Q&A Watchdog Timer Configuration for DRV3205-Q1

ABSTRACT

The DRV3205-Q1 device features a highly configurable watchdog timer used to monitor an external microcontroller unit (MCU). This application report describes the functionality of the DRV3205-Q1 watchdog timer.

Contents

List of Figures

List of Tables

1 Watchdog Timer Overview

A typical system implementation for the DRV3205-Q1, as shown in Figure 1, includes a microcontroller unit (MCU) that controls the three phases of the gate driver and communicates over serial peripheral interface (SPI).

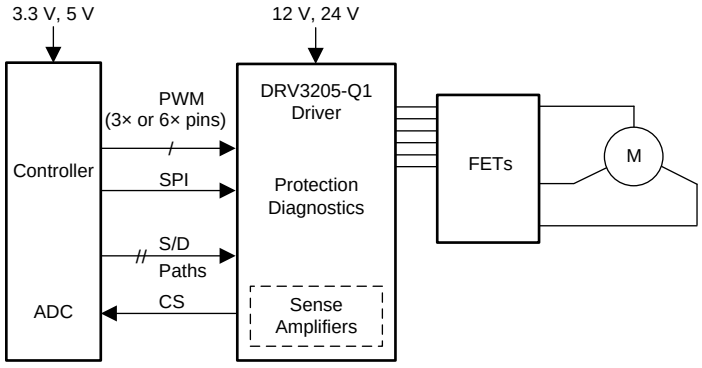

Figure 1. Typical Application Diagram

In this type of application, a failure in the MCU can have a catastrophic impact on the system. To help prevent further damage to the system, the DRV3205-Q1 device can be configured to monitor the health of the MCU using a watchdog timer. By periodically checking the MCU, the DRV3205-Q1 device can shut down the external MOSFETs and bring the system to a safe state.

1.1 Watchdog Operation

The question and answer watchdog operates on a periodic basis by sending specific message sequences through SPI. Upon the request from the MCU, the DRV3205-Q1 device provides a token (or question) to the MCU over SPI, latched in the WDT_TOKEN_VALUE register. The MCU performs a fixed series of arithmetic operations on the token value and returns the resulting token value answers to the ASIC over SPI by writing to the WDT_ANSWER register. The DRV3205-Q1 device verifies that the MCU returns the token-value results (answers) within the specified timing windows, and that the token value responses (answers) are correct.

When the MCU performs the watchdog-related SPI communications within the correct timing windows, and returns the correctly calculated responses (answers), the watchdog considers these *good events*.

When the MCU performs the watchdog-related SPI communications outside of the correct timing windows, or returns an incorrectly calculated token response (answers), or returns the correct answers in the wrong sequence, the watchdog considers these *bad events*.

When the MCU suspends watchdog-related SPI communications for the duration of the watchdog-timeout window, the watchdog considers this as a *no response event*.

An internal counter stores the number of bad responses in the WD_FAIL_CNT register, which triggers a failure if the WD_FAIL_CNT reaches a predefined limit. By specifying the limit in the WD_FAIL_MAX register, a buffer for the number of bad events can be set.

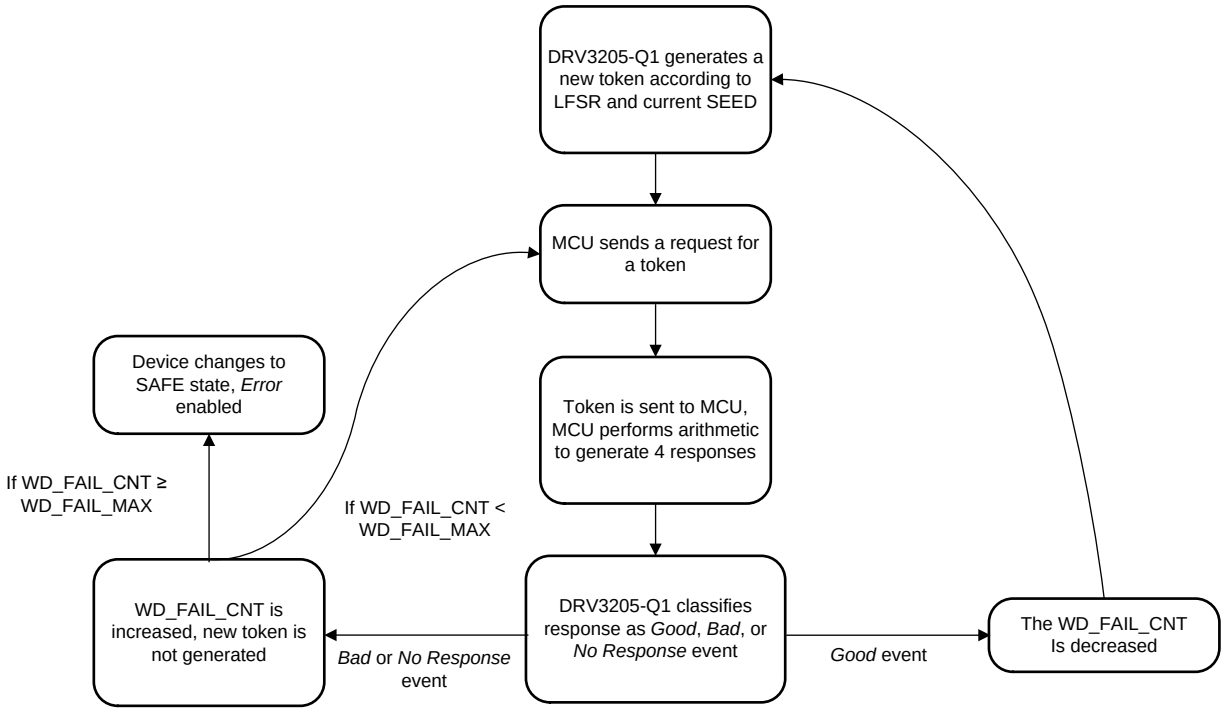

Figure 2. Simplified Watchdog-Timer Flow Chart

1.2 Timing Windows

The timing of the four responses is synchronized in two internal time windows. The first three responses must be sent during the first timing window (*open window*), configured in the RT field of the WDT_WIN1_CFG register. The last response must be sent during the second timing window (*close window*). The close window expires either after the programmed time, configured by the RW field of the WDT_WIN2_CFG register, or after the next clock cycle following the final response.

For detailed information, refer to the *DRV3205-Q1 Three-Phase Automotive Gate Driver With Three Integrated Current Shunt Amplifiers and Enhanced Protection, Diagnostics, and Monitoring*.

1.3 Response Events

1.3.1 Good Event

A good event occurs when the answers are sent in the correct order and within the correct timing windows. After a good event, the watchdog failure counter (WD_FAIL_CNT) decrements and a new token is generated by the DRV3205-Q1 device and the process restarts. The example in Figure 1 shows the value of the DRV3205-Q1 registers after each watchdog response.

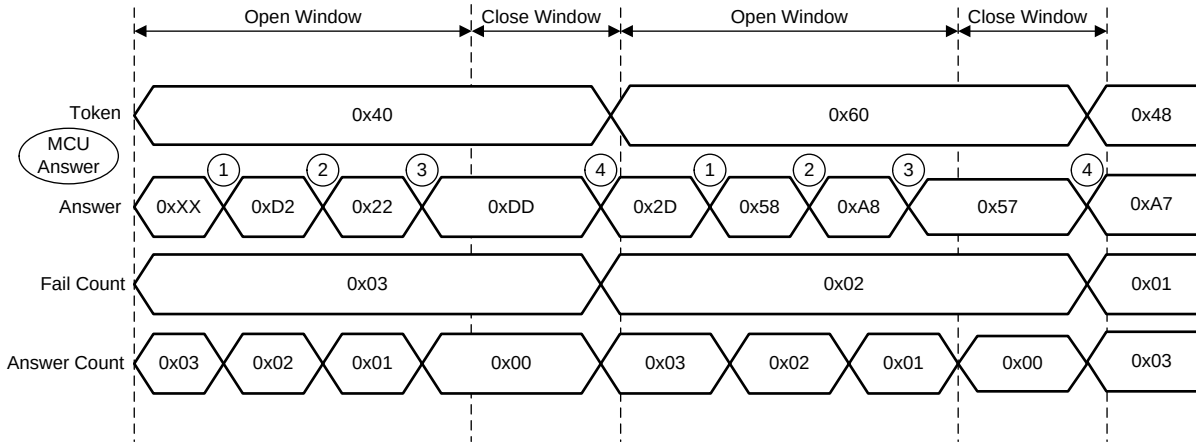

Figure 3. Watchdog-Timer Good Event

1.3.2 Bad Event

A bad event occurs when the MCU executes the watchdog sequence incorrectly. After any bad event, the DRV3205-Q1 device increases the watchdog fail count (WD_FAIL_CNT), and the token for the next frame remains the same. If the WD_FAIL_CNT counter exceeds the WD_FAIL_MAX value, the configured error condition is executed.

1.3.2.1 Incorrect Answer Value

An incorrect value written to the WDT_ANSWER register for one of the responses generates a token error and sequence error. The token error remains active until the next response is given and the sequence error remains set until the next correct sequence is executed.

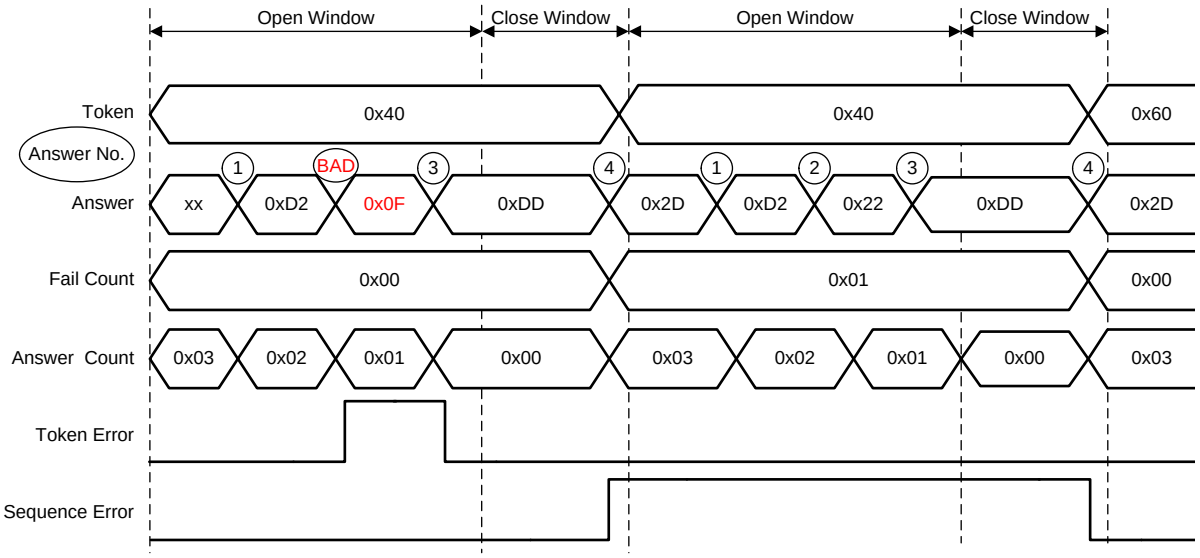

Figure 4. Incorrect Answer-Value Timing Diagram

1.3.2.2 Incorrect Answer Sequence

If a received token response is in the wrong order, the DRV3205-Q1 device responds similarly to the incorrect answer value case, except the token-error flag (TOKEN_ERR) remains set for every response that is out of order.

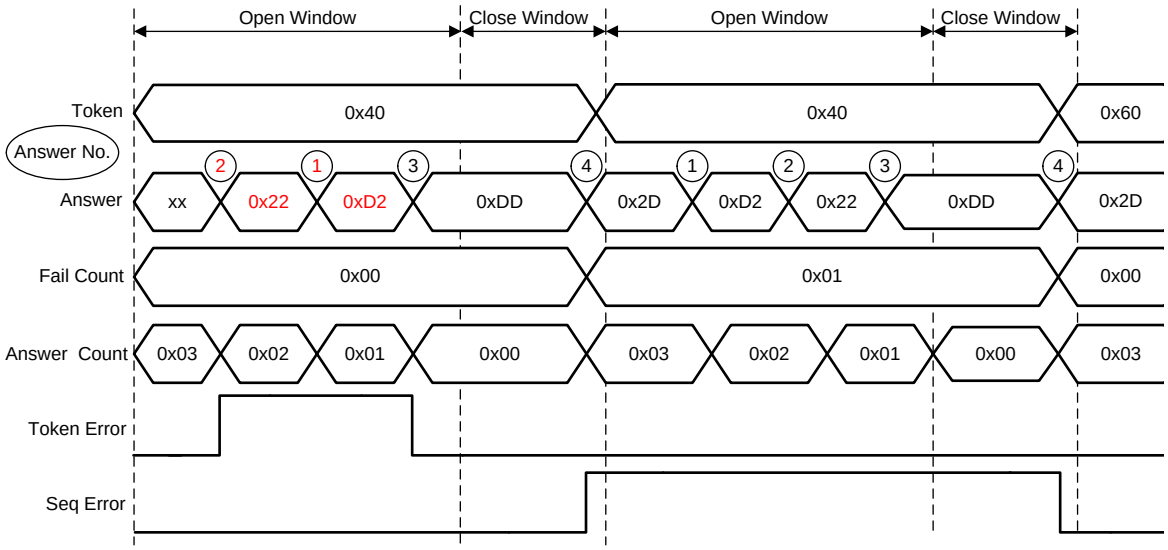

Figure 5. Incorrect Answer Sequence Timing Diagram

1.3.2.3 Out-of-Timing Window

Several possible response cases can result in responses being sent out of the correct window. These cases are as follows:

- Early response sent during the close window
- Fourth response in the open window
- Response received after the transaction

In the case that the fourth response is received in the open window, the DRV3205-Q1 device generates a token-early flag (TOKEN_EARLY), which is not cleared until the next good response event.

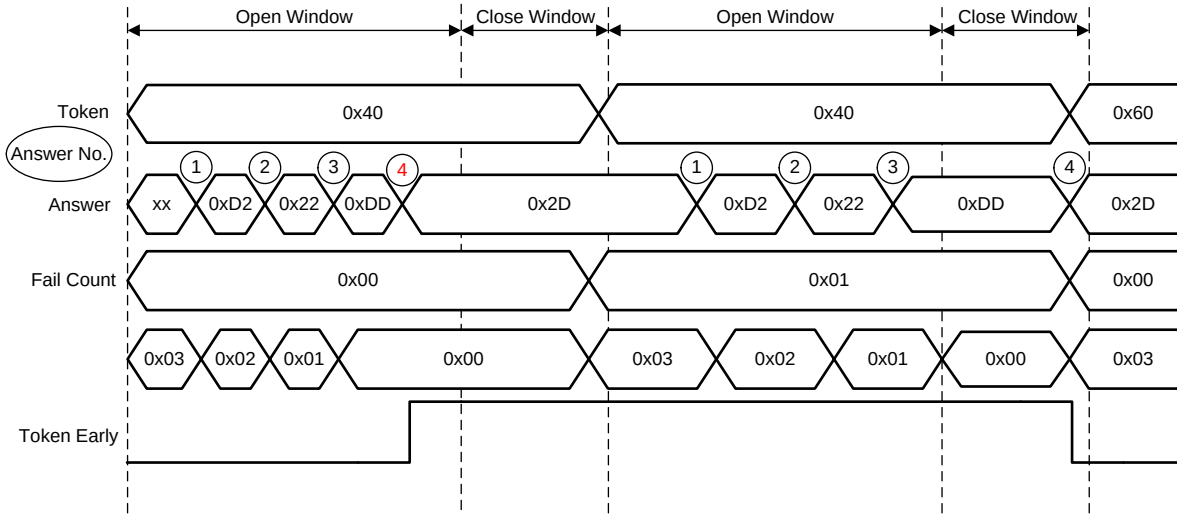

Figure 6. Out-of-Timing Window Timing Diagram

1.3.3 No Response Event

A no response event occurs if the MCU suspends the watchdog-related SPI communications for the duration of the watchdog window. During this event the TIME_OUT flag is set. This flag can be used by the MCU software to resynchronize the watchdog events on the required watchdog timing.

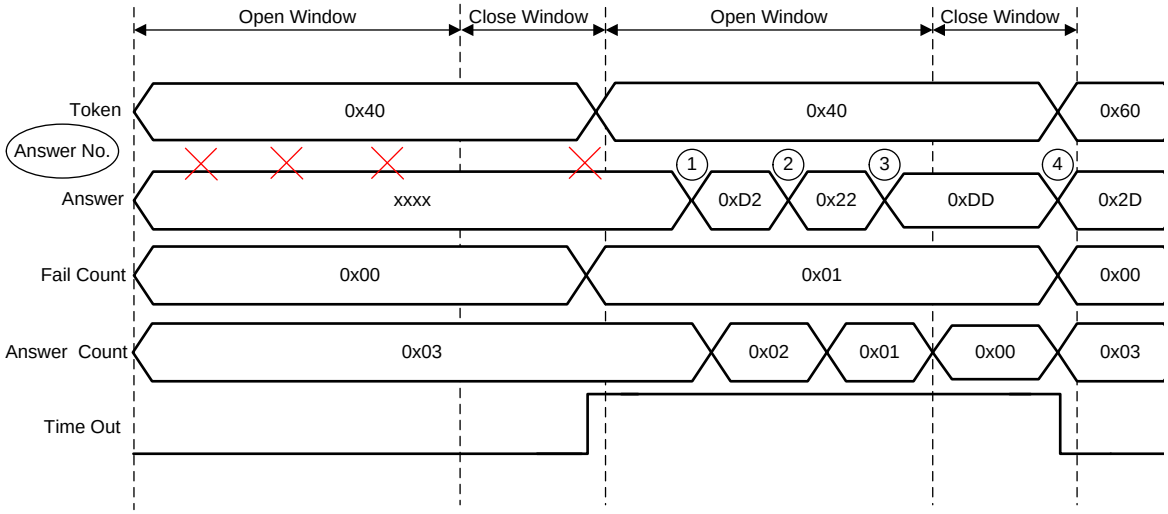

Figure 7. WDT No Response Event

1.4 Token Generation

The watchdog timer uses a linear-feedback shift-register (LFSR) circuit to generate a cyclic series of token values. The LFSR equation is set by default, and generates a total of 15 values for each cycle. The initial value (seed) of this LFSR is preloaded to a default value, and is set to a new value after every LFSR cycle.

After each good event, a four-bit internal WDT TOKEN counter is incremented. The combination of this counter value and the LFSR output is used to create the token value. When the WDT TOKEN counter overflows, the seed value is shifted based on the LFSR equation.

For detailed information, refer to the *DRV3205-Q1 Three-Phase Automotive Gate Driver With Three Integrated Current Shunt Amplifiers and Enhanced Protection, Diagnostics, and Monitoring*.

2 Implementing the Watchdog Timer

2.1 Initial Configuration

The default state of the DRV3205-Q1 device upon reset is with the watchdog timer disabled (WD_EN = 0). The watchdog timer must be configured and enabled in the DIAGNOSTIC state, as the registers become locked in the ACTIVE state. Table 1 lists summary of all watchdog-setting bit fields as a reference.

Table 1. Watchdog Bit Field Summary

Bit Field Name	Register	Address	Bits	Descriptions	Type
RT[6:0]	WDT_WIN1_CFG	0x31	[7:1]	Open window: 0.55 ms × (1 + setting)	Config
RW[4:0]	WDT_WIN2_CFG	0x32	[7:3]	Close window: 0.55 ms × (1 + setting)	Config
FDBK[3:0]	WDT_TOKEN_FDBCK	0x33	[7:4]	Adjust token feedback	Config
TOKEN_SEED[3:0]	WDT_TOKEN_FDBCK	0x33	[3:0]	Change token seed	Config
TOKEN[3:0]	WDT_TOKEN	0x34	[6:3]	Token value	Operation
WDT_ANSW[7:0]	WDT_ANSWER	0x35	[7:0]	Answer register	Operation
WD_RST_EN	SFCR1	0x09	[6]	Enable transition to RESET state after WD fail	Config
WD_EN	SFCR1	0x09	[0]	Enable the WDT	Config
WD_FAIL	STAT0	0x0A	[7]	WD_FAIL_CNT ≥ WD_FAIL_MAX	Status
WD_FAIL_CNT	STAT3	0x0C	[2:0]	Number of failed WDT operations	Status
NO_WRST	SFCC1	0x20	[6]	Disable transition to SAFE on WD_FAIL	Config
WDT_ANSW_CNT[1:0]	WDT_STATUS	0x36	[7:6]	Current answer received by the device	Status
TOKEN_ERR	WDT_STATUS	0x36	[5]	Invalid response	Status
SEQ_ERR	WDT_STATUS	0x36	[2]	Correct response in the wrong order	Status
TIME_OUT	WDT_STATUS	0x36	[1]	No response in the required window	Status
TOKEN_EARLY	WDT_STATUS	0x36	[0]	Final response in the first window	Status
WD_FAIL_MAX[2:0]	WD_FAIL_CFG	0x37	[7:5]	Max WD_FAIL_CNT to cause a fault	Config
WD_FAIL_DEFAULT[2:0]	WD_FAIL_CFG	0x37	[4:2]	Value to set WD_FAIL_CNT on transition from DIAG to ACTIVE	Config

To configure the device after reset, use the steps that follow:

1. Verify the device is in the DIAGNOSTIC state (ERR pin transitions from L to H).
2. Set the WDT_WIN1_CFG and WDT_WIN2_CFG registers for the desired open and closed window timings.
3. Set the WD_FAIL_MAX to the max fault tolerance allowed before error handling takes place. The maximum time to detect a fault is equal to Equation 1.

   WD_FAIL_MAX × ([(RT[6:0] + 1) + (RW[4:0] + 1)] × 0.55 ms). (1)

4. Change the token seed and feedback values if a different Q/A configuration is needed.

5. Set the WD_EN bit to 1 to enable the watchdog timer. Use this operation to synchronize the MCU with the internal watchdog clock.
6. Verify that the WD_FAIL_CNT counter is decremented and the following error flags are clear:
   - TIME_OUT
   - TOKEN_ERR
   - SEQ_ERR
   - TOKEN_EARLY
7. Set the desired error handling (listed as follows) in case of watchdog timer fault:
   - WD_RST_EN causes the device to transition to the RESET state in case of a fault.
   - NO_WRST causes the device to not transition out of the ACTIVE state in case of a fault. This error handling has higher priority than WD_RST_EN and is typically used for debugging.
8. Clear the WD_FAIL bit to 0 if it is set to 1.

2.1.1 Disabling the Watchdog

The watchdog timer is disabled by default, and can only be disabled in the DIAGNSOTIC state. When using the DRV3205-Q1 device without the watchdog timer, the WD_FAIL_DEFAULT[2:0] bit must be set to a higher value than the default of 3 to allow full state transitions according to the device data sheet.

When the device is in the DIAGNOSTIC state (ERR pin transitions from L to H), use the steps that follow to set this configuration:

1. Set the WD_EN bit to 0.
2. Enter CSM Mode
3. Set the WD_FAIL_DEFAULT[2:0] bits to 5.
4. Exit CSM Mode.
5. Configure the remaining SPI registers.

2.1.2 Running the Watchdog

2.1.2.1 Watchdog Startup

The watchdog timing window and logic begins when the WD_EN is set to 1 while the device is in the DIAGNOSTIC state. The WD_FAIL_CNT counter value changes depending on the answers received. When the DIAG_EXIT_MASK bit is set, the device transitions from the DIAGNOSTIC state to the ACTIVE state when the WD_FAIL_CNT counter value is below the programmed WD_FAIL_DEFAULT[2:0] value. When the device transitions to the ACTIVE state, the WD_FAIL_CNT counter is reset to the programmed WD_FAIL_DEFAULT[2:0] value.

Figure 8 shows this behavior in the case that the WD_FAIL_DEFAULT[2:0] bit is programmed to the default value, and the DIAG_EXIT_MASK bit is set before the first transaction completes. After the first successful watchdog transaction, the WD_FAIL_CNT counter decrements which satisfies the condition to transition to the ACTIVE state. This transition resets the WD_FAIL_CNT counter to the WD_FAIL_DEFAULT[2:0] value.

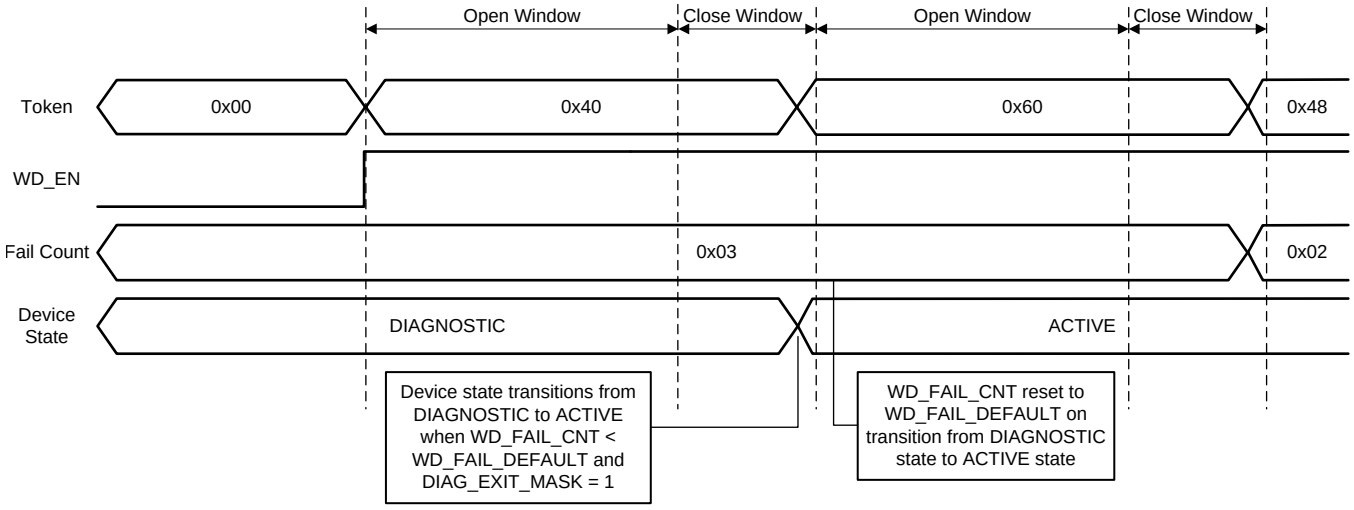

Figure 8. Watchdog Startup

2.1.2.2 MCU Response Calculation

Depending on the level of coverage required by the system, the MCU can be configured to either look up a precalculated response to a generated token, or to use the internal ALU to calculate the response.

For the arithmetic required for the MCU to perform, refer to the *DRV3205-Q1 Three-Phase Automotive Gate Driver With Three Integrated Current Shunt Amplifiers and Enhanced Protection, Diagnostics, and Monitoring*.

If this level of coverage is not required, a simpler lookup table approach can be used. By storing the precalculated answer values, the MCU can provide the answer to the DRV3205-Q1 device simply by using the token value and response number to look up the answer. For more information on the stored watchdog responses, refer to the *DRV3205-Q1 Three-Phase Automotive Gate Driver With Three Integrated Current Shunt Amplifiers and Enhanced Protection, Diagnostics, and Monitoring*

The following pseudo code written in C format illustrates how this can be implemented"

```
static const uint8_t DRV3205_WDT_TABLE[16][4] =
{
    {0xFF,    0x0F,    0xF0,    0x0},
    {0xB0,    0x40,    0xBF,    0x4F},
    {0xE9,    0x19,    0xE6,    0x16},
    {0xA6,    0x56,    0xA9,    0x59},
    {0x75,    0x85,    0x7A,    0x8A},
    {0x3A,    0xCA,    0x35,    0xC5},
    {0x63,    0x93,    0x6C,    0x9C},
    {0x2C,    0xDC,    0x23,    0xD3},
    {0xD2,    0x22,    0xDD,    0x2D},
    {0x9D,    0x6D,    0x92,    0x62},
    {0xC4,    0x34,    0xCB,    0x3B},
    {0x8B,    0x7B,    0x84,    0x74},
    {0x58,    0xA8,    0x57,    0xA7},
    {0x17,    0xE7,    0x18,    0xE8},
    {0x4E,    0xBE,    0x41,    0xB1},
    {0x01,    0xF1,    0x0E,    0xFE}
};
```

By storing the table in memory in this configuration, the answer can be accessed as follows:

```
DRV3205_WDT_TABLE[Token][ReponseNumber]
```

2.1.2.3 Response Scheduling

To stay synchronized with the watchdog timer, the MCU must schedule responses in the proper timing windows. This scheduling is typically implemented using a timer interrupt or inside a critical looping code section which has a strict timing requirement.

The response schedule can be done in multiple ways, as long as it satisfies the proper window timings. The recommended method of response scheduling is to use a method with a fixed period, *T*, in the MCU with a window timing set as shown in Equation 2.

$$T = [(RT[6:0]+1) + ((RW[4:0]+1) \times 0.5)] \times 0.55\text{ms} \quad (2)$$

At each period, T, the MCU writes the fourth response in the middle of the close window which triggers the end of the watchdog sequence and the beginning of a new open window. Immediately after the fourth response, the token is queried, the answers are calculated, and the first three responses are sent. The WDT_WIN2_CFG time should be set to the allowable variation in the MCU loop timing.

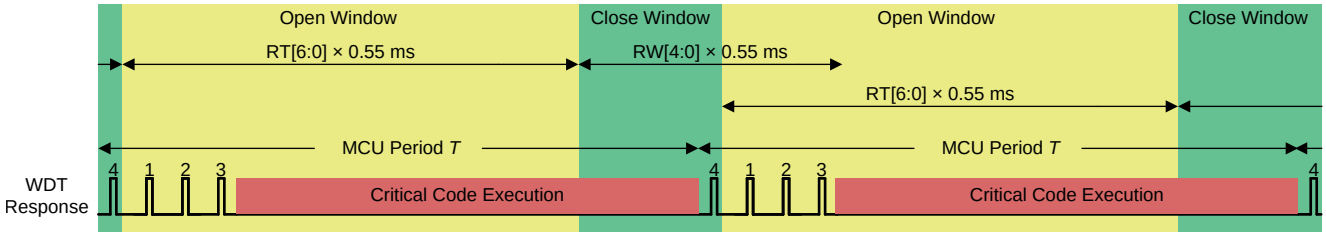

Figure 9. Watchdog Response Scheduling

2.1.2.4 Resynchronizing the Watchdog Timer

In the case that the MCU no longer captures the correct timing, the MCU must resynchronize with the internal timing of the DRV3205-Q1 device. The device provides functionality to help with resynchonizing. The functionality is listed as follows:

- Stop existing timing and response generation in the MCU.
- Clear the TIME_OUT flag to 0 by writing a dummy value to the WDT_WINx_CFG register.
- Poll the TIME_OUT flag for a transition from 0 to 1.
- Reset the MCU timer to this transition.
- Run the watchdog routine normally and verify that the WD_FAIL_CNT counter value is decremented.
- Clear the WD_FAIL flag.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers’ products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers’ products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265