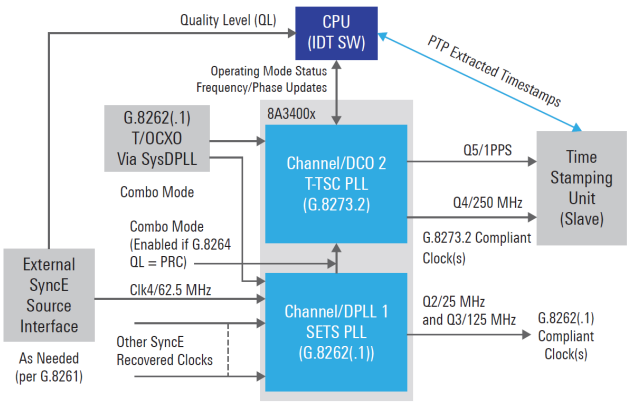
pcm4l Software

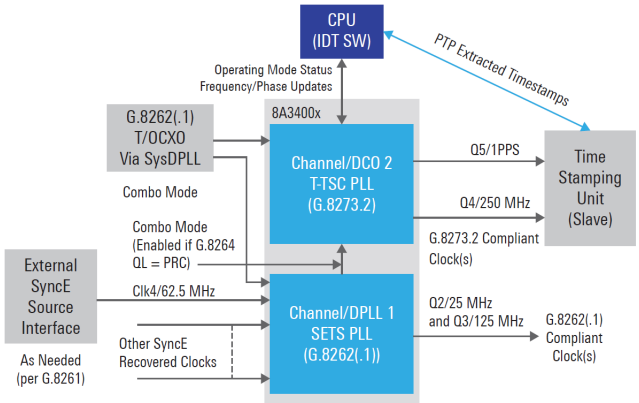
Materials highlighted in bold must be accessed by downloading the pcm4l files link from the Renesas pcm4l webpage.

Before starting, ensure that your time stamper has Linux PHC (PTP Hardware Clock) support. If it does not, you will either need to get one generated, or choose a different time stamper.

1. Clocking and Hardware. A functioning hardware platform is necessary. Complete the following directions to accomplish this.
   a. In the system, route the clocks (per your time stamper) from the Renesas DPLL to the time stamper (e.g., 250MHz/1 PPS or 8kHz/1 PPS).
   b. Confirm the TCS (ClockMatrix configuration file) is correct for the system.
   c. An example **Clock Matrix hardware configuration file with pure PTP "(8A3400x_G.8275.1_49p152MHz_v6_eEEC.tcs)"** is available. You may need or want to customize frequencies and other details using Timing Commander available from Renesas.
      i. Use combo mode from System DPLL (from OCXO/TCXO) to the PTP channel.
   d. Start with the example **ClockMatrix hardware configuration file with SyncE support (8A3400x_G.8275.1_49p152MHz_v6_purePTP.tcs)**.
      i. Use combo mode from System DPLL (from OCXO/TCXO) to the PTP and SyncE channels.
      ii. Use combo mode from SyncE channel to PTP channel.

      The following diagrams show both configurations with other system components.





   e. From the Timing Commander GUI, export the .bin file (ptp_idtcm driver configuration file) corresponding to the .tcs file (the .bin file is located in the "/lib/firmware" directory).
   Note: Bin file generation is part of Timing Commander Installer (v1.16.3) or newer, and the Timing Commander Personality File for ClockMatrix 8A340xx for version 8.4.1 and newer
   f. Before proceeding to the next step, confirm that the signals being generated from your hardware are correct. If they are not, start over. If you get stuck, please contact the Renesas web support team using the general support button on the web page.

2. PHC Driver for Renesas Clock Hardware. The next step is to confirm that the kernel driver is engaging the clocking device properly.
   a. The driver source is present in the Linux kernel.
   b. More information about the Linux ClockMatrix driver including adding it to your kernel and checking the operation is in the *Linux PTP Using PHC Adjust Phase Reference Manual*.
   c. Load the ptp_idtcm kernel module (via modprobe ptp_idtcm or similar) and confirm the firmware version as printed by the module matches the hardware. This confirms that I^2C read function is working. Note that it may not flag errors in I^2C write function.
3. Ts2phc/ptp4l. This step ensures that the TSU and Clock Hardware are interfacing properly with one another.
   a. Run ts2phc – Confirm that the seconds portion of the ToD is correct at the DPLL and time stamper.
      i. Observe the difference as reported by ts2phc. If the system is working the difference should be 0 unless starting up or locking to a new master.
      ii. If the difference is not 0, you need to troubleshoot:
         1. Confirm that the clock had enough time start up, or if going through switchover, ensure the clock had enough time to lock to the new master.
         2. Ensure the 1pps hardware clock is working properly.
         3. If the above does not resolve the issue, then something is likely wrong with the kernel driver for the TSU, and that vendor should be consulted.
      iii. An example **ts2phc.cfg** is available.
   b. An example configuration file for unicast mode is in **standalone_G.8275.2.cfg**.
   Note: While the standalone_G.8275.2.cfg configuration can process unicast packets using ptp4l with the linear filter in ClockMatrix™ hardware, this will likely yield results that are not compliant to the ITU performance specifications [G.8273.4].
   c. An example configuration file for multicast mode is in **standalone_G.8275.1.cfg**.
   d. Run ptp4l. Confirm performance by measuring the 1 PPS from the system versus 1 PPS from the master using “ts2phc -m -f ts2phc.cfg &” then “ptp4l -m -q -i eth0 -p /dev/ptp1 -s -f standalone_G.8275.1.cfg”.
      i. If the performance is insufficient, the protocol test network and/or master are likely faulty.
4. <STOP>. Do not start integrating pcm4l until ts2phc/ptp4l is running correctly since pcm4l uses the same hardware and drivers as pcm4l and it is easier to debug a new system.
5. Pcm4l/ptp4l. The final step is to get the pcm4l and ptp4l interface up and running.
   a. For a unicast master, the system should use ptp4l with **externServo_G.8275.2.cfg** and **reConfigPCM_G8273_4_P.json** with pcm4l. When using reConfigPCM.json and externServo_G.8275.2.cfg, pcm4l will be compliant to the ITU performance specifications [G.8273.4] due to the Renesas advanced servo present in pcm4l.
   b. For a multicast master, the system should use ptp4l with **externServo_G.8275.1.cfg** and **reConfigPCM_G8273_2.json** with pcm4l. This uses pcm4l in Write Phase mode providing improved accuracy in an aware network.
   c. Run pcm4l using “./pcm4l -f reConfigPCM.json” then “ptp4l -f externServo_G.8275.2.cfg”.
      i. Confirm performance by measuring the 1 PPS from the system versus 1 PPS from the master. Refer to “pcm4l: Basic Debugging Steps for Renesas PTP Clock Manager Software” to understand any messages and address possible bugs. If this fails to solve the problem, contact the Renesas web support team using the general support button on the web page.

1. Revision History

Revision	Date	Description
1.0	Jan.29.21	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 skilled in the art 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 for development of 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 out of 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.

(Rev.1.0 Mar 2020)

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/

Trademarks

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