MARVELL

System and Latency Requirements

Contribution to 802.3dm Task Force

September 17, 2024

TJ Houck (Marvell)
Paul Fuller (Marvell)

Supporters

Daniel Cashen (Stellantis)

Sasha Zbrozek (Cruise)

Kresimir Mirosavljevic (Cariad)

Thomas Hogenmueller (Bosch)

Sanjeev Jain (Nvidia)

Jay Cordaro (Analog Devices)

Heath Stewart (Analog Devices)

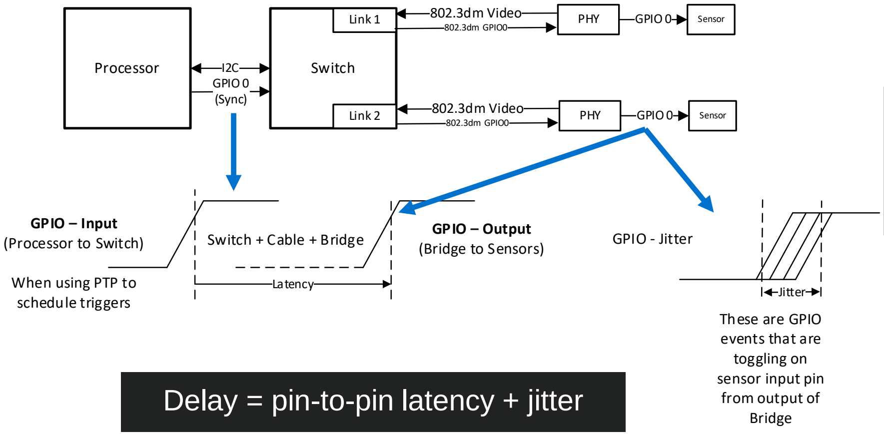
William Lo (Axonne)

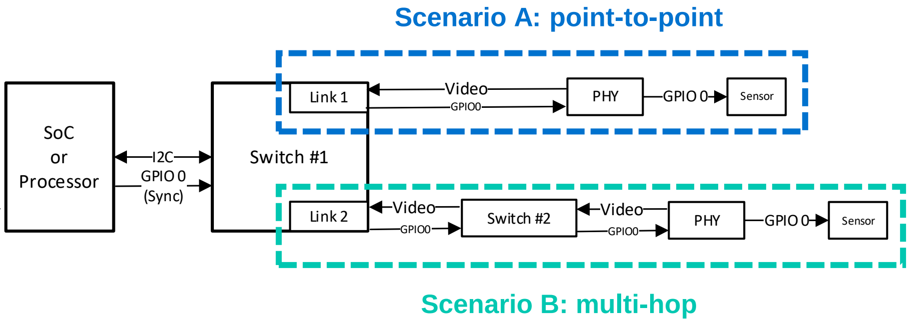
Topics

- Delay, latency and jitter
- Differences in point-to-point and multi-hop 802.3dm links
- Can PTP (IEEE 1588) handle all scenarios
- GPIO forwarding requirements
- I2C latency importance
- Link start-up delay
- Task force proposals: latency limits

Delay in point-to-point and multi-hop scenarios

- Delay = latency measured from pin to pin + jitter
  - Customers care about the worst-case time of arrival at the pin of the sensor
- Scenario A: refers to a point-to-point use case (link 1)
- Scenario B: refers to multi-hop(s) or a “daisy-chain” or reaggregation application (link 2)





Delay = pin-to-pin latency + jitter

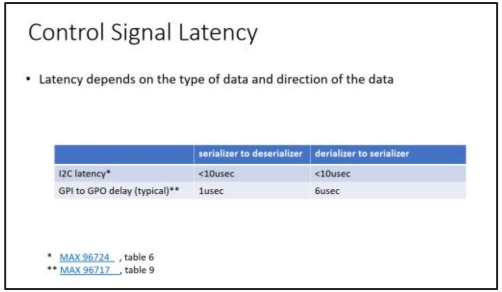
This presentation will focus on point-to-point delay (scenario A)

Referenced: https://www.ieee802.org/3/dm/public/0724/houck_fuller_3dm_01_0724.pdf

Reminder: latency requirements

- Latency and jitter are important for current production systems
  - These systems use low latency to avoid long sensor initializations
  - Systems can require unique frame synchronization with other ADAS sensors
- For automotive sensor applications, Ethernet latency should not exceed **10us**
  - Hard limit related for system designs (from SoC to camera)
    - This includes trigger (GPIO) events or a single I2C command without ACK/NACK response
    - This measurement is from Pin of Bridge to Pin of Switch as shown in the previous diagram

GMSL Latency



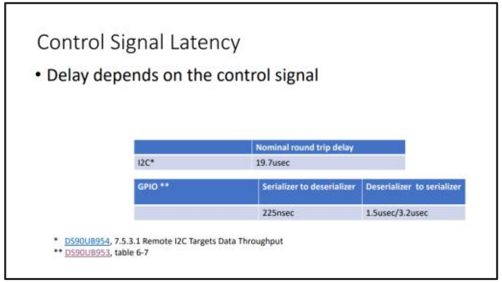
Control Signal Latency

- Latency depends on the type of data and direction of the data

	serializer to deserializer	derializer to serializer
I2C latency*	<10usec	<10usec
GPI to GPO delay (typical)**	1usec	6usec

* MAX 96724 , table 6
** MAX 96717 , table 9

FPD-Link Latency



Control Signal Latency

- Delay depends on the control signal

	Nominal round trip delay
I2C*	19.7usec

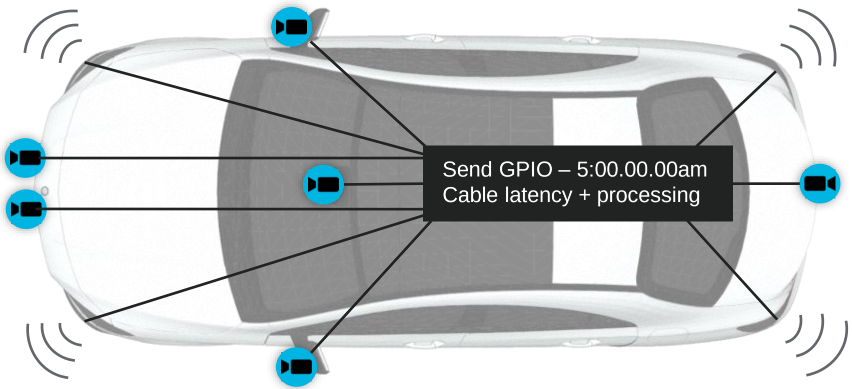
GPIO **	Serializer to deserializer	Deserializer to serializer
	225nsec	1.5usec/3.2usec

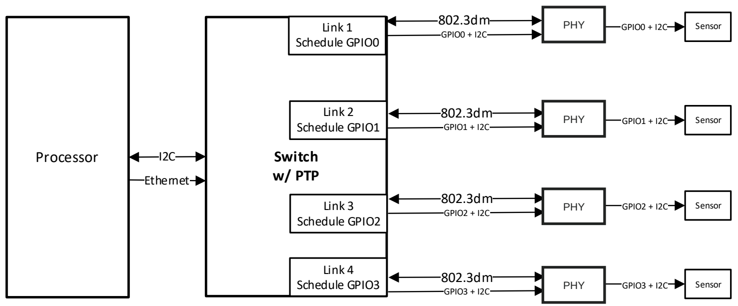
* DS90UB954, 7.5.3.1 Remote I2C Targets Data Throughput
** DS90UB953, table 6-7

Referenced: https://www.ieee802.org/3/dm/public/adhoc/090524/razavi_dm_09042024.pdf
Referenced: https://www.ieee802.org/3/dm/public/adhoc/082224/razavi_dm_08222024_v2.pdf

Can PTP handle every scenario?

- Why not use PTP to synchronize all the sensors?
  - PTP can be effective for general synchronization
  - Software overhead and proper alignment is difficult for all scenarios

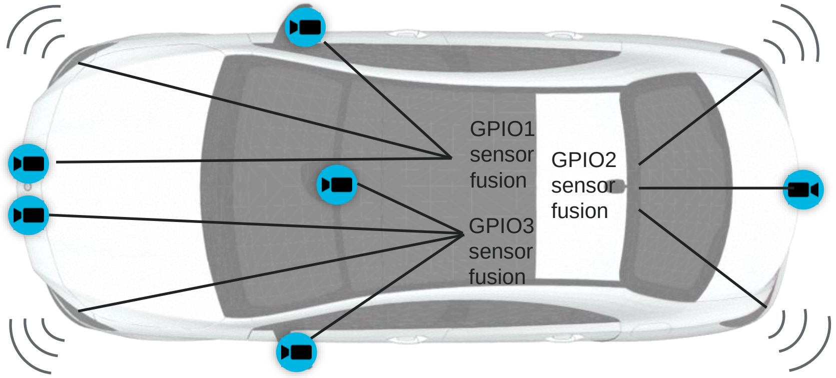


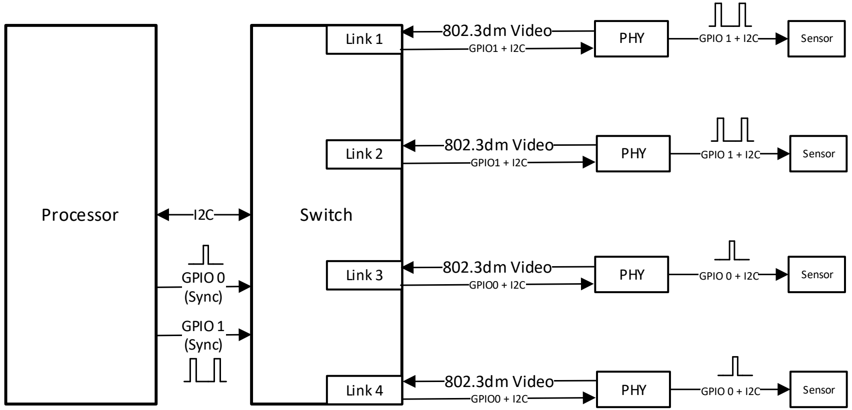


When scenarios are changing, controlling latency is paramount.
GPIO can provide precision delay control.

GPIO addresses system delay requirements

- GPIO forwarding provides the ability to rapidly respond to changing scenarios
  - GPIO allows for easy implementation with no software overhead
- Example application: camera and sensor fusion applications
  - LIDAR and radar have varying FPS and need to avoid interference

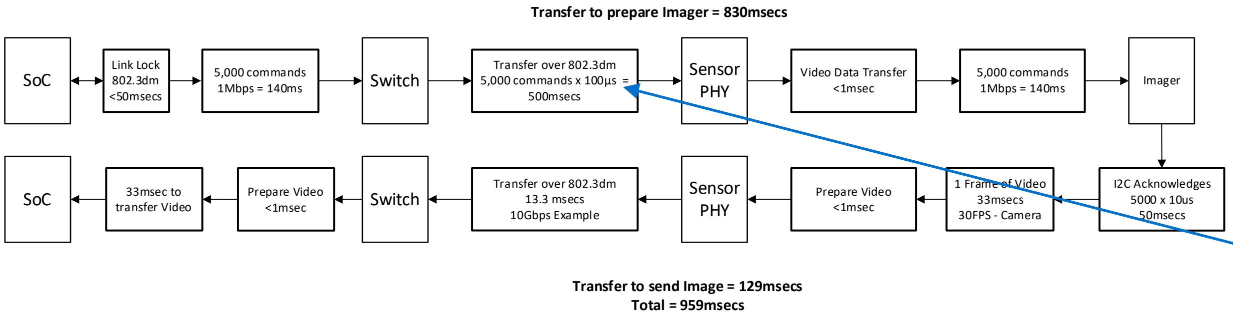


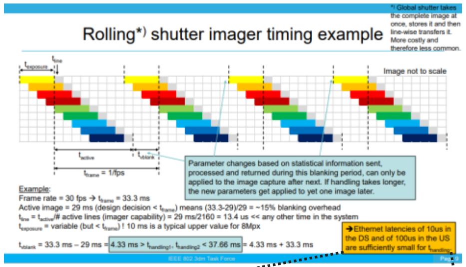


GPIO: for all sensors to sync properly, a delay <12-15us is required to properly align radar, LIDAR and camera data

Link initialization imposes strict requirements

- Additional latency will accumulate quickly for I2C commands
- Customers expect 1st frame in <300ms for rear-facing cameras due to FMVSS No.111
  - The regulations mandate that rearview camera systems must be operational within 2 seconds of the vehicle being placed in reverse
  - As more sensors are added customer expects quicker initialization
- Example total link budget impact is ~1 second





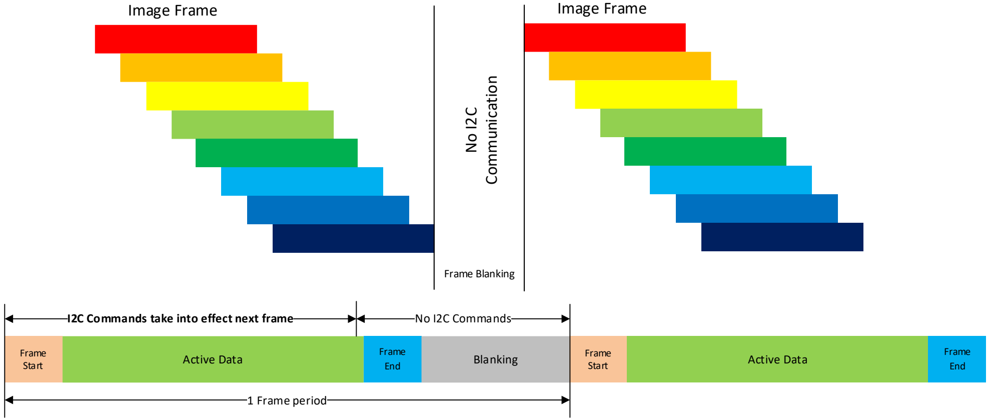
→Ethernet latencies of 10us in the DS and of 100us in the US are sufficiently small for $t_{handling}$.

Increasing link budget time impacts overall processing cycle timing budget. This example leaves only 1.04 seconds

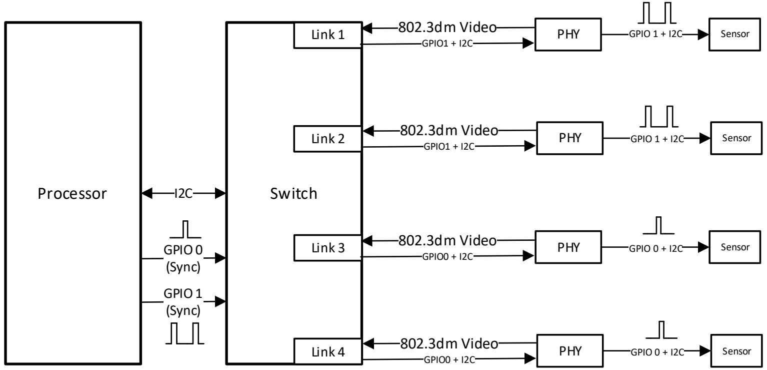
Referenced: https://www.ieee802.org/3/dm/public/0724/houck_fuller_3dm_01_0724.pdf
Image from: https://www.ieee802.org/3/dm/public/0724/matheus_dm_02b_latency_07152024.pdf

I2C real-time operating requirements

- Imagers have a fixed window to communicate
  - Making use of the given allocation time is critical to get the next image frame updated before the “window” of communication closes



- Sensors can take <10 to 30 I2C commands to update frame information
- 100us = 3msecs | 10us = 300usecs
- Next-generation sensors: Interior 90-120FPS | Exterior – 60FPS
- This means sensors will only have 11-15msecs/frame
- SoCs will want to process as much as the frame as possible before issuing commands



- When 4 sensors use the same I2C bus additional latency will be experienced
- When all 4 sensors are needing the bus at the same time the communication is now 1/4th of the allotted time of a single sensor

802.3dm links will now be

3msecs x 4 = 12msecs

300usecs x 4 = 1.2msecs

This would take 1 frame to complete all the transactions for the last sensor, NOT INCLUDING PTP SCHEDULING COMMANDS

Summary

1 It is essential to limit both **delay** and **start-up time** for automotive Ethernet links to meet customer and application requirements

2 It is proposed to **limit delay to <12µs** from Processor (SoC) to Sensor for I2C commands and GPIO trigger events

3 It is proposed to **restrict link lock time to <50ms** on initial / cold power up

4 Minimize Change on Existing SERDES (GMSL/FPD-Link)

MARVELL™

Essential technology, done right™