How to Use the FemtoClock 3 Wireless (FC3W) Time-of-Day Counter

Introduction

This document describes how to configure the Time-of-Day Counter for FemtoClock™ 3 Wireless (FC3W) devices.

Contents

1. Overview

FemtoClock 3 Wireless (FC3W) devices have an internal Time-of-Day Counter (TOD) for accurate and precise time stamping. The TOD can be configured to reference the output clock from any of the fractional output dividers (FOD). The TOD reference clock requires a minimum frequency of 500MHz and maximum 625MHz coming from the chosen FOD.

There are two internal counters labeled as the Time Clock (sync_counter) and Time Sync (sub_sync_counter). The Time Clock speed is configured using the **time_clock_gen_count (0xA00[5:0])** register set, which represents the number of FOD input clock cycles (minus one) in a Time Clock period. The Time Sync speed is configured using the **sub_sync_count (0xA04[30:0]**) register set, which represents the number of Time Clock cycles (minus one) in a Time Sync period.

The **time_clock_gen_count** has a maximum integer value of 64, allowing a range of frequencies between 7.8125MHz – FOD input frequency. The sub_sync_count register set has a maximum value of (2^31) – 1, allowing the counter to have frequencies less than 1Hz.

There is a 1µs update rate for both count values. The counts increment internally and can be read at the 1µs interval.

2. RICBox GUI Setup

To configure the TOD in RICBox, open the FemtoClock 3 Wireless user interface and go to the block diagram view.

1. Click on the *TOD and Synthesis* block.

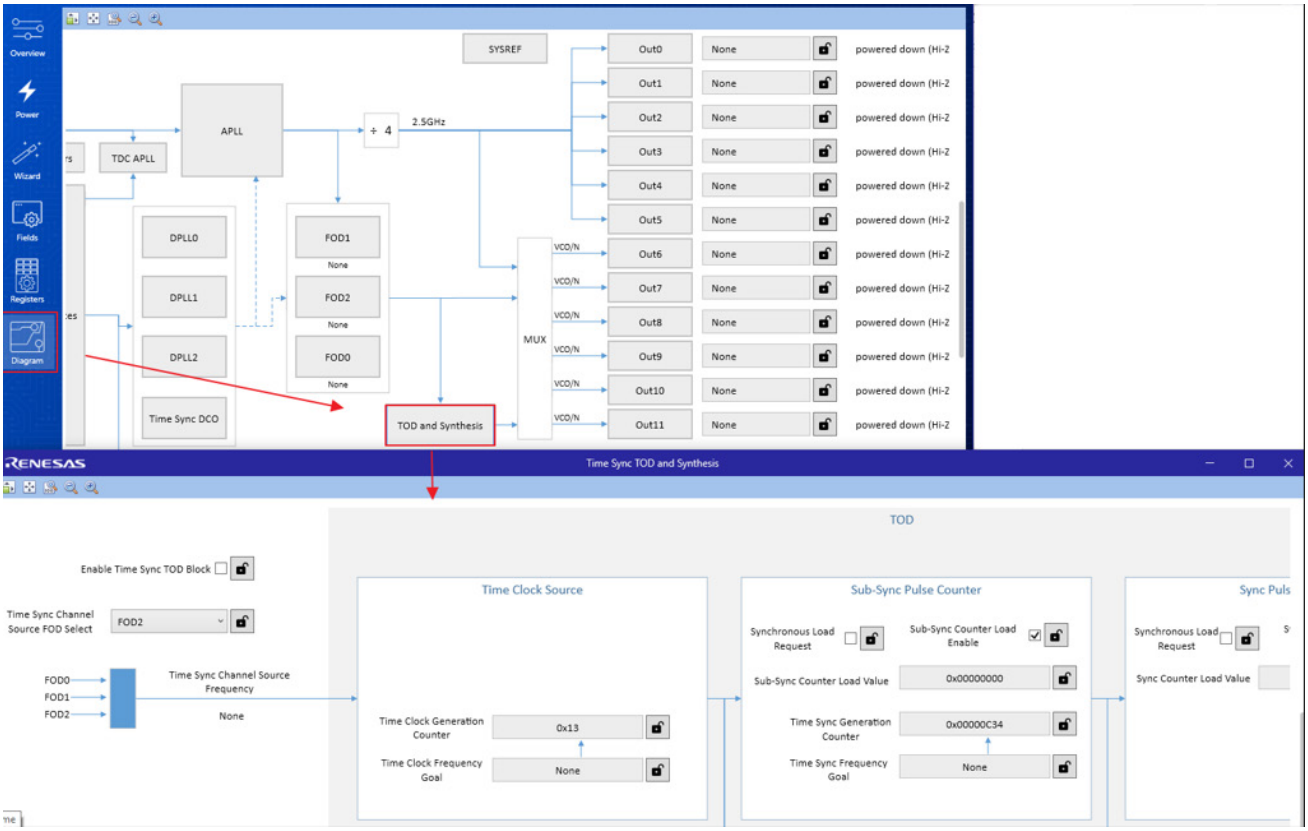

Figure 1. Open TOD and Synthesis Block

2. Click the Enable Time Sync TOD Block checkbox.

Note: This enables the TOD to start counting if configured during run-time.

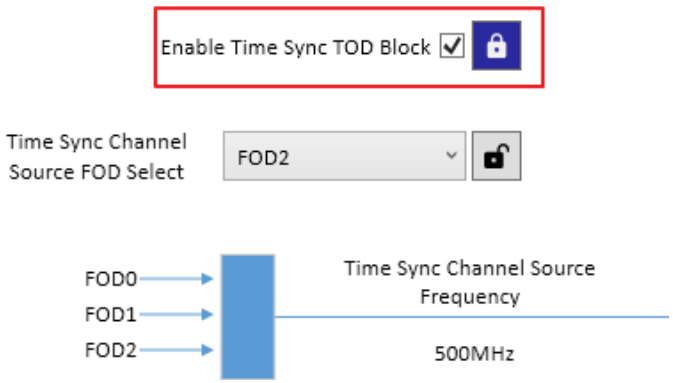

Figure 2. Enable Time Sync TOD Block

3. Select the FOD input reference from the dropdown menu in the upper left corner.

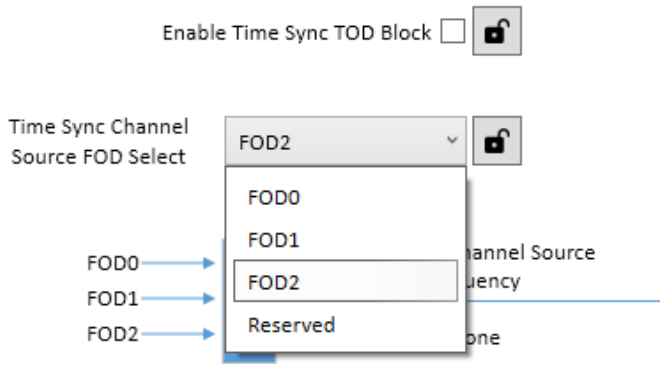

Figure 3. Select TOD Reference Clock

4. Configure the *Time Clock Frequency Goal* and *Time Sync Frequency Goal*. RICBox automatically fills in the value for the **time_clock_gen_count** and **sub_sync_count**.

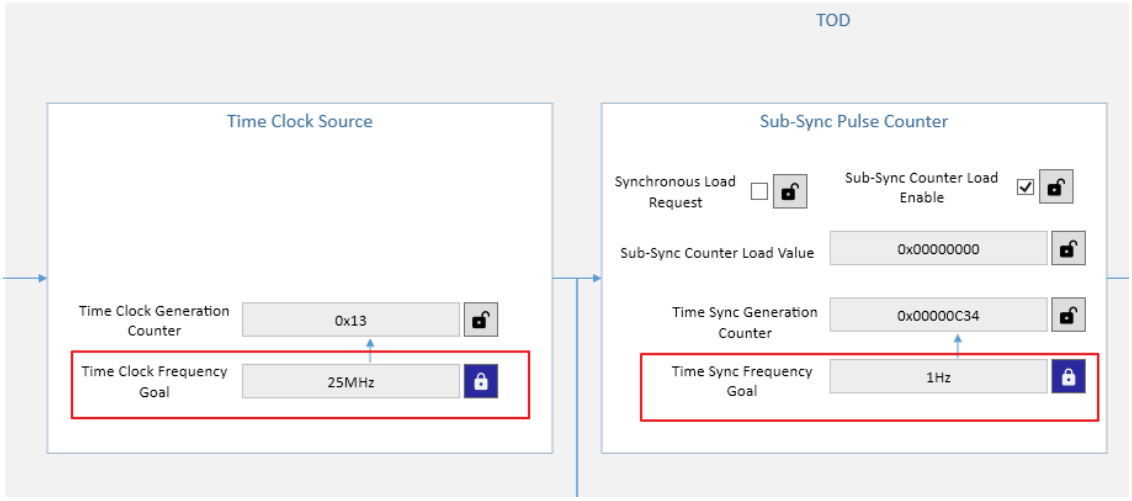

Figure 4. Set Time Clock and Time Sync Frequencies

3. Run-time

To start or stop the counters, write to the **TOP.TIME_SYNC_TOD.TOD_ENABLE_CTRL.enable** bit (0xA0A[0]). This is done in the RICBox by connecting to a device with the connection interface and using the command line interface (CLI).

1. After a connection is established, click on the *Tools* menu and select *CLI*.

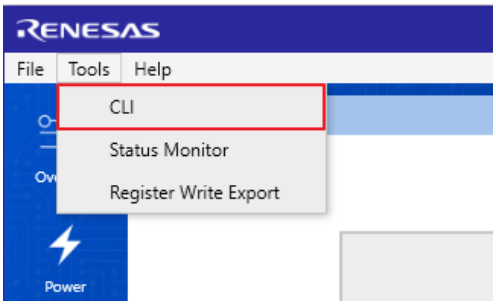

Figure 5. Open RICBox CLI

2. Type "**write TOP.TIME_SYNC_TOD.TOD_ENABLE_CTRL.enable 1**" to start the count.
3. Use "**write TOP.TIME_SYNC_TOD.TOD_ENABLE_CTRL.enable 0**" to stop the count.

Figure 6. Enable the TOD Count

3.1 Reading the Count with Serial Connection

Since the counters are updated every 1µs, it is possible the count could get updated while it is also being read. Issue a read to the **tod_counter_rd_req (0xA5F[0])** to temporarily stop counter updates, ensuring the updates are blocked and the data can be read atomically. This also ensures that the Time Clock and Time Sync values are consistent with one another.

- The Time Clock count is stored in **sync_counter_value (0xA60[82:31])**.
- The Time Sync count is stored in **sub_sync_counter_value (0xA60[30:0])**.

1. In RICBox CLI, type "**read TOP.TIME_SYNC_TOD.TOD_COUNTER_READ_REQ.tod_counter_rd_req**".

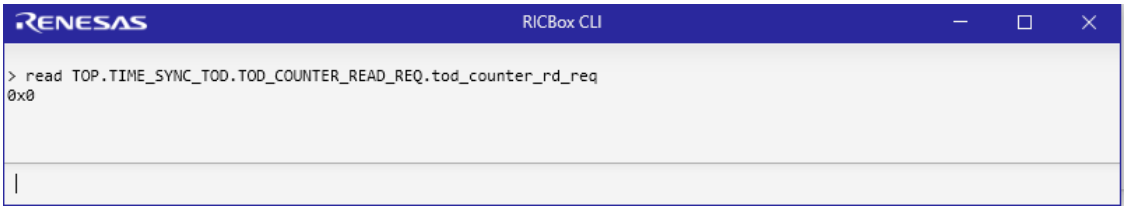

Figure 7. TOD Read Request

2. Then type "**read TOP.TIME_SYNC_TOD.TOD_COUNTER_STS.sync_counter_value**" to read the Time Clock count.

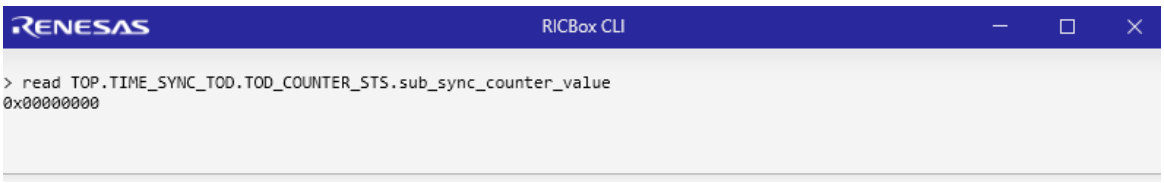

Figure 8. Read Time Clock Count

3. Type “**read TOP.TIME_SYNC_TOD.TOD_COUNTER_STS.sub_sync_counter_value**” to read the Time Sync count.

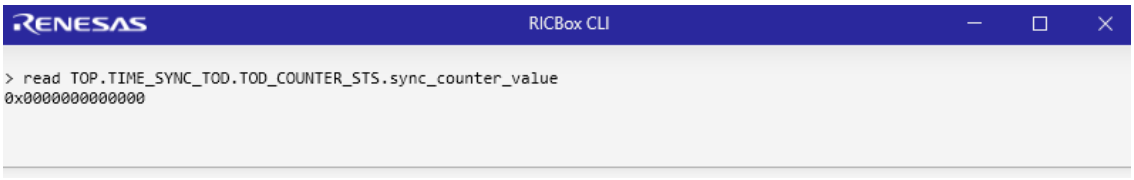

Figure 9. Read Time Sync Count

3.2 Updating the Counters Synchronously

The Time Clock and Time Sync counters can be updated synchronously. A new Time Clock counter value can be stored in the **sync_counter_load_val (0xA10[82:31])** register set.

1. Issue a write command in the RICBox CLI:

“**write TOP.TIME_SYNC_TOD.TOD_SYNC_LOAD_VAL_CTRL.sync_counter_load_val <NEW VALUE>**”

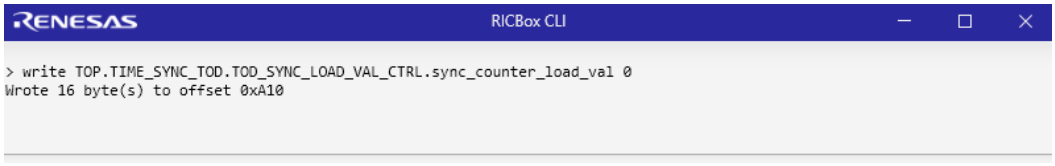

Figure 10. Set Time Clock Counter Value Buffer

2. Enable a synchronous load to the Time Clock with the **sync_load_enable bit (0xA20[1])**.

“**write TOP.TIME_SYNC_TOD.TOD_SYNC_LOAD_EN_CTRL.sync_load_enable 1**”

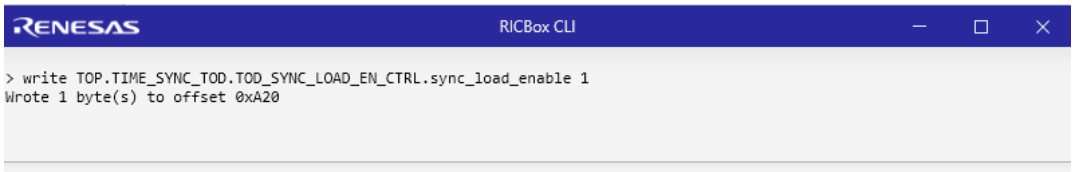

Figure 11. Enable Time Clock Synchronous Load

3. Store the new Time Sync counter value in the **sub_sync_counter_load_val (0xA10[30:0])** register set.

“**write TOP.TIME_SYNC_TOD.TOD_SYNC_LOAD_VAL_CTRL.sub_sync_counter_load_val <NEW VALUE>**”

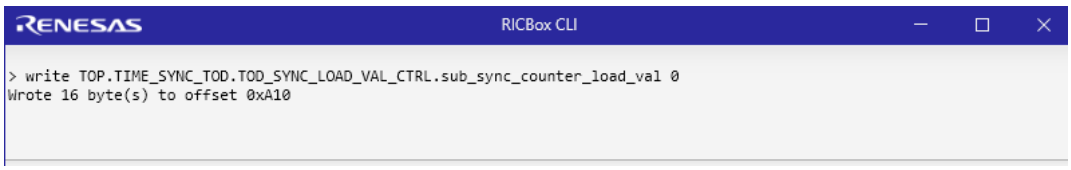

Figure 12. Set Time Sync Counter Value Buffer

4. Enable a synchronous load to the Time Sync with **sub_sync_load_enable bit (0xA20[0])**.

“**write TOP.TIME_SYNC_TOD.TOD_SYNC_LOAD_EN_CTRL.sub_sync_load_enable 1**”

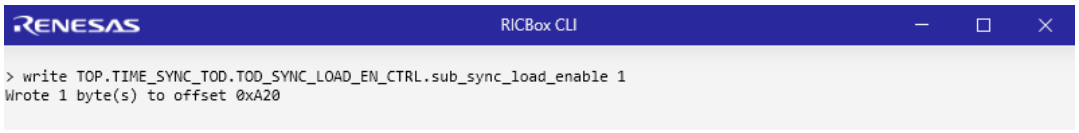

Figure 13. Enable Time Sync Synchronous Load

5. Issue a synchronous load request by writing a 1 to the **sync_load_req** bit **(0xA21[0])**. This will wait for the next Time Sync edge and synchronously load the value to both the Time Clock and Time Sync counters.

“**write TOP.TIME_SYNC_TOD.TOD_SYNC_LOAD_REQ_CTRL.sync_load_req 1**”

Figure 14. Issue Synchronous Load Request

3.3 Updating the Counters Asynchronously

The Time Sync can be asynchronously updated with the value stored in **async_adjust_val (0xA0C[12:0]**). Write a 1 to the **async_adjust_req** bit **(0xA0E[0])** to update the Time Sync counter as soon as it is possible to do so.

“**write TOP.TIME_SYNC_TOD.TOD_ASYNC_ADJUST_VAL_CTRL.async_adjust_val <NEW VALUE>**”

“**write TOP.TIME_SYNC_TOD.TOD_ASYNC_ADJUST_REQ_CTRL.async_adjust_req 1**”

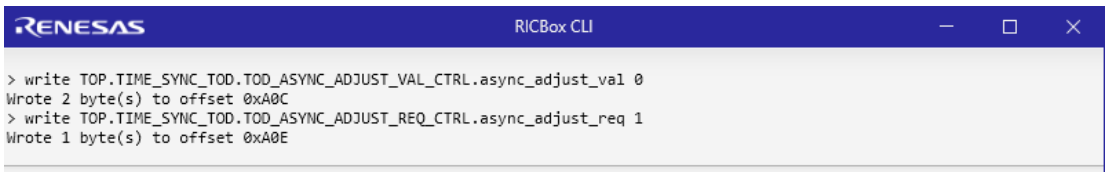

Figure 15. Asynchronous Time Sync Adjustment

4. Outputting the Count Signals

The counts can be outputted as clock signals from either the GPIO or output pins. The output pins can divide the frequency down further if needed, allow flexibility in output signal type, and have configurable Time Sync clock pulse widths. The GPIOs only output 1.8V CMOS signals with unchangeable pulse width.

4.1 Output From GPIO

1. Enable the time sync TDC.

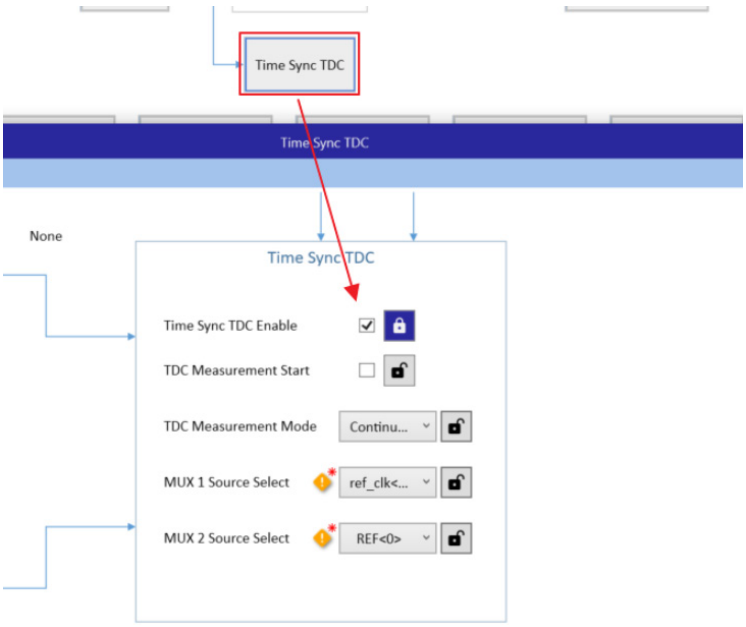

Figure 16. Enable Time Sync TDC

2. Enable TOD clocks to the TDC.

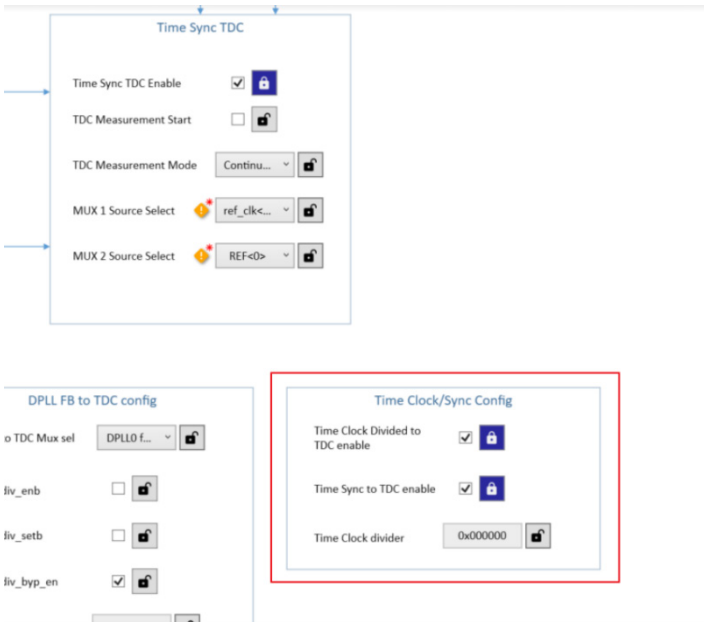

Figure 17. Allow TOD Clocks as Input to Time Sync TDC

3. Set the TDC references to the TOD counters.

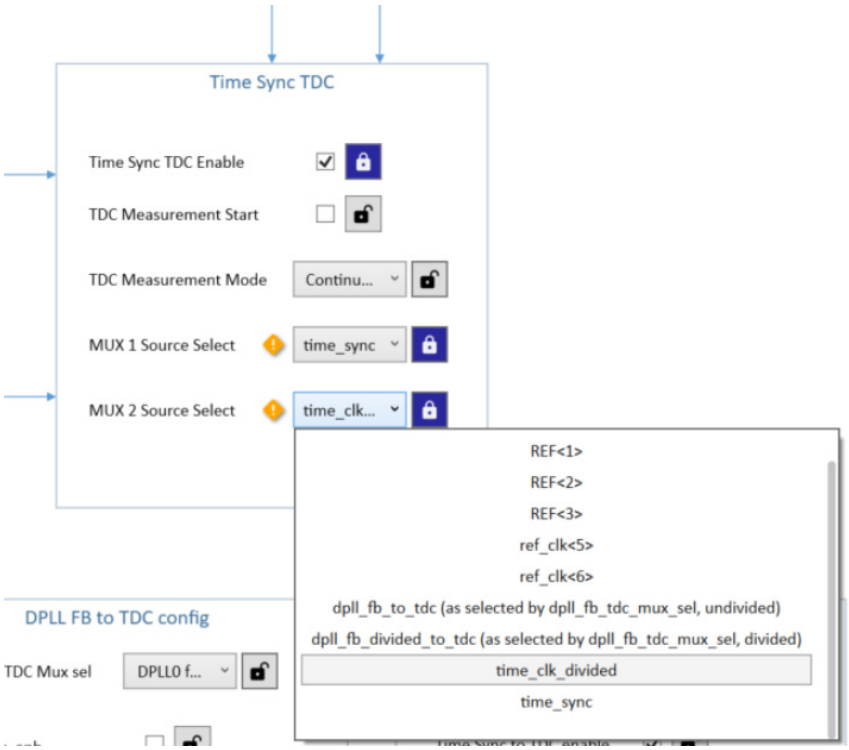

Figure 18. Select TDC Input Clocks

4. Set GPIO function to either **tdc_clk[6]** or **tdc_clk[7]** depending on which clock needs to be outputted from the GPIO. The **tdc_clk[6]** corresponds to the mux source 1 and **tdc_clk[7]** is mux source 2 in the TDC block.

Figure 19. Set GPIO Output as TDC Reference

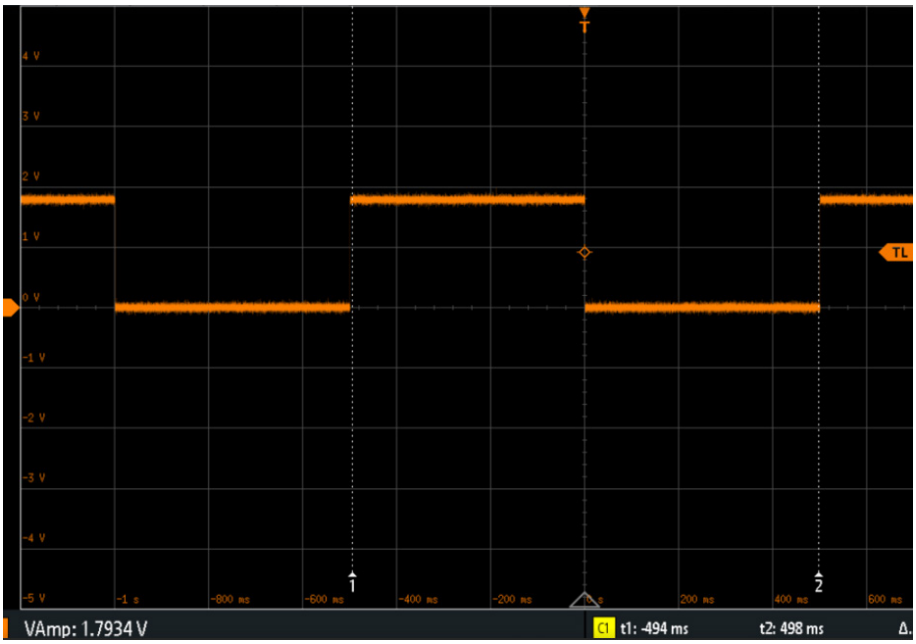

Figure 20. 1Hz Count GPIO Output

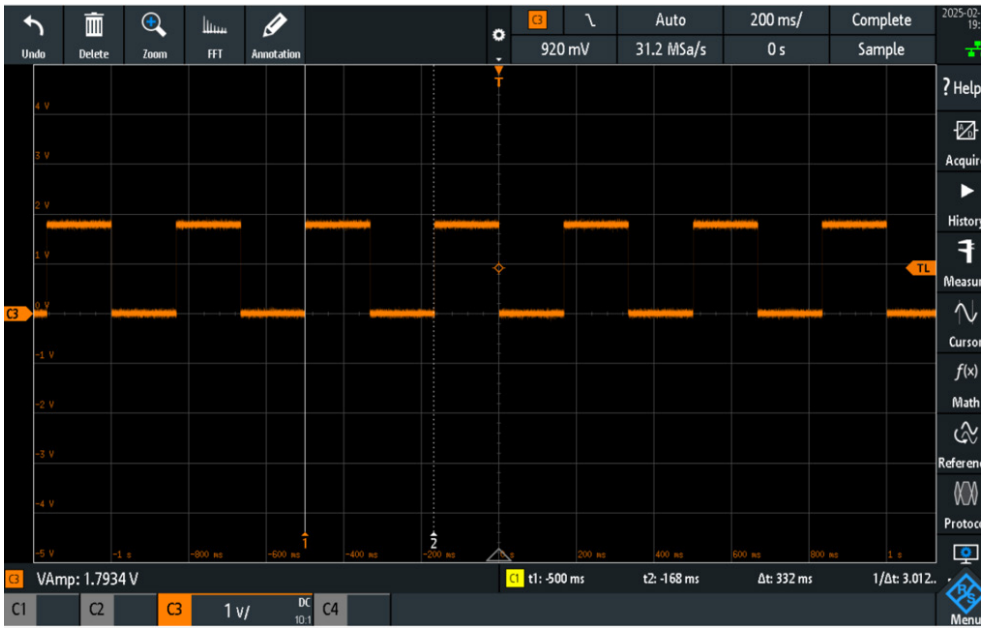

Figure 21. 3Hz Count GPIO Output

4.2 Output From Out Pin

The Time Clock and Time Sync counters can be outputted using Out8–11 pins.

1. Select the **iod_mux_sel** source of the corresponding output as Time Clock or Time Sync in RICBox. Then configure the output frequency goal in the main block diagram. The counters can be divided down using the corresponding integer output divider (IOD).
2. In RICBox, click on Out8–11 block and select the IOD input reference from the drop-down menu on the left side of the pop-up page.

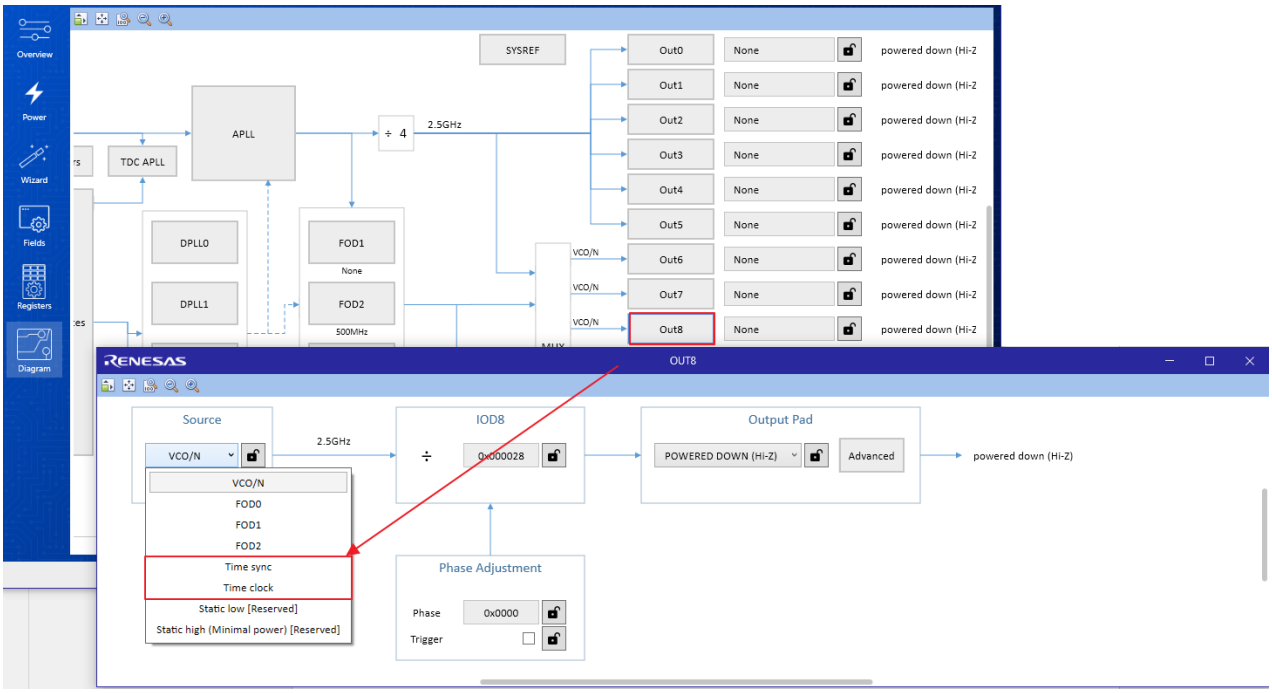

Figure 22. Configure Out8 to Use TOD Reference

3. The pulse width for the Time Sync outputs is configured using **time_sync_width (0xA02 [5:0])** register set. This can be configured in RICBox using the dropdown menu in the TOD and Synthesis block.

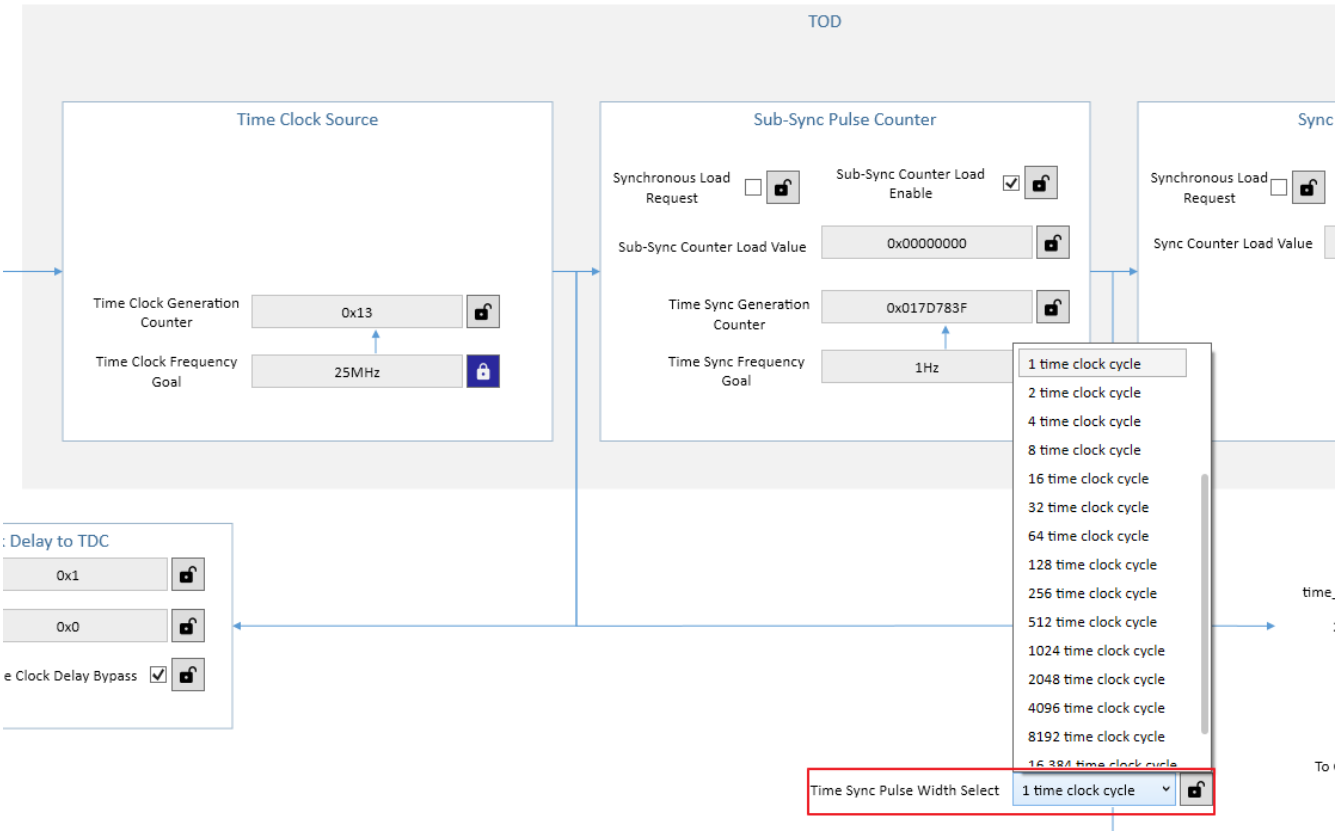

Figure 23. Configure Time Sync Pulse Width

5. Revision History

Revision	Date	Description
1.00	May 7, 2025	Initial release.

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS 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 OR 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 developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate 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. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/.

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.