How to capture and use EtherCAT trace data with WireShark

1. Objective

There is a general miss understanding about EtherCAT. EtherCAT is a transport tool only. We use it to carry data from a CNC (PLC, EtherCAT master) to the drive. CanOpen over EtherCAT (CoE), CanOpen is the actual protocol that commands the drive.

In my experience, the EtherCAT part of the fieldbus works 99% of the time. Most Customers' issues are with the setup of CanOpen. What data is needed for Interpolated Position or Cyclic synchronous position? How do I setup to read the motor torque?

And the most frustrating issue is troubleshooting why it doesn't work? Is the problem in the CNC or in the drive? Who can tell me what is wrong? How do I collect the needed data so someone can help me?

So, my objective is to outline a simple way to collect the CoE traffic, between the CNC and the drive, so it can be emailed to Kollmorgen technical support.

2. Hardware and Software

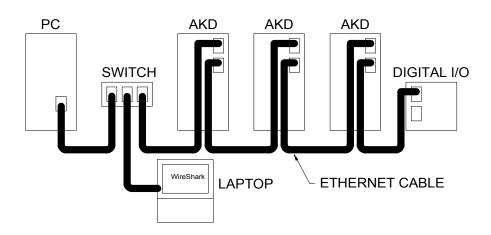
Unmanaged switch: You need a device that can be connected between the CNC and the I/O (drives, digital I/O, anything on the EtherCAT network). I have tested Netgear FS105 "Unmanaged Supper fast Ethernet Port" and it worked perfectly. No setup or "mirroring" needed. (Mirroring is needed for Modbus, EIP, anything TCP/IP based) At the time of writing this note, the FS105 was listed for \$18 online.

NOTE: EtherCAT is not a standard office TCP/IP protocol. You just need a low cost, unmanaged, dumb switch. The reason the FS105 works is that it doesn't try to read or manage the message. "What comes in goes out to all". A more complicated switch requires setup and/or could even be damaged.

Copy of **WireShark software**: It can be found (Free) at <u>https://www.wireshark.org/</u> Make sure you get it from the WireShark.org website. During the installation, it will ask you if it can install a couple of windows support programs. You should let it. If your unsure, get help! WireShark is well known with the IT profession and your IT people can answer your questions

3. Setup

Install the Unmanaged Switch between the CNC and the first node on the network. In my case, I had a PC with EtherCAT master software installed, three AKD drives, and a digital I/O module. The switch has to be in the middle and never at the end of the network.



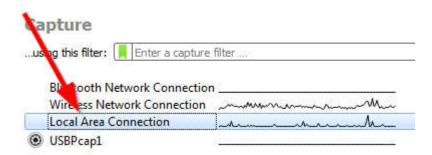
Plug in the laptop, with WireShark software, into the switch as well.

NOTE: Laptop is better than a Desktop PC. Desktop PC has an earth ground that tie into the Ethernet port. You could create a ground loop between the CNC and your PC.

NOTE: Never try and mix EtherCAT with Ethernet. Damage can happen. There should be only three cables in the switch. The laptop should be direct to the switch.

4. Making and saving the trace

When you open the WireShark software, you should see a screen asking you which port you want to capture. Double click on the "Local connection". If you have more than one "local connection", pick the simple one that is linked to the port you installed the cable to the switch. (Otherwise, try them all until you find one that works)



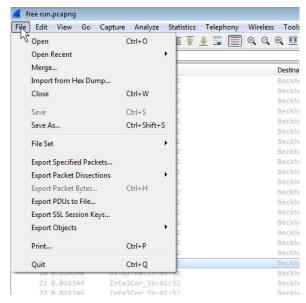
If all is well, you should see a screen that is full of EtherCAT messages

free run.pcapng	Co. Contras Analysis Statistics Talaska	Window Trate Units		
	Go Capture Analyze Statistics Telepho			
🕻 🔳 🖉 💽 🌗	🕼 🗙 🖾 । ९, 🗢 🕾 🗿 👲 其 ।	<u>ା</u> ପ୍ ପ୍ <u>୩</u>		
Apply a display filte	<ctrl-></ctrl->			
o. Time	Source	Destination	Protocol	Length Info
46650 39.795	038 92:e2:ba:1b:61:52	Beckhoff_01:00:00	ECAT	540 'FPRD': Len: 512, Adp 0x3e9, Ado 0x1c00, Wc 1 Mbx(CoE SDO Res: Scs 0)
46651 39.797	018 IntelCor_1b:61:52	Beckhoff_01:00:00	ECAT	108 5 Cmds, SumLen 32, 'NOP'
46652 39.797	020 92:e2:ba:1b:61:52	Beckhoff_01:00:00	ECAT	108 5 Cmds, SumLen 32, 'NOP'
46653 39.799	015 IntelCor_1b:61:52	Beckhoff_01:00:00	ECAT	108 5 Cmds, SumLen 32, 'NOP'
46654 39.799	018 IntelCor_1b:61:52	Beckhoff_01:00:00	ECAT	60 2 Cmds, 'FPWR': len 16, 'FPWR': len 1 Mbx(CoE SDO Req : 'Initiate Upload' (2) Idx=0x100a Sub=0
46655 39.799	019 92:e2:ba:1b:61:52	Beckhoff_01:00:00	ECAT	108 5 Cmds, SumLen 32, 'NOP'
46656 39.799	019 92:e2:ba:1b:61:52	Beckhoff 01:00:00	ECAT	60 2 Cmds, 'FPWR': len 16, 'FPWR': len 1 Mbx(CoE SDO Req : 'Initiate Upload' (2) Idx=0x100a Sub=0
46657 39.801	019 IntelCor_1b:61:52	Beckhoff 01:00:00	ECAT	108 5 Cmds, SumLen 32, 'NOP'
46658 39.801	021 92:e2:ba:1b:61:52	Beckhoff 01:00:00	ECAT	108 5 Cmds, SumLen 32, 'NOP'
46659 39.803	005 IntelCor_1b:61:52	Beckhoff 01:00:00	ECAT	108 5 Cmds, SumLen 32, 'NOP'
46660 39.803	008 92:e2:ba:1b:61:52	Beckhoff 01:00:00	ECAT	108 5 Cmds, SumLen 32, 'NOP'
46661 39.803	009 IntelCor 1b:61:52	Beckhoff 01:00:00	ECAT	540 'FPRD': Len: 512, Adp 0x3e9, Ado 0x1c00, Wc 0
46662 39.803	011 92:e2:ba:1b:61:52	Beckhoff 01:00:00	ECAT	540 'FPRD': Len: 512, Adp 0x3e9, Ado 0x1c00, Wc 1 Mbx(CoE SDO Res: Scs 2)
46663 39.804	867 IntelCor 1b:61:52	Beckhoff 01:00:00	ECAT	108 5 Cmds, SumLen 32, 'NOP'
46664 39.804	873 92:e2:ba:1b:61:52	Beckhoff_01:00:00	ECAT	108 5 Cmds, SumLen 32, 'NOP'
46665 39.807	006 IntelCor_1b:61:52	Beckhoff_01:00:00	ECAT	108 5 Cmds, SumLen 32, 'NOP'
46666 39.807	009 IntelCor_1b:61:52	Beckhoff_01:00:00	ECAT	60 2 Cmds, 'FPWR': len 16, 'FPWR': len 1 Mbx(CoE SDO Req : 'Upload Segment' (3))
46667 39.807	010 IntelCor_1b:61:52	Beckhoff_01:00:00	ECAT	60 'FPRD': Len: 2, Adp 0x3ec, Ado 0x1e00, Wc 0
46668 39.807	011 92:e2:ba:1b:61:52	Beckhoff_01:00:00	ECAT	108 5 Cmds, SumLen 32, 'NOP'
46669 39.807	012 92:e2:ba:1b:61:52	Beckhoff_01:00:00	ECAT	60 2 Cmds, 'FPWR': len 16, 'FPWR': len 1 Mbx(CoE SDO Req : 'Upload Segment' (3))
46670 39.807	013 92:e2:ba:1b:61:52	Beckhoff_01:00:00	ECAT	60 'FPRD': Len: 2, Adp 0x3ec, Ado 0x1e00, Wc 0
46671 39,809	003 IntelCor 1b:61:52	Beckhoff 01:00:00	ECAT	108 5 Cmds, SumLen 32, 'NOP'

Typical EtherCAT network makes 1000 ~ 2000 messages a second. That is a huge amount of data to weed through if your in a hurry. There is a Stop/Start button at the top of the WireShark screen so you can do a better job of sequencing the capture with the event you want to capture.

File Edit View Go	Capture Analyze Statistics Teleph	ony Wireless Tools Help					
🕻 🔳 🖉 🧿 🌗 🛅	🖹 🖸 🍳 🗢 🕾 Ŧ 🛓 🚍	📃 Q, Q, Q, 🎚					
Appl display nine <	Ctrl-/>						
lo. ime	Surce	Destination					
1 0,00000	Int	Beckhoff 01:00:00					
2.0 Stop	Continue	Beckhoff 01:00:00					
Stop	2:	8eckhoff_01:00:0					
1 0 mm	92:e2:ba:1b:61:52	Beckhoff 01:00:00					
Otart Marrie	IntelCor_1b:61:52	Beckhoff_01:00:00					
Start New	92:e2:ba:1b:61:52	Beckhoff_01:00:00					
	IntelCor_1b:61:52	Beckhoff_01:00:00					
8 0,004381	92:e2:ba:1b:61:52	Beckhoff_01:00:00					
9 0.006436	IntelCor_1b:61:52	Beckhoff_01:00:00					
10 0.006441	92:e2:ba:1b:61:52	Beckhoff_01:00:00					
11 0.008495	IntelCor_1b:61:52	Beckhoff_01:00:00					
12 0.008498	92:e2:ba:1b:61:52	Beckhoff_01:00:00					
13 0.010493	IntelCor_1b:61:52	Beckhoff_01:00:00					
14 0.010497	92:e2:ba:1b:61:52	Beckhoff_01:00:00					
15 0 012405	TotelCon th-61.57	Reckhoff 01-00-00					

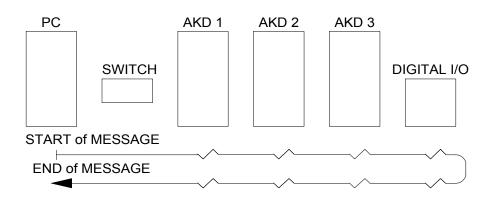
And, just like most Windows programs, there is a "FILE" on the pull down menu that will let you "save as" the file.



If you want to read the CoE data yourself

5. EtherCAT Operating Principle and Topology

The "secret sauce" of EtherCAT is that there is only one message, each node having a data region in that message, and that each node process the message "on the fly". The flow of this "one message" is out to the last node and back. The "step" in my drawing shows that, even though the message passes through the node, there is a slight delay of the data entering the node to when that same data exits the node. (nS but it exists)



In this topology, the front of the message could be passing through the 3ed AKD while the tail of the message is passing through the 1st AKD.

The WireShark trace, that you captured, will show the message twice. Once on the way out and once on the way back.

OUT	Source	Destination	Protocol	Length	Info				
RETURN	IntelCor_1b:61:52	Beckhoff_01:00:00	ECAT	108	5 Cmds,	SumLen 3	32, '1	VOP'.	
Single Republic Contraction	92:e2:ba:1b:61:52	Beckhoff 01:00:00	ECAT	108	5 Cmds,	SumLen 3	32, '1	NOP'.	
OUT	IntelCor_1b:61:52	Beckhoff 01:00:00	ECAT	108	5 Cmds,	SumLen 3	32, '1	VOP'.	
RETURN	92:e2:ba:1b:61:52	Beckhoff 01:00:00	ECAT	108	5 Cmds,	SumLen 3	32, '1	VOP'.	

6. Reading the CoE SDO (Mailbox) data from a WireShark trace (Expert)

In CoE, SDO are delivered and returned using a Data Link type called "MailBox" or "MailBox protocol". There can be multiple messages in one MailBox. The EtherCAT message, with the MailBox, is sent separate from the Process data message or PDO.

You will need to capture a trace with SDO activity in it. SDO messages don't happen very often in CoE. In a typical application, there is a lot of SDO activity at the startup of the system and then only if your program specifically does a SDO read/write.

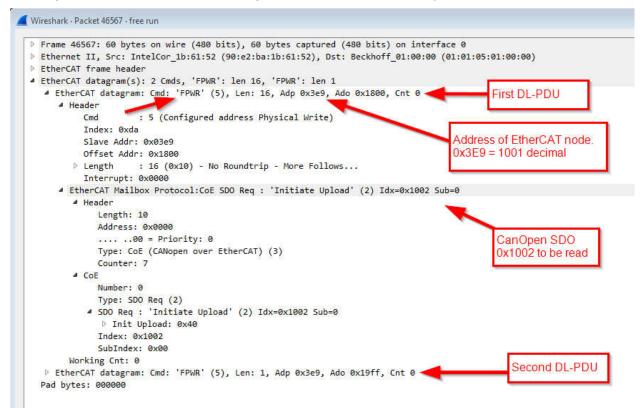
Once you have EtherCAT data being captured, apply the filter command "ecat_mailbox"

ecat_maibox		Olan de Oat Ellerte	_	
Time	Source	Step 1: Set Filter to	local	Length Info
46526 39.717005	IntelCor_1b:61:52	"ecat mailbox"	T	60 2 Cmds, 'FPWR': len 16, 'FPWR': len 1 Mbx(CoE SD0 Step 2: Click on the
46528 39.717008	92:e2:ba:1b:61:52	ecal_mainux	T	60 2 Cmds, 'FPWR': len 16, 'FPWR': len 1 Mbx(CoE 5D0
46534 39.721010	92:e2:ba:1b:61:52	2.2 M	T	548 'FPRD': Len: 512, Adp 8x3e9, Ado 8x1c88, Nc 1 Mbx(EXECUTE ATTOW
46542 39.727020	IntelCor_1b:61:52	DECKNOTT_01180180	EUAT	60 2 Cmds, 'FPWR': len 16, 'FPWR': len 1 Mbx(CoE SDO
46544 39.727823	92:e2:be:1b:61:52	Beckhoff_01:00:00	ECAT	60 2 Cmds, 'PPWR': len 16, 'PPWR': len 1 Mbx(CoE SDO
46550 39.731066	92:e2:ba:1b:61:52	Beckhoff_01:00:00	ECAT	540 'FPRD': Len: 512, Adp 0x3e9, Ado 0x1c00, Wc 1 Mbx(CoE 5DD Res: Scs 2)
46554 39.735016	IntelCor_1b:61:52	Beckhoff_01:00:00	ECAT	60 2 Cmds, 'FPWR': len 16, 'FPWR': len 1 Mbx(CoE SDO Req : 'Initiate Upload' (2) Idx=0x1001 Sub=0)
46556 39.735018	92:e2:ba:1b:61:52	Beckhoff 01:00:00	ECAT	60 2 Cmds, 'FPWR': len 16, 'FPWR': len 1 Mbx(CoE SDO Reg : 'Initiate Upload' (2) Idx=0x1001 Sub=0)
46562 39.739824	92:e2:ba:1b:61:52	Beckhoff 01:00:00	ECAT	540 "FPRD': Len: 512, Adp 0x3e9, Ado 0x1c00, Wc 1 Mbx(CoE SDD Res: Scs 2)
46567 39.743014	IntelCor_1b:61:52	Beckhoff 81:00:00	ECAT	60 2 Cmds, 'FPWR': len 16, 'FPWR': len 1 Mbx(CoE SDO Reg : 'Initiate Upload' (2) Idx=0x1002 Sub=0)
46570 39.743020	92:e2:ba:1b:61:52	Beckhoff 01:00:00	ECAT	60 2 Cmds, 'FPWR': len 16, 'FPWR': len 1 Mbx(CoE SDO Reg : 'Initiate Upload' (2) Idx=0x1002 Sub=0)
46576 39,747027	92:e2:ba:1b:61:52	Beckhoff 01:00:00	ECAT	540 'FPRD': Len: 512, Ado 0x3e9, Ado 0x1c00, Wc 1 Mbx(CoE SDO Res: 5cs 2)

The read the information, you first need to identify

vo.	Time	Source	Destination	Protocol	Length	Info										
From Mas	ster (Read Request)	IntelCor_1b:61:52	Beckhoff_01:00:00	ECAT	6	0 2 Cmds,	'FPWR':]	len 16,	'FPWR':	len 1	Mbx(CoE	SDO Req	'Initiate L	pload'	(2)	Idx=0x1000 Sub=0
Return (A	ACK of read request)	92:e2:ba:1b:61:52	Beckhoff_01:00:00	ECAT	6	0 2 Cmds,	'FPWR':]	len 16,	'FPWR':	len 1	Mbx(CoE	SDO Req	'Initiate L	pload'	(2)	Idx=0x1000 Sub=0
Data repl	ly from node	92:e2:ba:1b:61:52	Beckhoff_01:00:00	ECAT	54	0 'FPRD':	Len: 512,	, Adp Øx	x3e9, Ad	0 0x1c0	0, Wc 1	Mbx(CoE S	DO Res: Scs	2)		
465	542 39.727020	IntelCor_1b:61:52	Beckhoff_01:00:00	ECAT	6	0 2 Cmds,	'FPWR':]	len 16,	'FPWR':	len 1	Mbx(CoE	SDO Req	'Initiate U	pload'	(2)	Idx=0x1000 Sub=0
465	544 39.727023	92:e2:ba:1b:61:52	Beckhoff_01:00:00	ECAT	6	0 2 Cmds,	'FPWR':]	len 16,	'FPWR':	len 1	Mbx(CoE	SDO Req	'Initiate L	pload'	(2)	Idx=0x1000 Sub=0
465	550 39.731066	92:e2:ba:1b:61:52	Beckhoff 01:00:00	ECAT	54	Ø 'FPRD':	Len: 512.	, Adp Øx	x3e9, Ad	0 0x1c0	0, Wc 1	Mbx(CoE S	DO Res: Scs	2)		

Double click on the message from the EtherCAT master. The read request uses the "Data Link Protocal Data Unit", or DL-PUD for short, type "FPWR". There can be more than one SDO message in the one EtherCAT message. Here, I have expanded just one of two.



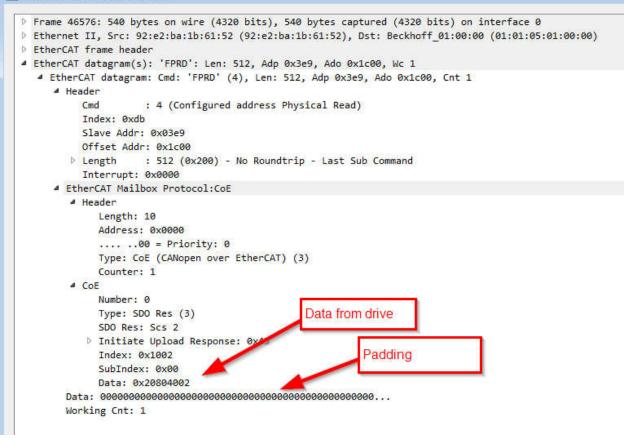
And there is a response from the node (slave) that it received the message. Becaue it takes the node time to process the request, the data is send in a separate message.

Wireshark · Packet 46570 · free run

```
Frame 46570: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
Ethernet II, Src: 92:e2:ba:1b:61:52 (92:e2:ba:1b:61:52), Dst: Beckhoff_01:00:00 (01:01:05:01:00:00)
EtherCAT frame header
EtherCAT datagram(s): 2 Cmds, 'FPWR': len 16, 'FPWR': len 1
  EtherCAT datagram: Cmd: 'FPWR' (5), Len: 16, Adp 0x3e9, Ado 0x1800, Cnt 1
     ▲ Header
          Cmd
                     : 5 (Configured address Physical Write)
          Index: 0xda
          Slave Addr: 0x03e9
          Offset Addr: 0x1800
                  : 16 (0x10) - No Roundtrip - More Follows...
        Length
          Interrupt: 0x0000
     EtherCAT Mailbox Protocol:CoE SDO Req : 'Initiate Upload' (2) Idx=0x1002 Sub=0
        ▲ Header
             Length: 10
             Address: 0x0000
             .... ..00 = Priority: 0
             Type: CoE (CANopen over EtherCAT) (3)
             Counter: 7
        ▲ CoE
             Number: 0
             Type: SDO Req (2)
           SDO Req : 'Initiate Upload' (2) Idx=0x1002 Sub=0
              Init Upload: 0x40
            Index: 0x1002
             SubIndex: 0x00
       Working Cnt: 1
  EtherCAT datagram: Cmd: 'FPWR' (5), Len: 1, Adp 0x3e9, Ado 0x19ff, Cnt 1
  Pad bytes: 000000
```

And the data response follows. The AKD's default MailBoax size is 512 bytes. The drive needs to pad or fill in the unused space in the message with "00".

Wireshark · Packet 46576 · free run



7. Reading CoE PDO data from a WireShark trace (Expert)

You will need to remove the "ecat_mailbox" filer (if you set it reading SDOs).

PDO communication means the producer and the consumer already knows what the data means and how to truncate it into useful parameters. In the case of the AKD used in this example: The drive receives 6 bytes of data (0x6040 and 0x60C1sub1) and the drive sends 6 bytes of data (0x6041 and 0x6064)

Your looking for EtherCAT datagram "NOP" messages.

	-	
From Master IntelCor_1b:61:52	Beckhoff_01:00:00 ECAT	108 5 Cmds, SumLen 32, 'NOP'
Return To Master 92:e2:ba:1b:61:52	Beckhoff_01:00:00 ECAT	108 5 Cmds, SumLen 32, 'NOP'
From Master IntelCor_1b:61:52	Beckhoff_01:00:00 ECAT	108 5 Cmds, SumLen 32, 'NOP'
46566 39.743012 IntelCor_1b:61:52	Beckhoff_01:00:00 ECAT	60 'FPRD': Len: 2, Adp 0x3eb, Ado 0x1c00, Wc 0
46567 39.743014 IntelCor_1b:61:52	Beckhoff_01:00:00 ECAT	60 2 Cmds, 'FPWR': len 16, 'FPWR': len 1 Mbx(CoE SDO Req : 'Initiate Upload' (2) Idx=0x1002 Sub=0)
Return To Master 92:e2:ba:1b:61:52	Beckhoff_01:00:00 ECAT	108 5 Cmds, SumLen 32, 'NOP'
46569 39.743019 92:e2:ba:1b:61:52	Beckhoff_01:00:00 ECAT	60 'FPRD': Len: 2, Adp 0x3eb, Ado 0x1c00, Wc 0
46570 39.743020 92:e2:ba:1b:61:52	Beckhoff_01:00:00 ECAT	60 2 Cmds, 'FPWR': len 16, 'FPWR': len 1 Mbx(CoE SDO Req : 'Initiate Upload' (2) Idx=0x1002 Sub=0)

Double clicking on the "From Master" message, there is a lot of different DL-PDUs being sent to the drives. The PDO data is located in the "LRW".

```
Frame 46563: 108 bytes on wire (864 bits), 108 bytes captured (864 bits) on interface 0
Ethernet II, Src: IntelCor_1b:61:52 (90:e2:ba:1b:61:52), Dst: Beckhoff_01:00:00 (01:01:05:01:00:00)
EtherCAT frame header
EtherCAT datagram(s): 5 Cmds, SumLen 32, 'NOP'...
  EtherCAT datagram: Cmd: 'NOP' (0), Len: 4, Adp 0x0, Ado 0x900, Cnt 0
  EtherCAT datagram: Cmd: 'ARMW' (13), Len: 4, Adp 0x0, Ado 0x910, Cnt 0
  EtherCAT datagram: Cmd: 'LRW' (12), Len: 4, Addr 0x1000000, Cnt 0
  EtherCAT datagram: Cmd: 'LRW' (12), Len: 18, Addr 0x1000800, Cnt 0
    ▲ Header
                   : 12 (Logical ReadWrite)
        Cmd
        Index: 0x00
        Log Addr: 0x01000800
                 : 18 (0x12) - No Roundtrip - More Follows...
       Length
        Interrupt: 0x0000
      Data: a89511011f00a1c303001f00489efaff1f00
                                                                         Data
      Working Cnt: 0
  EtherCAT datagram: Cmd: 'BRD' (7), Len: 2, Adp 0x0, Ado 0x130, Cnt 0
```

What your looking at is all the data to all the nodes. (the nodes know how to truncate the data) So, to make this data meaningful, you need to divide it into groups of two hexadecimal numbers and then reversed. So 0xA89511011F00A1C303001F00489EFAFF1F00 is read as 00 1F FF FA 9E 48 00 1F 01 30 30 1C 0A F0 11 10 51 95 A8. My AKD drive's PDO mapping is 6 bytes each. This long string of hexadecimal will need to be divided again in to groups of 6 octets.

Drive 1: 00 1F FF FA 9E 48 (0x6040 = 0x001F and 0x60C1sub1 = 0xFFFA9E48) Drive 2: 00 1F 01 30 30 1C Drive 3: 0A F0 11 10 51 95 Digital I/O: A8

And the return message is read the same way.

Frame 46564: 108 bytes on wire (864 bits), 108 bytes captured (864 bits) on interface 0
Ethernet II, Src: 92:e2:ba:1b:61:52 (92:e2:ba:1b:61:52), Dst: Beckhoff_01:00:00 (01:01:05:01:00:00)
EtherCAT frame header
EtherCAT datagram(s): 5 Cmds, SumLen 32, 'NOP'
> EtherCAT datagram: Cmd: 'NOP' (0), Len: 4, Adp 0x0, Ado 0x900, Cnt 0
EtherCAT datagram: Cmd: 'ARMW' (13), Len: 4, Adp 0x4, Ado 0x910, Cnt 3
EtherCAT datagram: Cmd: 'LRW' (12), Len: 4, Addr 0x1000000, Cnt 3
EtherCAT datagram: Cmd: 'LRW' (12), Len: 18, Addr 0x1000800, Cnt 9
Header
Cmd : 12 (Logical ReadWrite)
Index: 0x00
Log Addr: 0x01000800 Data
Length : 18 (0x12) - No Roundtrip - More Follows
Interrupt: 0x0000
Data: a19511013712a5c3030037124d9efaff3712
Working Cnt: 9
EtherCAT datagram: Cmd: 'BRD' (7), Len: 2, Adp 0x4, Ado 0x130, Cnt 4

Raw Data	a1	95	11	01	37	12	a5	с3	03	00	37	12	4d	9e	fa	ff	37	12	
Reversed and truncated	12	37	ff	fa	9e	4d	12	37	00	03	с3	a5	12	37	01	11	95	a1	
CanOpen Object	0x6041 0x6064				0x6	0x6041 0x6064					0x6041 0x6064								
Node	Drive 1						Drive 2						Drive 3						

(The I/O module was only setup as "outputs" and is not mapped for inputs)

8. References

WireShark: https://www.wireshark.org/

"ecat" filter commands: https://www.wireshark.org/docs/dfref/e/ecat.html

EtherCAT: https://www.ethercat.org CanOpen: <u>https://www.can-cia.org/</u>