Industrial Interface Standards Overview:
RS-485/422, PROFIBUS, RS-232, CAN, LIN, I2C, IO-Link

October 2018
Transceiver Interface Products
Agenda

Following standards will be covered
• RS-485
• RS-422
• ProfiBus
• RS-232
• CAN
• LIN
• I2C
• IO-Link
Communication Hierarchy

- **Industrial Ethernet**
- **Field Bus (RS-485, CAN, RS-232)**
- **Device level/Sub-bus (IO-LINK, LIN, etc.)**
The **Open Systems Interconnection model** (OSI Model) is a conceptual model that characterizes and standardizes the internal functions of a communication system by partitioning it into abstraction layers.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Application</td>
<td>Message format, human-machine interface</td>
</tr>
<tr>
<td>2</td>
<td>Presentation</td>
<td>Coding into 1s and 0s, encryption, compression</td>
</tr>
<tr>
<td>3</td>
<td>Session</td>
<td>Authentication, permissions, session restoration</td>
</tr>
<tr>
<td>4</td>
<td>Transport</td>
<td>End-to-end error control</td>
</tr>
<tr>
<td>5</td>
<td>Network</td>
<td>Network addressing, routing</td>
</tr>
<tr>
<td>6</td>
<td>Data Link</td>
<td>Error detection, flow control (CAN controller, Profibus)</td>
</tr>
<tr>
<td>7</td>
<td>Physical</td>
<td>Physical representation of bits (RS-485, LIN, etc.)</td>
</tr>
</tbody>
</table>
RS-485, RS-422, PROFIBUS
What is RS-485?

- “Recommended Standard” jointly published by Telecommunications Industry Association (TIA) and Electronic Industries Alliance (EIA).
- Defines the electrical characteristics of interface circuits (line drivers/transmitters and receivers) used for serial communications over a multipoint network. Examples include:
  - Signal amplitude
  - Input sensitivity
  - Input impedance
- Does not define:
  - Cabling
  - Connectors
  - Data protocol
Why RS-485?

1. Balanced interface
   → increases noise immunity and decreases emissions

2. Multipoint, bi-directional communication on a single pair of wires
   → lower cabling costs

3. Large differential signal, large common-mode range
   → allows for communication over long distances and with large ground potential differences

4. Can achieve signaling rates up to 50 Mbps
   → suitable for a wide array of applications
RS-485

• RS-485 is a differential signaling standard which defines the electrical characteristics of drivers and receivers used to implement a balanced, multi-point transmission line.

• Key features include a large differential signal (1.5V across 54Ω) and driver and receiver operation over a wide common-mode range (-7V to +12V).

• The standard is suitable for serial data transmission at moderate data rates (up to ~20Mbps) over long distances (up to ~1,000m).
An RS-485 compliant driver must produce at least 1.5V across a 54Ω load.

\[ V_D = V_A - V_B \]

\[ |V_D| = V_{CC} - 2(V_F + V_{R-ON}) \]
RS-485 Receiver

The resistor divider network comprised of R2 and R3:
1. Biases the receiver input relative to the local \( V_{CC} \) and ground, allowing the receiver to operate without a ground wire.
2. Attenuates the voltages appearing at the A and B terminals (which may range from -7V to +12V) to operable levels between \( V_{CC} \) and ground.
RS-485 Applications Tips

1. Use twisted pair cable.  
   - $Z_0 = 120\Omega$ or $100\Omega$
2. Connect nodes via daisy-chain.
3. Terminate unused conductors.  
   - $R_T = Z_0/2$
4. Terminate one end.  
   - $R_{T1} = Z_0$
5. Apply failsafe biasing to the other end.
6. Terminate this end.  
   - $R_{T2} = [2R_{FS} \cdot Z_0] / [2R_{FS} – Z_0]$
7. Determine maximum distance/data rate.
8. Minimize stub lengths.  
   - $L_{stub} \leq 3 \cdot 10^{-4} \cdot t_r \cdot v$
9. 3V and 5V XCVRs are interoperable.
10. Use SM712 for ESD, EFT, and surge protection.
11. Limit currents with $10\Omega$ pulse-proof resistors.
12. Filter signal between XCVR and UART.
13. Isolate when GPD $\geq 7V$. 

11  

Texas Instruments
RS-485 Grounding

a) The system is vulnerable to high ground potential differences (GPD)
   – If the GPD is greater than the limit of device, the device could stop working
     or even be damaged

b) If a high GPD a large group loop currents could form, which can be coupled into
   the data lines as common-mode noise

c) Recommended by RS485 standard, adding the series resistors lower the loop
   current, but the noise could still exist
RS-485 Functions & Features

FUNDAMENTALS
• Supply Voltage: 3.3V or 5V... or 1.8V?
• Duplex: Half or full duplex?
• Data Rate: 10kbps, 1Mbps, ... , 20Mbps?

INTEGRATED ESD PROTECTION
• Low: HBM
• Medium: IEC 61000-4-2 (ESD), IEC 61000-4-4 (EFT)
• High: IEC 61000-4-5 (Surge)

SPECIAL FEATURES
• Automatic polarity correction
• High standoff/bus-fault protection
• Receiver equalization
• 1.8V I/O levels
• Wide common-mode
• Large differential output voltage

Tolerating cross-wire faults: (E-metering and longhaul networks)
High-speed data over long distance: (Encoders, seismic, traffic monitoring)
High output voltage: (Long distance and noisy environment)
High ESD/EFT (3.3V and 5V): (Factory and building automation)
Lightning protection: (Industrial networks)
Running data adjacent to power cable: (Factory and building automation)
Selecting low/high data rates at 1.8 V_{IO}: (Telecom linecards)
Profibus applications: (Factory automation)

THVD1505
SN65HVD23 / 24
SN65HVD05 / 50
THVD15xx/THVD14xx
THVD1419
SN65HVD178x → THVD24xx
SN65HVD01
THVD14xx
RS-422

Like RS-485, RS-422 is a differential signaling standard which defines the electrical characteristics of drivers and receivers used to implement a balanced transmission line.

Unlike RS-485, RS-422 is a multi-drop standard, rather than multi-point, allowing only one driver and up to ten receivers to be connected to the bus.

Any RS-485 compliant transceiver is compatible with an RS-422 application, though it may not be strictly compliant with the RS-422 standard.
## RS-422 and RS-485 Comparison

<table>
<thead>
<tr>
<th></th>
<th>RS-422</th>
<th>RS-485</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bus topology</strong></td>
<td>Multi-drop</td>
<td>Multi-point</td>
</tr>
<tr>
<td><strong>Number of Drivers</strong></td>
<td>1</td>
<td>Many</td>
</tr>
<tr>
<td><strong>Number of Receivers</strong></td>
<td>10</td>
<td>Minimum 32, up to 256</td>
</tr>
<tr>
<td><strong>Differential Output Voltage</strong></td>
<td>2V across 100Ω</td>
<td>1.5V across 54Ω</td>
</tr>
<tr>
<td><strong>Driver Output Common-Mode Range</strong></td>
<td>Unspecified</td>
<td>-7V to +12V</td>
</tr>
<tr>
<td><strong>Driver Short Circuit Current</strong></td>
<td>150mA</td>
<td>250mA</td>
</tr>
<tr>
<td><strong>Minimum Receiver Input Impedance</strong></td>
<td>4kΩ</td>
<td>12kΩ</td>
</tr>
<tr>
<td><strong>Receiver Input Common-Mode Range</strong></td>
<td>-7V to +7V</td>
<td>-7V to +12V</td>
</tr>
</tbody>
</table>

1 In RS-422, driver short circuit current is specified from each A and B output to ground. In RS-485, driver short circuit current is specified from A to B, B to A, and from each A and B output to -7V to +12V.
ProfiBus

- ProfiBus is a widely used fieldbus standard in factory automation applications.
- IEC 61158-2 defines multiple transmission standards, of which RS-485 is the most common.
- While similar to standard RS-485 in many ways, some additional specifications are imposed, and existing specifications are extended.
ProfiBus and RS-485 Comparison and Physical Layer and Bus Requirements

<table>
<thead>
<tr>
<th></th>
<th>RS-485</th>
<th>ProfiBus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Rate</td>
<td>Not specified</td>
<td>9.6kbps to 12Mbps</td>
</tr>
<tr>
<td>Minimum Differential Output Voltage</td>
<td>1.5V</td>
<td>2.1V</td>
</tr>
<tr>
<td>Cable Characteristic Impedance</td>
<td>120Ω</td>
<td>135Ω to 165Ω¹</td>
</tr>
<tr>
<td>Termination Resistance</td>
<td>120Ω</td>
<td>220Ω</td>
</tr>
<tr>
<td>Failsafe Resistance</td>
<td>Not specified</td>
<td>390Ω to V_CC and GND</td>
</tr>
<tr>
<td>Receiver Input Impedance</td>
<td>12kΩ (min.)</td>
<td>12kΩ (min.)</td>
</tr>
<tr>
<td>Maximum Bus Capacitance</td>
<td>Not specified</td>
<td>Varies across signaling rate</td>
</tr>
<tr>
<td>Bus Pin Abs. Max Voltage</td>
<td>-7V to +12V</td>
<td>Often -9V to +14V</td>
</tr>
</tbody>
</table>

¹ Typically choose Z₀ close to 165Ω to reduce line reflections, as equivalent termination is approximately 171Ω (220Ω || [ 390Ω + 390Ω ]).
RS-232
RS-232

- RS-232 is a single ended communication standard for serial communication. It conveys data over a simple unterminated multi-conductor cable. The original specification was designed to connect the serial port of a computer to a modem or other peripheral devices.
- The current version of the standard is TIA-232-F issued in 1997.

Easy to implement, long distance communication, no software necessary, reliable, low noise sensitivity

Slow, no power transmission, 1-to-1 transmission only, large connector
The image at left shows the transmission of a data word using the common “9600 – 8 – none – 1” UART format. This means that the baud rate is 9600 bps, the word length is 8 bits, no parity bit is used, and one start/stop bit is used:

- The start bit is indicated by a “0” bit (i.e., a positive voltage)
- Next, the data is transmitted LSB first. (This example shows transmission of 0100001, or the letter “A.”)
- The stop bit is indicated by a “1” bit (i.e., a negative voltage)
## RS-232 TIA standard

<table>
<thead>
<tr>
<th>Specifications</th>
<th>RS-232</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode of Operation</td>
<td>Single Ended</td>
</tr>
<tr>
<td>Number of Drivers and Receivers on Line</td>
<td>1 driver, 1 receiver</td>
</tr>
<tr>
<td>Maximum Cable Length</td>
<td>50ft (20k baud)</td>
</tr>
<tr>
<td>Maximum Data Rate</td>
<td>20kB/s</td>
</tr>
<tr>
<td>Maximum Voltage Applied to Driver Output</td>
<td>+/- 25V</td>
</tr>
<tr>
<td>Driver Output Voltage</td>
<td>+/-5V (min)</td>
</tr>
<tr>
<td></td>
<td>+/- 25V (max)</td>
</tr>
<tr>
<td>Output slew rate</td>
<td>30V/µs (max)</td>
</tr>
<tr>
<td>Receiver Input Voltage Range</td>
<td>+/- 25V (max)</td>
</tr>
<tr>
<td>Receiver Input Sensitivity</td>
<td>+/-3V</td>
</tr>
<tr>
<td>Receiver Input Resistance</td>
<td>3kΩ to 7kΩ</td>
</tr>
</tbody>
</table>
RS-232 requires only two wires as a bare minimum to transmit data (three wires to transmit and receive), but a full 9-pin connector can increase accuracy and speed of transmission through handshaking and control signals:

- **Pin 2 [RD]:** Receive data line
- **Pin 3 [TD]:** Transmit data line
- **Pin 4 [DTR]:** Data terminal ready line
- **Pin 5 [G]:** Signal ground
- **Pin 6 [DSR]:** Data set ready line
- **Pin 7 [RTS]:** Request to send line
- **Pin 8 [CTS]:** Clear to send line

(Pins 1 and 9 are not used)
Charge Pump

• Charge pump simplified schematic

- The charge pump operates in a discontinuous mode using an internal oscillator and a voltage regulator (set at 5.5V). If the output voltages are less than a magnitude of 5.5V, the charge pump is enabled. If the output voltages exceed a magnitude of 5.5V, the charge pump is disabled.

- The capacitor value ratios C1/C3 and C2/C4 govern the ripple at V+ and V-.
RS-232 Product Features

• Triple supply vs. single supply
  – Multiple generated supplies vs. charge pumps
• Data rate (standard: 20kbps, 30V/us; now: 1Mbps; future: 3Mbps)
  – Load capacitance: For 2500pF cable capacitance, as per IEA 232D for data rates less than 20k baud. For data rates greater than 20k baud, $C_{LOAD} = 1000pF$.
• ESD Level
  – Suffix E (15kV HBM, 8kV IEC contact, 15kV IEC air gap)
• Auto power down and auto power down plus
  – Voltage level based vs. time based
• VL (logic pin supply)
• Low voltage power supply (TRS3122E)
  – Voltage tripler vs. DC/DC converter
RS-232 Product Portfolio

Need more? See Parametric Search:

[RS-232]
Controller Area Network (CAN)
CAN and LIN in Automotive Applications

**CAN** is a main bus
- Differential
- Two-wire
- 1Mbps

**LIN** is a sub-bus
- Single-ended
- One-wire
- 20kbps
CAN (Controller Area Network)

- The CAN standard defines both a protocol and a physical layer for asynchronous, serial communication in multi-point bus applications.
- Each node consists of a CAN transceiver and CAN controller (MCU).
- A unique driver structure results in differential signaling levels different from RS-485, which accommodate additional features in protocol.
ISO11898 (Automotive) CAN Standard

- Part 1: Data link layer and physical signaling
- **Part 2: High-speed medium access**
- Part 3: Low-speed, fault-tolerant, medium-dependent interface
- Part 4: Time-triggered communication
- **Part 5: High-speed medium access unit with low-power mode**
  - Includes all requirements of Part 2, adds low power mode requirements.
- **Part 6: High-speed medium access unit selective wake (Partial networking)**
CAN-Based Additional Standards and Protocols

**HIGHER LAYER PROTOCOLS**

- **ARINC825**: Airborne systems
- **CANaerospace**: Aerospace
- **CAN Kingdom**: Fieldbus
- **CANopen**: Embedded control
- **DeviceNet**: Industrial automation
- **ISO11783**: Agriculture and forestry
- **MilCAN**: Military
- **NMEA2000**: Marine
- **SafetyBUS p**: Safety critical automation

**ADDITIONAL STANDARDS**

- **IEC 62132-4**
- **IEC 61967-4**
- **ISO 11452**
- **CISPR 22**
- **CISPR 25**
- **EN 55022**
- **FCC Part 15**
A CAN compliant driver must produce at least 1.5V across a 50Ω load.
CAN Bus Termination

• CAN is designed to be used with twisted pair cabling with 120-Ω characteristic impedance
• The network should be wired in a bus topology (limiting stubs as much as possible)
• The bus should be properly terminated at both ends with resistors that match the impedance of the network
  – This help to reduce signal integrity issues like reflections.
  – If nodes may be removed from the bus, care must be used on where to place the termination resistors
• Termination may be a single 120-Ω resistor at each end of the bus or “split termination” may used
  – Utilizing split termination improves signal integrity and electromagnetic emissions
  – Split termination eliminates fluctuations in the bus common mode voltage levels (not differential)
• Since CAN networks may be shorted to voltage sources the power ratings of the termination resistors should take
  into account the short circuit current protection of the CAN transceivers in the network.

![Standard Termination Diagram](image)

- Standard Termination
- Split Termination

- $R_{TERM} = 120\Omega$
- $C_{SPLIT} = 4.7\text{nF to } 100\text{nF}$
CAN Data Frame

- **Start of Frame** – A dominant bit begins the frame and initiates arbitration
- **Message Identifier** – 11 or 29 bit identifier used for arbitration priority
- **Control Field** – Specifies the length of the data to be transmitted
- **Data Field** - Data
- **CRC Sequence** – Cyclical recovery checking
- **ACK** – Acknowledges the CRC status of receiving nodes
- **End of Frame** – Marks the end of data and remote frames
**CAN Data Arbitration**

- The CAN physical layer allows for priority based arbitration based on the 11-bit identifier of each module.
  - 000 0000 0000 is the highest priority identifier.
  - 111 1111 1111 is the lowest priority identifier.

- During each bit of the identifier frame, each node will monitor the bus and compare the bus state with the state it is driving.
  - If the XCVR transmits a logic “1” and receives a “0”, it will stop transmitting.
  - The node will attempt to access the bus again after the next inter-frame spacing occurrence.
Loop Time in CAN

The maximum length and operating rate of a CAN bus depend on several factors:

- CAN controller IO (TXD/RXD) is negligible
- CAN Transceiver Loop Delay
- Cable length
- Other delays introduced by additional series components

Arbitration is the key to a CAN Network – Knowing the loop and round trip delay is critical

- Each Node must know the total round trip delay for sufficient sampling timing
- Each component in the system contributes to the total round trip delay
  - Round trip delay = 2 x \( T_{\text{PROPAB}} \) + Isolator Delay + Transceiver delay + Controller IO delay
- Faster loop times allow more propagation delay budget for isolation devices and/or longer cables without compromising data rate
- Specific parameters related to timing and synchronization can be setup in the CAN controller to accommodate propagation delays
CAN Bit Timing (at Controller)

- **SYNC_SEG**
  - Hard synchronization forces rising edge in first segment

- **PROP_SEG**
  - Compensates for propagation delays

- **PHASE_SEG1**
  - SEG 1 may be lengthened or SEG 2 may be shortened for resynchronization

- **PHASE_SEG2**

Nominal Bit Length (Unit Interval)

Sample Point
CAN Node Configuration Example – Car

Each node has a host processor, CAN controller and transceiver.

V_{\text{Battery}}

V_{\text{IN}}

V_{\text{3.3V_REG}}

V_{\text{OUT}}

V_{\text{IN}}

V_{\text{5.0V_REG}}

V_{\text{OUT}}

V_{\text{CC}}

V_{\text{CAN}}

V_{\text{RXD}}

V_{\text{TXD}}

V_{\text{S/STB}}

V_{\text{CANH}}

V_{\text{CANL}}

GND

V_{\text{OUT}}

V_{\text{OUT}}

V_{\text{OUT}}

Optional^{1}: ESD or Transient Protection on CAN bus outside CAN Transceiver Spec Range

Optional^{2}: SPLIT Node Termination (Typically 60Ω, 60Ω and 4.7nF)

Optional^{3}: SPLIT Node Termination, with common mode bus stabilization output

Optional^{4}: MCUs without internal pull-up resistors need an external pull-up for fast data-rates

Texas Instruments
CAN With Flexible Data Rate (CAN FD)

• Defines timing specifications at two higher data rates: 2 Mbps and 5 Mbps
• Two key differences between CAN and CAN FD protocol:
  – Data Rate: CAN FD frames have the option to have separate data rates for the arbitration portions and the data portions. Arbitration portions adhere to CAN, while data may be transmitted at a higher data rate.
  – Data Field Length: CAN FD frames allow for data fields up to 64 bytes, 8 times more than standard CAN.
Benefits of CAN FD

INCREASED BANDWIDTH
• The 8 byte limitation of the CAN data rate is constraining for some applications, and requires multiple messages to send the requisite data.

LOWER RELATIVE COST & COMPLEXITY
• Small incremental cost to increase bandwidth.
• Less complexity than implementing major network changes, such as FlexRay or Ethernet.

FAST FLASH PROGRAMMING
• CAN FD allows for end-of-line flash programming of modules and ECUs, reducing manufacturing costs.
**CAN 8-Pin Standard Pinout**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TXD</td>
<td>CAN transmit data input</td>
</tr>
<tr>
<td>RXD</td>
<td>CAN receive data output</td>
</tr>
<tr>
<td>CANH</td>
<td>High level CAN bus input/output</td>
</tr>
<tr>
<td>CANL</td>
<td>Low level CAN bus input/output</td>
</tr>
</tbody>
</table>

**Additional 8-Pin CAN Features**

- **STB (Standby Mode)**
  - In standby mode, the driver is disabled while the receiver is in a low-power wake-up mode.

- **S (Silent Mode)**
  - In silent mode, the receiver is enabled and mirrors the bus, while the driver is disabled.

- **V\textsubscript{IO}**
  - The V\textsubscript{IO} pin provides a separate supply voltage for the transceiver I/O pins, TXD and RXD.

- **SPLIT**
  - The SPLIT pin provides a VCC/2 output to stabilize the bus common-mode voltage for applications utilizing split termination.

- **FAULT**
  - For use in redundant bus topologies, the FAULT pin issues a fault signal when a dominant time-out (DTO) occurs at the receiver.
14-pin CAN transceivers keep the same base functionality as 8-pin transceivers, but add several additional functions (such as the ability to run off of a battery in low-power mode and to signal the rest of the system to start-up based on wake-up commands issued via the CAN bus).
Local Interconnect Network (LIN)
Features of LIN

- Operating voltage of 12V (for automotive)
- Single wire communication bus
- Guaranteed latency times (for system debugging)
- Variable data frame length (overhead control)
- Data checksum and error detection
- Detection of defective nodes
- Low cost silicon implementation (UART)
- Allows for hierarchical network design

Example of LIN Bus
LIN (Local Interconnect Network)

- LIN is a broadcasting, serial, one-wire interface, typically implemented as a sub-bus of a CAN network.
- Allows automotive manufacturers to reduce cost by offloading low-speed (<20kbps), non-safety critical functions from a two-wire CAN bus to a one-wire bus.
- One master coordinates communication between up to 16 slaves.
LIN Physical Layer

\[ V_{LIN-REC} = V_{SUP} - V_F \]

Dominant when \( V_{LIN} \leq 0.4 \times V_{SUP} \)

Recessive when \( V_{LIN} \geq 0.6 \times V_{SUP} \)

\[ V_{LIN-DOM} = V_{R-ON} \]
The LIN protocol specification defines:

- All types of frames that may be sent on the LIN bus
- The fields that make up each type of frame
- The order of the bits in each field

The physical layer specification is unchanged for specification versions 1.3 through 2.2A
8-Pin LIN Transceiver

- RXD – Pin that reports the current state of the LIN bus voltage, dominant (0) or recessive (1)
- EN – Input to toggle device between Normal and Sleep mode
- NWake – Input that places the device in intermediate power saving Standby mode
- TXD – Pin that controls the state of the LIN output pin
- INH – Output to control external regulator with an inhibit input (turns off regulator in sleep mode)
- Vsup – Device supply connected to battery
- LIN – Pin that is connected to the LIN bus
I2C
• I2C is a two-line (clock and data) interface that allows for bidirectional bus communication between master and slave device
• Open-drain signaling is used, and signal levels scale with Vcc
I²C Physical Layer

- **V_SCL(t)**
  - **V_CC**
  - **V_IH**
  - **V_IL**
  - **V_OL**
  - **V_ON=LOW**

- **Capacitor (C_BUS)**

- **Resistors (R_PULLUP)**

- **i²C Control**

- **MASTER**
  - SDA
  - SCL

- **SLAVE**
  - SDA
  - SCL

- **VCC**
I²C Physical Layer

- **ON=LOW**
- **OFF=HIGH**

- **VCC**
- **VIH**
- **VIL**
- **VOL**

- **SDA**
- **SCL**

- **RPULLUP**

- **MASTER**
- **SLAVE**

- **C_BUS**

- **t**
I²C Physical Layer

![I²C Physical Layer Diagram]

- **Masters**
  - SDA
  - SCL
  - Pull-up resistor (RPULLUP)

- **Slaves**
  - SDA
  - SCL
  - Pull-up resistor (RPULLUP)

- **Signals**
  - \( V_{CC} \)
  - \( V_{IL} \)
  - \( V_{OL} \)
  - \( V_{HH} \)
  - \( V_{OL} \)

- **Rise Time**
  - \( t_{RISE} \)
  - \( (\text{ns}) \)

- **BusCap**
  - \( C_{BUS} \)
### Features of I²C Interface

<table>
<thead>
<tr>
<th></th>
<th>Standard Mode</th>
<th>Fast Mode</th>
<th>Fast Mode Plus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
<td>0 to 100 kHz</td>
<td>0 to 400 kHz</td>
<td>0 to 1,000 kHz</td>
</tr>
<tr>
<td>$C_{\text{BUS MAX}}$</td>
<td>400 pF</td>
<td>400 pF</td>
<td>550 pF</td>
</tr>
<tr>
<td>$t_{\text{RISE MAX}}$</td>
<td>1,000 ns</td>
<td>300 ns</td>
<td>120 ns</td>
</tr>
</tbody>
</table>
Byte Format

8-bit (byte) format

MSB - - - - - - - LSB

MASTER Controlled

SLAVE Controlled

S=START Condition

A=ACK Acknowledge
IO Link
IO-Link

• IO-Link is a serial, bi-directional, point-to-point protocol and interface standard for sensors and actuators in factory automation applications.
• Standardized in IEC 61131-9, and is the first worldwide standard for communication with sensors and actuators.
• Standardized cabling and connectors provide power (2 wires) and data (1 wire).
• Extends existing implementations by providing process data, as well as parameterization, diagnostic information, and configuration programming.
IO-Link specifies three speeds of operation: **230.4kbps**, **38.4kbps**, or **4.8kbps**.

In v1.1, the master port must support the highest rate of operation, but the device node may support only one. Communication is initiated by the master node transmitting test sequences to adapt to the data rate of the device node.

For failed transmission, the frame is repeated two additional times. On the third failure, the master signals a failure to the higher-level controller.
• IO-Link nodes may operate a back-up, standard I/O (SIO) to ensure backward compatibility with non-IO-Link compatible binary sensors, specified in IEC 61131-2.

• Many modern sensor nodes use IO-Link for programming of the device on manufacturing, but operate only in SIO mode for the life of the device.

• A change in operating mode is initiated by a Wake-Up Request from the master.

• The device node returns to SIO mode after a “Fallback” command from the master.
### IO-Link Standard Connectors

Unshielded 3- or 5-conductor cables are used to connect the slave nodes to the master via a standard M12 plug connector.

Cables may be up to 20 meters in length.

<table>
<thead>
<tr>
<th>Class A</th>
<th>Pin</th>
<th>Class B</th>
</tr>
</thead>
<tbody>
<tr>
<td>L+</td>
<td>1</td>
<td>L+</td>
</tr>
<tr>
<td>Not specified</td>
<td>2</td>
<td>V+₁</td>
</tr>
<tr>
<td>L-</td>
<td>3</td>
<td>L-</td>
</tr>
<tr>
<td>C/Q</td>
<td>4</td>
<td>C/Q</td>
</tr>
<tr>
<td>Not specified</td>
<td>5</td>
<td>V+₂</td>
</tr>
</tbody>
</table>
Thank You!