EtherCAT Introduction

Texas Instruments

EtherCAT Introduction

- Ethernet for Control Automation Technology (EtherCAT) is an Ethernet-based fieldbus system
  - Invented by Beckhoff Automation™ in 2003
  - Beckhoff created the EtherCAT Technology Group (ETG) in 2004 to promote the protocol
  - ETG holds the rights to EtherCAT
- Open Technology covered under international standards (IEC61158,61784,61800,& ISO 15745)
- Multiple vendor implementations of the EtherCAT slave controller

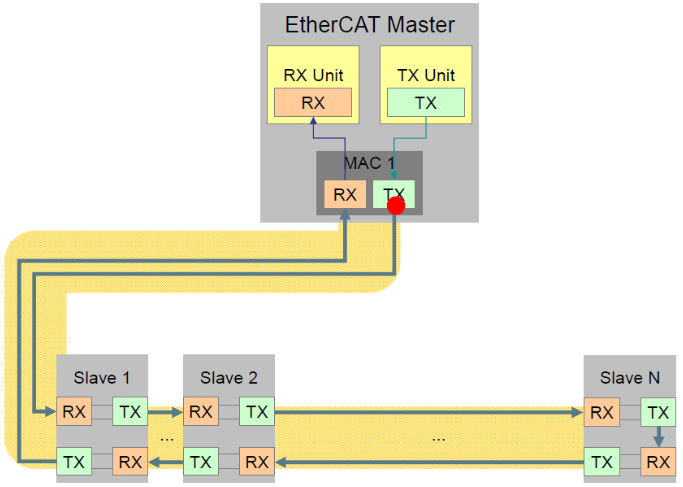
Beckhoff and EtherCAT Technology Group (ETG)

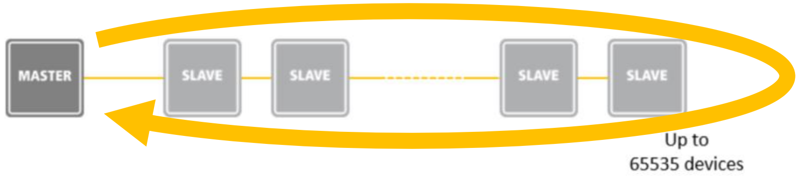
	Beckhoff	ETG
Who are they?	Inventor of EtherCAT	An industrial EtherCAT user organization (required to join as an EtherCAT developer)
What do they do?	• Offer EtherCAT hardware, FPGAs, etc • Developer of the EtherCAT Slave Stack code • Developer of the TwinCAT, EtherCAT SSC and CTT applications	• Assign and organize EtherCAT vendor-IDs • Offer training, developer forums, Plug Fests, etc
What do they provide?	• Downloads for TwinCAT (EtherCAT master) • License for using CTT	• Downloads for SSC and CTT • EtherCAT specifications
	https://beckhoff.com/	https://www.ethercat.org/

SSC = Slave Stack Code Tool **CTT** = Conformance Test Tool

EtherCAT Introduction: What is EtherCAT?

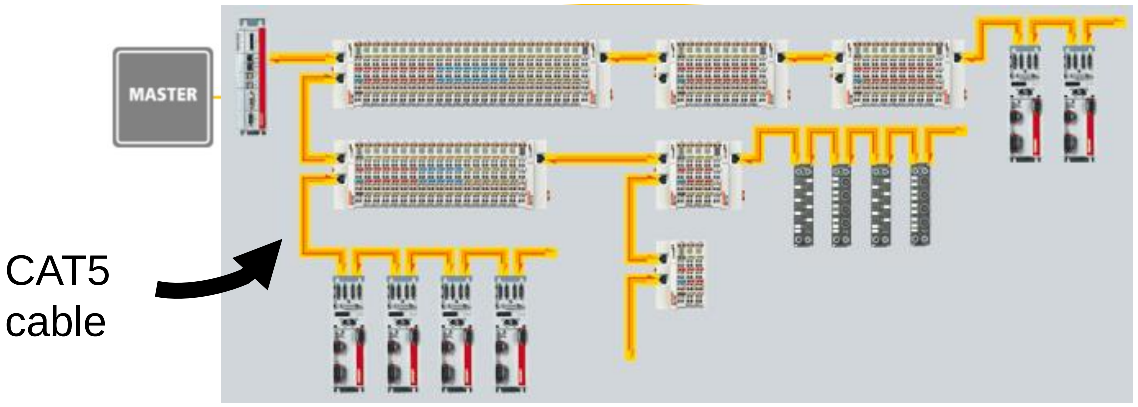
- Involves master and slave(s) setup where slave nodes are physically connected daisy-chain style but logically operate on a loop
- EtherCAT specializes in precise, low jitter synchronization across slave nodes (≤ 1 µs)
- Each slave processes message data "on the fly" as the frame passes from one node to the next
- Uses IEEE 802.3 Ethernet physical layer and standard Ethernet frames
- Can carry other protocols over EtherCAT (i.e. CANopen)





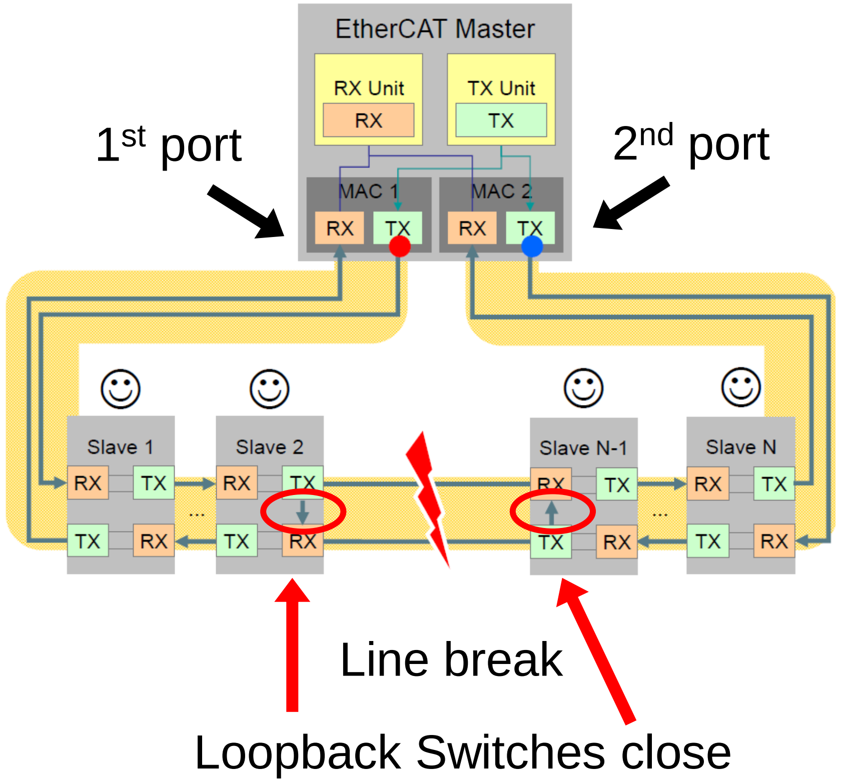
Data Flow

EtherCAT Physical Network Structure



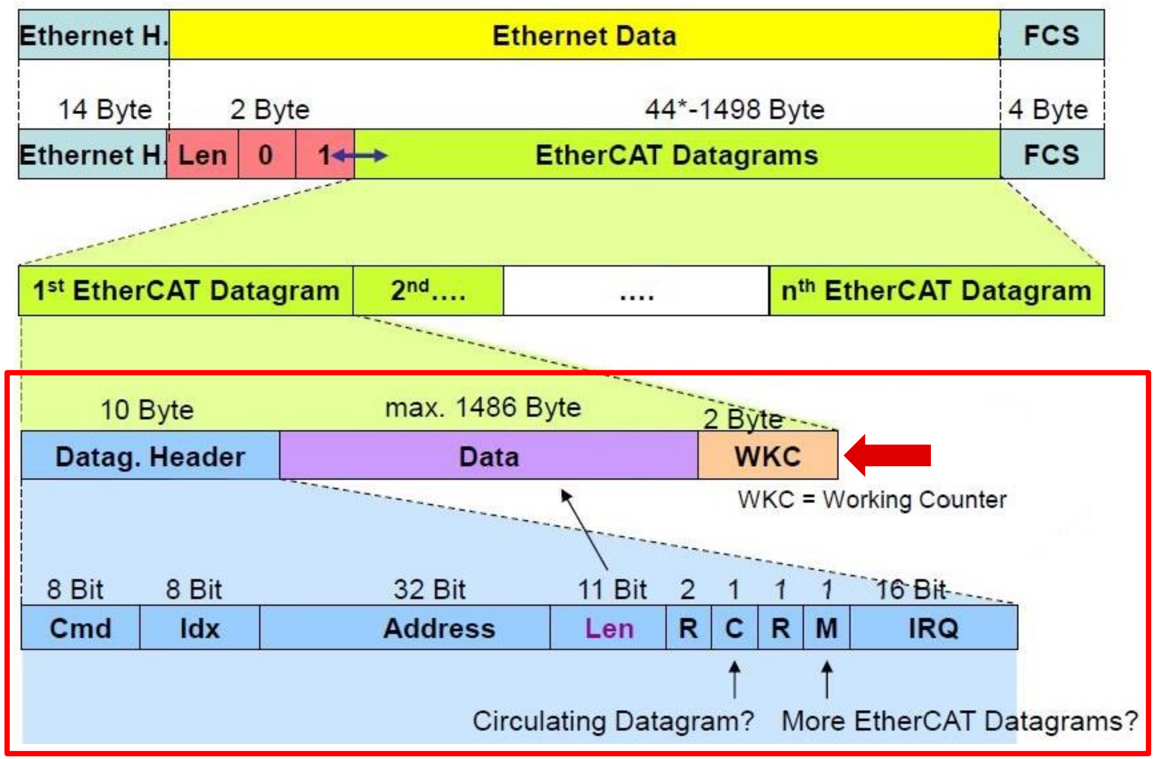
- Physically: Slave nodes can have multiple configurations (Line, Tree, Star, etc)
- Logically: Slave nodes are connected as a daisy-chain and operate on a loop
  - Duplex communication: CAT5 (Ethernet) cable has two differential pairs (outgoing pair and return path)
  - Only 1 Ethernet port needed at the master to connect to network for EtherCAT to run

EtherCAT Network Redundancy



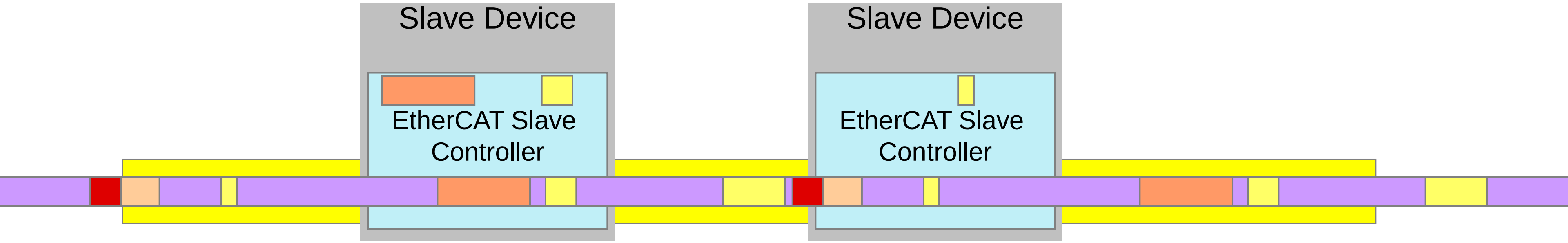
- Requires 2nd Master Ethernet port
- Redundant data on 2nd port
  - never used unless there's a line break
- **Loopback switches** in slave nodes close to maintain the loop in the event "downstream" nodes fail

EtherCAT Frame Structure



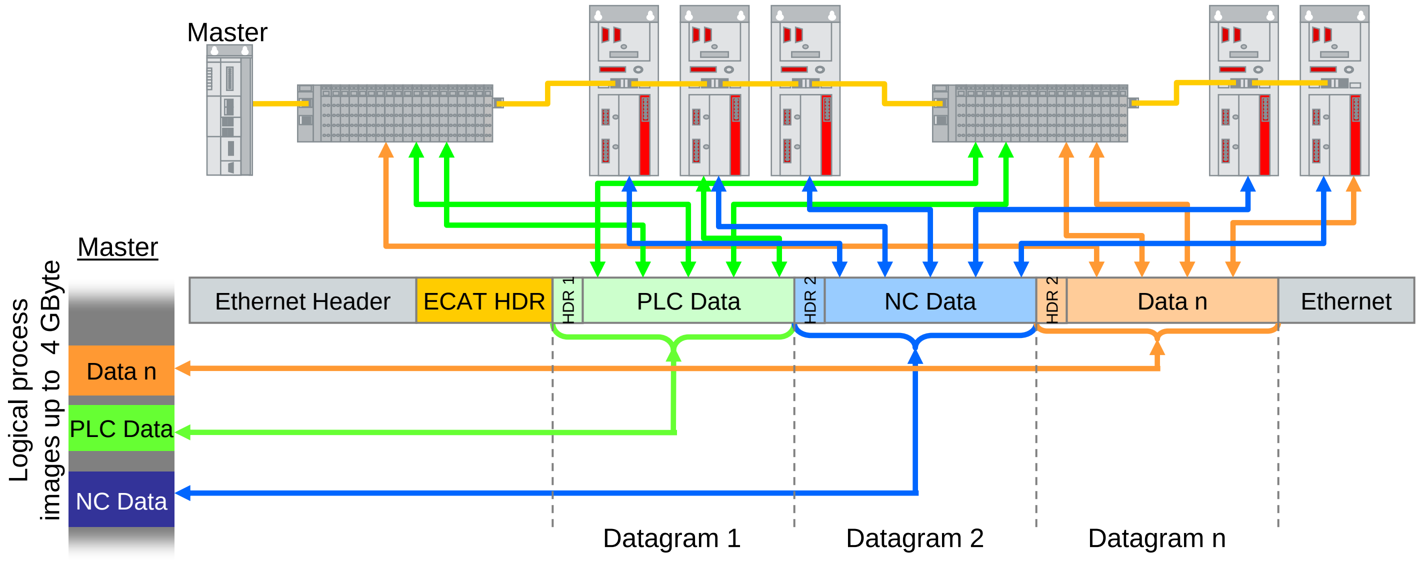
- 1 or more Datagrams per frame.
- Datagram consists of a Header, Data, and “Working Counter”
- Header contains:
  - Command, address, len, and various check bits
- The Working Counter (WKC) is the # of interactions contained in a given datagram.
  - WC is incremented appropriately by each slave. If the WKC in the frame returned to the master isn’t what’s expected, then there’s a problem somewhere in the network.

EtherCAT Communication “On-the-Fly”



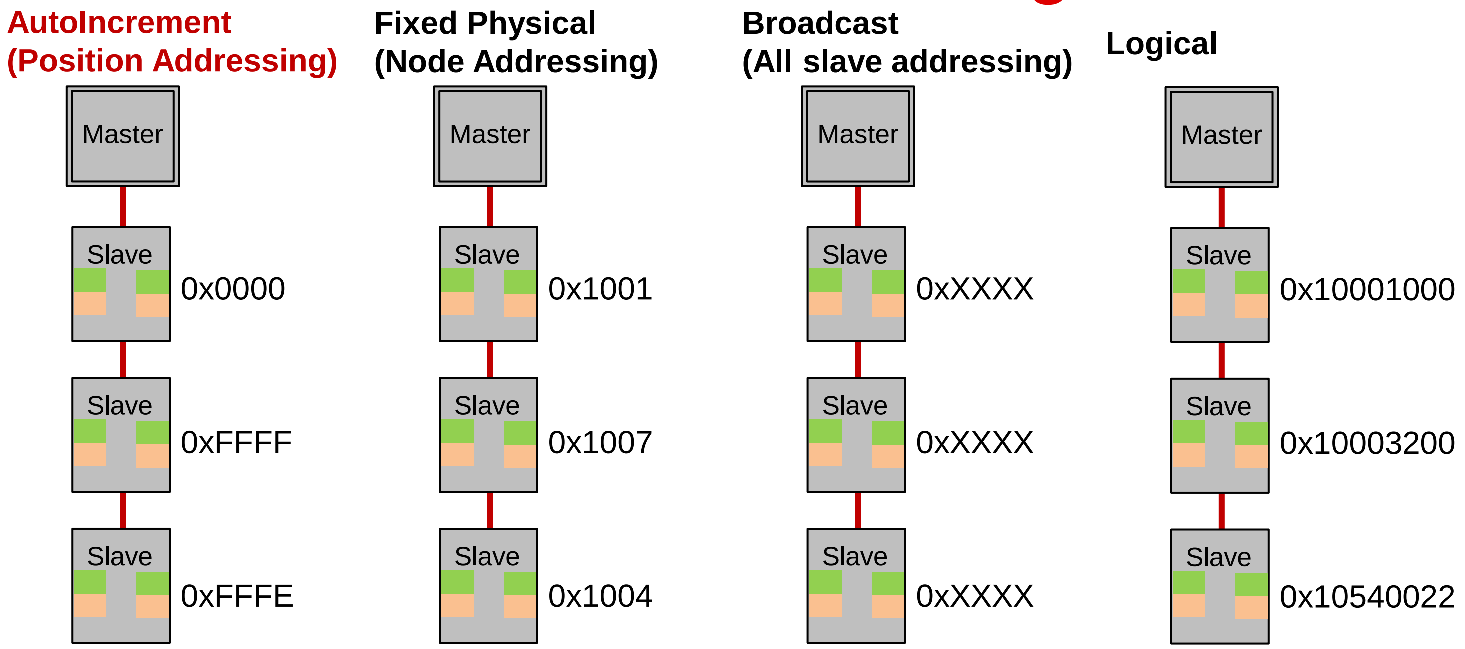
- EtherCAT Process data is extracted and inserted into the frame while the frame passes through the node at full speed
- EtherCAT slave **does not** require software interaction for data transmission or reception
- Each slave’s Datagram **size is almost unlimited**
  - from 1bit to 60kByte (if needed, using multiple frames)
- The structure of process data can change each cycle.
  - short datagram intervals for axis control updates
  - longer datagram intervals for I/O update
  - Asynchronous, event-triggered communication is also possible

EtherCAT Memory Mapping



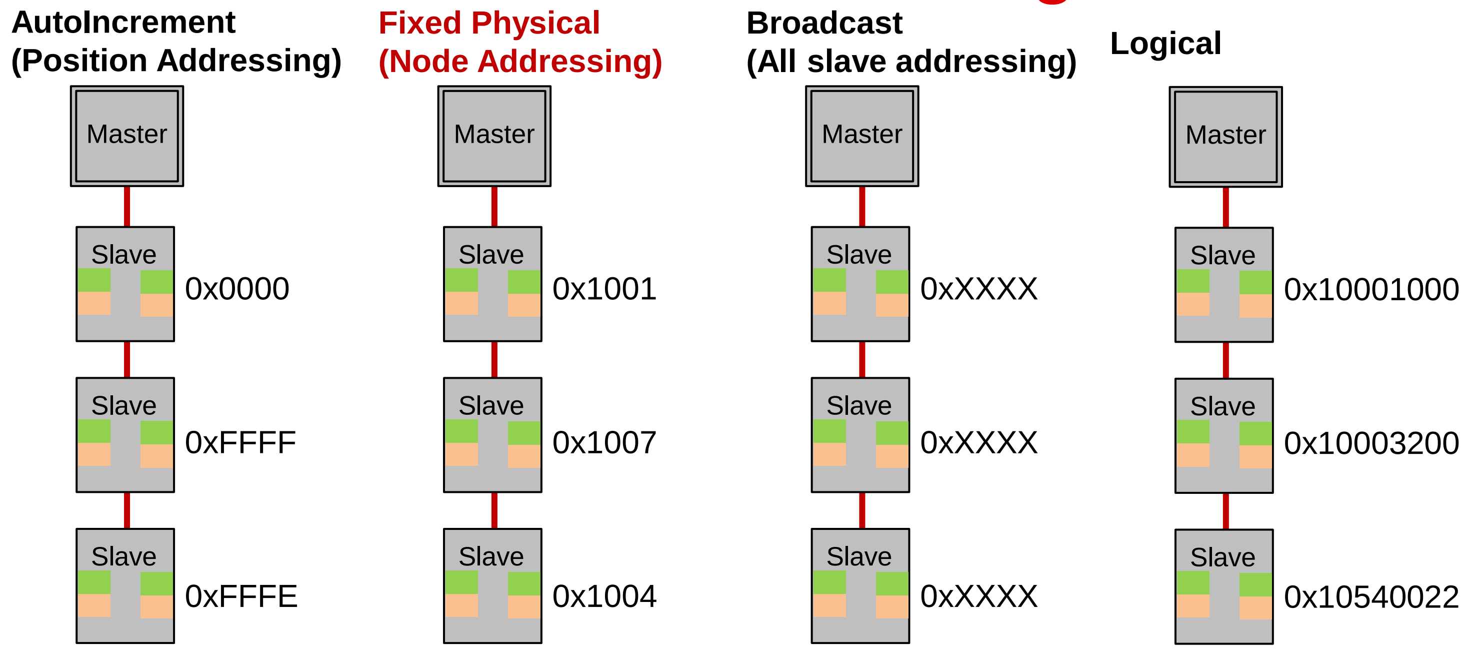
- Typically a single frame contains data for multiple slaves (if not the entire network)
- Each Slave has an FMMU (Fieldbus Memory Management unit) to extract data from the packet and map the logical addresses to the physical addresses in the ESC.
- Data is transmitted according to the application requirements: extremely fast, flexible and efficient

EtherCAT Slave Node Addressing



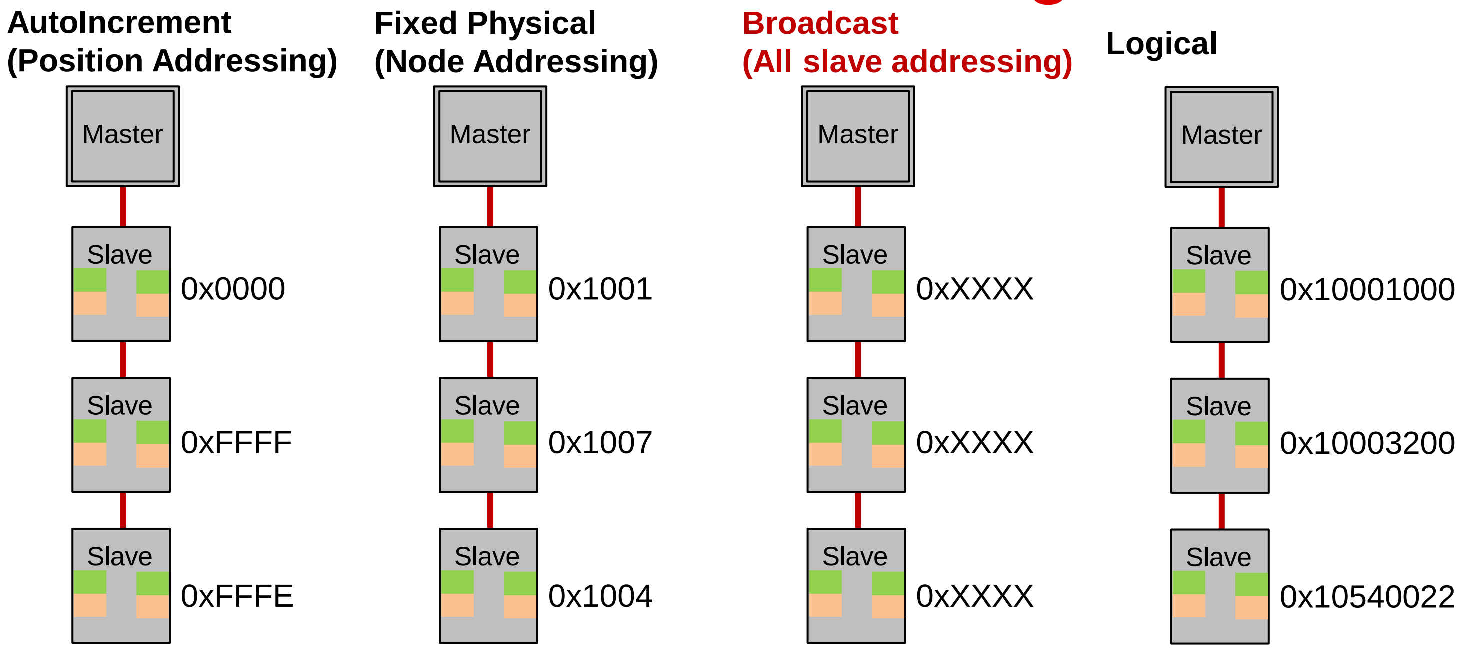
- AutoIncrement (Position Addressing)
  - Used typically only during start-up to scan the network
  - Position address of the addressed slave is stored as negative value
  - Each slave increments the address. The slave that reads this address as zero is addressed

EtherCAT Slave Node Addressing



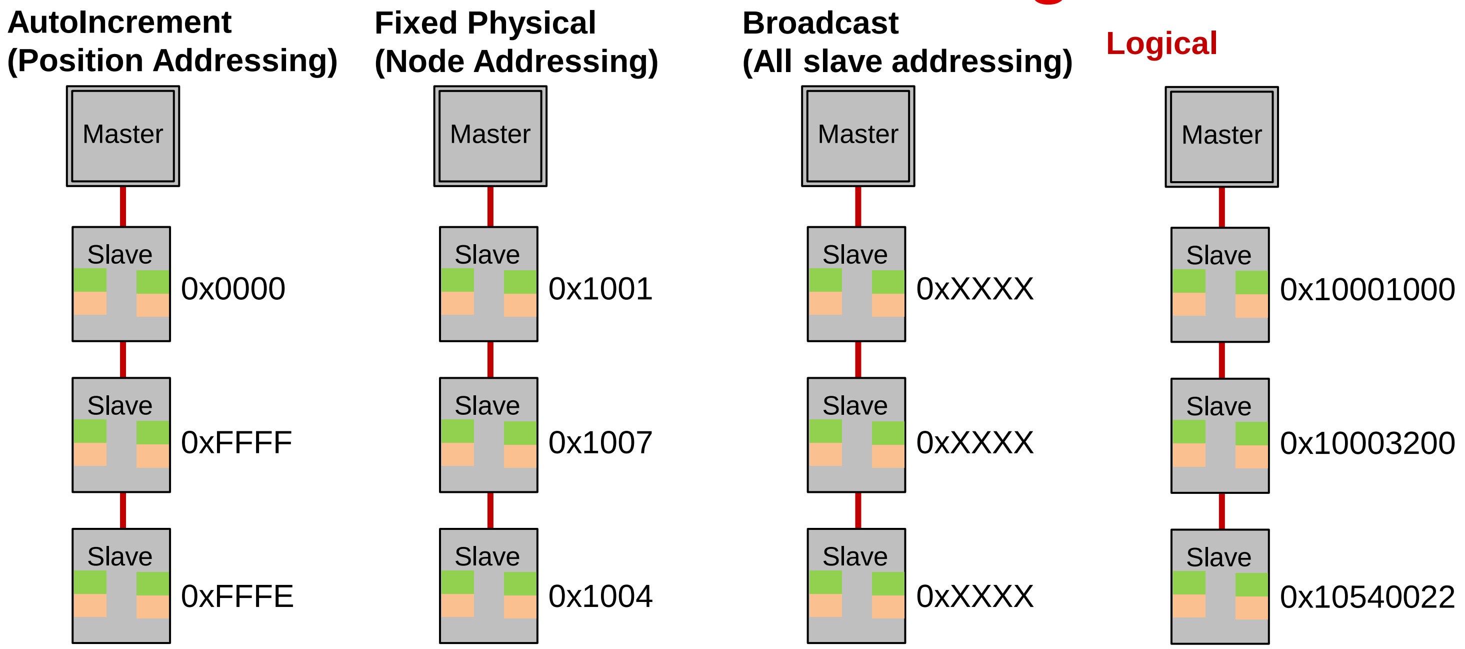
- Fixed Physical (Node Addressing)
  - Typically used for register access to individual slaves that have already been identified
  - The configured slave address is assigned by the master at start up and cannot be changed by the EtherCAT slave

EtherCAT Slave Node Addressing



- Broadcast Addressing
  - Used for initializing all slave devices
  - Addresses all slaves in the network

EtherCAT Slave Node Addressing



- Logical Addressing
  - Used to reduce unnecessary content in process data communication
  - All slaves read from and write to the same logical address range of the EtherCAT datagram
  - Each slave uses the FMMU to map data from the logical address to the local physical memory address

More Information on EtherCAT

EtherCAT® Technology Group

www.ethercat.org