Salman Azim
Student ID: 243850

Advanced Course in Factory Communication Systems

Assignment 2

EtherCAT Telegram Analysis

1. The EtherCAT Master and slaves are connected via standard Ethernet cable. The network configuration information is stored in the EtherCAT network information file (ENI), which is created based on the EtherCAT slave network information files. The EtherCAT configuration tool is used to generate the network configuration. There is hardware and software needed for EtherCAT Master. A network interface controller is the only hardware needed for the network. A real time runtime environment is needed to run the slaves in the network. The EtherCAT Slave devices mainly consists of 3 major parts. The network interface as the Physical layer, Slave controller and EEPROM as the Data link layer, host controller as the application layer. **[1]**

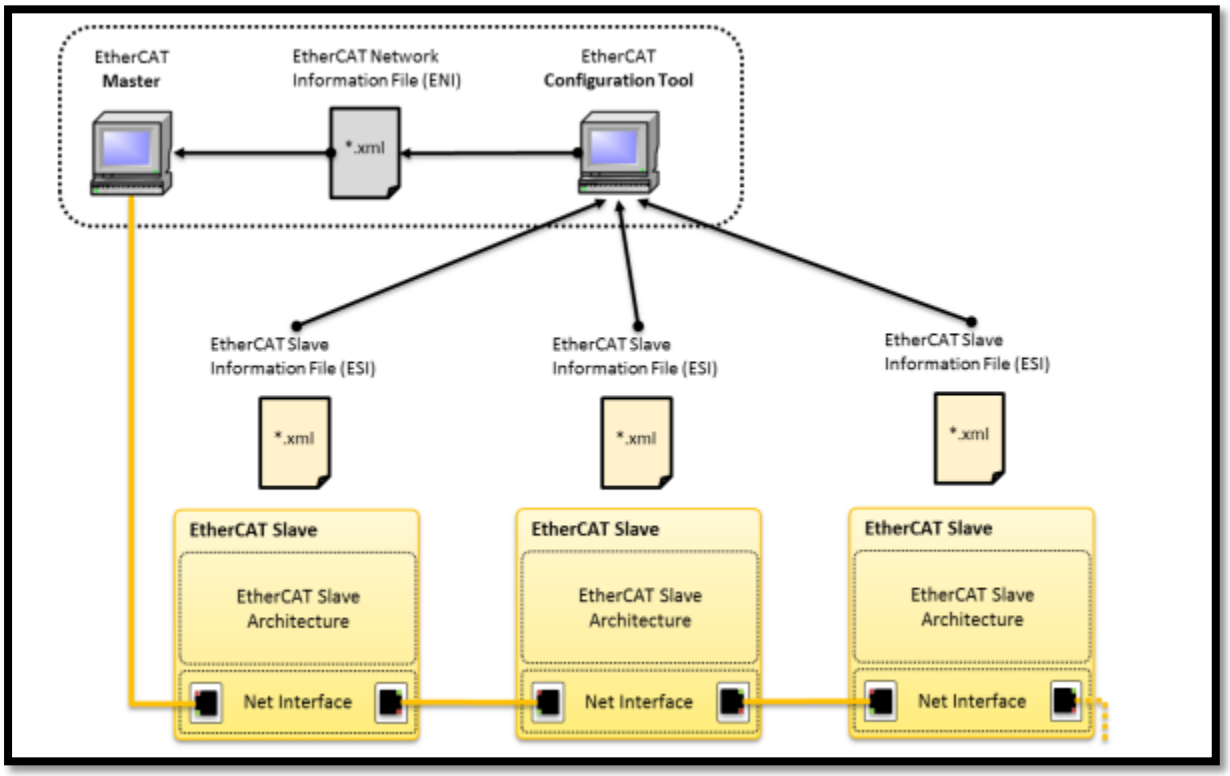


Fig 1: Hierarchical structure of EtherCAT network [1]

2. EtherCAT is by far the fastest and largest Industrial Ethernet technology capable of consisting unlimited nodes and also synchronizes in nanosecond accuracy. As the full duplex feature of Ethernet is used, the maximum data rate is over 90%, theoretically over 100 Mbits/sec. In EtherCAT network master is the only node that is allowed to send an

EtherCAT telegram, while all the other nodes (slaves) can forward the frames downstream. For this feature it filters unwanted delays and provides real time capabilities.

3. EtherCAT operates on three layers of the OSI model. It only uses the Application, Data link and Physical layer of the OSI model.

No.	OSI-Layer	EtherCAT	
7	Application Layer	http*, ftp*	Cyclic Data Exchange Mailbox Acyclic Data Access
6	Presentation Layer	-	-
5	Session Layer	-	-
4	Transport Layer	TCP*	-
3	Network Layer	IP*	-
2	Data Link Layer	Mailbox/Buffer Handling, Process Data Mapping, Extreme Fast Auto-Forwarder	
		Ethernet MAC	
1	Physical Layer	100BASE-TX, 100BASE-FX	
		*optional	

Fig 2: EtherCAT OSI model representation
[Source: https://en.wikipedia.org/wiki/EtherCAT]

4. In the Lab environment TwinCAT is working as a master node, and if the TwinCAT software is closed, the communication between the master and the salve is interrupted and the slaves go into a state of error. After certain time the master is not in OP mode. Then the slaves go in error mode. In the advanced settings of Master/Slave if the ReInit after communication error option is unchecked then the master/slave goes to error when any communication error happens.

5. From the Wireshark application EtherCAT packets were captured. Taking one frame the packet can be analyzed. It consists of several segments each defining some components of the protocol. The specification of the frame can be found in the message. The frame size, Interface ID, Encapsulation type, Epoch time, Time from previous frame, Frame number, Frame Length etc. are specified in this part. The MAC address of Source and Destination are specified in next segment. The EtherCAT telegram header and its length is provided in the message. And the EtherCAT datagram is specified in the last segment. There are 6 sub datagrams present. Each of the sub datagrams serves as particular memory area of the logical process image that can be up to 4 Gigabytes in size. There is also header, logical address and command type specified for sub datagrams. There are three types of command found. Logical Read, Logical Read/Write and Broadcast Read.

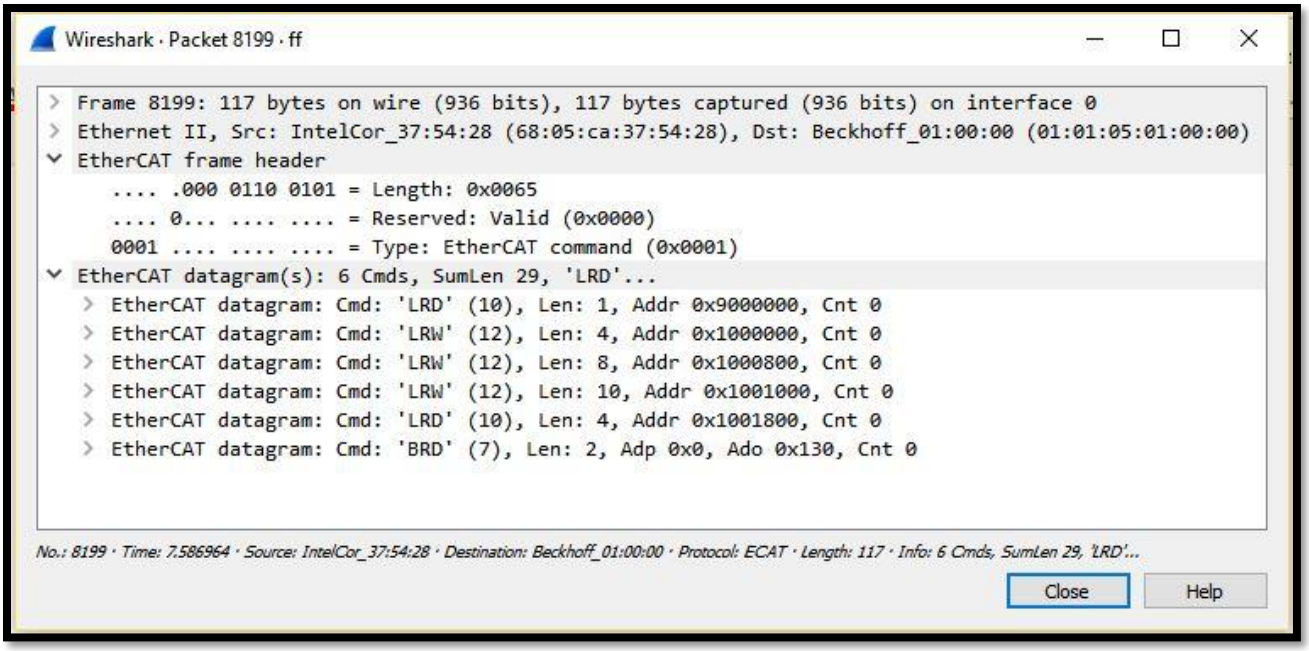
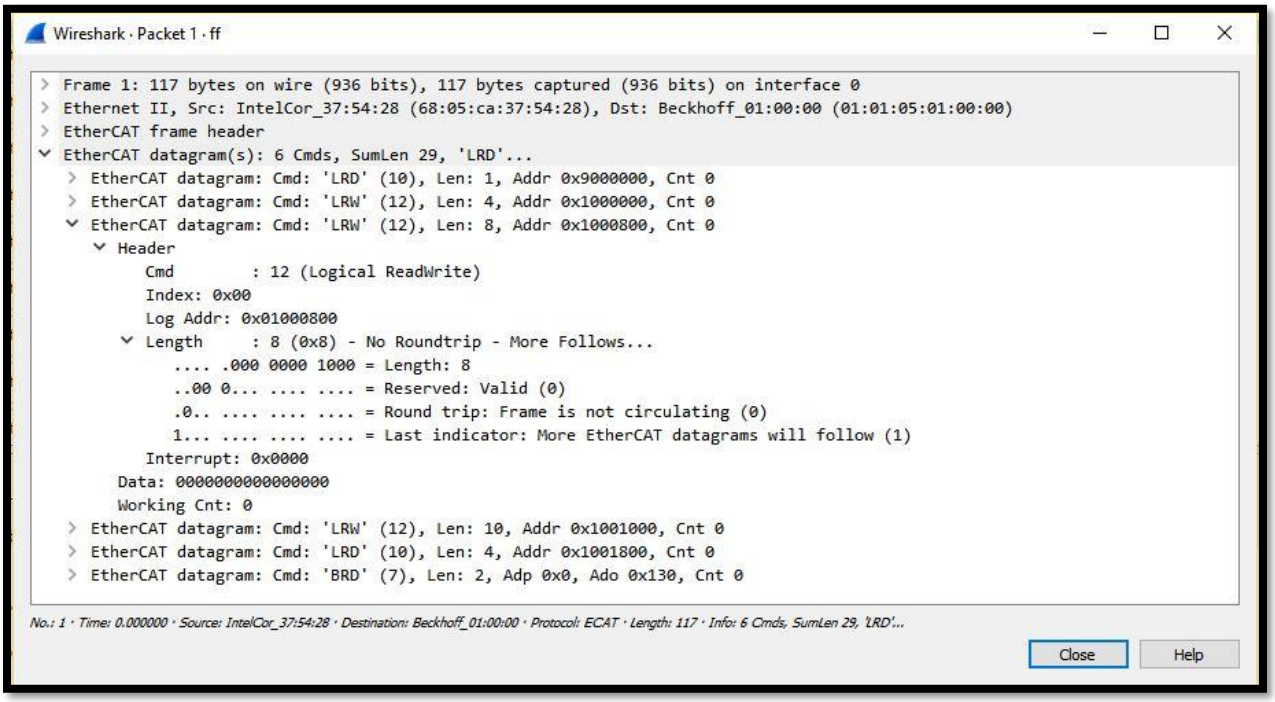


Fig 3: Captured EtherCAT message from Wireshark

6. **a.** The EtherCAT datagram changes when we write a data on the channel of CANopen slave device. When the data was not written the data field shown was 0. After writing there was some data appearing on the packet of the EtherCAT. The change was also observed physically on the specific channel of the Slave device.



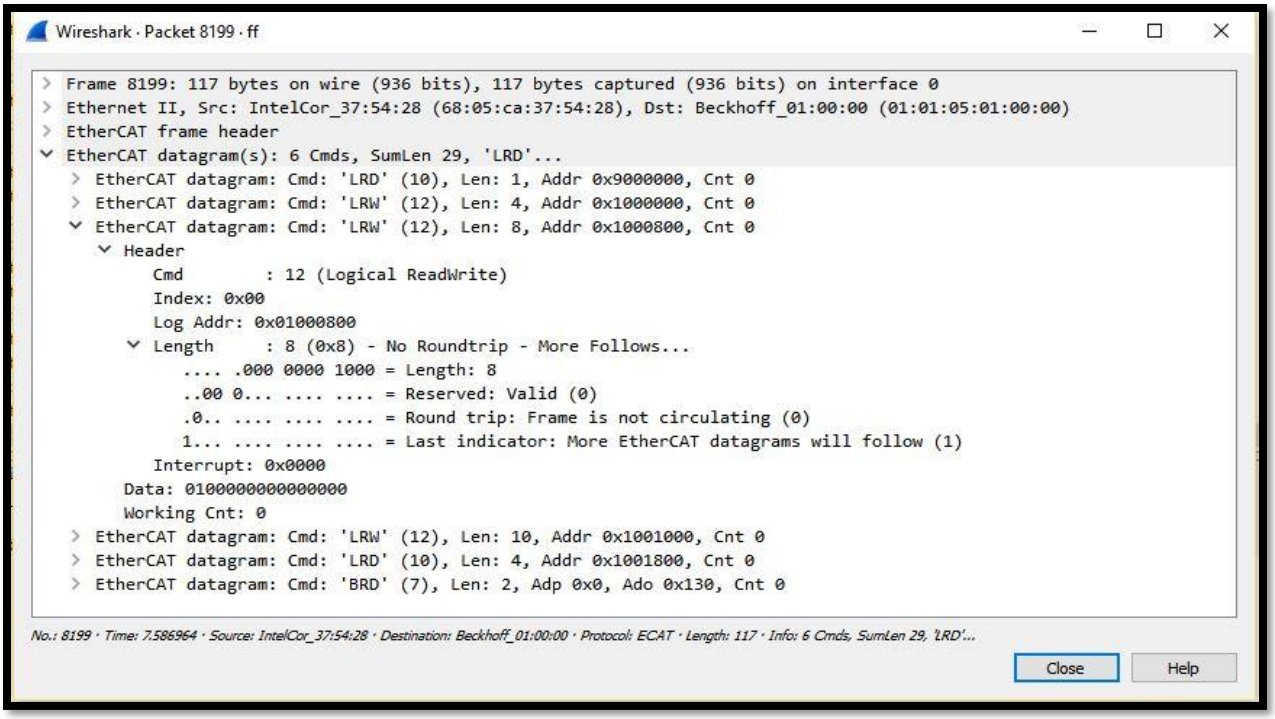


Fig 4: Data changes after writing data on the slave device

b. From the Wireshark log we can search for packets that have changes in the output data. IF the data is not zero in any sub datagram of any packet then there the write command took place. Analyzing the messages randomly can give the data change moment.

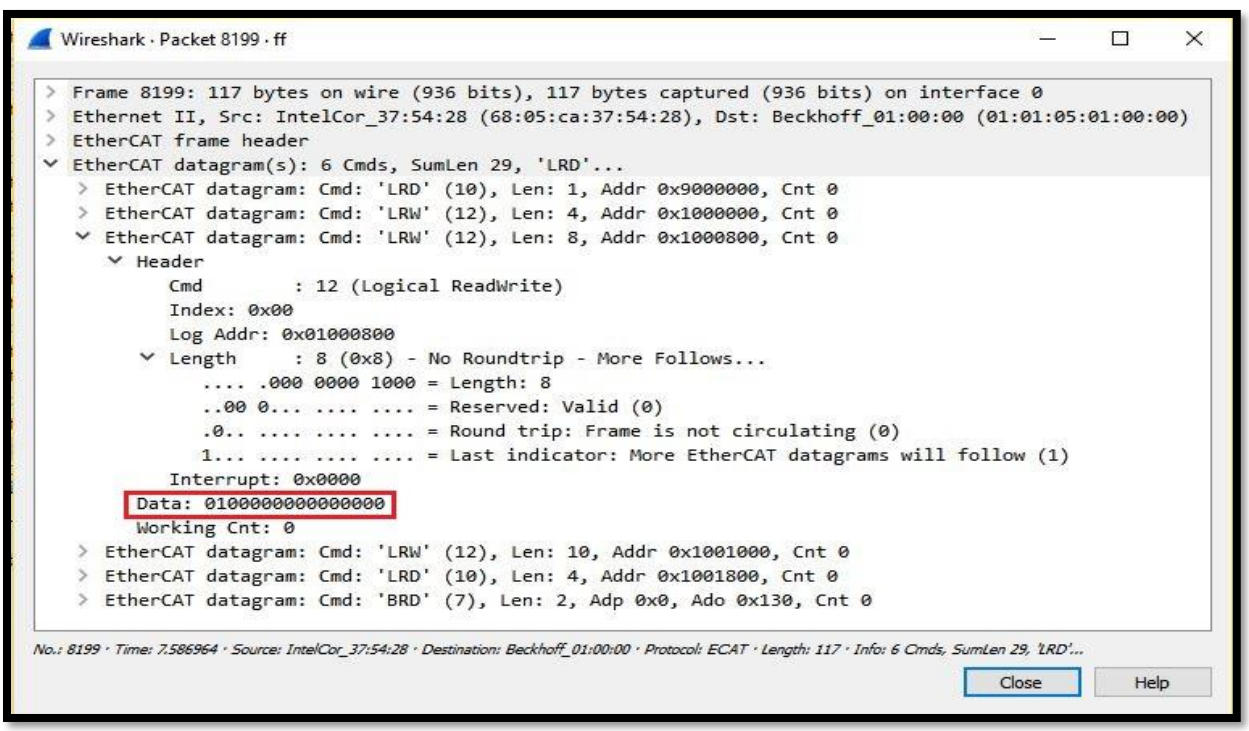


Fig 5: EtherCAT frame change after writing data on Slave device

c. To find the moment when the write action takes place, first I searched randomly the first packet of the capture and the last capture. In the 1st message of the capture, there was no written data. And the last message there was change in the sub datagram of the datagram. As there was a huge load of packets I took the exact time of the PC before writing data on the channel. I checked some of the packets before the time and after the time. Then I got the same result that I observed earlier as the data changes on the datagram.

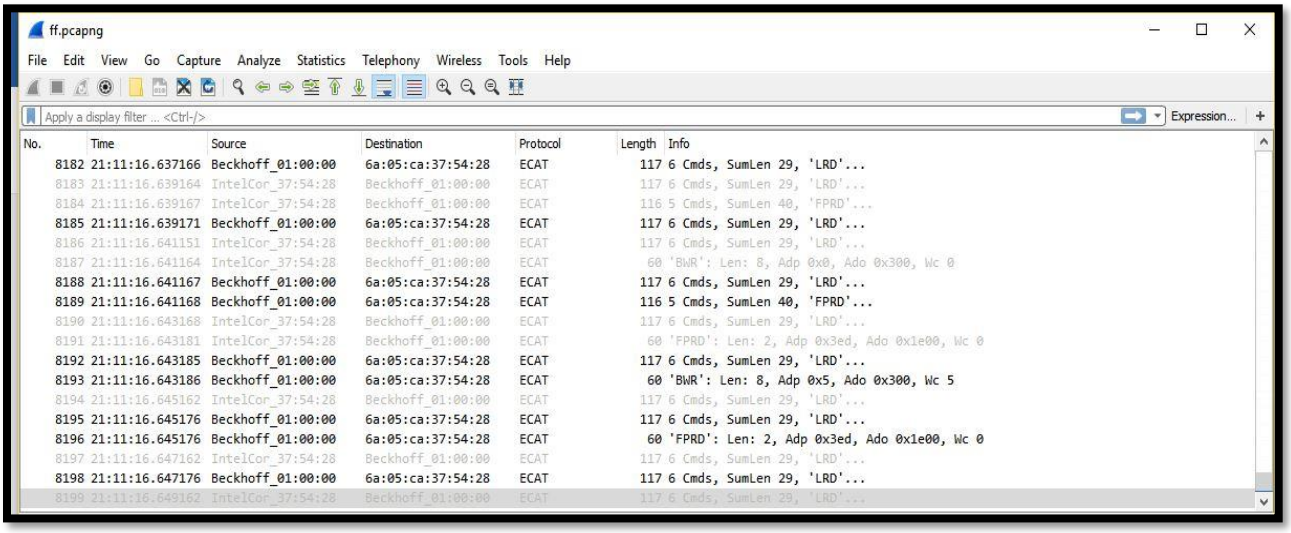


Fig 6: Wireshark data log of the captured EtherCAT messages

References:

1. http://www.ethercat.org/pdf/english/ETG2200_V2i0i0_SlaveImplementationGuide.pdf
2. https://en.wikipedia.org/wiki/EtherCAT
3. https://wiki.wireshark.org/Protocols/ethercat