Technical Article

Implementing an IO-Link Master With Deterministic Timing

Steffen Graf

In industrial factory automation designs, IO-Link has evolved into a popular protocol for sensors and actuators to improve factory efficiency and reduce downtime. The IO-Link master is a crucial component of an IO-Link system, as it initiates the complete communication cycle, handles the cycle timing, and exchanges process data between a programmable logic controller and connected devices.

IO-Link uses half-duplex serial communication with a COM1 (4,800 baud), COM2 (38,400 baud) or COM3 (230,400 baud) data rate at 24 V. The simplicity of the lower protocol layers makes it possible to build an IO-Link device comprising a small microcontroller (MCU) with an integrated universal asynchronous receiver transmitter peripheral and an external physical layer (PHY). The external PHY acts as a level shifter and detects the wakeup pulse to switch the IO-Link device from digital in/out mode into IO-Link mode.

The master side of an IO-Link system is more complex. The PHY is similar, with a small current sink at the communication line (CQ) to support simple transceivers on the device side; it must be able to send the wakeup pulse. The power supply has to provide a 24-V voltage to all connected devices such as sensors and actuators. The master needs to support all three data rates, and typically comes in configurations with four or eight ports.

In this article, I will address how to implement high-performance IO-Link masters in industrial systems to achieve precise cycle times and deterministic latency.

IO-Link Timing

The IO-Link standard specifies several types of timing to build interoperable systems working with devices from different manufacturers. Cycle time (t_{CYC}) is the most important; the data sheet of each device and master will always specify t_{CYC}. The master initiates every process data exchange and the device responds. $t_{C\ YC}$ is the time between two messages from the master.

Depending on the data rate, the fastest cycle time is 400 µs (COM3), 6.4 ms (COM2) or 32 ms (COM1), specified with a tolerance of 0% to 10%. The latest version of the standard extended the minimum tolerance from 0% to –1%, as a tolerance of 0% is impossible to implement. There will always be some tolerance, and even one bit time too fast means that the master does not meet the timing specified by the standard.

Implementing a master requires precise timing on the port and jitter significantly less than one bit time. Figure 1 shows the CQ line in blue and the interrupt request (IRQ) to the host MCU in yellow. The connected IO-Link device uses a cycle time of 133 ms and a data rate of COM2. Configuring the trigger to delay by one cycle time and the display to infinite persistence enables a view of the jitter. The message sent from the master is visible first, and the response from the device is blurred after it. This jitter delays the ready interrupt slightly, depending on when the master receives the last bit from the device.

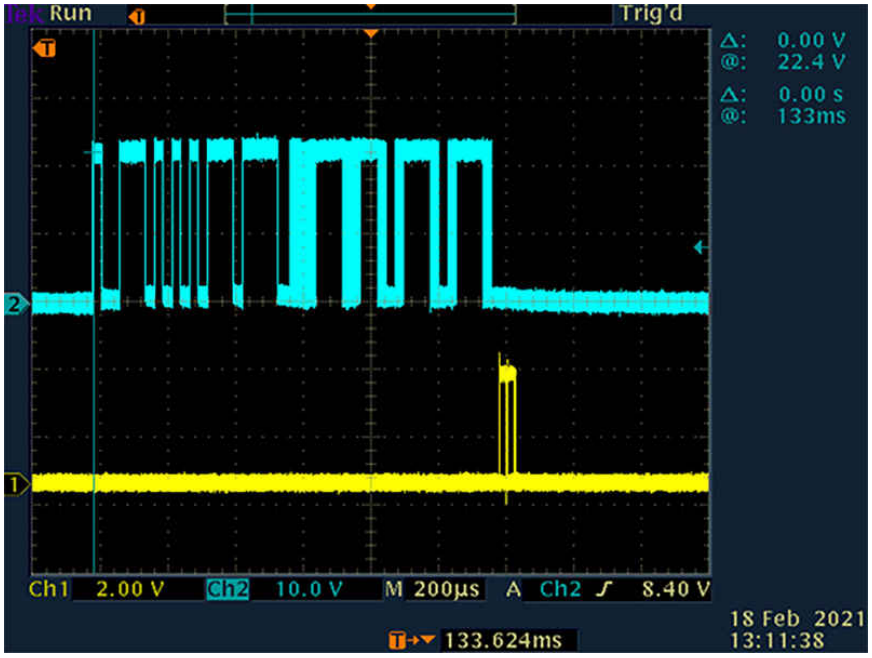

Figure 1. IO-Link Communication Jitter and Receive Interrupt

The jitter on the cycle time shown in Figure 1 is significantly less than one bit time and is not visible on this particular scope shot. The device has more jitter when replying to the message, but as this time (t_A) is given by the IO-Link standard with three to 10 bit times, the measured time is well in its limits.

Looking at delays is also interesting. The shorter the delay between the last bit and the issuing of the IRQ, the more time the central processing unit (CPU) has to process the data and prepare for the next cycle. The scope shot in Figure 1 looks as if there is one bit time between the last falling edge on the CQ line and the rising edge of the IRQ signal. The last bit of the transmission is the stop bit (low and not properly visible on the figure), indicating the issuing of the IRQ immediately after that last bit.

Figure 2 illustrates the start of a transaction. The falling edge (the magenta line) is the start signal of the frame handler. The CQ line is turquoise and the internal time base is yellow. I captured this image in persistence mode to see whether some of the signals were unstable.

The scope shows that there is no significant jitter when the master starts sending out a message.

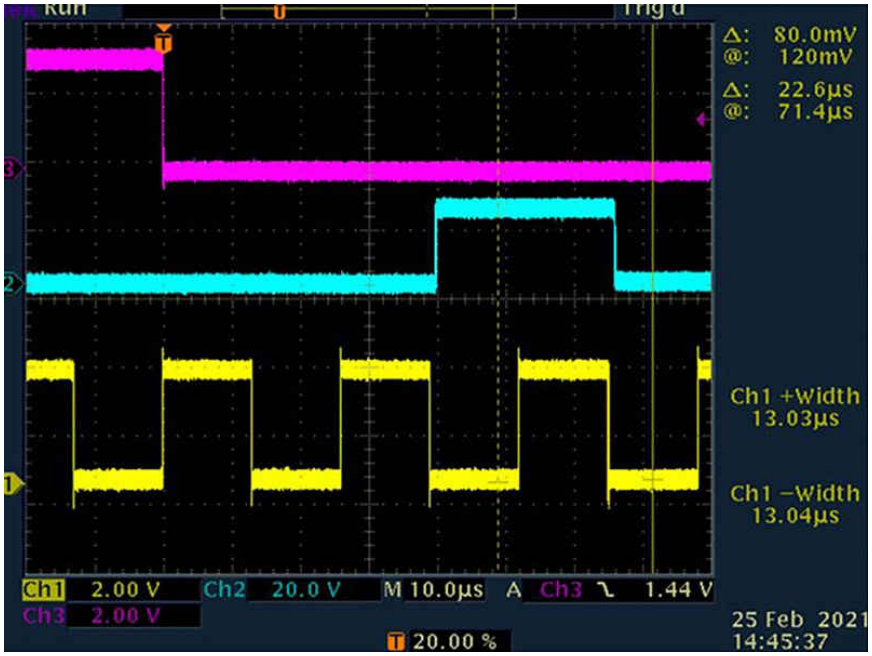

Figure 2. Start of IO-Link Communication Cycle and Internal Signals

When creating a high-performance master system with deterministic timing, the frame handler should have no significant jitter when sending out a message, and no jitter on the cycle time.

The implementation features eight ports. An industrial communications subsystem generates the cycle times shown in Figure 1, while Figure 2 shows more detail. One programmable real-time unit (PRU) handles the IO-Link frames of the eight ports simultaneously at different baud rates, along with a second PRU that generates precise timing. This implementation enables accurate cycle times and deterministic latency between the ports that can synchronize connected IO-Link devices such as sensors or actuators.

The ability to configure times as well as read-and-write data frames occurs through register access inside the Sitara AM4379 device, completed in nanoseconds. The combination of a high-performance Arm® core with a real-time CPU core and frame handler firmware sets the foundation for a highly integrated IO-Link master implementation.

Additional Resources

- Check out the design guide for the Eight-Port IO-Link Master Reference Design.
- See the AM437x Sitara processor data sheet.
- Read the application reports:
  - "Adding Real-Time Communication to Linux."
  - "Advanced High-Side Switch with Dynamic Current Limit for IO-Link Master."

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), 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, regulatory 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 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.

TI objects to and rejects any additional or different terms you may have proposed.

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