OPTIREG™ TLF3068x

Service and configuration of Window Watchdog (WWD) via SPI

Protected registers

- Certain device configuration registers need to be protected against being unintentionally overwritten. Some of these registers are related to the WWD configuration.
- In order to write to protected registers, the procedure described within data sheet chapter 7.2.2 must be applied.
- Write access to protected registers is only possible after a dedicated 32 bit UNLOCK sequence has been sent via SPI to the PROTCFG register. The four bytes need to be sent without any other SPI register write access in between. Errors will generate an interrupt with GSF.SPI and SPISF.LOCK flags set and a new whole UNLOCK sequence has to be sent again. Once the UNLOCK sequence has been performed successfully, any write protected register can be written.
- The protected registers content are captured by the respective target modules only after a successful LOCK sequence has been performed and an internal configuration time of max. 60µs is elapsed.
- A successful LOCK sequence consists of four bytes which have to be sent with no other SPI register write access in between. Errors in doing so will generate an interrupt (same GSF.SPI and SPISF.LOCK flags are set) and a new complete LOCK sequence has to be sent again.
- Process of how to unlock protected registers.
  1. Unlock sequence (write PROTCFG)

CMD	A5	A3	A3	A2	A1	A0	D7	D6	D5	D4	D3	D2	D1	D0	Parity
1	0	0	0	0	1	1	1	0	1	0	1	0	1	1	0
1	0	0	0	0	1	1	1	1	1	0	1	1	1	1	0
1	0	0	0	0	1	1	0	1	0	1	0	1	1	0	1
1	0	0	0	0	1	1	0	0	0	1	0	0	1	0	1

Adress	Data	SPI hex
03_h	AB_h	8756_h
03_h	EF_h	$87DE_h$
03_h	56_h	$86AD_h$
03_h	12_h	8625_h

  2. Check protection status (read PROTSTAT)
- Process of how to lock protected registers
  1. Lock sequence (write PROTCFG)

CMD	A5	A3	A3	A2	A1	A0	D7	D6	D5	D4	D3	D2	D1	D0	Parity
1	0	0	0	0	1	1	1	1	0	1	1	1	1	1	0
1	0	0	0	0	1	1	0	0	1	1	0	1	0	0	0
1	0	0	0	0	1	1	1	0	1	1	1	1	1	0	1
1	0	0	0	0	1	1	1	1	0	0	1	0	1	0	1

Adress	Data	SPI hex
03_h	DF_h	$87BE_h$
03_h	34_h	8668_h
03_h	BE_h	$877D_h$
03_h	CA_h	8795_h

  2. Check protection status (read PROTSTAT)

Copyright © Infineon Technologies AG 2022. All rights reserved.

WWD configuration

- The WWD of the device is configured during the “long open window” as stated in chapter 7.5 of datasheet using the registers PWDCFG0, PWDCFG1 and PWDCFG2.
- These registers are protected and thus the procedure shown in slide 2 must be applied.
- Afterwards it is possible to set general timings, the error counter threshold, whether SPI or pin triggering is used and the WWD can be enabled as well as disabled.
- As soon as the WWD has been configured by writing to the PWDCFG0/1/2, the correct takeover into the device can be checked by means of the RWDCFG0/1/2 read-only registers.

- Thus a correct WWD configuration follows the procedure below:
  1. Unlock protected registers (PROTCFG)
  2. Check protection status (PROTSTAT)
  3. Perform all WWD configurations (PWDCFGx)
  4. Lock protected registers (PROTCFG)
  5. Check protection status (PROTSTAT)
  6. Check WWD configurations (RWDCFGx)
  7. Service WWD
- For servicing the WWD, the sampling time of the selected trigger event shall be considered in order to avoid a false triggering (e.g. servicing the WWD too close to the end of the open window could lead to a not recognized trigger event).
  - For SPI triggering, internally the WWDSCMD.TRIG is sampled depending on the selected WWD cycle time:
    - WDCYC=100us: WWDSCMD.TRIG is sampled with 200us
    - WDCYC=10us: WWDSCMD.TRIG is sampled with 20us
  - For WWD pin triggering, the watchdog sampling time is given with parameter t_{WDI_filter} in the datasheet.

Copyright © Infineon Technologies AG 2022. All rights reserved.

WWD service via SPI

› As stated in datasheet chapter 7.5 the WWD can be serviced either via SPI or WDI pin. Following the description how to service the WWD via SPI.

› In order to service the WWD via SPI, a correct write operation to register WWDSCMD has to be performed.

WWDSCMD
Window watchdog service command **(33_H)** **Reset Value:00_H**

7	6	5	4	3	2	1	0
TRIG_STATUS	nu						TRIG
r	r						rw

Field	Bits	Type	Description
TRIG_STATUS	7	r	**Window watchdog last trigger received via SPI** Reset: 00_H
nu	6:1	r	**Not used**
TRIG	0	rw	**Window watchdog trigger command** Reset: 00_H

› As shown in the picture left, the WWDSCMD register is not protected and thus the lock/unlock procedure must not be performed.

› Correct write operation using WWDSCMD:
1. read WWDSCMD.TRIG_STATUS
2. write WWDSCMD.TRIG=! WWDSCMD.TRIG_STATUS

Copyright © Infineon Technologies AG 2022. All rights reserved.



Part of your life. Part of tomorrow.