Application Report
SLLA430–October 2018

TLINx441x Watchdog Configuration Guide

Eric Hackett

ABSTRACT

This application report presents the configuration, function, and use of the watchdog feature of the TLINx441x. The TLINx441x device and the general watchdog idea is explained, as well as the watchdog function on the TLINx441x along with cases and examples to further explain how it is used.

Contents

List of Figures

List of Tables

Trademarks

1 Introduction

The automotive industry continues to move in the direction of more intelligent technology in vehicles. With this trend comes the need for space-saving circuit boards and components, and thus integration of multiple functions into one packaged integrated circuit. In come CAN and LIN system basis chips (SBCs). These are CAN and LIN transceivers with low-dropout (LDO) supplies, watchdog, high-side switches, etc. in one device to be the basis for Electronic Control Units (ECU). SBCs allow the system designer to save board space by combining all of the necessary functions of an ECU into one chip, instead of several devices containing multiple passive components of their own.

The TLINx441x devices are one of TI's general purpose LIN SBC families, integrating a LIN transceiver, SPI interface, watchdog, and an LDO. In this application report, the TLINx441x watchdog function, configuration, and use is explained.

2 General Description of Watchdog

A watchdog function is common in the electronics industry, and is a timer meant to verify that the controller of a device is working correctly. This is done by the controller periodically updating or resetting a timer in the controlled device based upon a specific timing sequence. If this is not executed correctly, an interrupt or timeout signal is generated, and depending on the system design, some kind of corrective action take place. The idea is that in systems where human intervention is not easy, possible, or cannot take place quick enough, the system will autonomously fix any issues with its main controller by forcing a reset on it.

3 Timeout Watchdog

The timeout watchdog is the simplest implementation of the watchdog function. The general concept goes as follows: a timer is started as soon as the first watchdog trigger is sent, and that same trigger must be sent within the configured time limit, or an error is asserted. In most cases, an error counter is incremented when the timer runs out before a trigger occurs, and when this counter reaches a certain value, a fault (interrupt) is sent to the controller to initiate the diagnostic and/or fixing mechanisms.

4 Window Watchdog

The window watchdog is similar to the timeout watchdog, except the timer is split into an opened and closed window. As the naming implies, the open window is when the watchdog trigger can be sent and accepted, and the closed window is the time range when the watchdog trigger cannot be sent. If the trigger is sent during the closed window, the error counter is incremented. The idea behind this version of a watchdog timer is that the microcontroller has to be a bit more precise, and avoids a loop lock situation, where the microcontroller can still service a timeout watchdog while being stuck in a never-ending loop.

5 Explanation of Watchdog Specific to TLINx441x

The TLINx441x has two operational modes in which the watchdog function is implemented: PIN mode and SPI mode. These modes are configurable by the state of pin 9 at power up, and are not able to be changed unless there is a power cycle of the IC. When pin 9 is floating or held to a logic high, TLINx441x is in SPI mode; when pin 9 is grounded, it starts up in PIN mode.

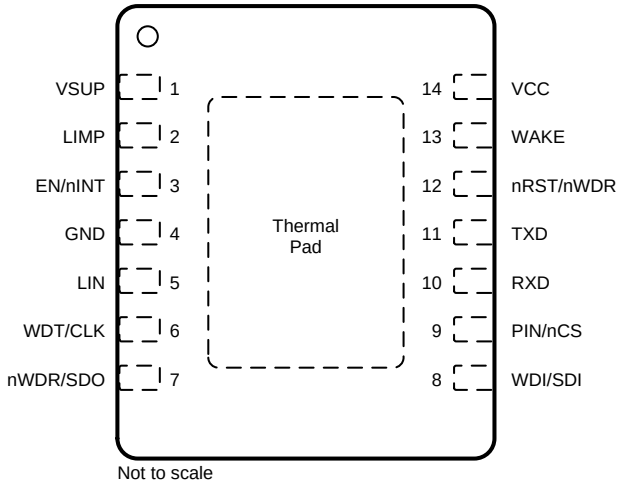

Figure 1. TLINx441x Pinout

Depending on each mode, pins 3, 6, 7, 8, 9, and 12 will take on different roles. By the names of the modes themselves, it can be deduced that SPI mode is the mode with the digital SPI interface, and PIN mode is the more analog mode, where there are less configuration options for the Watchdog, LIMP output, and other features of the device. Both modes have the Window watchdog feature, and the SPI mode has an additional Timeout watchdog feature.

On the TLINx441x, the window is a 50%/50% split of the configured timer in both PIN and SPI mode; that is, whatever time length is configured, 50% of that value will be open and 50% is closed. The timer is based on an internal oscillator with a ±10% accuracy range. This variance needs to be taken into account when determining the trigger point and absolute safe zone. For instance, if 40 ms is chosen as the timer, then your open window and closed windows would be 20 ms each. Considering the variance, your total window would be 36 ms (t_{WINDOW} min) or 44 ms (t_{WINDOW} max), and your open/closed window time would be 18 ms (t_{WDOUT} min) or 22 ms (t_{WDOUT} max). In the “worst” case, your open window would be 14 ms, considering 22 ms closed window and 36 ms total window time. Figure 2 illustrates this concept.

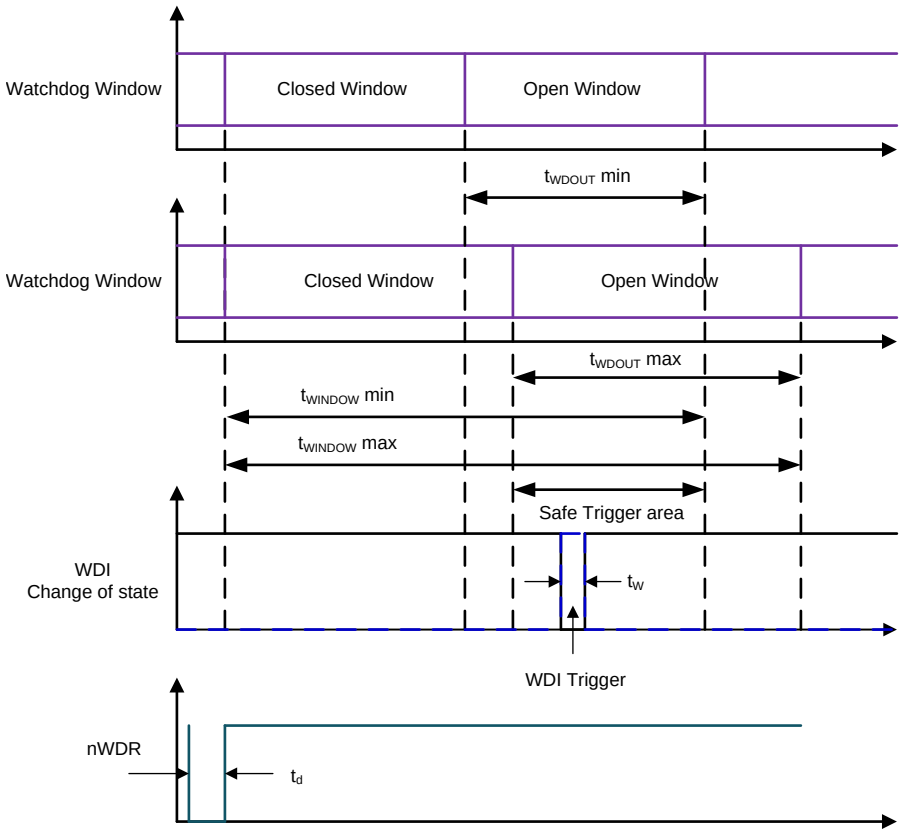

Figure 2. Watchdog Window Variance and Absolute Safe Trigger Zone

Because this is the variance of an oscillator, and not a random-phase generator, this real-time change in variance is not an actual case that can happen. However, the actual variance of the oscillator won’t be known by the user, so instead of assuming +10%, -10%, or something in between, it is best to trigger in the “safe zone” so that the correct trigger is ensured.

6 PIN Mode Watchdog

While in PIN mode, the window length is set by the state of the WDT pin. The different watchdog Window time ranges are show in Table 1.

Table 1. Watchdog Timer Values Based on WDT State in PIN Mode

	Minimum	Typical	Maximum	Units
WDT = GND	36	40	44	ms
WDT = Floating	5400	6000	6600	ms
WDT = VCC	540	600	660	ms

As stated before, in each case the open window is 50% of the configured time, and the closed window is 50% of the configured time. So for the WDT = GND case, the window is open for 20 ms and closed for 20 ms. It is recommended that the trigger is sent in the absolute safe zone, shown in Figure 2.

The watchdog timer starts once the WDI pin receives a pulse the first time in Normal mode. Once the timer starts, the window is closed for the configured time, and then opens for the allotted time to allow for triggering. To trigger the watchdog, a pulse must be sent to the WDI pin during the open window time. The digital logic of the TLIN device recognizes the edge transition of the pulse as the event to service the watchdog, and this is considered a valid service of the watchdog. The processor then must continue to update the WDI pin within the open window to avoid causing a watchdog error and incrementing the error counter.

If the pulse is sent during the closed window time, or not sent at all, the nWDR pin will pulse low for 4 ms, informing the processor that a watchdog error has occurred, and the LIMP pin goes high. The error counter decrements every time the watchdog is correctly serviced and the LIMP pin does not go back low until the error counter reaches 0 again.

7 SPI Mode Watchdog

SPI mode has two watchdog choices: Window and Timeout. This is chosen by writing to bit[6] in register ‘h13. By default, the bit value is 0 and this selects the Window watchdog; to choose the Timeout watchdog, this bit must be changed to 1. Configuration of all Watchdog settings can happen in any mode that SPI is active.

7.1 Window Watchdog

The Window watchdog in SPI mode functions the same as the Window watchdog in PIN mode, but there are more configuration options. The timer is set by setting bits [5:4] in register ‘h13, and bits [7:5] in register ‘h14. Table 2 shows the resulting timer values based on the combinations. It is important to note that these timer values also have a 10% variance, so there is a minimum and maximum time associated with the windows and an absolute safe trigger zone.

Table 2. Watchdog Timer Configuration Register Values in SPI mode

WD_Timer (ms)	reg13[5:4]				
Reg14[7:5]	00	01	10	11	Units
000	4	8	12	16	ms
001	32	64	96	128	
010	128	256	384	512	
011	256	384	512	768	
100	512	1024	1536	2048	
101	2048	4096	6144	8192	
110	10240	20240	RSVD	RSVD	

In order to service the watchdog in SPI mode, bits [7:0] in register ‘h15 must be written to ‘hFF. This must be done within the open window segment of the total window time; otherwise, a watchdog error is asserted. When an error is asserted, by default (same as PIN mode), nWDR, now pin 12, will pulse low and the LIMP pin goes high. However, this can be configured to respond differently in SPI mode. The condition for LIMP to reset to low can also be configured.

It is important to note that the function of pin 12 is configured through the SPI registers. Register ‘h10, bits 3 and 2 are the configuration bits for this function. By default, the pin is configured as nRST, if the user desires for the pin to have the nWDR function, there are two options, shown in Table 3.

Table 3. Pin 12 Configuration Bits

Bits [3:2] register 'h10	Pin 12 configuration
00	nRST (Default)
01	nWDR
10	Both nRST for UVcc and nWDR for watchdog failure event

In order for the watchdog error to assert an external indicator, the interrupt mask for the watchdog error must be disabled. By default, all interrupts are masked on power up, and these masks are located in register ‘h0E. To disable the interrupt mask, a 0 must be written to the bit corresponding to the interrupt that needs to be seen. For the Watchdog error interrupt, this is bit 3, so to disable the Watchdog interrupt mask, ‘h08 must be written to register ‘h0E.

When the error is thrown, the WDERR Interrupt flag is asserted in register ‘h0C. This triggers the external indicator for which the device is configured. Once this flag is asserted, it must be cleared in order for the external indicator to work again. That is, if the Watchdog error occurs, then the Watchdog is restarted and errors out again without the flag being cleared from the first error, the external indicator is not executed. To clear the WDERR Interrupt flag, a 1 must be written to bit 3 in register ‘h0C, or ‘h08 must be written to register ‘h0C.

Bits [4:3] in register ‘h0B configure how many watchdog correct triggers will put the LIMP pin back in the low state.

Table 4. LIMP Reset Condition Configuration Register Values

Bits [3:2] register ‘h13	Watchdog Error Counter Event Trigger
00	On the third successful input trigger the error counter receives
01	First correst input trigger
10	SPI write 1 to bit [1] in register 'h0B

Bits [3:2] in register ‘h13 can be set to configure how many watchdog errors triggers a watchdog event. Table 5 shows these configurations.

Table 5. Error Counter Value to Trigger Watchdog Event Register Values

Bits [1:0] register ‘h13	Watchdog Event Action Trigger
00	Immediate trigger on each watchdog event
01	Triggers on fifth error event
10	Triggers on ninth error event

Bits [1:0] in register ‘h13 can be set to configure what action occurs when a watchdog event is triggered by incorrect servicing. If the nINT configuration is chosen, this always responds on the first Watchdog error event, regardless of the watchdog event action trigger configuration.

Table 6. Watchdog Event Action Register Values

Bits [1:0] register ‘h13	Watchdog Event Action
00	nINT will be pulled low
01	Vcc will be turned off for 100 ms and turned back on
10	nWDR will be toggled high > low > high

7.2 Timeout Watchdog

The timeout watchdog on the TLINx441x is selected by writing a 1 to bit [6] in register ‘h13. The timeout watchdog has the same watchdog event and timing configurations as the window watchdog in SPI mode, the difference is how it is serviced and what triggers an error.

In Window mode, the timer value is chosen and the window in which the watchdog can be serviced is set to be 50% of that value. In Timeout mode, the timer value is chosen and the watchdog can be serviced at any time until the timer duration runs out.

Once bits [7:0] in register ‘h15 are written to ‘hFF for the first time, the watchdog timer will start. When the watchdog is correctly serviced, the timer resets the moment it was correctly serviced; this must be considered for correctly timing the next watchdog service. This is illustrated in Figure 3.

The processor then must continue to write to these before the timer runs out each time, or the watchdog error is asserted. Depending on the error configuration chosen, the counter will be incremented or a watchdog event occurs if the timer is allowed to expire without a watchdog service.

It is recommended that if a timer value of less than 64 ms is chosen, the timeout watchdog should be used. The window variance (due to the internal oscillator variance) could cause the open window time to be too small, making it difficult for the processor to reliably service even if it is working correctly. However, it is not required that timeout watchdog is used if a faster timer value is chosen, provided that the processor can handle the service speed.

8 Case Examples of Watchdog

8.1 Window Watchdog in PIN Mode

- To put the device in PIN mode, pin 9 is tied to GND on power up.
- 600 ms timer is chosen by connecting the WDT pin (pin 6), to VCC (pin 14).

With the ±10% oscillator, this means that the minimum and maximum full timer value are 540 ms and 660 ms, and the minimum and maximum open/closed window times are 270 ms and 330 ms.

Figure 3 shows an oscilloscope screenshot of the WDT, WDI, and nWDR pins of the TLIN14415 device while in the configuration described in this section.

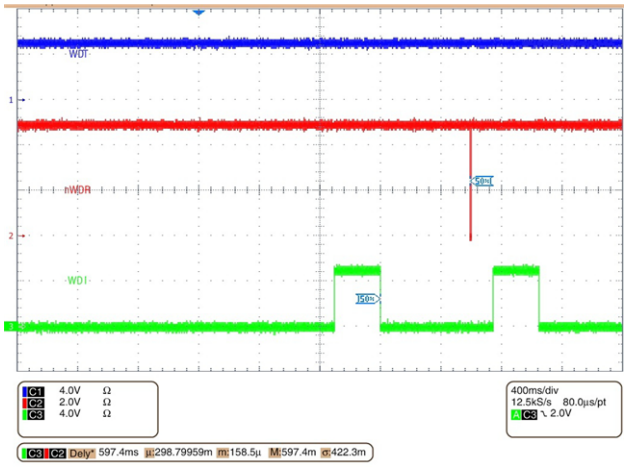

Figure 3. TLIN14415-Q1 PIN Mode Watchdog Example

In this screenshot, WDT is held at 5 V to configure the window timer to 600 ms. The green waveform shows the WDI pin being toggled to begin the watchdog timer, and the red waveform shows the nWDR responding to the first watchdog failure by pulsing low. The WDI is exercised with the incorrect timing intentionally to illustrate that the window timer is working correctly according to the WDT pin setting. The nWDR pin toggles 597.4 ms after the initial WDI service, as shown by the delay measurement on the oscilloscope. If the space between the pulses on the WDI pin were shortened, by 150 ms or more, the watchdog would be correctly serviced, and the nWDR pin would stay high.

8.2 Window Watchdog in SPI mode

To put the device into SPI mode, the state of pin 9 must be floating or logic high at power up. Floating will put the IO voltage to 3.3 V, and a logic high will put the IO voltage to 5 V

For this example the device will be placed in SPI mode with the following configuration options:

- Window watchdog selected
- 1536 ms timer
- Watchdog event will trigger on the 5th error event
- nWDR will be toggled high -> low -> high
- LIMP resets on SPI write 1 to bit[1] in register 'h0B

These options will take three separate SPI writes to accomplish. For the SPI protocol implementation on the TLINx441x, refer to the datasheet. The three SPI writes will be:

- Write 'h26 to register 'h13: 'b0010 0111 0010 0110 or 'h2725
  - Watchdog is enabled
  - Window mode is chosen
  - Watchdog timer prescaler bit setting of 'b10 is selected
  - Amount of errors to trigger a watchdog event is set to 5
  - Watchdog event action is set to nWDR toggling high -> low -> high
- Write 'h80 to register 'h14: 'b0010 1001 1000 0000 or 'h2960
  - Watchdog timer multiplier of 100 is selected
- Write 'hB0 to register 'h0B: 'b0001 0111 1011 0000 or 'h17B0
  - LIMP is enabled
  - LIMP is configured to reset by writing 1 to 'h0B[2]
- Write 'h02 to register 'h10: 'b0010 0001 0000 0010
  - Pin 12 configuration is set to nWDR

Figure 4 is an oscilloscope screenshot of a TLIN14415 device in this configuration using the watchdog.

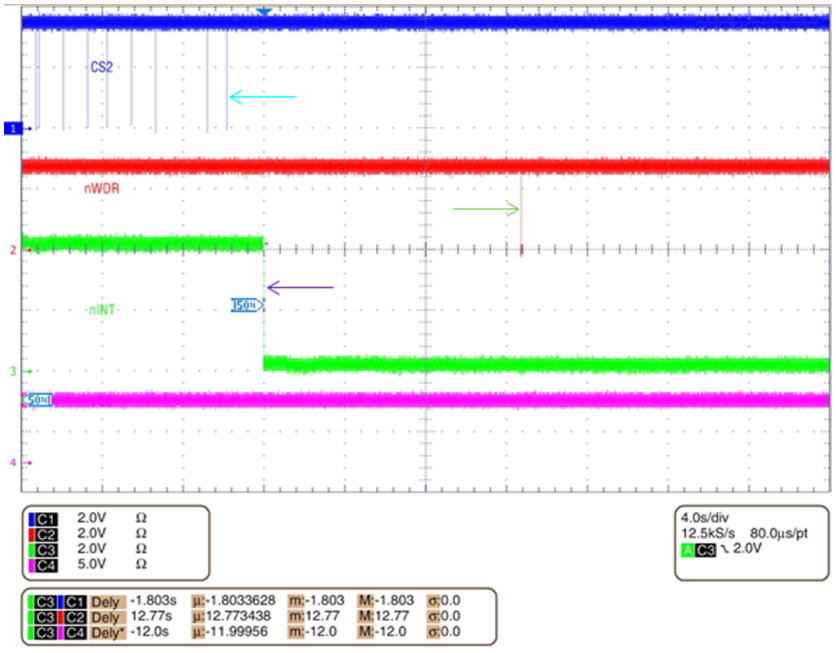

Figure 4. TLIN14415-Q1 Window Watchdog Service and Failure

Figure 4 shows the TLIN14415-Q1 device watchdog service and then failure in the configuration described in this section. The nCS, nWDR, nINT, and VCC waveforms are shown, and it is important to note that the time scale is large to accommodate for the long duration window chosen, so the pulses of SPI writes and failure indicators look like one-pixel wide lines. The nCS waveform (blue) shows several of these lines, which indicate SPI writes and reads. The SPI writes are to service the watchdog (writing 'hFF to register 'h15), and the SPI reads are to monitor the error count. For this example, these SPI writes are decrementing the error counter to 0 from the default value of 4. Once the writes, and thus the servicing of the watchdog, the timer expires and timeout is hit; this is indicated by nINT waveform (green).

As previously mentioned, no matter which failure event is chosen to be indicated, nINT will always indicate the first timeout of the Watchdog, one full window duration. The purple arrow points to the point when the timer runs out, and the delay measurement at the bottom of the screenshot shows the delay between the last service (indicated by the light blue arrow) and the nINT voltage dropping to 0, around 1.8 s. This indicates one window time. Then the delay between nINT dropping and nWDR (red waveform) dropping is shown by the delay between channel 2 and 3, 12.77 s. From nINT toggling, to nWDR toggling, this is equal to 8 time windows, since nINT toggles after the first window has expired. Dividing 12.77 s by 8 gives 1.59625 s, proving the configured window time Since nWDR was configured to be the watchdog failure event indicator, this is the event that will happen 9 windows after the last service.

8.3 Timeout Watchdog in SPI Mode

Again, the device will be placed into SPI mode. And the following configuration options will be in place:

- Timeout watchdog selected
- 128 ms timer
- Watchdog event will trigger on the 9th error event
- VCC will be turned off for 100 ms then turned back on
- The third successful watchdog trigger will reset LIMP

These options will also take three separate SPI writes to accomplish.

- Write 'h75 to register 'h13: 'b0010 0111 0111 0101 or 'h2775
  - Watchdog is enabled
  - Timeout mode is chosen
  - Watchdog timer prescaler bit setting of 'b11 is selected
  - Amount of errors to trigger a watchdog event is set to 5
  - Watchdog event action is set to VCC turning off then back on
- Write 'h20 to register 'h14: 'b0010 1001 0010 0000 or 'h2920
  - Watchdog timer multiplier of 001 is selected
- Write 'h80 to register 'h0B: 'b0001 0111 1000 0000 or 'h1780
  - LIMP is enabled
  - LIMP is reset on the third successful watchdog trigger

Figure 5 is a screenshot of the TLIN14415-Q1 device in this configuration, using the watchdog.

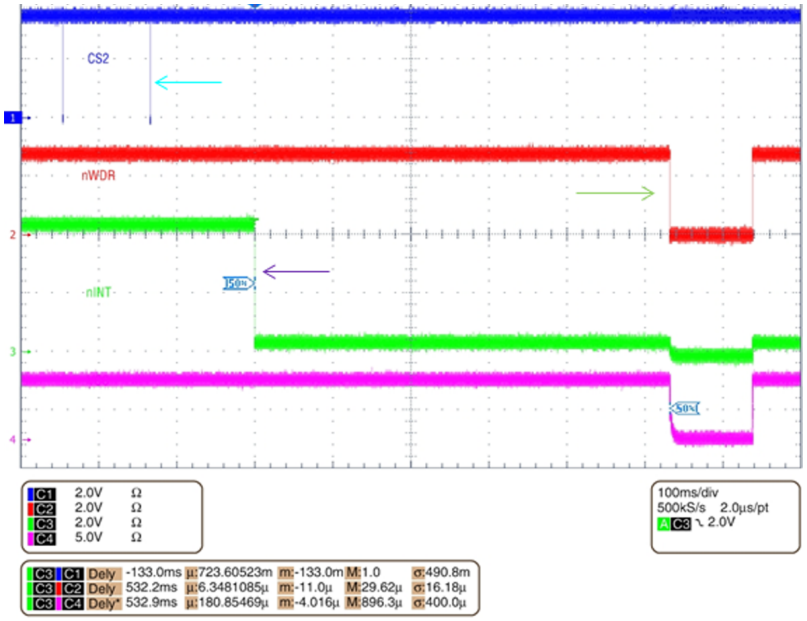

Figure 5. TLIN14415-Q1 Timeout Watchdog Service and Failure

Figure 5 shows the TLIN14415-Q1 device watchdog service and then failure in the configuration described in this section. The screenshot is in the same format as Figure 4, but with a different configuration for the watchdog. Again, the nCS is showing the SPI writes and reads to decrement the counter and service the Watchdog. The light blue arrow points at the last service of the watchdog, and the purple arrow indicates the first point of failure, shown by nINT. The timing here is 133 ms, and then the green arrow shows where the nWDR and VCC toggle 4 windows later, as the 5th watchdog event was configured for this example. The delay between these two points is 532 ms, adding these together, the 5 windows take about 660 ms. Since 128 ms window was chosen, this makes sense, as 133 ms is within the 10% clock variance, and 5 x 128 = 640, with the actual time being 660 ms.

9 Summary

The watchdog configuration and service can be a daunting task for any device. Hopefully this report sheds some light on the function of the watchdog module on the TLIN14415-Q1 LIN SBC, and will allow the development to be as smooth and straightforward as possible.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

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