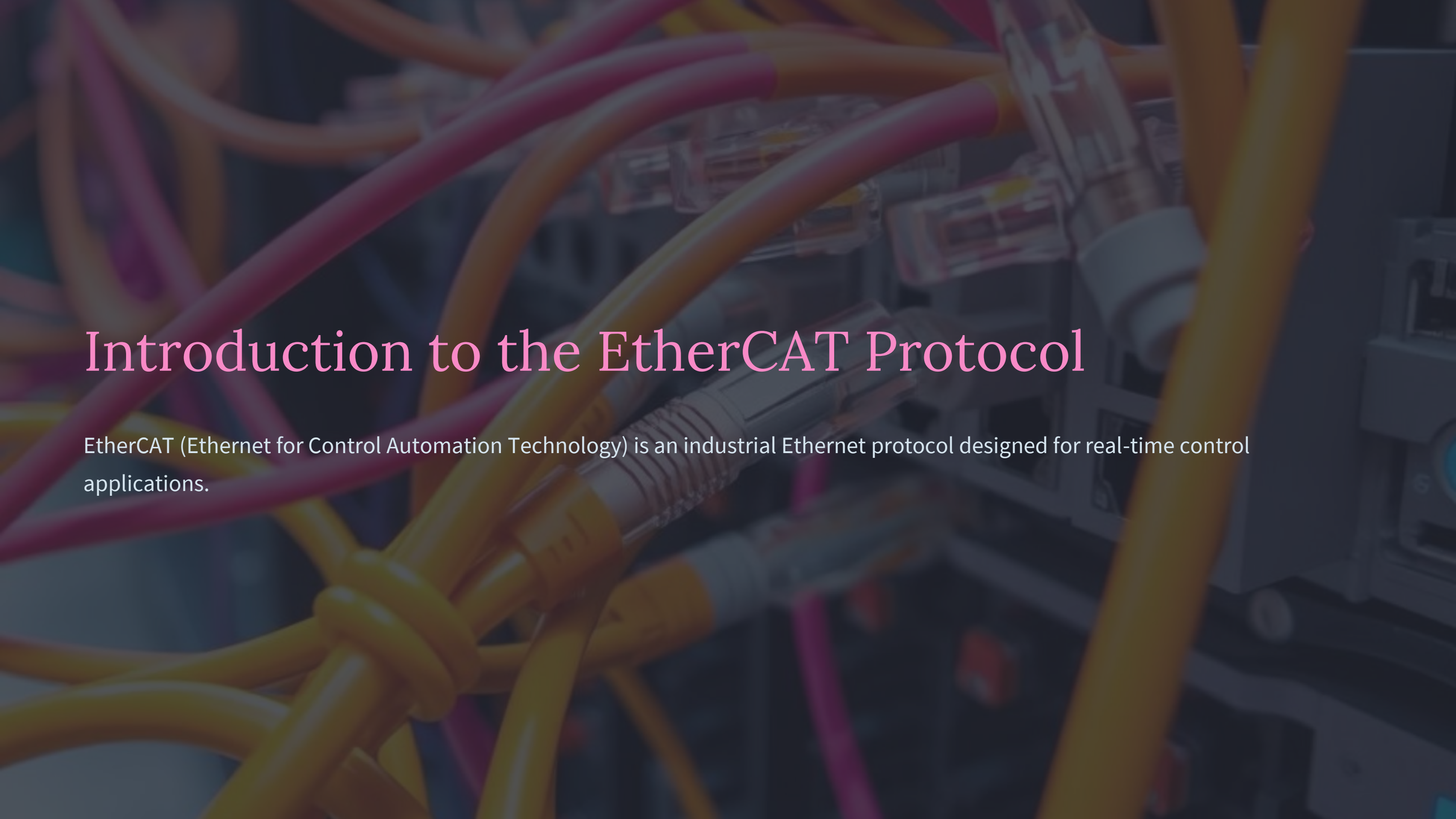
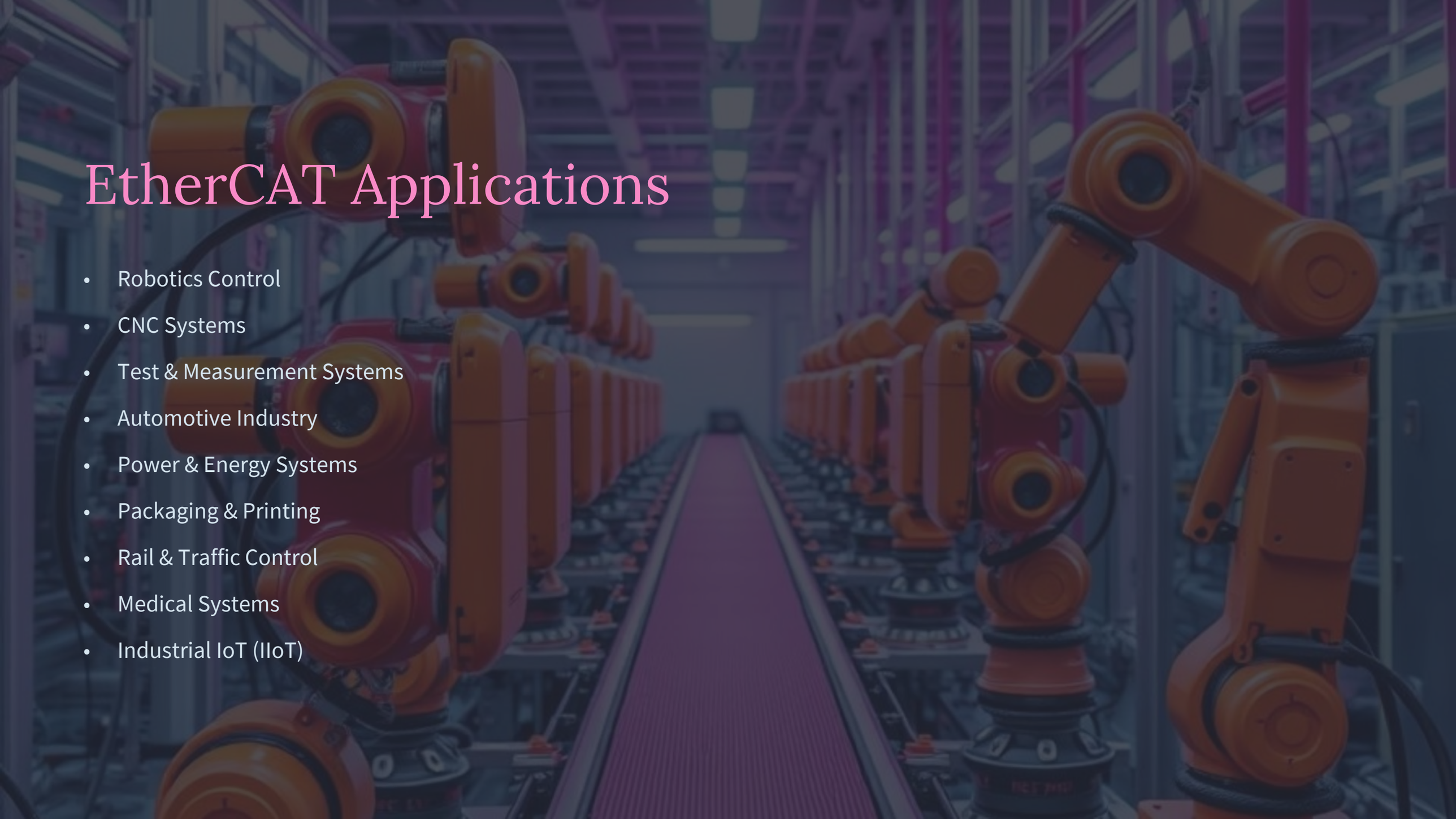


Introduction to the EtherCAT Protocol

EtherCAT (Ethernet for Control Automation Technology) is an industrial Ethernet protocol designed for real-time control applications.



EtherCAT Applications

- Robotics Control
- CNC Systems
- Test & Measurement Systems
- Automotive Industry
- Power & Energy Systems
- Packaging & Printing
- Rail & Traffic Control
- Medical Systems
- Industrial IoT (IIoT)

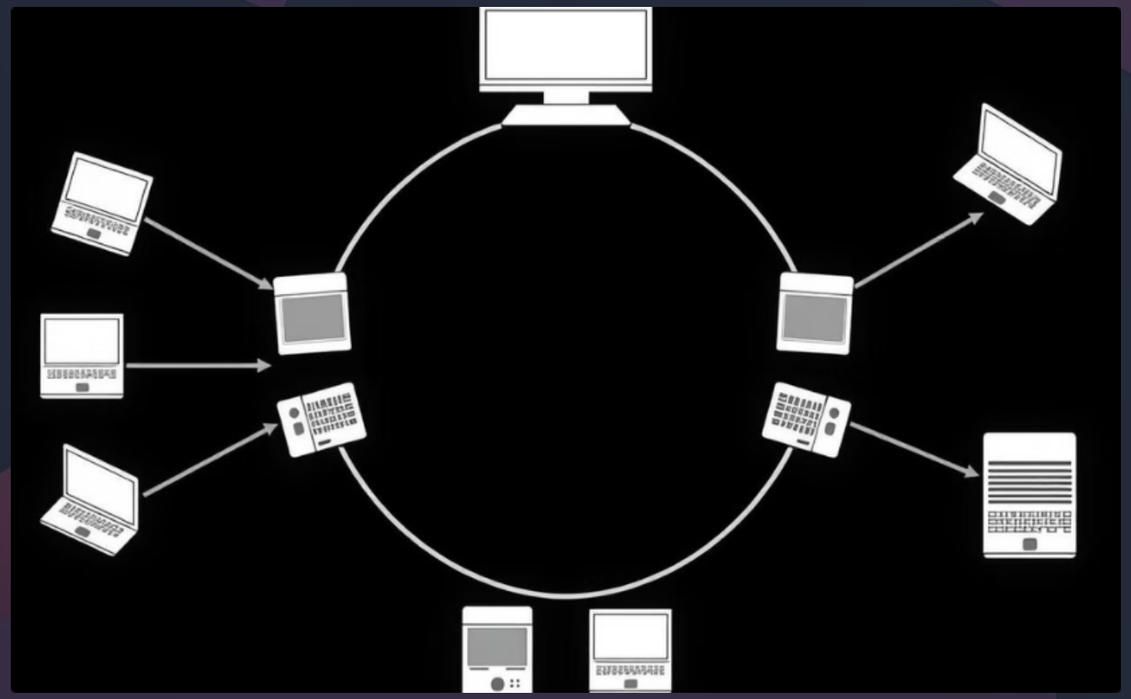
Reasons for the Development of the EtherCAT Protocol

1. Need for High Speed and Low Latency in Industrial Automation
2. Increasing Complexity and Number of Connected Devices in Industrial Systems
3. Need for High Precision in Control System Synchronization
4. Reducing Hardware Costs and Optimizing Resource Utilization
5. Compatibility with Standard Ethernet and the Need for System Integration
6. Need to Improve Security and Reduce Communication Errors

Physical Layer Connections and Circuits

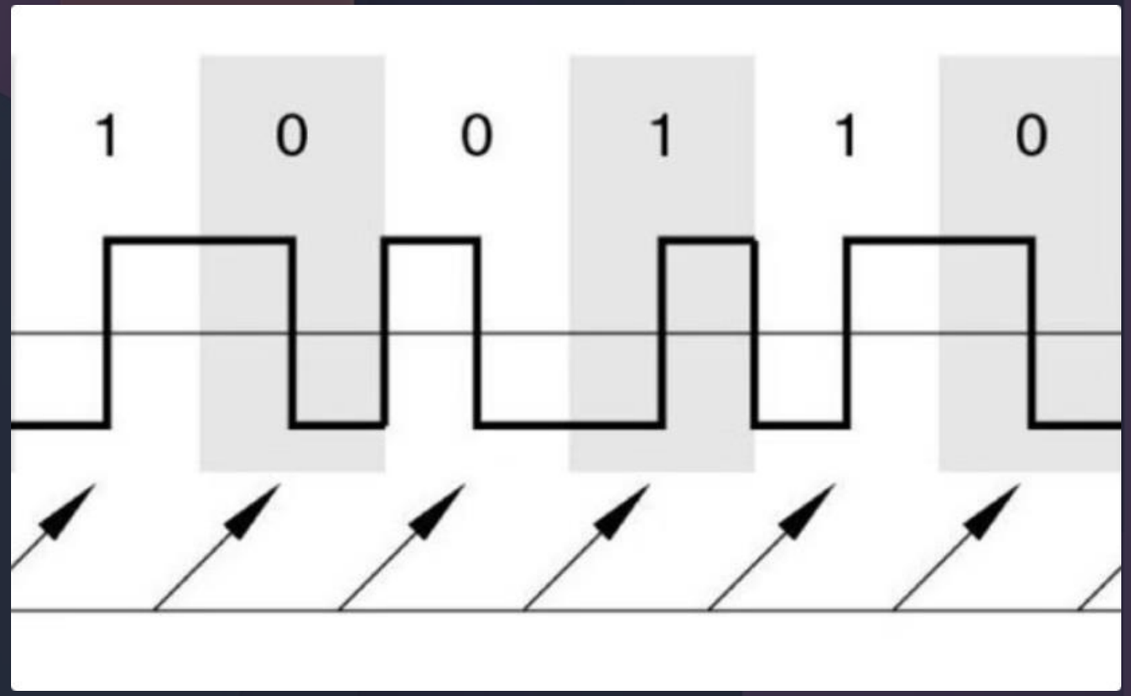
- EtherCAT uses the IEEE 802.3 Ethernet standard and differential signaling for reliable communication.
- It transmits data through two twisted pairs of wires with opposite electrical potentials for noise reduction.
- Connections include Ethernet (RJ45 or industrial M8/M12), power supply (24V DC), and differential signals (TX+/TX- and RX+/RX-).
- Optional connections include digital/analog I/O, fiber optic cable, and additional ports for redundancy.

Connection Type and Encoding



Connection Type

EtherCAT is a serial protocol that transmits data in a ring or daisy-chain structure through Ethernet cable



Encoding Type

EtherCAT uses Manchester encoding, which is defined in the 100BASE-TX Ethernet standard

How to Connect Devices

Ability to Connect Multiple Devices

The EtherCAT protocol allows you to connect multiple different devices or hardware modules. Using this protocol, you can connect various types of devices to a single network.

This feature is particularly useful for complex industrial automation systems with a large number of devices.

Collision Management

Collision management in EtherCAT is done using a precise and efficient mechanism. In this protocol, there is a Master device responsible for controlling and managing the network.

The Master is responsible for the precise timing and Real-Time communication loop, as well as managing the data position structure within the frame. This approach helps prevent collisions and ensures efficient data transfer.

Data Management

Addressing in EtherCAT

- Auto-Increment Addressing
- Configured Addressing
- Ethernet MAC Addressing

Routing in EtherCAT

- Chain structure and direct frame processing
- Loopback mechanism if needed
- EtherCAT does not require complex routing

Data Flow Management

1

1. Initial Configuration

System Configuration Tool generates network info.

2

2. Information Processing

ENI sent to XML Parser.

3

3. Command Execution

Master Driver processes and prepares data.

4

4. Network Transmission

Master Driver transmits via Ethernet MAC.

Error Detection and Correction Approach

1 **Physical Layer**

Errors caused by noise, interruption or signal quality degradation

2 **Data Link Layer**

Strong protective mechanisms for detecting and correcting data errors

3 **Network Layer**

Monitoring network status and detecting unwanted changes

4 **Transport Layer**

Detecting communication problems and supporting redundant ring topology

5 **Application Layer**

Configuration, software or issues in the control program

Types of Messages in EtherCAT

Process Data Messages (PDM)

Used for sending and receiving I/O data from slave devices. These messages are optimized for low latency and Real-Time control.

Service Data Messages (SDM)

Used for configuring, reading, or writing parameters in slave devices using the SDO (Service Data Object) model. Primarily used for initial setup and parameter changes, with less stringent response time requirements.

Mailbox Messages

Enable asynchronous communication and the transmission of additional data, including CAN over EtherCAT (CoE) and Ethernet over EtherCAT (EoE). These messages facilitate alerts, log transmissions, and more complex settings.

Error Detection and Correction Approach

Error Detection

EtherCAT is designed to minimize errors through robust detection mechanisms.

- Master/Slave architecture limits error potential
- CRC checksum verifies data integrity
- Working Counter (WKC) tracks data packet sequence

Error Correction

Unlike TCP/IP, EtherCAT relies on immediate error detection and fast retransmission.

- Master automatically re-sends faulty packets
- Redundant Masters and communication paths enhance reliability
- Bit error correction prevents data frame corruption

Resources

- EtherCAT.org - Why Use EtherCAT
- EtherCAT.org - Technology
- Dewesoft - What is EtherCAT Protocol
- EtherCAT.org - Ethernet Introduction
- Wikipedia - Ethernet Physical Layer
- EtherCAT.org - Diagnosis for Users

Thanks!

Do you have any question?

Presenter: Mohammadfarhan Bahrami

Contact me: mmdfarhanbahrami@gmail.com

Course Instructor: Dr. Foshati