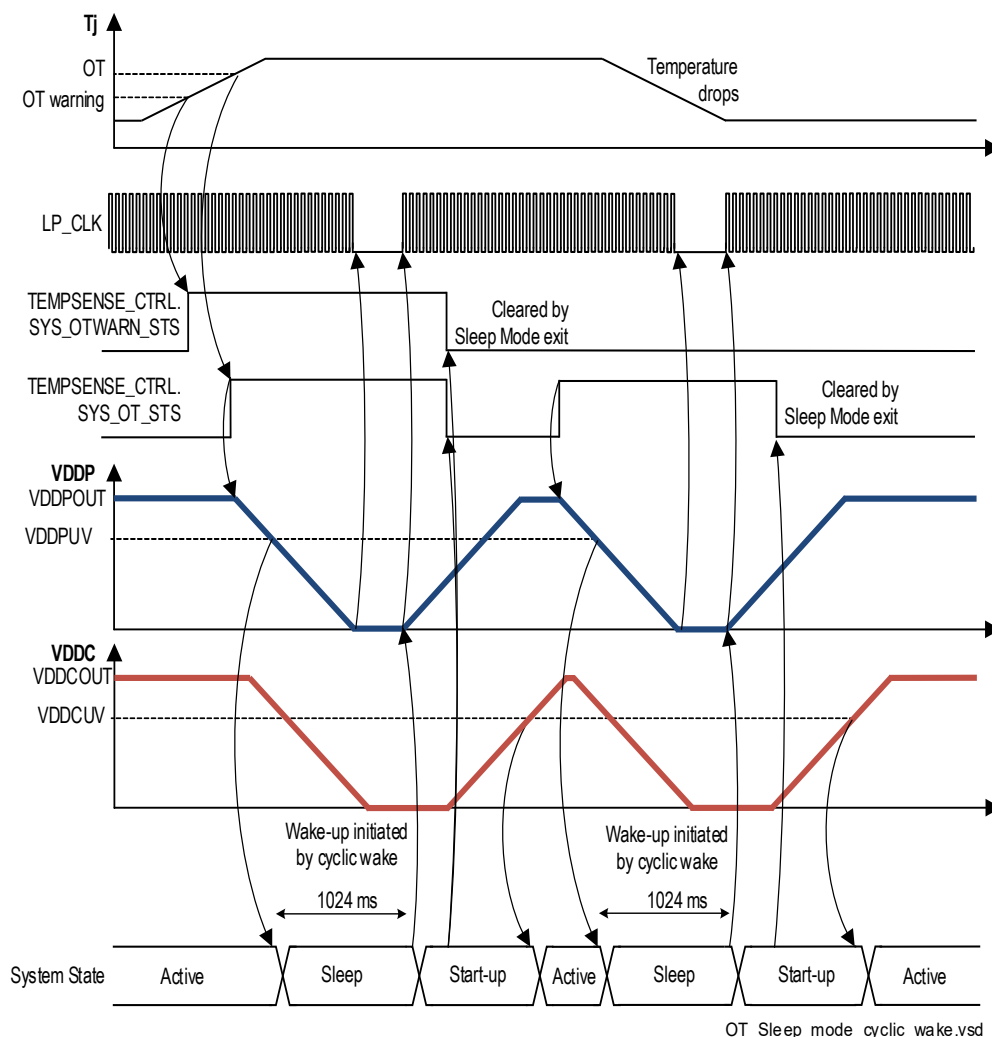
Figure 7   System overtemperature sytem state transition

Device state after system overtemperature

7.2 Implementation: Cyclic wake period

When entering Sleep Mode after system overtemperature, the period for cyclic wake will automatically be configured, any user settings for cyclic wake will be overwritten:

Table 8 Stop Mode after system overtemperature

Field	Bit	Value	Describtion
PMU-> CNF_CYC_WAKE.E01	5:4	0b11	Exponent value is 3
PMU-> CNF_CYC_WAKE.M03	3:0	0b0111	Mantissa value is 7

Resulting in a fixed wake-up period in case of system overtemperature: (7.1)

$$t_{period} = 4^{E01} \times (M03 + 1) \times 2ms = 4^3 \times (7 + 1) \times 2ms = 1024ms$$

8 BEMF comparator demag-pulse filter

What is the demagnetization filter and how does it work?

The demagnetization-pulse (demag-pulse) filter is used to get rid of wrongly detected BEMF zero-crossings caused by demag-pulses.

A demag-pulse is caused by the currents forced by the commutation of an excited motor armature out of or into the phase inductor. As long there is current flowing in the phase inductor, the phase at SHx is pulled up or down and the Back-EMF is not visible.

8.1 Describtion: Demag-pulse filter

The demag-filterd can be bypassed by setting:

Table 9 Disable demag filter

Field	Bit	Value	Describtion
MF-> BEMFC_CTRL_STS.DEM GFILTDIS	3	0b1	Demag filter is bypassed

The filter is a simple asynchronous state machine, which behaves as described in **Figure 8**:
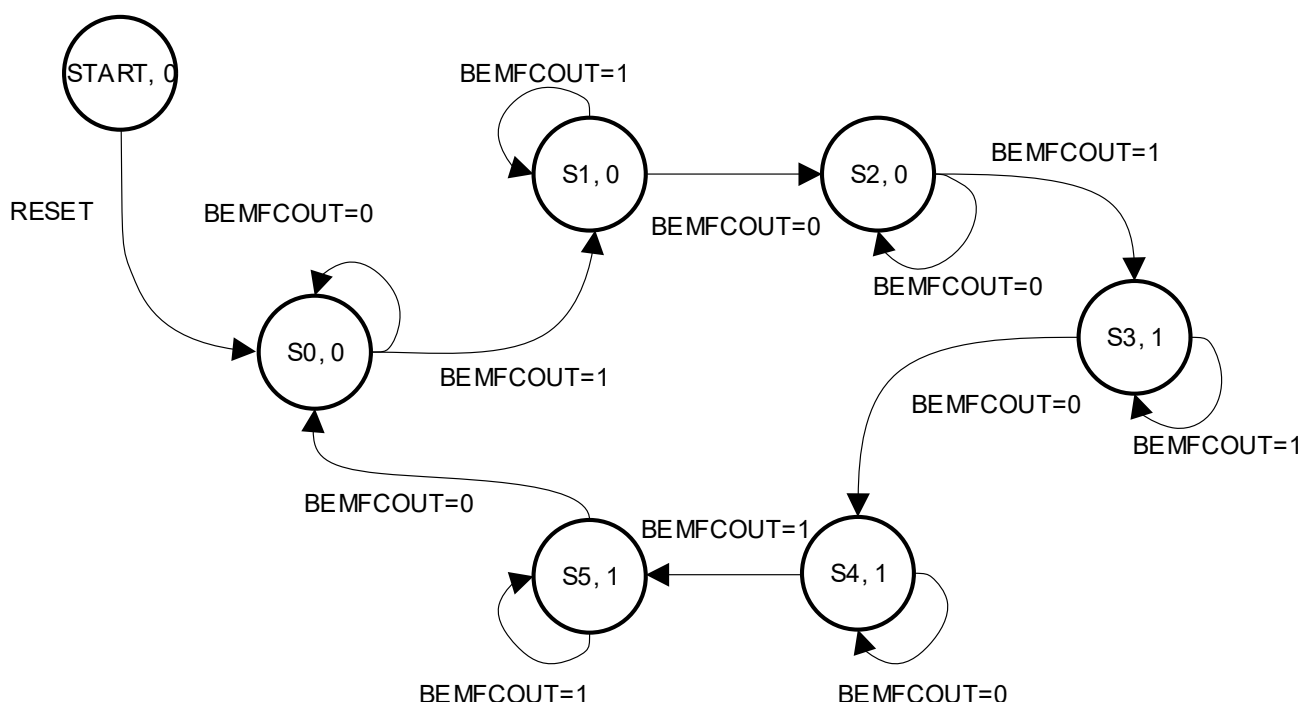
Figure 8 BEMF demag-pulse filter state diagram

The state is expressed by SX,Y (X is the state, Y is the state output). BEMFOUT is the BEMF comparator output.

8.2 Implementation: Demag-pulse filter output

Since not all motors show demag-pulses and for some conditions one of the two expected pulses might be weaker or stronger, the demag-pulse filter ouput may differ. The state transitions and resulting demag-pulse filter ouputs, depending on the presence of a demag-pulses, are illustrated below. In case the motor does only show one or no demag-pulse at all, the demag-filter has to be bypassed.

BEMF comparator demag-pulse filter

The scenarios are:

• BEMF with two demag-pulses (**Figure 9**)
• BEMF with only one demag-pulse (**Figure 10**)
• BEMF without demag-pulse (**Figure 11**)
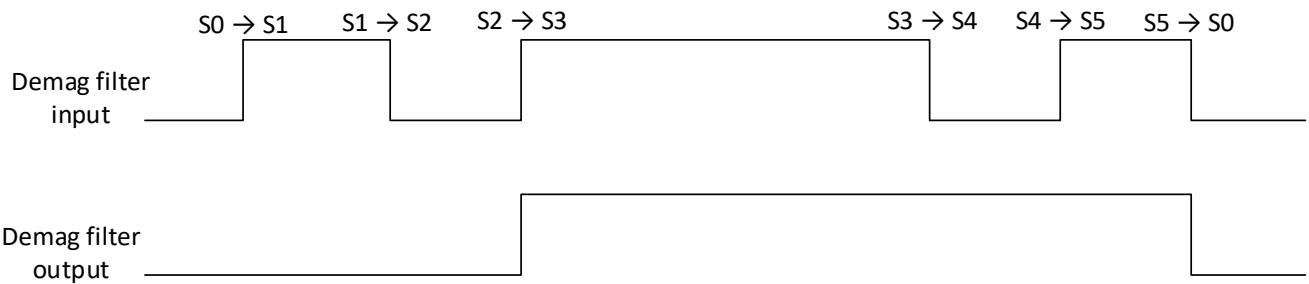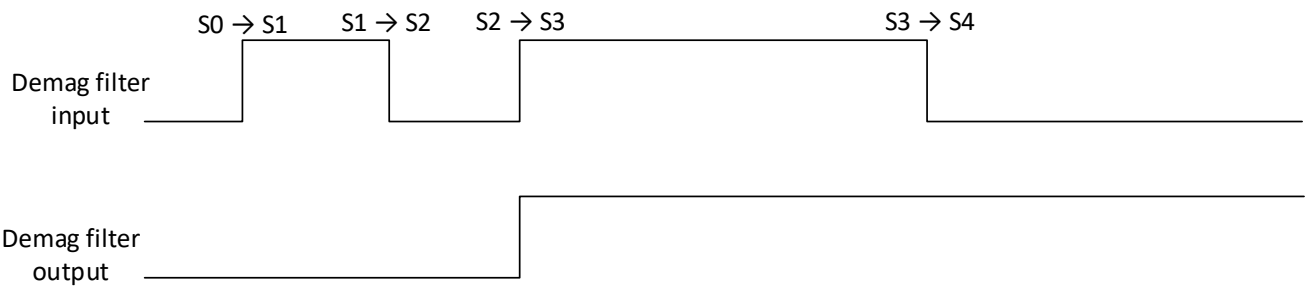

Figure 9   BEMF with two demag-pulses
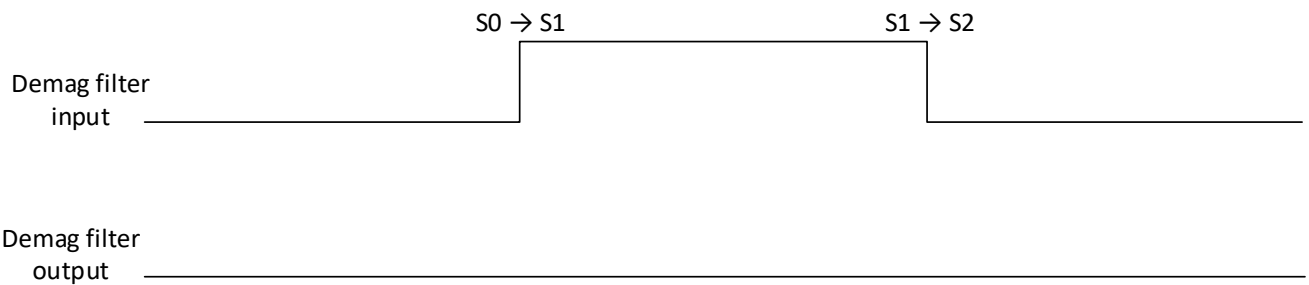

Figure 10   BEMF with only one demag-pulse

Figure 11   BEMF without demag-pulse

9 VQFN vs TQFP package

What is the difference between the VQFN and TQFP package variants?

While the VQFN package is leadless, the TQFP package does have a leadframe, which makes it resilient to cyclic thermal stress, and fit for optical inspection and soldering.

9.1 Implementation: Additional parameters and test conditions

The main difference between VQFN and TQFP are the different packages, but additonally some parameters and test conditons were added for the TQFP variants. The additional parameters and test conditions are listed below.

Table 10 Added parameter and test conditions

Parameter	Symbol	Values			Unit	Note or test condition	Number	Grade
		Min.	Typ.	Max.				
Voltage range at SL	VSL	-8.0		48	V	-	P_1.1.48	all
Junction to top	RthJTOP 2s2p	-	8		K/W	1) Not subject to production test, specified by design 2)According to Jedec JESD51-2,-5,-7 with natural convection on a FR4 2s2p board. Board: 76.2 x 114.3 x 1.5 mm^3 with two inner copper layers (35μm strong), with thermal dissipation via array under the exposed pad contacting the first inner copper layer and 300 mm^2 of cooling area on the bottom layer (70μm)	P_1.4.3	all
Accuracy_2	Acc_1	-10		10	°C	125°C <Tj <= 175°C	P_8.2.6	175°C
Wake-up/monitoring threshold voltage	V$_{MONth}$	0.4 x Vs	0.5 x Vs	0.6 x Vs	V	Without external serial resistor Rs (with Rs:DV = IPD/PU x Rs);VS = 5.5 V to 18 V; -40°C <= Tj <= 85°C	P_11.1.1	150°C
Maximum total charge driver capability (three-phase PWM)	Qtot_max, 20kHz			150	nC	1) Not subject to production test, specified by design. Due to charge pump current capability, six MOSFETs and additional external capacitors with a total charge of maximal 150 nC can be driven simultaneously at a PWM frequency of 20 kHz. VSD,min >= 6.5V for VGS,min >= 7V	P_12.1.120	all

10 BDRV on-state vs off-state diagnosis

What is the difference between the on and off-state diagnosis in the bridge-driver module?

The off-state diagnosis is only available if the external MOSFETs are off, while the on-state diagnos is only available if the external MOSFETs are on.

10.1 Description: diagnosis features in the BDRV

The BDRV of the TLE986x/7x does have several diagnosis features. It can be distinguished between on-state diagnosis features and off-state diagnosis features.

10.1.1 On-state diagnosis

The on-state diagnosis is available if the bridge-driver is active and the external MOSFETs are being switched. Each MOSFET is monitored by a drain-source voltage comparator. In case the drain-source voltage is higher than the limit configured in BDRV->CTRL3.DSMONVTH during the on-phase of the MOSFET, the affected MOSFET driver or all MOSFET drivers are switched off. This diagnosis feature is used to detect over current and protect the external MOSFETs from thermal overstress.

10.1.2 Off-state diagnosis

The off-state diagnosis is available if the bridge driver is active but no external MOSFETs are being switched. It is performed by an internally generated test current and the drain-source voltage comparators. This diagnosis feature is used to detect open load connections and shorts to ground or battery between the load connections.

10.2 Implementation: Register settings and status registers

Depending on the desired diagnosis feature, the according registers have to be configured differently and different status registers have to be monitored.

Table 11   On-state diagnosis vs off-state diagnosis

Register.bitfield	On-state diagnosis	Off-state diagnosis
CTRL3.IDISCHARGE_TRIM	Set to 0b00001	Not necessary
CTRL3.DSMONVTH	MOSFET dependent	Set to 0b000
CTRL1/2.HSx_DCS_EN	Not necessary	Set to 0b1 or 0b0
CTRL1/2.LSx_DCS_EN	Not necessary	Set to 0b1 or 0b0
CTRLL1/2.HS_DS_STS	Not necessary	Clear & Read
CTRL1/2.LS_DS_STS	Not necessary	Clear & Read
CTRL1/2.HSx_OC_STS	Read	Not necessary
CTRL1/2.LSx_OC_STS	Read	Not necessary

11 Internal voltage regulators

What can the internal voltage regulators be used for?

The TLE986x/7x do have serval integrated voltage regulators inside their power supply generation unit for the pad supply (VDDP), core supply (VDDC) as well as for external supply (VDDEXT). With the ADC reference voltage (VAREF), which is dervied from the band-gap voltage (VBG), these are the only internally generated voltages that are externally accesible.

11.1 Description: internal voltage generation

The voltages have different use-cases and have different ratings.

Table 12   Internal voltage regulators

	VDDEXT	VDDP	VDDC	VAREF
Voltage	5.0 V	5.0 V	1.5 V	5.0 V
Ioutmax	20/40 mA	30/50 mA	40 mA	
VMAX	7 V	7V	1.6 V	
Use as supply	Yes	Possible	No	No
Overload protection	Yes	Yes	Yes	No
Under voltage detection	Yes	Yes	Yes	No
Off-board usage	Possible	No	No	No

12 ASIL rating of TLE986x/7x

Which ASIL rating does the TLE9876x/7x have?

The TLE986x/7x is developed according to ASIL as a QM device.

12.1 Description: Functional safety with TLE986x/7x

Infineon does provide serval documents to support integration of the TLE986x/7x in functional safety critical applications. These documents can be provided on request:

- Pin FMEA (TLE987x only)
- FIT rate
- Area split
- Watchdog independency

13 Power dissipation inside TLE986x/7

How can the power dissipation inside the TLE986x/7x be calculated?

The power dissipation can be caculated by using the Power Dissipation Tool available in the Infineon Toolbox. (http://softwaretools.infineon.com/tools/com.ifx.tb.legacy.PowerDissipation.feature.feature.group). If more detailed calulations or analysis is required, please reach out to your Infineon contact.

14 Very high/low duty cycle PWM

How can very high/low PWM duty cycles be achieved?

When operating in PWM control the duty cycle is skewed by the dead time necessary to prevent cross-current between the the active PWM MOSFET and the active free-wheeling MOSFET. This does have an influence one the minimal or maximal duty cycle.

14.1 Description: CCU6 PWM generation

The internal Capture Compre Unit 6 (CCU6) generates the PWM to controll the internal bridge driver to drive external MOSFETs. The CCU6 has two timers, one of which is Timer12 (T12). T12 has several independent output channels, each with a complementary signal for active free-wheeling half-bridge operation. A compare value can be defined for each output channel x=[0;3] (usually only 0,1,2 are used to drive a B6 powerstage). The output "function" is based on the T12 counter-value passing by the compare value configured for every channel, resulting in a so called compare match (CM). It is completely user-definable on how the two complementary output channels (CC6x and COUT6x) for each channel x behave. The T12 counter-value is reset once it reaches its period-match (PM).

T12 can either run in edge aligned or center aligned mode. The example below shows the edge aligned mode, but the idea is also valid for center aligned mode and block commutation.

The dead time, the duty-cycle (DC) as well as the channel state before or after the compare match can be configured. Different settings for channels 0-2 are shown in **Figure 12**.

CCU6_deadtime_signal_state_edge.vsd
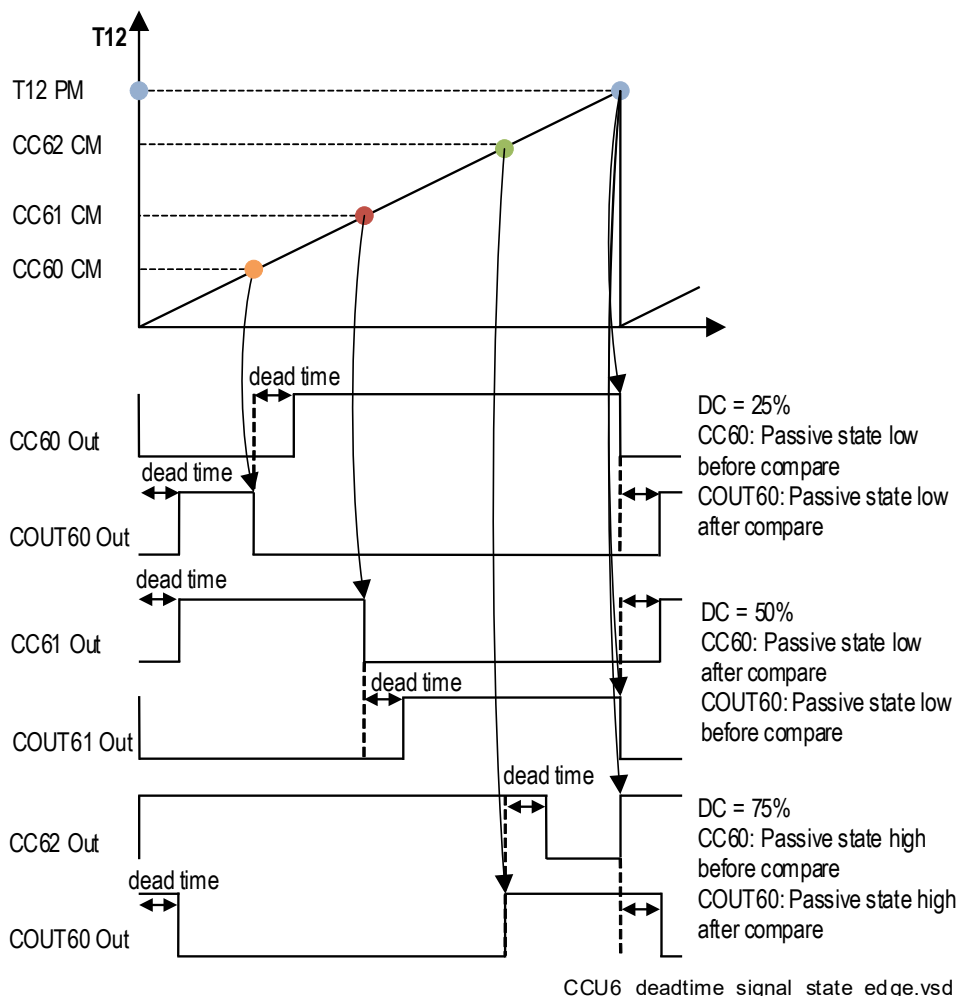
Figure 12 Different PWM generation patterns

14.2 Implementation: 100% duty-cycle

When operating with very high or very low DCs, the PWM generation pattern described below will behave differently. For example , one of the complementary channels will be constantly kept off or on if the configured dead time is greater than the configured compare match value. The maximal duty-cycle is limited by the configured dead time. If smaller/higher duty-cylces are to be configured, the dead time has to be decreased.

There are different CM settings resulting in different corner conditinons. These corner conditions are:

• CC6x CM < dead time

• CC6x CM + dead time > T12 PM

• CC6x CM= T12 PM

The resulting PWM pattere can be seen in **Figure 13**.

CCU6_deadtime_signal_state_high_low_DC.vsd
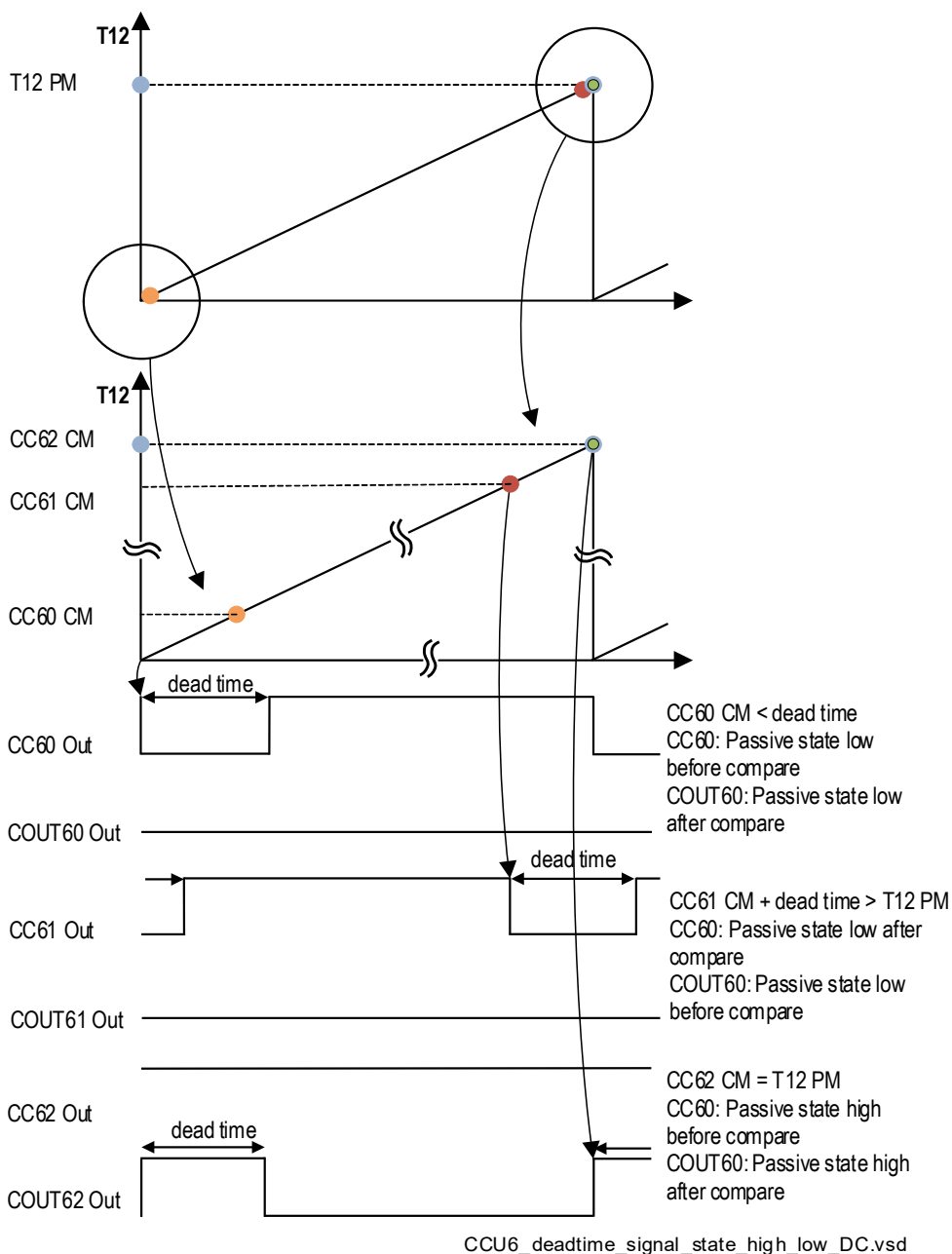
Figure 13   Very high/low DC generation

Very high/low duty cycle PWM

If CC6x CM = T12 PM the DC will be smaller than 100%, since the dead time will still be triggered. To achieve 100% PWM the CC6x CM has to be set to:

$$CC6x\ CM(100\%\ DC) > T12\ PM$$

This way the CM will never be triggered and the dead time counter never starts. There are different possibilities to achieve 100% DC PWM. All possibilities are displayed in **Table 13**.

Table 13   Configurations for 100% DC PWM

100% DC MOSFET	CC6x state	COUT6x state	Channel x compare value
LS	Low after compare	Low before compare	0
LS	Low after compare	Low before compare	> T12 PM
HS	Low before compare	Low after compare	> T12 PM
HS	Low before compare	Low after compare	0
HS	High after compare	High before compare	> T12PM
HS	High after compare	High before compare	0
LS	High before compare	High after compare	0
LS	High before compare	High after compare	> T12 PM

Attention: Passive state high is only recommended if the CCU6 signals are used with an external bridge driver that uses an inverted logic

15 Revision History

Revision	Date	Changes
1.1	2020-12-08	**Chapter 5.1.3**, **Chapter 7**, **Chapter 8**, **Chapter 9**, **Chapter 10**, **Chapter 11**, **Chapter 12**, **Chapter 13**, **Chapter 14** added
		Table 4 updated
		Chapter 2 updated
1.0	2017-04-21	Released version