BE.services

CODESYS SysEthernet driver for PRU-ICSS

Software.
Embedded.

Target Board: TMDXIDK5728

CONTENT

1 Overview

The SysEthernet driver for PRU-ICSS enables the CODESYS runtime to access PRU-ICSS for Ethernet communication. This can be used for standard Ethernet based industrial protocols such as EtherCAT. The SysEthernet also supports the time triggered send (TTS) functionality available in the PRU-ICSS to provide a communication setup with very small transmission jitter. This document will present the architecture of the final system along with test results.

2 System Architecture

The System architecture for the final design is shown in figure 1. The CODESYS IDE interacts with the TI Sitara Development Kit (TMDXIDK5728) using one of the Gigabit Ethernet ports. The CODESYS IDE provides the user interface to the user to configure the target application based on the requirement using the IEC IEC 61131-3 programming. The final application is downloaded on the board which is executed with the CODESYS runtime system.

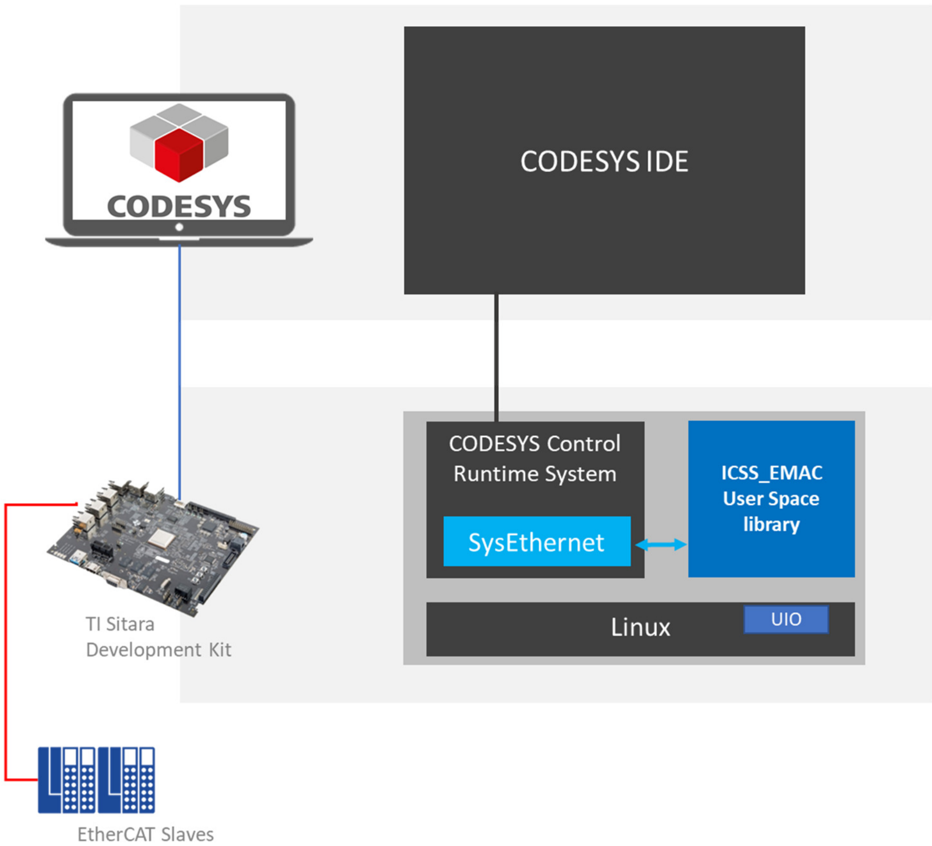

Figure 1: System Architecture with PRU-ICSS

The CODESYS Control runtime system for the TI Sitara development kit includes an updated *SysEthernet* component. The updated SysEthernet implements an interface to the user space ICSS-EMAC driver which provides the API to control the PRU. The ICSS-EMAC driver uses the UIO kernel space module to map the PRU-ICSS memory into user space and along with interrupts via events. The driver is capable of handling multiple features which includes storm control, MAC address filtering, Quality of Service using multiple Queues and the time triggered send. The detailed software architecture of this setup is shown in figure 2. The ICSS-EMAC user space low level driver (LLD) uses the PRU user space LLD and OSAL library to access the PRU hardware. The UIO kernel module is used to map the PRU HW memory into the Linux user space as well as provide an interrupt handling mechanism in the user space using file descriptors.

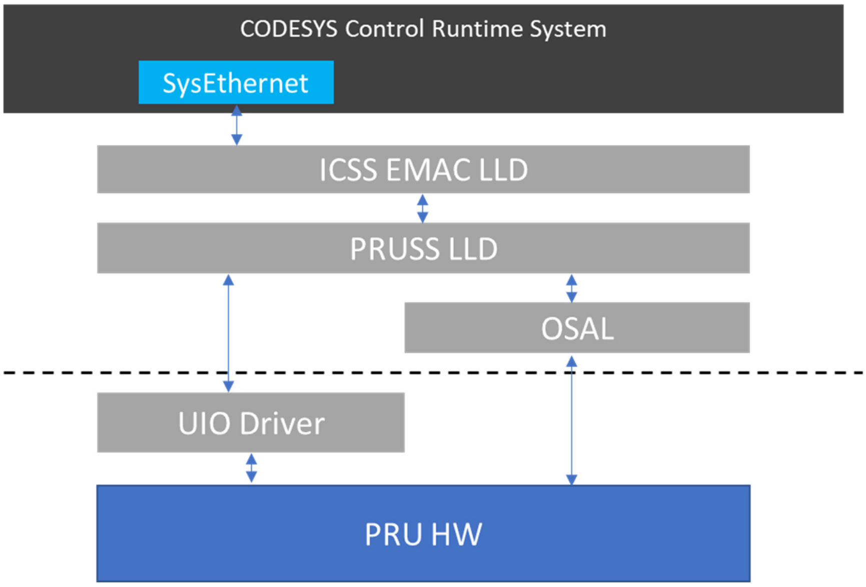

Figure 2: ICSS-EMAC User Driver interface to CODESYS

The CODESYS Control Runtime System for Linux space for ARM platforms was used as the base. The existing SysEthernet component in the runtime used Linux socket APIs to access the Ethernet interfaces which are prone to high transmission jitter and round trip time. The updated SysEthernet component implements an interface to the ICSS-EMAC driver to utilize the APIs to access the PRU directly from the CODESYS runtime. This allows to control the jitter and the round trip time by avoiding the longer and slower socket based communication. Additionally, the SysEthernet component is designed to utilize the time triggered send (TTS) functionality of the PRU to send the frames. TTS provide a way to achieve deterministic communication by reducing the transmission jitter for cyclic communication. The next section gives an overview of the TTS functionality.

2.1 Time Triggered Send

TTS can be utilized to transmit cyclic packets which have pre-define cyclic interval for transmission of the frames. The PRU HW implements the cyclic trigger logic to send frames at a pre-defined intervals on priority basis which can be used by application to reduce the transmission jitter and achieve deterministic communication.

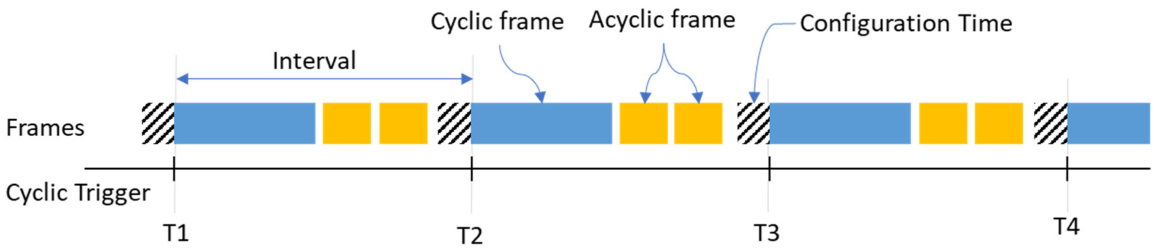

Figure 3: Cyclic communication with Time Triggered Send

The cyclic communication of frames is shown in figure 3. To perform the cyclic communication the PRU HW needs to be configured with following parameters:

- **cyclePeriod**: TTS Cyclic interval
- **cycleStartTime**: A future IEP counter value at which the first cyclic packet will be transmitted.
- **configTime**: Configuration time for PRU to setup upcoming TTS cycle.

These parameters allow the PRU to create cyclic time triggers which is used to transmit frames. The **configTime** is an application specific parameter which should be set to a value large enough so that the application is able to copy the frame to be transmitted in the PRU memory before the cyclic trigger occurs. If the frame is not available at the time of the trigger a worst case transmission time of one cycle can be expected.

Along with cyclic communication acyclic packets can also be transmitted when operating in the TTS mode. This is done by having separate priority for the Queues in the PRU HW. TTS frames should be queued in the Queue 0 which has the highest priority whereas the acylic packets should be queued into non-priority queues (Queue 1-3). The acyclic packets are transmitted only if enough time is available for the frame to be transmitted successfully without violating the cyclic interval. This is enusred by estimating the transmission time of the acyclic frames and computing the remaining interval in each cycle.

The PRU HW provides two ways to ensure that the application successfully copies the frame into the PRU memory before the next cyclic trigger.

Polling mode: In this mode, the PRU firmware sets a status bit (**insertCycFrameNotification**) in the control register when it is time to start queueing the frame. The time for the notification is controlled using the **configTime** parameter as explained before. Once the packet is written into the queue, the firmware clears this bit. The application can control is bit to identify the time at which it should start copying the frame.

Interrupt Mode: In this mode, the PRU firmware addition to setting the status bit, also generates an interrupt when its time to copy the cyclic frame. The application should correctly configure the interrupt during initialization of the ICSS EMAC driver by enabling this interrupt setting the **ICSS_EmacTTSEnableCycPktInterrupt** flag in the **ICSS_EmacInitConfig**. Also an interrupt handler should be registered using the **ICSS_EmacRegisterHwIntTTSCyc** routine. The application should then copy the frame into the Queue 0 in the interrupt handler of this cyclic interrupt.

3 Evaluation end Results

The network performance of the implementation was evaluated using BE.services Real Time Test Framework (RTTF) which uses the CODESYS test manager to perform a series of tests on the target system varying the system and network load and evaluating the transmission jitter and the round trip time of the system. The transmission performance was evaluated for three variants using PRU in different modes.

- ICSS-EMAC without TTS: First variants uses the SysEthernet along with the ICSS-EMAC driver but the TTS is disabled
- ICSS-EMAC with TTS: Similar to first but with the TTS enabled
- PRU with native driver: This variant uses the native linux driver available for the PRU. The SysEthernet uses socket to communicate via the PRU.

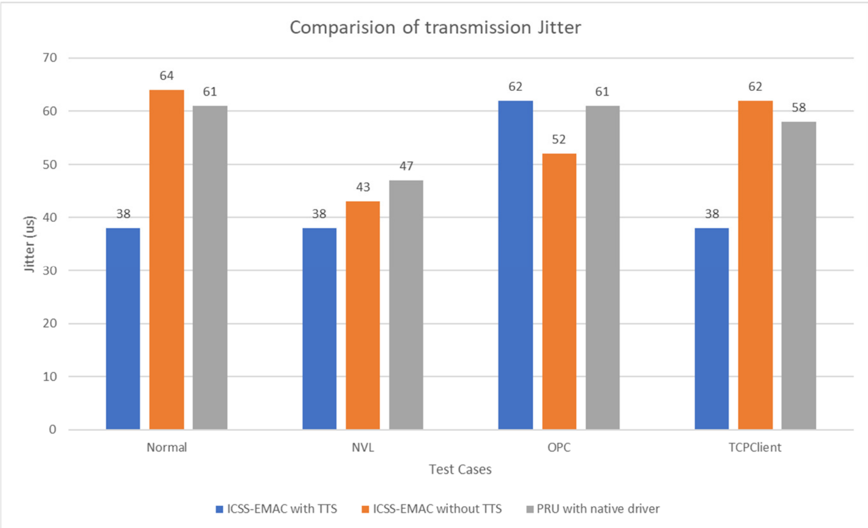

Figure 4: Comparision of Transmission jitter

The figure 4 shows the plot comparing the transmission jitter of the three variants using PRU. The ICSS-EMAC version with TTS enabled has the best performance with the maximum jitter of 38 us in all the test cases. Also the max jitter is reported for a single packet as shown in the figure 5. The higher jitter for a single packet is reported as the TTS is initialized dynamically which causes delay in transmission of the first packet. The jitter of the other two variants is comparable but the variant using native driver has higher variance of the jitter.

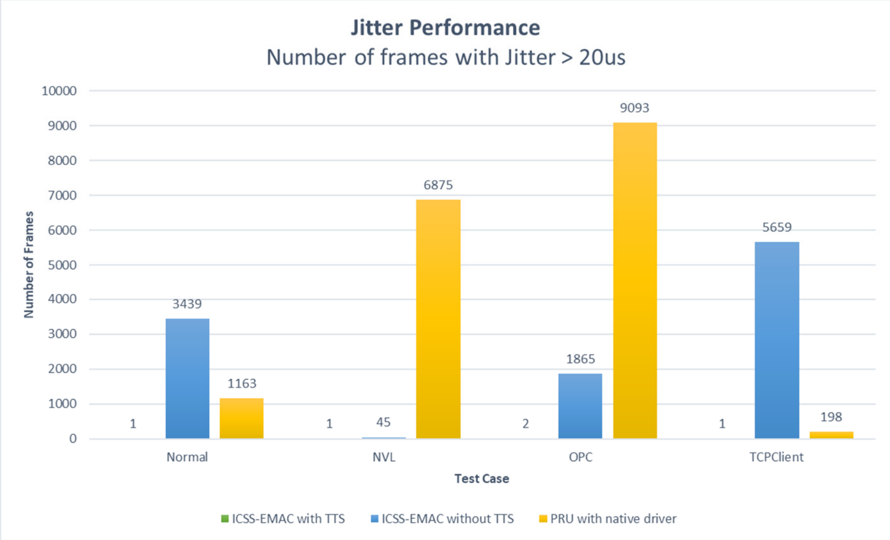

Figure 5: Jitter Performance of the variants

Comparing the round trip times, the ICSS-EMAC without TTS has the best performance reporting a maximum RTT <800us all test cases. ICSS-EMAC with TTS also does not have bad performance and has maximum RTT < 900 us. The slightly higher RTT can be attributed to 250 us configuration time which is required to ensure that the maximum jitter of the transmission remains below 20us for all frames. This was confirmed by reducing the configuration time to lower values and performing the tests again. The lower configuration time produce similar RTT like the without TTS but few packets have higher jitter. The main reason being the variance of task jitter which results failure to copy the frame into PRU memory before the TTS trigger. Figure 6 shows the plot comparing the maximum RTT values for each variant.

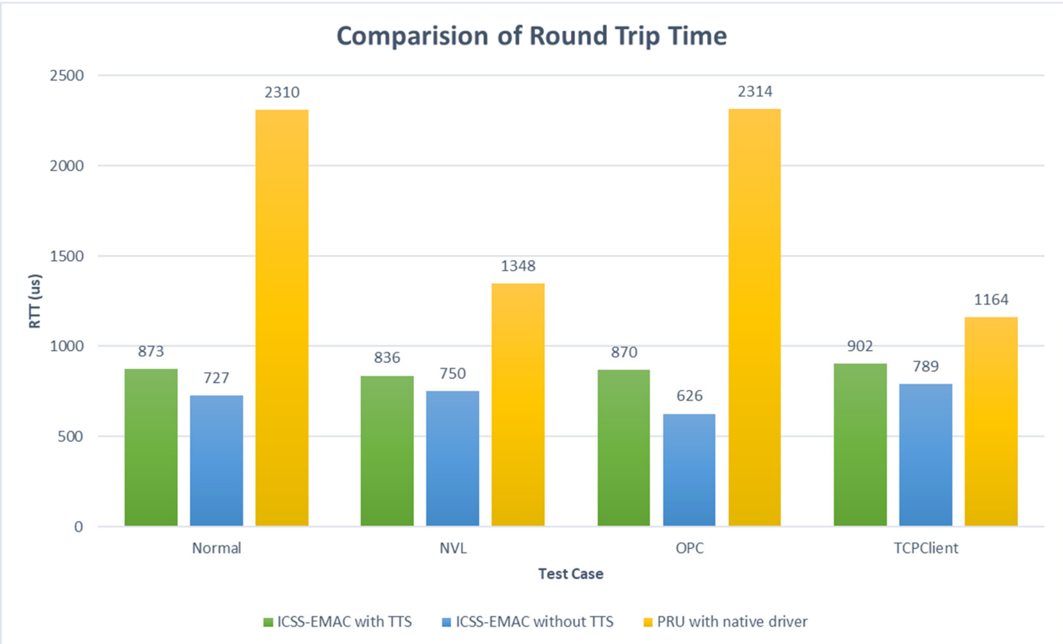

Figure 6: Rount Trip Time comparision

Overall the PRU with TTS is able to deliver a network communication with very low jitter and small RTT times which are ideal for Industrial communication environment.

Result Summary consolidating the vaues of the evaluation:

Variant	Test	MinRTT	MaxRTT	Jitter
ICSS-EMAC with TTS	Normal	408	873	38
	NVL	393	836	38
	OPC	387	870	62
	TCPClient	394	902	38
ICSS-EMAC without TTS	Normal	253	727	64
	NVL	257	750	43
	OPC	254	626	52
	TCPClient	252	789	62
PRU with native driver	Normal	241	2310	61
	NVL	241	1348	47
	OPC	241	2314	61
	TCPClient	241	1164	58

Bibliography

[1] ICSS EMAC LLD developers guide
http://processors.wiki.ti.com/index.php/ICSS_EMAC_LLD_developers_guide#Time_Triggered_Send

[2] PRU-ICSS / PRU_ICSSG
http://software-dl.ti.com/processor-sdk-linux/esd/docs/latest/linux/Foundational_Components_PRU-ICSS_PRU_ICSSG.html

[3] EtherCAT® Reference Design on Sitara AM57x Gb Ethernet and PRU-ICSS with Time Triggered Send
http://www.ti.com/tool/TIDEP0079
http://www.ti.com/lit/ug/tidubz1b/tidubz1b.pdf

[4] Processor SDK Linux ICSS-EMAC User Space
http://processors.wiki.ti.com/index.php/Processor_SDK_Linux_ICSS-EMAC_User_Space

[5] EtherCAT Master Evaluation for TI Processors
https://www.acontis.com/en/ethercat-for-ti-processors.html