工控系統資安
Industry Control System (ICS) Cybersecurity

W5：工控系統網路模擬實驗與封包分析

Course name: ICS Cybersecurity

Associate Professor: C.T. Lin

2024.10.01

Course Outline

5.工控系統網路模擬實驗與封包分析

- 常見工業控制網路與協定
- 工業控制網路協定-Modbus介紹
- (Lab) Modbus資料存儲區
- (Lab) 工控網路封包擷取實驗與工控命令分析解讀 作業 1-A
- (Lab) 工控系統網路封包分析 作業 1-B

常見工業控制網路與協定

Common ICS 網路協定

- Universal ICS Protocols
  - Modbus TCP: TCP/502
  - OPC UA: TCP/4840
  - OPC UA XML: TCP/80, TCP/443
- Building Automation Specific Protocols
  - BACnet/IP: UDP/47808
  - LonTalk: UDP/1628, UDP/1629
  - Fox (Tridium/Niagara): TCP/1911
- Process Automation Specific Protocols
  - EtherCAT: UDP/34980
  - Ethernet/IP: TCP/44818, UDP/2222,44818
  - FL-net: UDP/55000 to 55003
  - Fieldbus HSE: TCP/1080-1091, UDP/1089-1091
  - HART-IP: TCP/5094, UDP/5094
  - PROFINET: TCP/34962-34964, UDP/34962-34964
- Energy Sector Specific Protocols
  - DNP3: TCP/20000, UDP/20000
  - DLMS/COSEM: TCP/4059, UDP/4059
  - ICCP: TCP/2404
  - IEEE C37.118: TCP/4712, UDP/4713
  - MMS: TCP/102

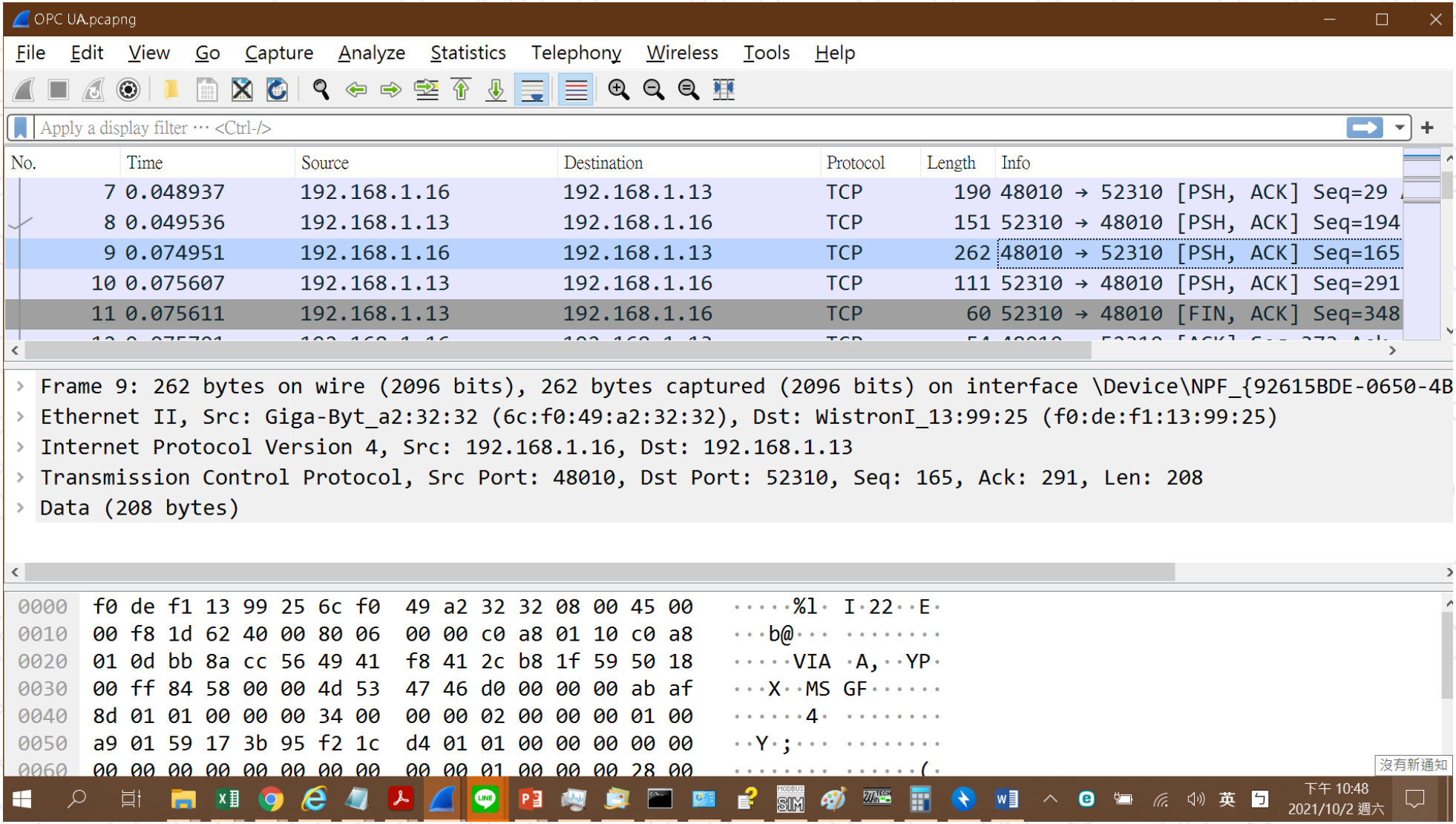
Source : SANS.org

Wireshark and ICS Protocols

- 支援大多數通用之ICS協定
  - 自1.8版即支援Modbus

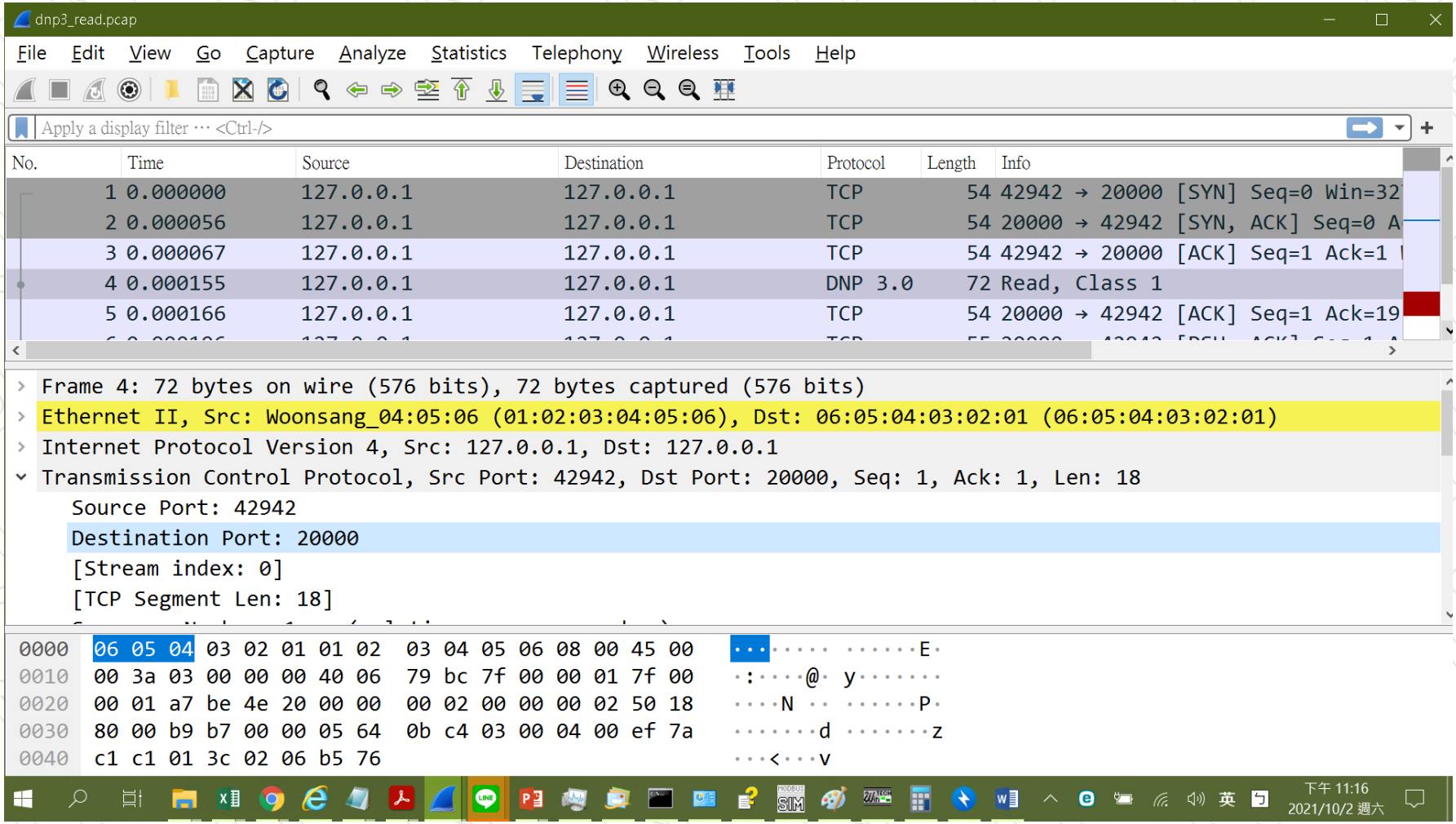
Siemens S7	OPC UA	EtherCAT	BACnet
MMS(IEC61850)	HART-IP	SERCOS III	KNXnet/IP
GOOSE(IEC61850)	CoAP	RTPS	Lontalk
SV(IEC61850)	Omron FINS	TTEthernet	CANopen
Modbus	openSAFETY	CDT	SAE J1939
OPC DA	Profinet	EtherNet/IP	USITT DMX512-A
FF HSE	DNP3	CIP	BSSAP/BSAP
IEC 104	Sinec H1	CIP Safety	Gryphon
Ethernet POWERLINK	EGD(Ethernet Global Data)	DeviceNet	ZigBee

OPC UA

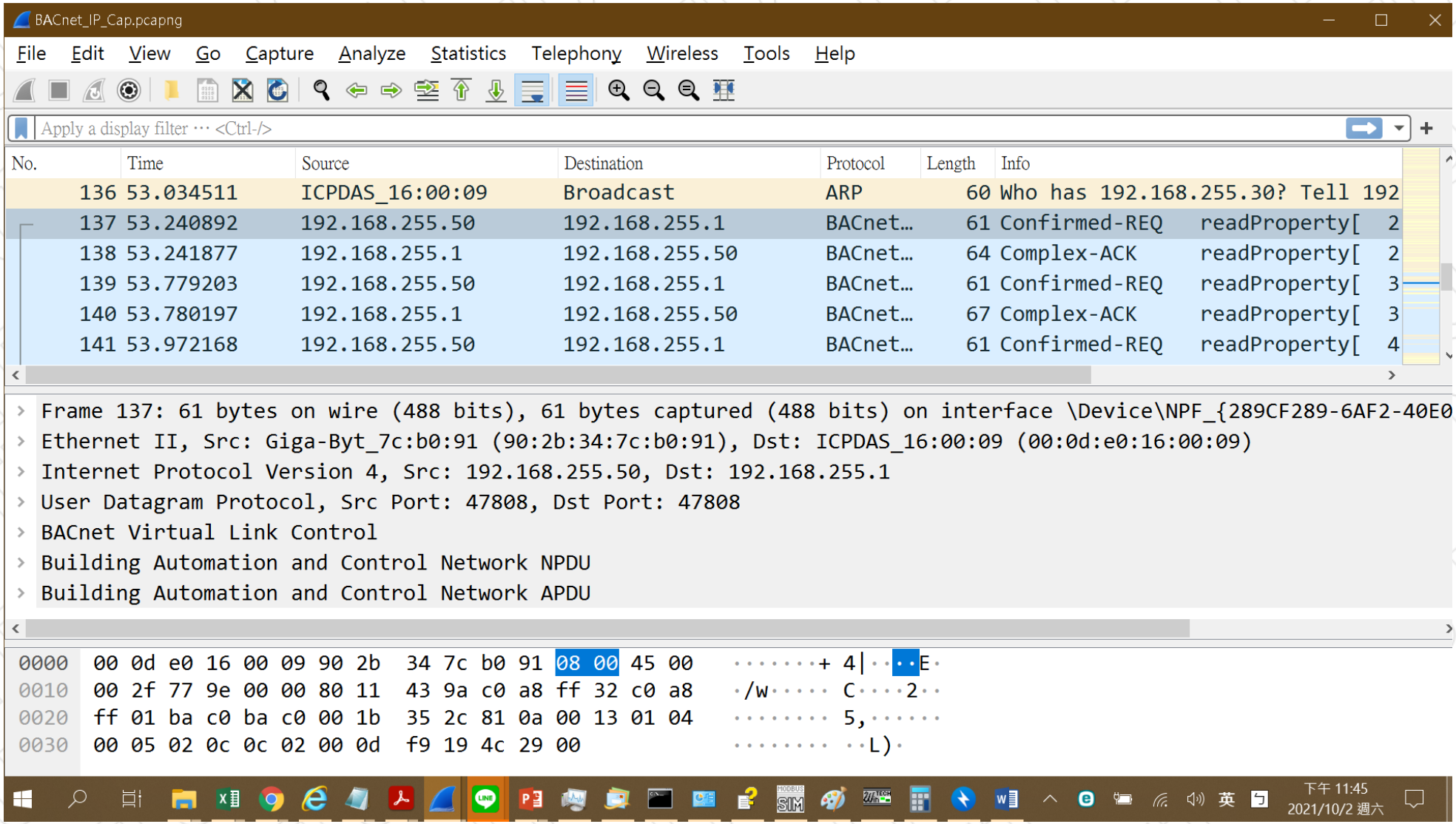

OPC UA.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ··· <Ctrl-/>
No. Time Source Destination Protocol Length Info
7 0.048937 192.168.1.16 192.168.1.13 TCP 190 48010 → 52310 [PSH, ACK] Seq=29
8 0.049536 192.168.1.13 192.168.1.16 TCP 151 52310 → 48010 [PSH, ACK] Seq=194
9 0.074951 192.168.1.16 192.168.1.13 TCP 262 48010 → 52310 [PSH, ACK] Seq=165
10 0.075607 192.168.1.13 192.168.1.16 TCP 111 52310 → 48010 [PSH, ACK] Seq=291
11 0.075611 192.168.1.13 192.168.1.16 TCP 60 52310 → 48010 [FIN, ACK] Seq=348
Frame 9: 262 bytes on wire (2096 bits), 262 bytes captured (2096 bits) on interface \Device\NPF_{92615BDE-0650-4B
Ethernet II, Src: Giga-Byt_a2:32:32 (6c:f0:49:a2:32:32), Dst: WistronI_13:99:25 (f0:de:f1:13:99:25)
Internet Protocol Version 4, Src: 192.168.1.16, Dst: 192.168.1.13
Transmission Control Protocol, Src Port: 48010, Dst Port: 52310, Seq: 165, Ack: 291, Len: 208
Data (208 bytes)
0000 f0 de f1 13 99 25 6c f0 49 a2 32 32 08 00 45 00
0010 00 f8 1d 62 40 00 80 06 00 00 c0 a8 01 10 c0 a8
0020 01 0d bb 8a cc 56 49 41 f8 41 2c b8 1f 59 50 18
0030 00 ff 84 58 00 00 4d 53 47 46 d0 00 00 00 ab af
0040 8d 01 01 00 00 00 34 00 00 00 02 00 00 00 01 00
0050 a9 01 59 17 3b 95 f2 1c d4 01 01 00 00 00 00 00
0060 00 00 00 00 00 00 00 00 00 00 01 00 00 00 28 00
沒有新通知
下午 10:48
2021/10/2 週六

DNP 3

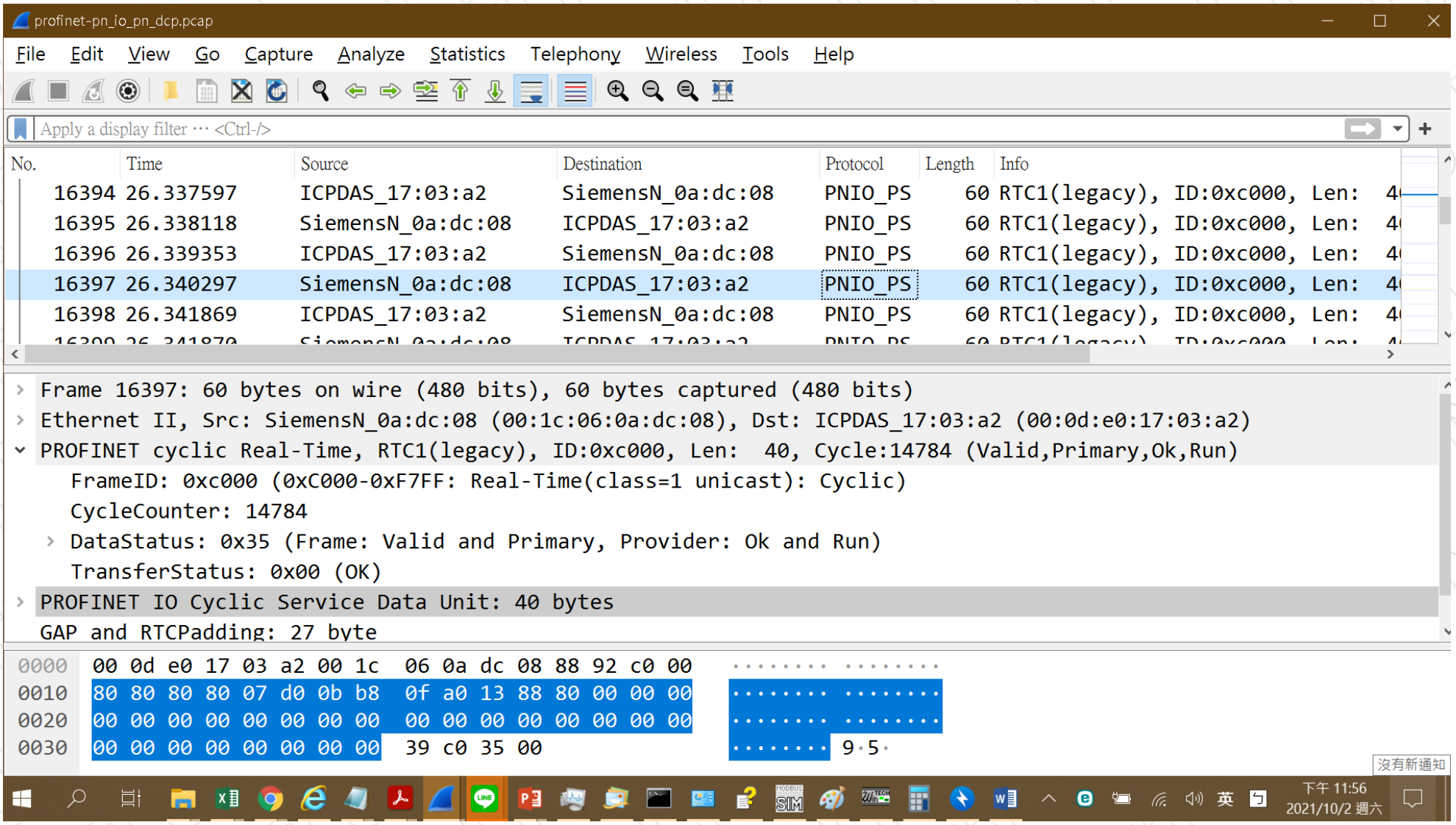
- 參考：http://amrstandard.tca.org.tw/upload/20201030060836 4.pdf



BACnet


BACnet_IP_Cap.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ··· <Ctrl-/>
No. Time Source Destination Protocol Length Info
136 53.034511 ICPDAS_16:00:09 Broadcast ARP 60 Who has 192.168.255.30? Tell 192
137 53.240892 192.168.255.50 192.168.255.1 BACnet… 61 Confirmed-REQ readProperty[2
138 53.241877 192.168.255.1 192.168.255.50 BACnet… 64 Complex-ACK readProperty[2
139 53.779203 192.168.255.50 192.168.255.1 BACnet… 61 Confirmed-REQ readProperty[3
140 53.780197 192.168.255.1 192.168.255.50 BACnet… 67 Complex-ACK readProperty[3
141 53.972168 192.168.255.50 192.168.255.1 BACnet… 61 Confirmed-REQ readProperty[4
Frame 137: 61 bytes on wire (488 bits), 61 bytes captured (488 bits) on interface \Device\NPF_{289CF289-6AF2-40E0
Ethernet II, Src: Giga-Byt_7c:b0:91 (90:2b:34:7c:b0:91), Dst: ICPDAS_16:00:09 (00:0d:e0:16:00:09)
Internet Protocol Version 4, Src: 192.168.255.50, Dst: 192.168.255.1
User Datagram Protocol, Src Port: 47808, Dst Port: 47808
BACnet Virtual Link Control
Building Automation and Control Network NPDU
Building Automation and Control Network APDU
0000 00 0d e0 16 00 09 90 2b 34 7c b0 91 08 00 45 00
0010 00 2f 77 9e 00 00 80 11 43 9a c0 a8 ff 32 c0 a8
0020 ff 01 ba c0 ba c0 00 1b 35 2c 81 0a 00 13 01 04
0030 00 05 02 0c 0c 02 00 0d f9 19 4c 29 00
下午 11:45
2021/10/2 週六

PROFINET


profinet-pn_io_pn_dcp.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ··· <Ctrl-/>
No. Time Source Destination Protocol Length Info
16394 26.337597 ICPDAS_17:03:a2 SiemensN_0a:dc:08 PNIO_PS 60 RTC1(legacy), ID:0xc000, Len: 4
16395 26.338118 SiemensN_0a:dc:08 ICPDAS_17:03:a2 PNIO_PS 60 RTC1(legacy), ID:0xc000, Len: 4
16396 26.339353 ICPDAS_17:03:a2 SiemensN_0a:dc:08 PNIO_PS 60 RTC1(legacy), ID:0xc000, Len: 4
16397 26.340297 SiemensN_0a:dc:08 ICPDAS_17:03:a2 PNIO_PS 60 RTC1(legacy), ID:0xc000, Len: 4
16398 26.341869 ICPDAS_17:03:a2 SiemensN_0a:dc:08 PNIO_PS 60 RTC1(legacy), ID:0xc000, Len: 4
Frame 16397: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
Ethernet II, Src: SiemensN_0a:dc:08 (00:1c:06:0a:dc:08), Dst: ICPDAS_17:03:a2 (00:0d:e0:17:03:a2)
PROFINET cyclic Real-Time, RTC1(legacy), ID:0xc000, Len: 40, Cycle:14784 (Valid,Primary,Ok,Run)
FrameID: 0xc000 (0xC000-0xF7FF: Real-Time(class=1 unicast): Cyclic)
CycleCounter: 14784
DataStatus: 0x35 (Frame: Valid and Primary, Provider: Ok and Run)
TransferStatus: 0x00 (OK)
PROFINET IO Cyclic Service Data Unit: 40 bytes
GAP and RTCPadding: 27 byte
0000 00 0d e0 17 03 a2 00 1c 06 0a dc 08 88 92 c0 00
0010 80 80 80 80 07 d0 0b b8 0f a0 13 88 80 00 00 00
0020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0030 00 00 00 00 00 00 00 00 39 c0 35 00
9·5·
沒有新通知
下午 11:56
2021/10/2 週六

UDP

udp-nm_anon.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ··· <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.16	224.0.0.1	UDP	64	12435 → 12435 Len=8

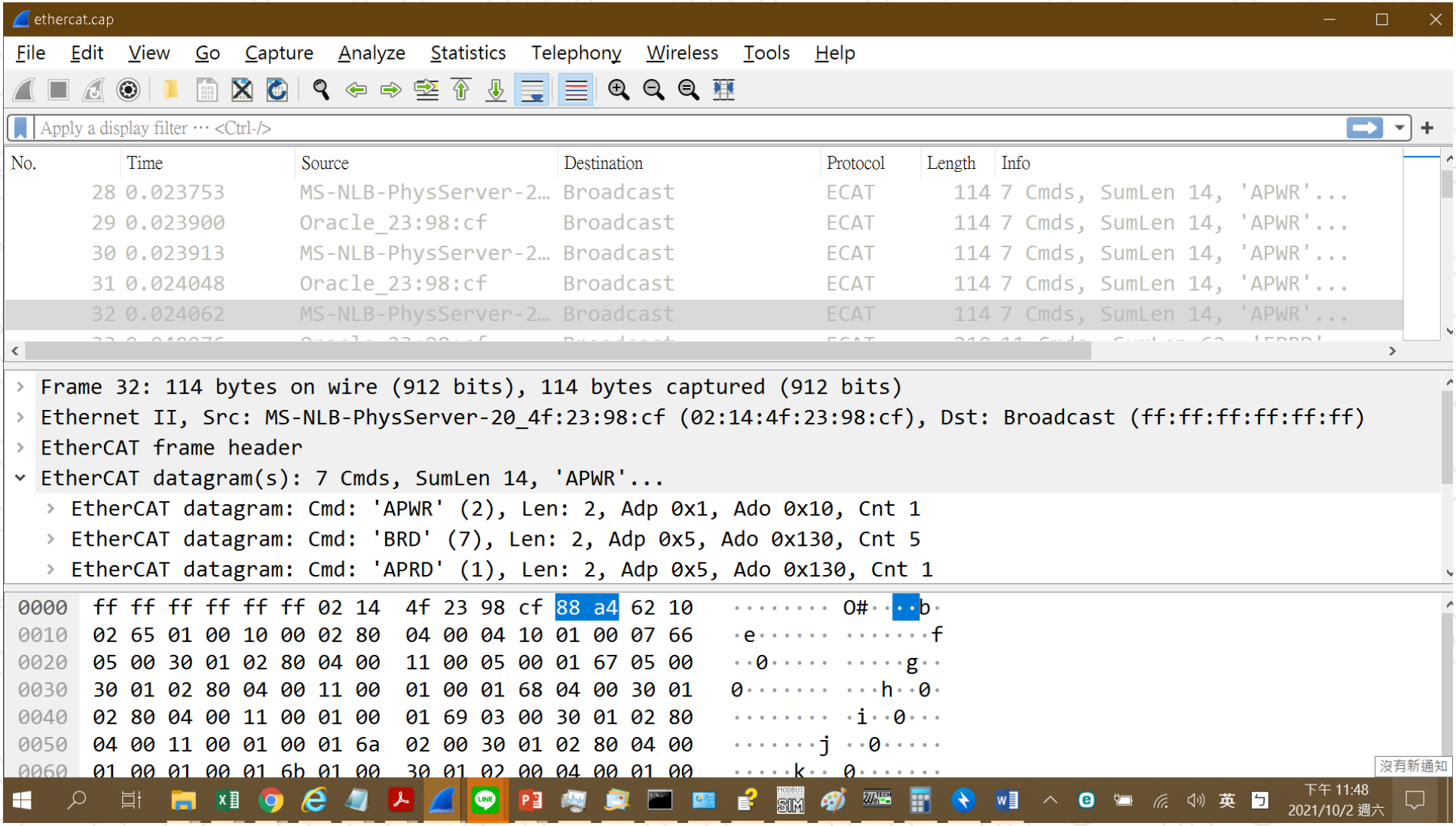
```
> Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits)
> Ethernet II, Src: d0:00:00:00:00:01 (d0:00:00:00:00:01), Dst: IPv4mcast_01 (01:00:5e:00:00:01)
> 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 5
> 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 1
> Internet Protocol Version 4, Src: 192.168.0.16, Dst: 224.0.0.1
> User Datagram Protocol, Src Port: 12435, Dst Port: 12435
> Data (8 bytes)
```

```
0000  01 00 5e 00 00 01 d0 00  00 00 00 01 91 00 00 05   ··^····· ········
0010  81 00 00 01 08 00 45 00  00 24 d4 2b 00 00 01 11   ····E· ·$·+····
0020  00 00 c0 a8 00 10 e0 00  00 01 30 93 30 93 00 10   ···· ··0·0···
0030  00 00 40 10 02 00 00 01  12 02 00 00 00 00 00 00   ··@····· ······
```

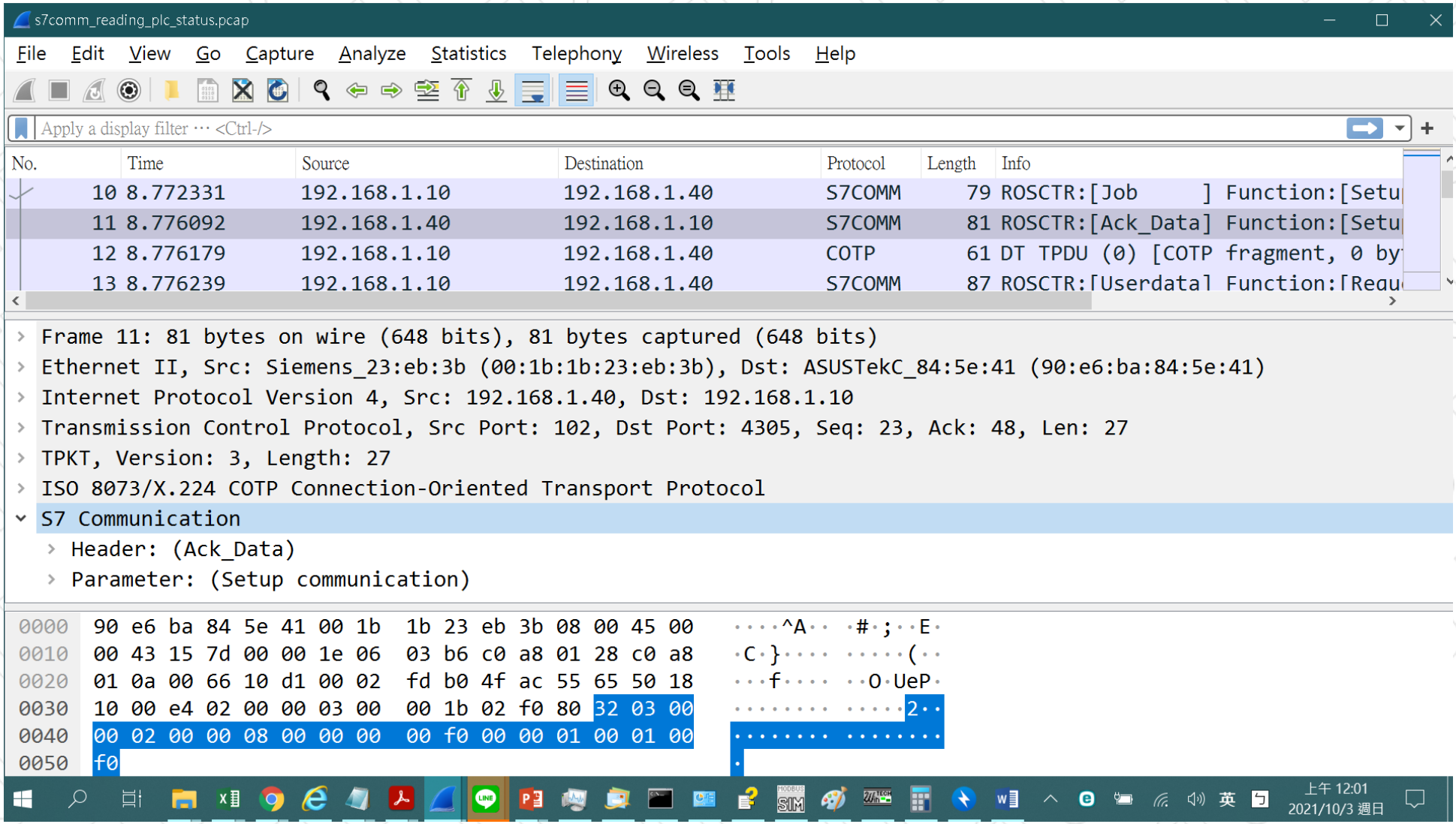
沒有新通知

下午 11:58
2021/10/2 週六

EtherCAT


ethercat.cap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ··· <Ctrl-/>
No. Time Source Destination Protocol Length Info
28 0.023753 MS-NLB-PhysServer-2… Broadcast ECAT 114 7 Cmds, SumLen 14, 'APWR'...
29 0.023900 Oracle_23:98:cf Broadcast ECAT 114 7 Cmds, SumLen 14, 'APWR'...
30 0.023913 MS-NLB-PhysServer-2… Broadcast ECAT 114 7 Cmds, SumLen 14, 'APWR'...
31 0.024048 Oracle_23:98:cf Broadcast ECAT 114 7 Cmds, SumLen 14, 'APWR'...
32 0.024062 MS-NLB-PhysServer-2… Broadcast ECAT 114 7 Cmds, SumLen 14, 'APWR'...
Frame 32: 114 bytes on wire (912 bits), 114 bytes captured (912 bits)
Ethernet II, Src: MS-NLB-PhysServer-20_4f:23:98:cf (02:14:4f:23:98:cf), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
EtherCAT frame header
EtherCAT datagram(s): 7 Cmds, SumLen 14, 'APWR'...
EtherCAT datagram: Cmd: 'APWR' (2), Len: 2, Adp 0x1, Ado 0x10, Cnt 1
EtherCAT datagram: Cmd: 'BRD' (7), Len: 2, Adp 0x5, Ado 0x130, Cnt 5
EtherCAT datagram: Cmd: 'APRD' (1), Len: 2, Adp 0x5, Ado 0x130, Cnt 1
0000 ff ff ff ff ff ff 02 14 4f 23 98 cf 88 a4 62 10
0010 02 65 01 00 10 00 02 80 04 00 04 10 01 00 07 66
0020 05 00 30 01 02 80 04 00 11 00 05 00 01 67 05 00
0030 30 01 02 80 04 00 11 00 01 00 01 68 04 00 30 01
0040 02 80 04 00 11 00 01 00 01 69 03 00 30 01 02 80
0050 04 00 11 00 01 00 01 6a 02 00 30 01 02 80 04 00
0060 01 00 01 00 01 6b 01 00 30 01 02 00 04 00 01 00
沒有新通知
下午 11:48
2021/10/2 週六

S7COMM

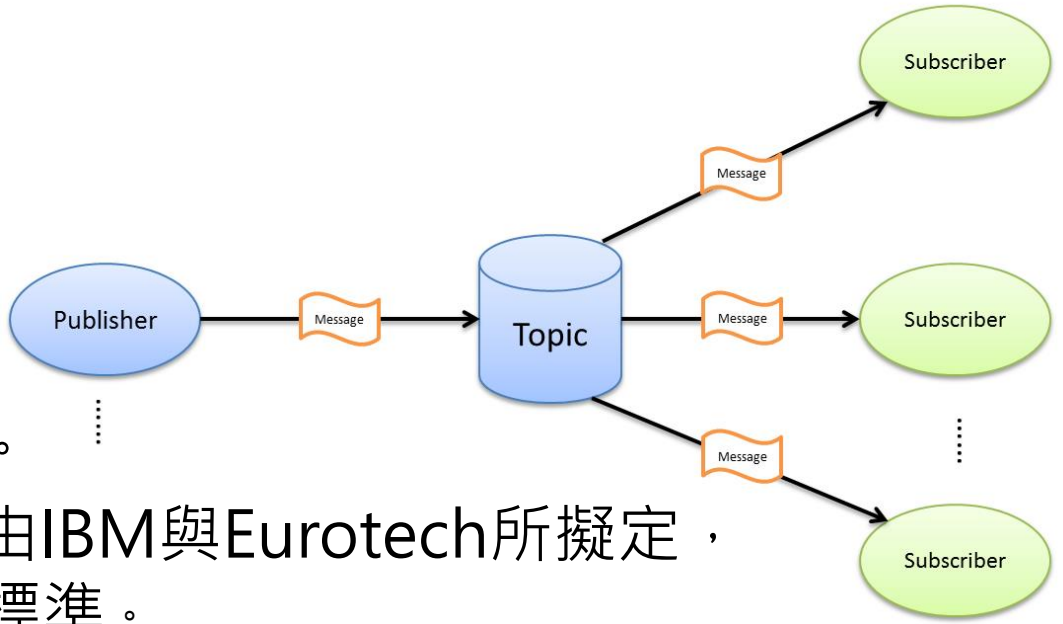

s7comm_reading_plc_status.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ··· <Ctrl-/>
No. Time Source Destination Protocol Length Info
10 8.772331 192.168.1.10 192.168.1.40 S7COMM 79 ROSCTR:[Job] Function:[Setu
11 8.776092 192.168.1.40 192.168.1.10 S7COMM 81 ROSCTR:[Ack_Data] Function:[Setu
12 8.776179 192.168.1.10 192.168.1.40 COTP 61 DT TPDU (0) [COTP fragment, 0 by
13 8.776239 192.168.1.10 192.168.1.40 S7COMM 87 ROSCTR:[Userdata] Function:[Requ
Frame 11: 81 bytes on wire (648 bits), 81 bytes captured (648 bits)
Ethernet II, Src: Siemens_23:eb:3b (00:1b:1b:23:eb:3b), Dst: ASUSTekC_84:5e:41 (90:e6:ba:84:5e:41)
Internet Protocol Version 4, Src: 192.168.1.40, Dst: 192.168.1.10
Transmission Control Protocol, Src Port: 102, Dst Port: 4305, Seq: 23, Ack: 48, Len: 27
TPKT, Version: 3, Length: 27
ISO 8073/X.224 COTP Connection-Oriented Transport Protocol
S7 Communication
Header: (Ack_Data)
Parameter: (Setup communication)
0000 90 e6 ba 84 5e 41 00 1b 1b 23 eb 3b 08 00 45 00
0010 00 43 15 7d 00 00 1e 06 03 b6 c0 a8 01 28 c0 a8
0020 01 0a 00 66 10 d1 00 02 fd b0 4f ac 55 65 50 18
0030 10 00 e4 02 00 00 03 00 00 1b 02 f0 80 32 03 00
0040 00 02 00 00 08 00 00 00 00 f0 00 00 01 00 01 00
0050 f0
上午 12:01
2021/10/3 週日

智慧應用通訊協定

- Machine-to-Machine (M2M) 協定目前常見的包括有 CoAP、XMPP、RESTful HTTP及MQTT 。
  - CoAP (Constrained Application Protocol)受限應用協定，是採用UDP方式傳送用在受限制的資源上的一個類似HTML觀念的應用層協定。其協定最小資料為4 byte header
    也就是說當你下一個類似
    coap://example.com:5683/~sensors./temp1.xml這樣的CoAP URI你就可以得到該感測器的資訊。
  - XMPP (Extensible Messaging and Presence Protocol)大家應該就比較熟悉了，這是一個採用TCP連接並且可以透過XML進行雙向溝通的協定。經常用在即時通訊之類的軟體上。
  - RESTful 符合REST(Representational State Transfer)原則的系統統稱為RESTful，REST同樣架構在HTTP over TCP上的一個協定，比較適合在雲端運算之類的環境。
- Android 常見的推播方式有GCM(Google Cloud Messaging)、XMPP、HTTP輪循方式(Web Service)以及MQTT協定。

智慧家庭通用協定-MQTT

- MQTT是一個 machine-to-machine (M2M) 的發佈(Publish)/訂閱(Subscribe)訊息的傳輸協定，簡單來說當發佈者將訊息送至Topic平台，而Topic會將這個訊息送到所註冊的訂閱者。一般來說發佈者可以是一個Sensors也可以是一個推播訊息的入口。訂閱者可以是個伺服器上的應用服務也可以是個手機。其關係如下圖所示：



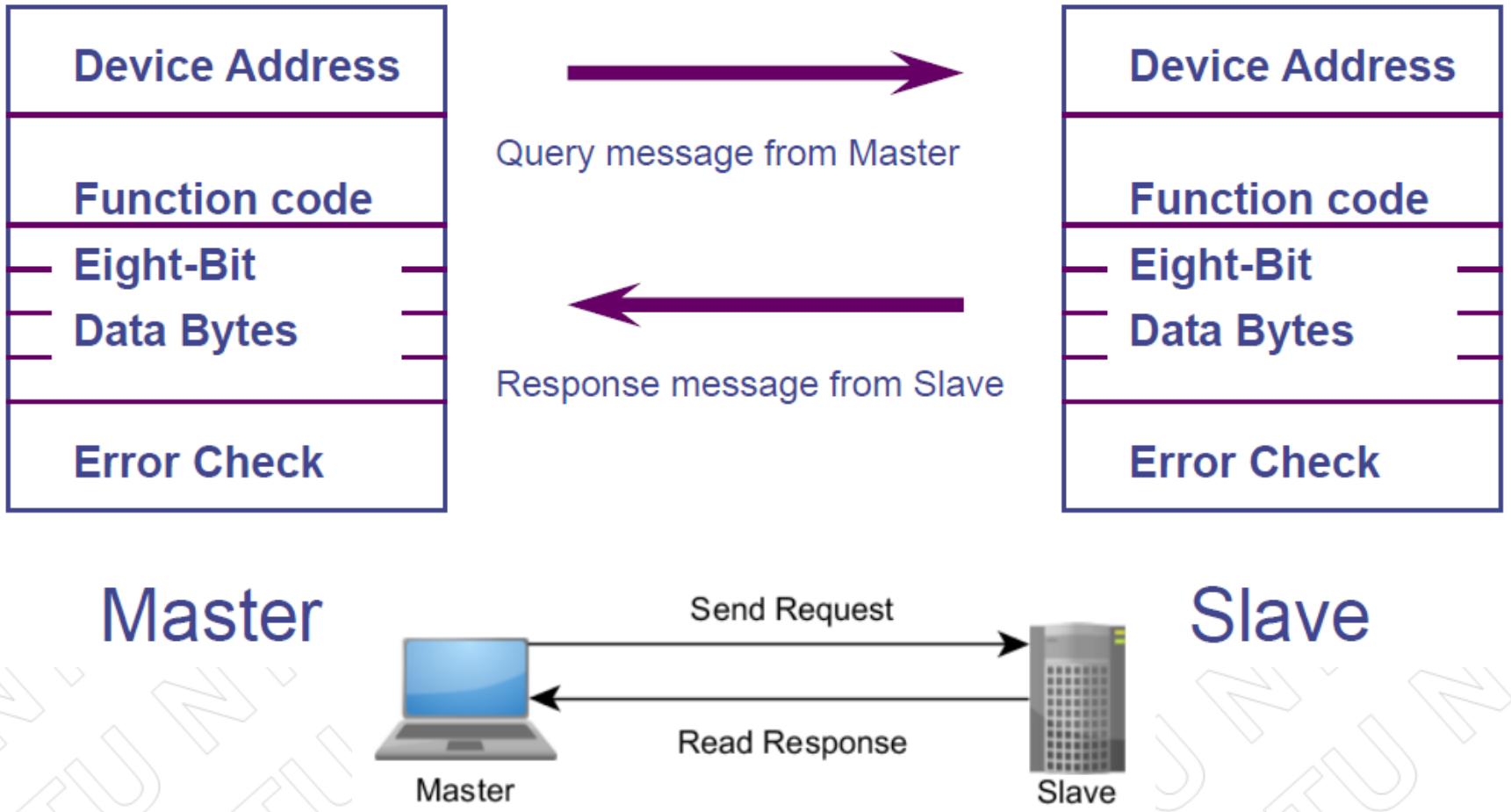
- 早在1999年由IBM的Andy Stanford-Clark及Arcom (現為Eurotech)的Arlen Nipper這位所一起創造MQTT這個協定。MQTT協定的原始規格是由IBM與Eurotech所擬定，並捐贈給OASIS作為開放標準。
- MQTT Specification: http://public.dhe.ibm.com/software/dw/webservices/ws-mqtt/mqtt-v3r1.html

工業控制網路協定-Modbus介紹

MODBUS概述

- 一種串行通信協議，Modicon公司於1979年發表，最初用於可編程邏輯控制器（PLC）通信
- 廣泛應用之協定，被數百業者開發用於眾多工業
- 是一種需求-回應協定
- 採用主從架構--主(Master)對從(Slave)之通訊
  - Master 需主動推播給現場設備
  - 現場設備不能發動通訊請求
  - Slave設備必須有唯一的位址(範圍1:247)
  - 主要裝置會是人機介面 (HMI) 或監控與資料擷取 (SCADA) 系統
  - 附屬裝置則是感測器、程式化邏輯控制器 (PLC) 或程式化自動控制器 (PAC)
  - 協定本身並未擴及安全性設計

MODBUS運作原理


Device Address
Function code
Eight-Bit
Data Bytes
Error Check
Query message from Master
Response message from Slave
Device Address
Function code
Eight-Bit
Data Bytes
Error Check
Master
Send Request
Read Response
Slave
Master
Slave

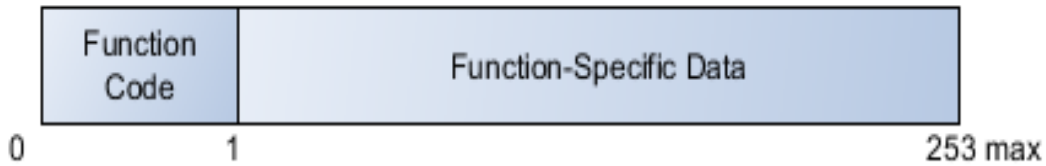
Modbus協定

- 協定資料單元 PDU (Protocol Data Unit)
  -每個 PDU 則包含了函式代碼和相關資料

- 應用資料單元 ADU (Application Data Unit)
  -每個 ADU 都有一個協定資料單元 (PDU)
  TCP/IP ADU

PDU(1/2)

- 早期Modbus 以序列為基礎的單一應用層協定，只有協定資料 單元(PDU)，無法分成好幾層
- PDU (Protocol Data Unit)：
  - 函式代碼+ 252 個位元的函式專屬資料
  - 附屬裝置(如PLC)會檢驗函式產生器、資料位址和資料範圍等輸入內容， 接著會執行所需的動作
  - 每個附屬裝置都要檢驗函式代碼、輸入數量、起始位址、整體範圍，還 有實際執行讀取、由附屬裝置定義的函式



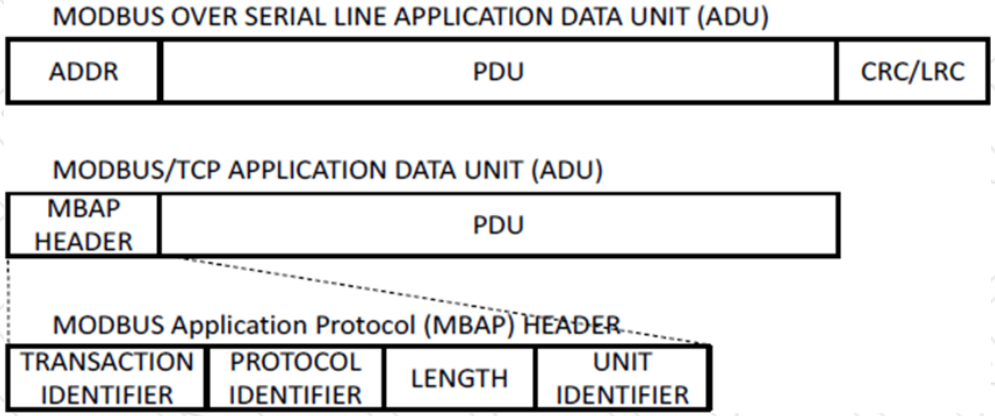
PDU (2/2)

- PDU為Modbus 應用協定規格的核心，用於定義協定所使用的資料類型、如何使用存取資料。
- Example : 讀取40001~40002的值

Function Code	Function-Specific Data			
Read Holding Registers	Data Address		Number	
03	00	00	00	02

ADU(1/3)

- 常見的ADU類型
  - Modbus-TCP
  - Modbus RTU
  - Modbus ASCII
- 常用的通訊模組
  - TCP/IP(Modbus-TCP)
  - RS485(Modbus RTU, Modbus ASCII)



ADU(2/3)

Modbus Application Protocol (MBAP) Header						
Transaction Identifier		**Protocol identifier**		**Length**		**Unit Identifier**
00	01	00	00	00	06	01
2 Bytes		2 Bytes		2 Bytes		1 Byte
用以辨識傳送與接 收封包對應		MODBUS固定為0		後續資料長度		遠端Slave設備辨識碼 (Slave ID 1~247)

ADU(3/3)

- 除了 Modbus 協定的 PDU 核心所定義的功能之外，根據不同類型包含其指定內容。

- Example : Modbus-TCP

讀取Device ID為1的設備上40001~40002的值

Modbus Application Protocol (MBAP) Header							Modbus PDU				
Transaction ID		Protocol		Length		UnitID	FC	Data Address		Number	
00	01	00	00	00	06	01	03	00	00	00	02

Function Code(1/3)

- 等級 0 代碼
  - 等級 0 代碼通常被視為實用 Modbus 裝置的極限最小值，能夠讓主要裝置讀寫資料模式。
- 03 Read Holding Registers(讀取多個保存暫存器)
- 16 Write Multiple Registers(寫入多個保存暫存器)

Function Code(2/3)

- 等級 1 代碼
  - 等級 1 函式代碼包含了存取資料模式所有類型所需的其他代碼。 原始定義的列表包含了函式代碼 7 (讀取例外)。 然而，目前的規格把此代碼定義為僅限序列的代碼。
- 01 Read Coils(讀取 Coil)
- 02 Read Discrete Inputs(讀取離散輸入)
- 04 Read Input Registers(讀取輸入暫存器)
- 05 Write Single Coil(寫入單一 Coil)
- 06 Write Single Register(寫入單一暫存器)
- 07 Read Exception Status-Serial Line only(讀取例外狀態)

Function Code(3/3)

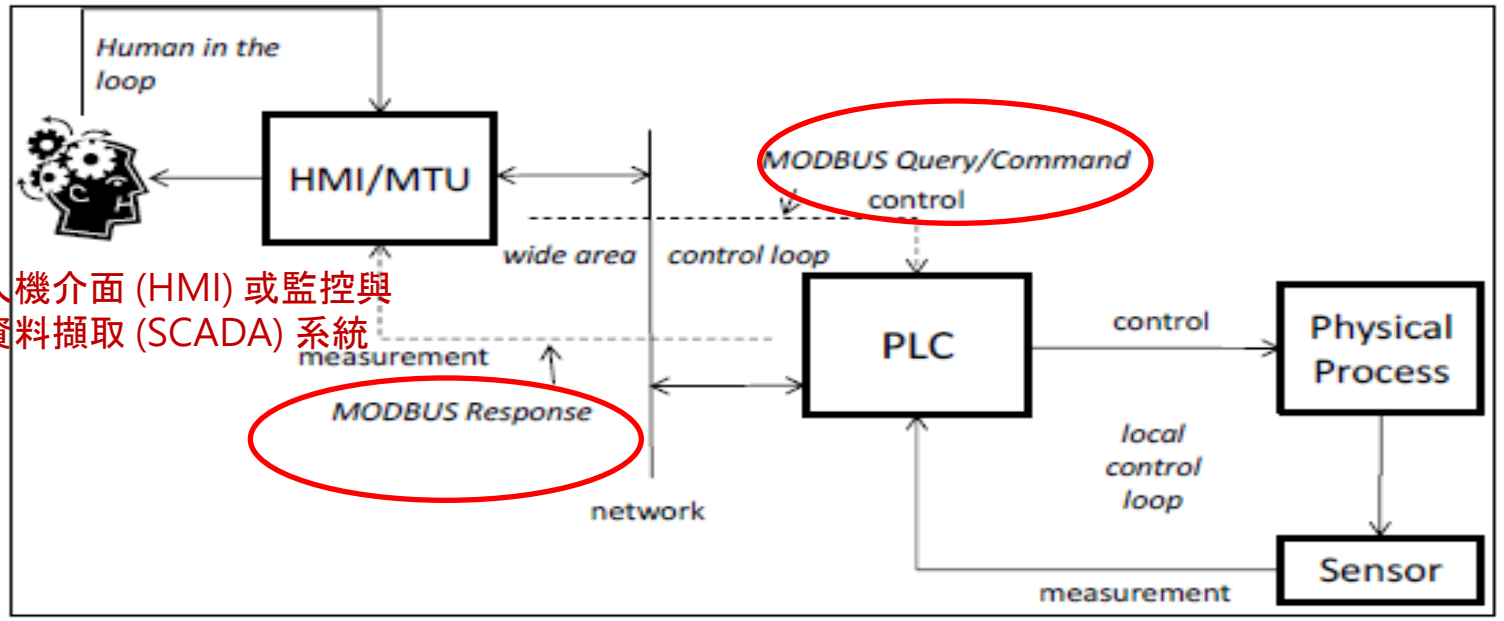
- 等級 2 代碼
  - 等級 2 函式代碼是更專門的代碼，實作機率也較低。 舉例來說，讀/寫多個暫存器可能有助於減少需求-回應週期的整體數量，但是仍然可以透過等級 0 代碼來實作此行為。
- 15 Write Multiple Coils(寫入多個 Coil)
- 20 Read File Record(讀取檔案紀錄)
- 21 Write File Record(寫入檔案紀錄)
- 22 Mask Write Register(遮罩寫入暫存器)
- 23 Read/Write Multiple Registers(讀/寫多個暫存器)
- 24 Read FIFO Queue(讀取 FIFO)

Modbus協定摘要

Modbus Function Code	說 明
01	Read Coil Status (output relay)
02	Read Input Status (input relay)
03	Read Holding Registers (output register)
04	Read Input Registers
05	Force Single Coil
06	Preset Single Register
15	Force Multiple Coils
16	Preset Multiple Registers
65 to 72	開放給使用者定義
100 to 110	開放給使用者定義

Typical SCADA 應用範例

- Modbus 是一種需求-回應協定，採用主從架構實作而成，主從架構的通訊作業會成雙成對的出現，必須有個裝置啟動需求並等候回應



master terminal unit (MTU)

Typical Industrial Control System Feedback Control Loops

Gao, W., Morris, T., "On Cyber Attacks and Signature Based Intrusion Detection for MODBUS Based Industrial Control Systems.", The Journal for Digital Forensics, Security and Law (JDFSL)., Volume 9, No. 1. 2014.

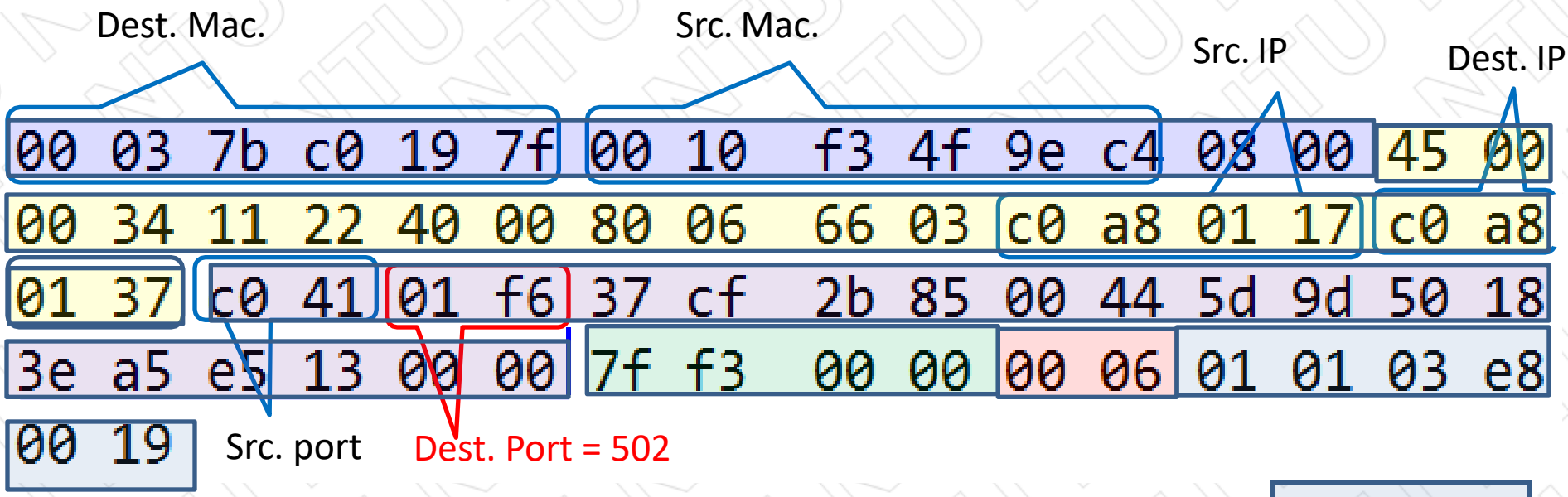
Modbus協定範例

```
01 01 03 e8 00 19
```

Query Message	Content byte	
Device Address	01	
Function Code	01	
Start Address (Hi byte)	03	1000
Start Address (Lo byte)	E8	
No. of points (Hi byte)	00	25
No. of points (Lo byte)	19	

發出request指令，向Device Address 01詢問讀取Relay Address 1001 – 1025 的Digit Input 資料，Start Address為1001，讀取點數25

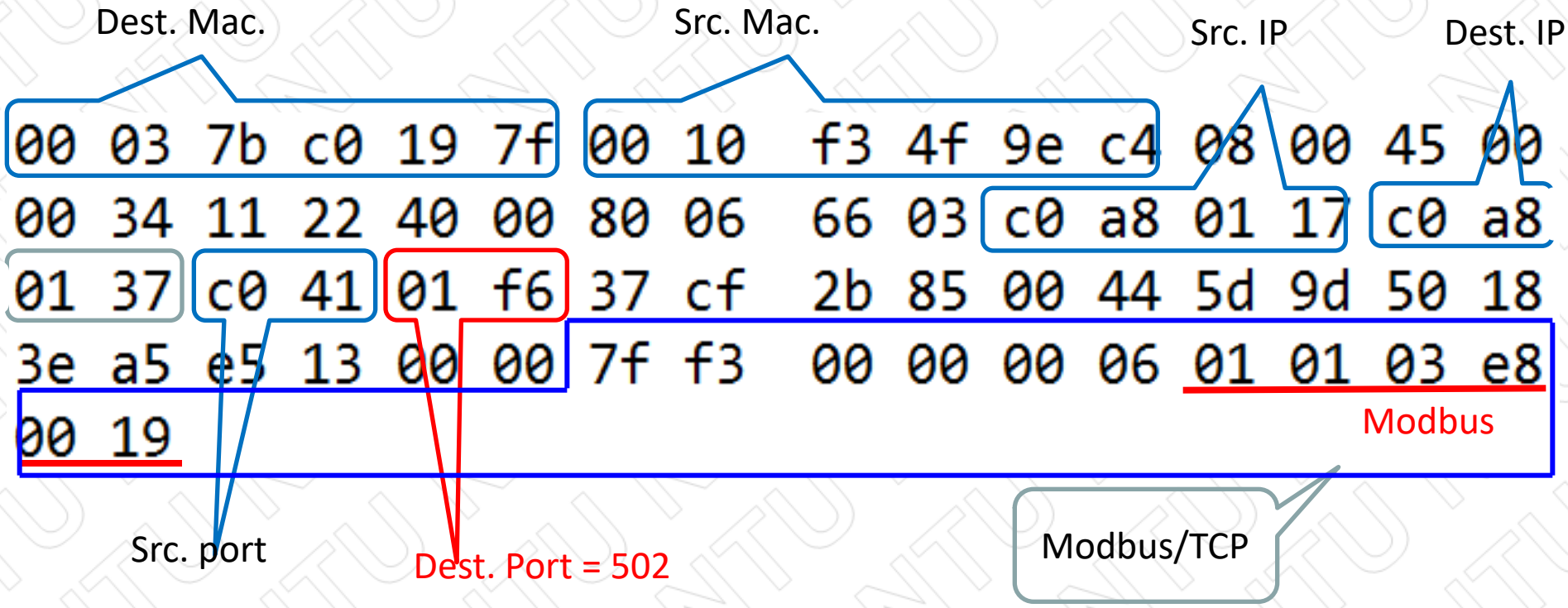
網路封包格式(1/2)



OSI Layer 包裝處理

Application	準備跟PLC請求之Modbus協定內容
Presentation	對Modbus協定內容進行hex encodes
Session	增加 Transaction ID
Transport	增加 TCP header
Network	增加 IP header
Data Link	增加 Ethernet header
Physica	實體逐bytes送出

網路封包格式(2/2)


Dest. Mac.
Src. Mac.
Src. IP
Dest. IP
00 03 7b c0 19 7f 00 10 f3 4f 9e c4 08 00 45 00
00 34 11 22 40 00 80 06 66 03 c0 a8 01 17 c0 a8
01 37 c0 41 01 f6 37 cf 2b 85 00 44 5d 9d 50 18
3e a5 e5 13 00 00 7f f3 00 00 00 06 01 01 03 e8
00 19
Modbus
Src. port
Dest. Port = 502
Modbus/TCP

Modbus Function Code-1 Example

Query

Query Message	通信內容 十六進位
Device Address	11
Function Code	01
Start Address (Hi byte)	00
Start Address (Lo byte)	13
No. of points (Hi byte)	00
No. of points (Lo byte)	0D

向Device Address 17 讀取Relay Address 20 – 32 的DI 資料，通信規約內Start Address 為19($13)，讀取點數13 ($0D) 。

Response

Response Message	通信內容 十六進位
Device Address	11
Function Code	01
Byte count	02
Data (Relay 27 – 20)	D3
Data (Relay 32 – 28)	17

回傳 2 bytes資料，以一個Byte (8 bits) 為一組，每一個Bit 表示一點Relay On/Off 狀態

- Data 1(Relay 27 – 20)的狀態($D3)為 ON-ON-OFF-ON-OFF-OFF-ON-ON (11010011)。
- Data(Relay 32 – 28)的狀態($17)為 OFF-ON-OFF-ON-ON-ON。（前面3 bit 不算，00010111 ）

Modbus Function Code-3 Example

Query

Query Message	通信內容 十六進位
Device Address	29
Function Code	03
Start Address (Hi byte)	02
Start Address (Lo byte)	FC
No. of registers (Hi byte)	00
No. of registers (Lo byte)	06

Query: 向由Device Address 41 讀取 Register Address 40765 – 40770 的 A0 資料，通信規約內Start Address 為 764($02FC)，讀取點數6。

Response: 回覆6個整數值共12bytes，值分別為99、12336、-1417、789、767、1 。

Response

Response Message	通信內容 十六進位
Device Address	29
Function Code	03
Byte count	0C
Data-1 (Hi byte)	00
Data-1 (Lo byte)	63
Data-2 (Hi byte)	30
Data-2 (Lo byte)	30
Data-3 (Hi byte)	FA
Data-3 (Lo byte)	77
Data-4 (Hi byte)	03
Data-4 (Lo byte)	15
Data-5 (Hi byte)	02
Data-5 (Lo byte)	FF
Data-6 (Hi byte)	00
Data-6 (Lo byte)	01

Modbus Function Code-5 Example

寫入單點DO 資料

Query

Query Message	通信內容 十六進位
Device Address	0A
Function Code	05
Start Address (Hi byte)	00
Start Address (Lo byte)	0B
Force Data (Hi byte)	FF
Force Data (Lo byte)	00

Response

Response Message	通信內容 十六進位
Device Address	0A
Function Code	05
Start Address (Hi byte)	00
Start Address (Lo byte)	0B
Force Data (Hi byte)	FF
Force Data (Lo byte)	00

向Device Address 10 寫入Relay Address 10012 的D0 資料，通信規約內Start Address 為11($0B)。如果設定為ON 於Force Data 設定十六進位0xFF00，如果設定為OFF 於Force Data 設定十六進位0x0000。

以Query Message 作為Response Message 傳回

Modbus Function Code-6 Example

寫入單點AO 資料

Query

Query Message	通信內容 十六進位
Device Address	0D
Function Code	06
Start Address (Hi byte)	00
Start Address (Lo byte)	6F
Preset Data (Hi byte)	03
Preset Data (Lo byte)	E7

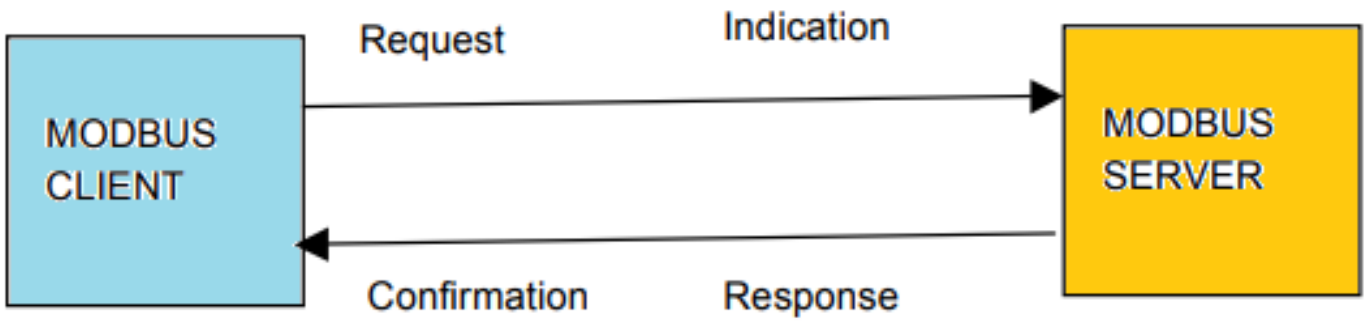
向Device Address 13 寫入Register Address 40112 的A0 資料，通信規約內Start Address 為111($006F)。設定16 bits 整數值為999 即是十六進位0x03E7。

Response

Response Message	通信內容 十六進位
Device Address	0D
Function Code	06
Start Address (Hi byte)	00
Start Address (Lo byte)	6F
Preset Data (Hi byte)	03
Preset Data (Lo byte)	E7

以Query Message 作為Response Message 傳回

Modbus Communication



Request

By the client to initiate a transaction

Indication

Request message received on the Server side

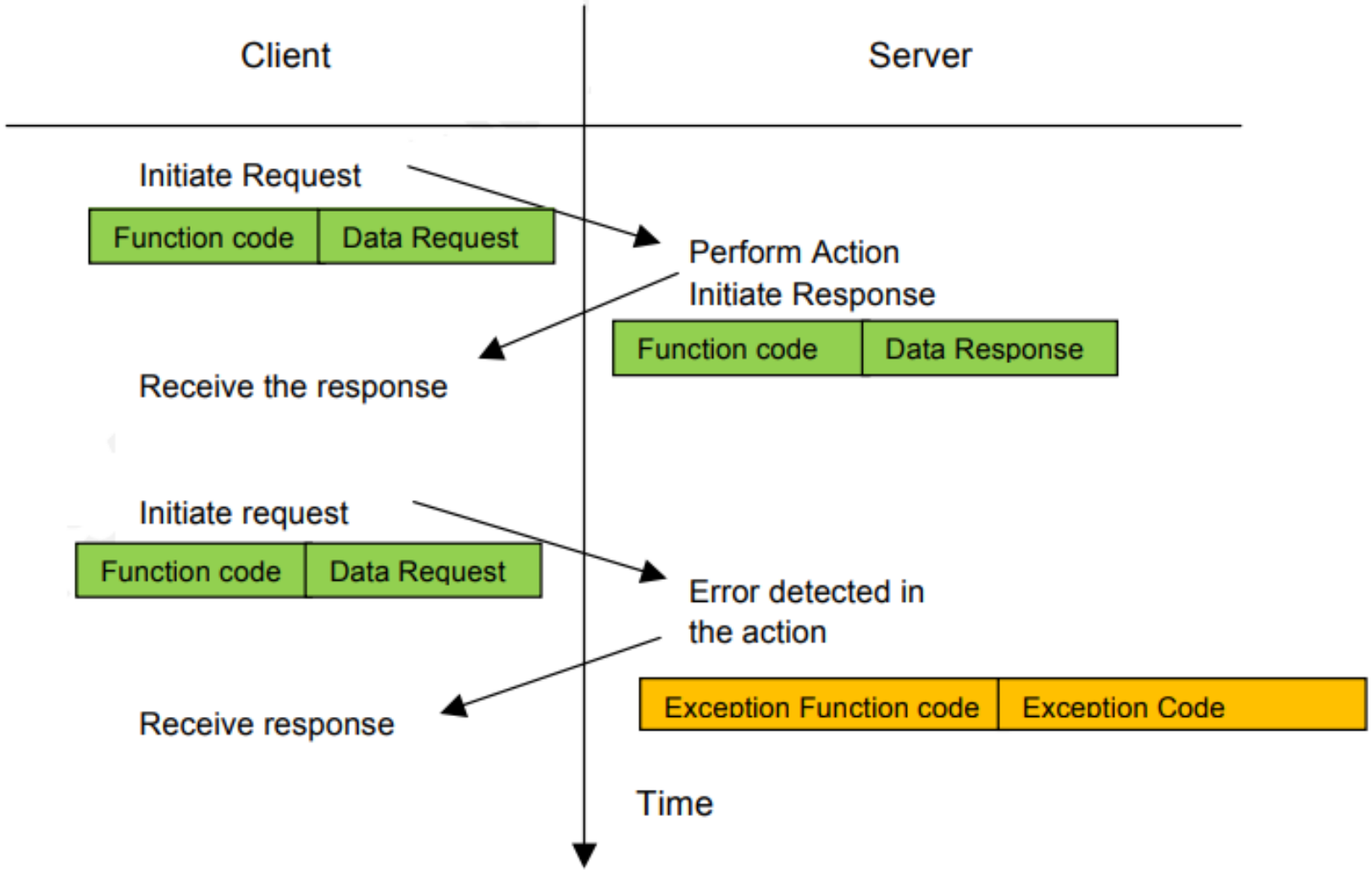
Response

Response message sent by the Server

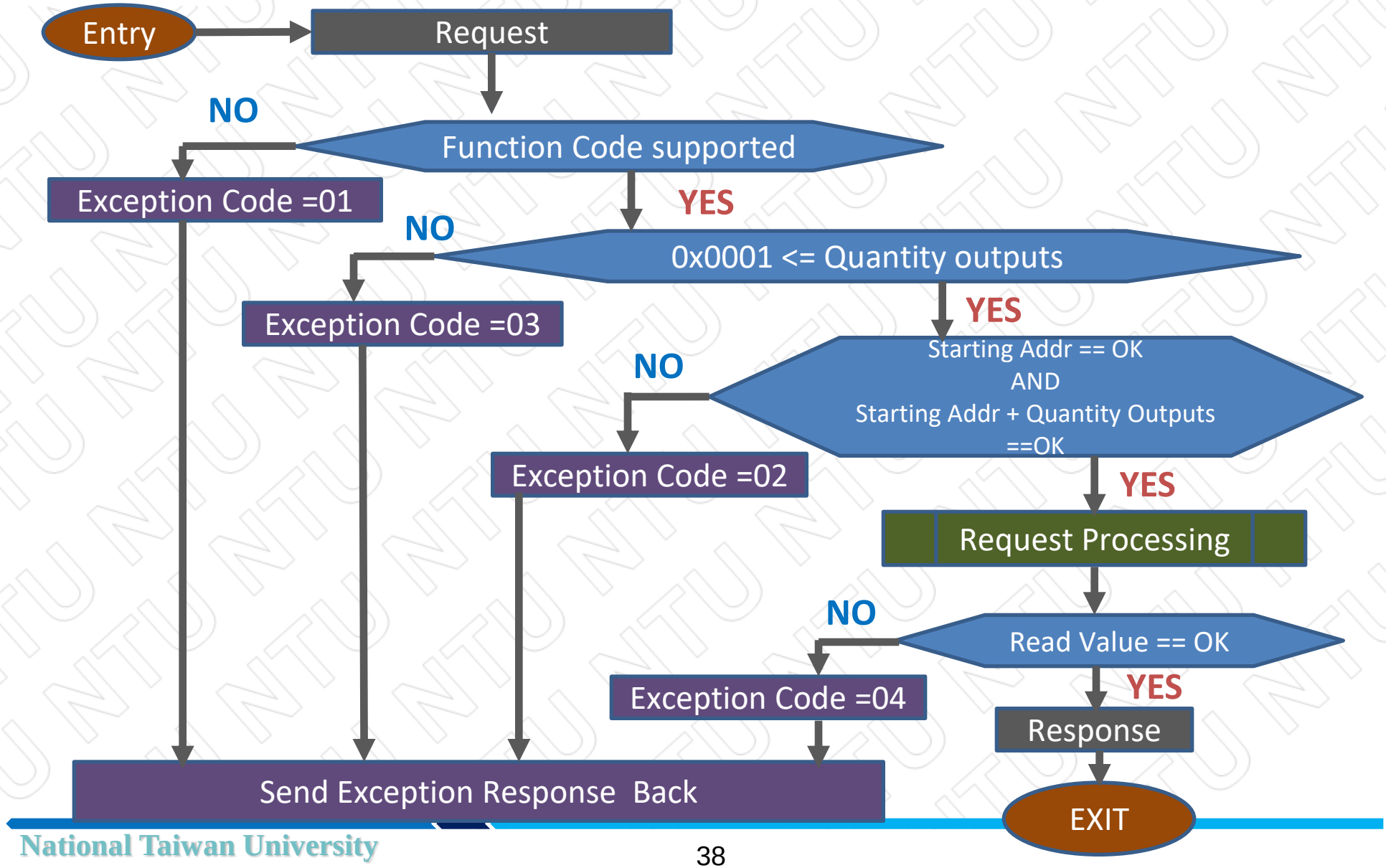
Confirmation

Response message received on the client side

Modbus Transactions



Modbus Function Code Flow



Modbus Exception Code

功能碼	名稱	說明
01	Illegal function	不支援的功能
02	Illegal data address	不合法的地址
03	Illegal data value	不合法的資料值
04	Slave Device failure	Slave 設備失效
05	Acknowledge	確認(命運執行中,但需要更久時間)
06	Slave device busy	Slave 設備忙碌

Modbus FC-2 (Read Discrete Inputs)

Query:

定義	通信內容(Hex)
Slave Address	11
Function Code	02
Data Address(Hi)	00
Data Address(Low)	C4(196,offset #10197)
Number(Hi)	00
Number(Low)	16(hex 22→197~218)

Response:

定義	通信內容(Hex)
Slave Address	11
Function Code	02
Number	03
Data	AC(1010-1100)
Data	DB(1101-1011)
Data	35(0011-0101)

向Device Address 17 讀取Address 197~218的DI 資料，
Start Address 為10197(0xC4)，讀取點數22(0x16) 。

Modbus FC-3 (Read Holding Registers)

Query:

定義	通信內容(Hex)
Slave Address	11
Function Code	03
Data Address(Hi)	00
Data Address(Low)	6B(hex107, +offset = #40108)
Number(Hi)	00
Number(Low)	03(40108~40110)

Response:

定義	通信內容(Hex)
Slave Address	11
Function Code	03
Number	06
Data(Hi)	AE
Data(Low)	41
Data(Hi)	56
Data(Low)	52
Data(Hi)	43
Data(Low)	40

Query: 向Device Address 17(0x11)讀取Register Address 40108 – 40110 的資料，通信規約內Start Address 為107($006B)，讀取點數3。Response: 回覆3個值共6 bytes

Modbus Fc-4 (Read Input Registers)

Query:

定義	通信內容(Hex)
Slave Address	11
Function Code	04
Data Address(Hi)	00
Data Address(Low)	6B(107)
Number(Hi)	00
Number(Low)	03

Query:
向Device Address 17(0x11) 讀取 Register Address 30108 – 30110 的資料，通信規約內Start Address 為107(0x6B)，讀取點數3。

Response:

定義	通信內容(Hex)
Slave Address	11
Function Code	04
Number	06
Data(Hi)	AE
Data(Low)	41
Data(Hi)	56
Data(Low)	52
Data(Hi)	43
Data(Low)	40

Response: 回覆3個值共6 bytes

Modbus Fc-5 (Write Single Coil)

Query:

定義	通信內容(Hex)
Slave Address	11
Function Code	05
Data Address(Hi)	00
Data Address(Low)	AC(hex172 →#173)
Status(Hi)	FF
Status(Low)	00

ON : 0xFF00
OFF : 0x0000

Response:

定義	通信內容(Hex)
Slave Address	11
Function Code	05
Data Address(Hi)	00
Data Address(Low)	AC
Status(Hi)	FF
Status(Low)	00

以Query Message 作為 Response Message 傳回

向Device Address 17(0x11) 寫入Address 00173 的資料
Start Address 為172(0xAC)。
設定資料值0xFF00 (ON)

Modbus Fc-6 (Write Single Register)

Query:

定義	通信內容(Hex)
Slave Address	11
Function Code	06
Data Address(Hi)	00
Data Address(Low)	01 (+offset 40001 = #40002)
Value(Hi)	00
Value(Low)	03

Response:

定義	通信內容(Hex)
Slave Address	11
Function Code	06
Data Address(Hi)	00
Data Address(Low)	01
Value(Hi)	00
Value(Low)	03

以Query Message 作為 Response Message 傳回

向Device Address 17(0x11)寫入Address 40002 的資料
Start Address 為01(0x01)。
設定資料值0x0003

Modbus Fc-15 (Write multiple Coils)

Query:

定義	通信內容(Hex)
Slave Address	11
Function Code	0F
Data Address(Hi)	00
Data Address(Low)	13 (hex19 , #20)
Number(Hi)	00
Number(Low)	0A
Number of Data	02
Value	CD(1100-1101)
Value	01(0000-0001)

不夠的位數補0

Response:

定義	通信內容(Hex)
Slave Address	11
Function Code	0F
Data Address(Hi)	00
Data Address(Low)	13
Number(Hi)	00
Number(Low)	0A

向Device Address 17(0x11) 寫入Address 20 的資料
Start Address 為19(0x13)。
寫入10點數資料值(0x0A) ,
一個Byte 即8 bits 為一組，10 bits需2 bytes
設定資料值0x CD01

Modbus Fc-16 (Write Multiple Registers)

Query:

定義	通信內容(Hex)
Slave Address	11
Function Code	10
Data Address(Hi)	00
Data Address(Low)	01(#40002)
Number(Hi)	00
Number(Low)	02
Number of Data	04
Value(Hi)	00 (write to #40002)
Value(Low)	0A (write to #40002)
Value(Hi)	01 (write to #40003)
Value(Low)	02(write to #40003)

Response:

定義	通信內容(Hex)
Slave Address	11
Function Code	10
Data Address(Hi)	00
Data Address(Low)	01
Number(Hi)	00
Number(Low)	02

向Device Address 17(0x11)
寫入Address 40002與40003 的資料
Start Address 為01(0x01)
設定2個數值,
一個數值為2bytes,共需4 bytes
設定40002資料值為0x000A
設定40003資料值為0X0102

(Lab)
Modbus資料存儲區

練習準備

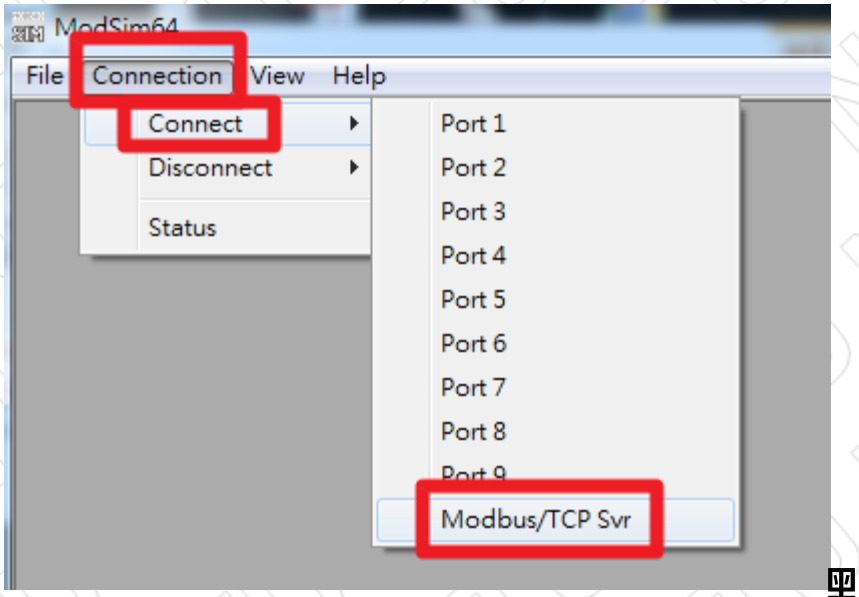
- 需開啟兩個VM 或者在不同電腦執行
  C:\Users\root\Desktop\modsim64\ModSim64.exe

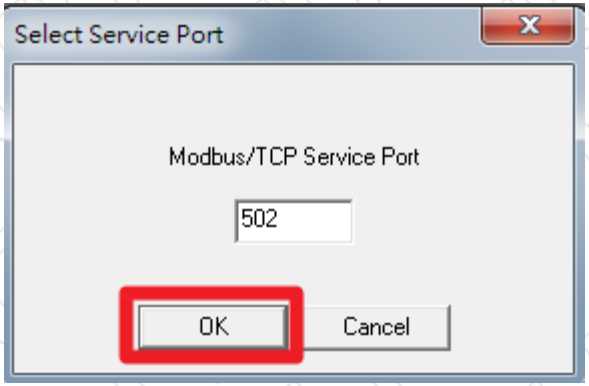
- ModScan
  模擬執行SCADA的HMI介面軟體，可下Modbus request 命令更改或讀取PLC (ModSim)記憶體位址之值

- ModSim
  模擬執行PLC，接受Modbus指令，更改設定值，並回應回傳值

啟動ICS模擬環境

- ModSim　　啟動與操作
  - 於VM1執行modsim64.exe
  - Connection/connect 選 Modbus/TCP Svr port=502

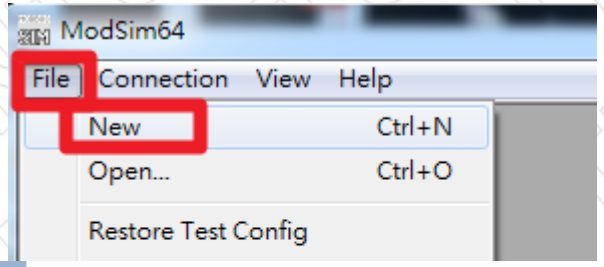




點選OK即完成ModSim初步設定

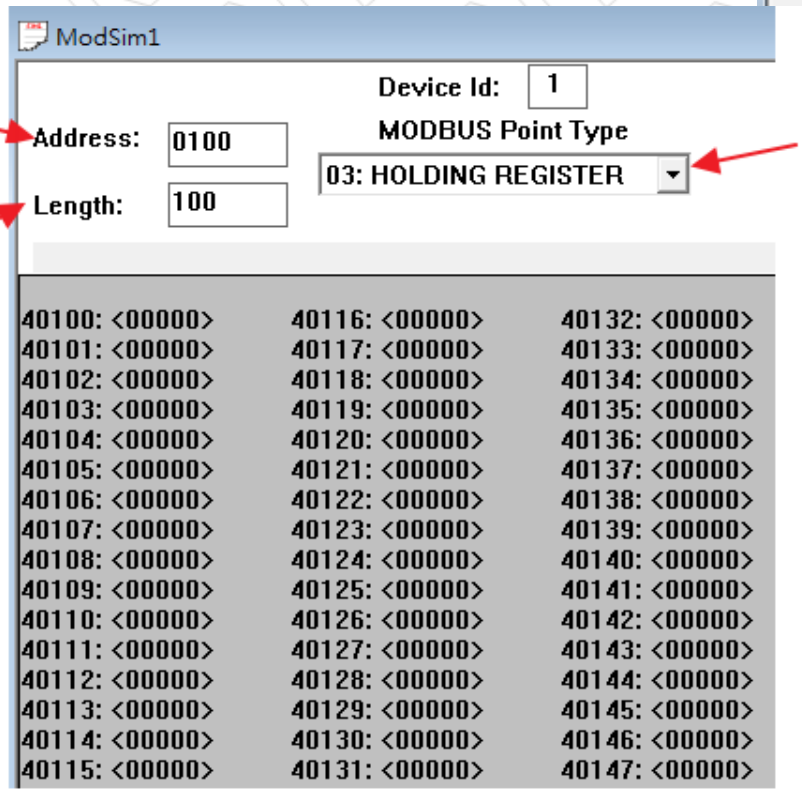
啟動ICS模擬環境

- ModSim 啟動與操作
  - File/new
  - 模擬PLC即開始執行



記憶體開始位址

顯示值位址數
此時顯示
40100~40199



01:COIL STATUS (0/1)
02:INPUT STATUS (0/1)
03:HOLDING REGISTER
04:INPUT REGISTER

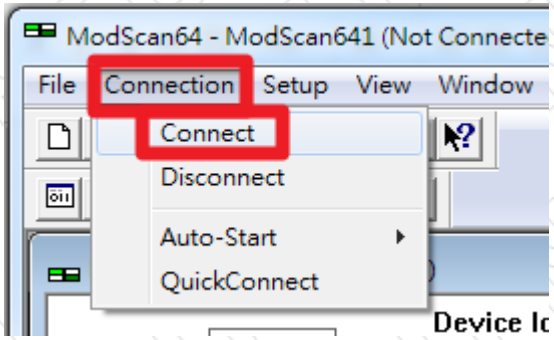
若出現

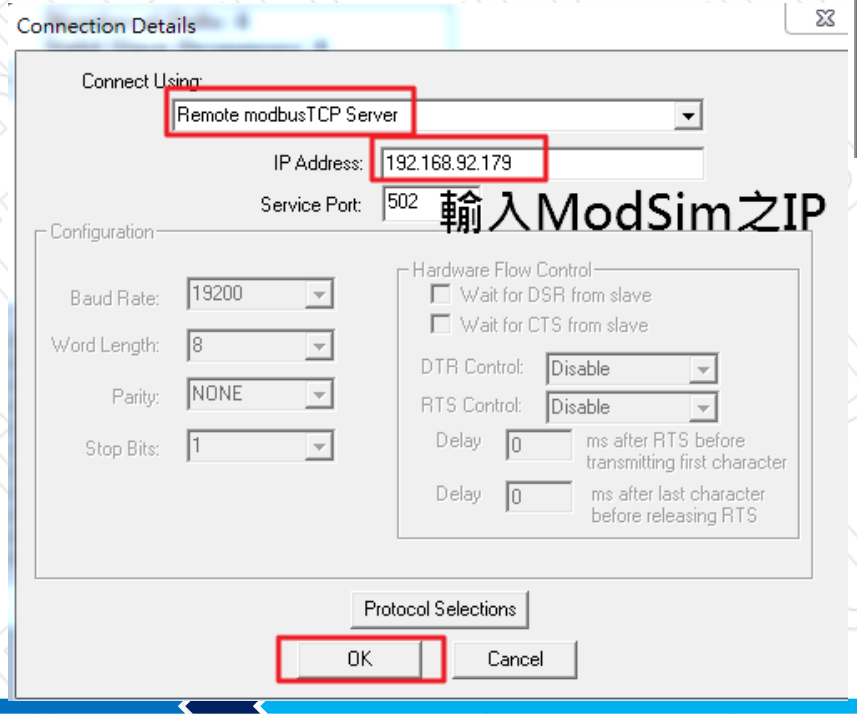
*** DEMO TIME ELAPSED! ***

試用時間到,
需重啟 File/new

啟動ICS模擬環境

- ModScan 啟動與操作
  - Check ModSim之IP : (ipconfig)
  - 在VM2執行ModScan64.exe
  - Connection/connect

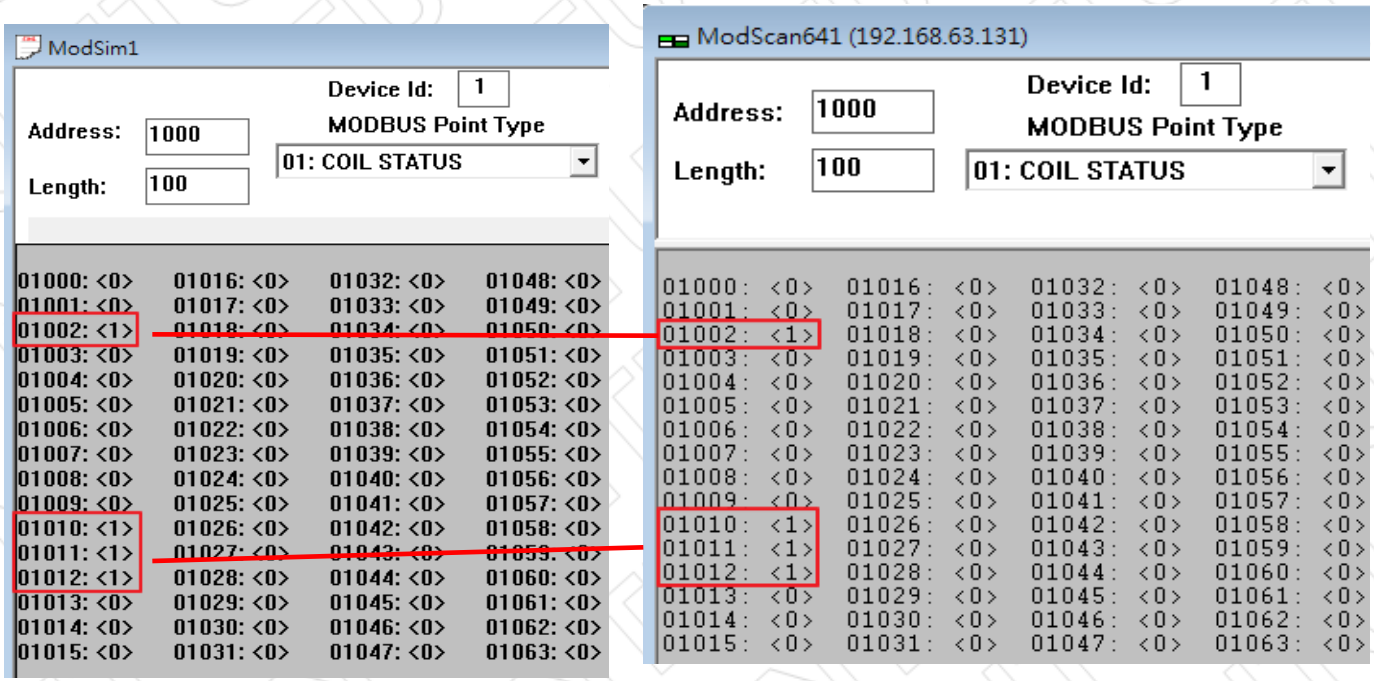




修改設定與ModSim同步

啟動ICS模擬環境

- ModScan　　啟動與操作
  - 點選任一位址輸入數值，確認ModSim是否有同步更新
  - ModScan與ModSim設定相同記體體區塊及位址，Modscan即可修改值內容，模擬送命令給PLC修改設定值



練習 (Read Coils)

Query:

定義	通信內容(Hex)
Slave Address	
Function Code	
Data Address(Hi)	
Data Address(Low)	
Number(Hi)	
Number(Low)	

Response:

定義	通信內容(Hex)
Slave Address	
Function Code	
Number	
Data(#27~#20)	
Data(#35~#28)	

向Device Address 17 讀取Relay Address 20 – 32 的DI 資料，
Start Address 為19($13)，讀取點數13 ($0D) 。

練習 (Read Coils)-A

Query:

定義	通信內容(Hex)
Slave Address	11
Function Code	01
Data Address(Hi)	00
Data Address(Low)	13
Number(Hi)	00
Number(Low)	0D

Response:

定義	通信內容(Hex)
Slave Address	11
Function Code	01
Number	02
Data(#27~#20)	D3
Data(#35~#28)	17

向Device Address 17 讀取Relay Address 20 – 32 的DI 資料，
Start Address 為19($13)，讀取點數13 ($0D) 。

練習寫到ModSim中

練習 (Read Holding Registers)

Query

定義	通信內容(Hex)
Slave Address	
Function Code	
Data Address(Hi)	
Data Address(Low)	
Number(Hi)	
Number(Low)	

Response

定義	通信內容(Hex)
Slave Address	
Function Code	
Number	
Data(Hi)	
Data(Low)	
Data(Hi)	
Data(Low)	

Query: 向由Device Address 41 讀取Register Address 40765 – 40770 的A0 資料，通信規約內Start Address 為764($02FC)，讀取點數6。

練習 (Read Holding Registers) -A

Query

定義	通信內容(Hex)
Slave Address	29
Function Code	03
Data Address(Hi)	02
Data Address(Low)	FC
Number(Hi)	00
Number(Low)	06

Query: 向由Device Address 41
讀取Register Address 40765 – 40770 的A0 資料，
Start Address 為764($02FC)，讀取點數6。

練習寫到ModSim中

Response

定義	通信內容(Hex)
Slave Address	29
Function Code	03
Number	0C
Data(Hi)-1	00
Data(Low)-1	63
Data(Hi)-2	30
Data(Low)-2	30
Data(Hi)-3	FA
Data(Low)-3	77
Data(Hi)-4	03
Data(Low)-4	15
Data(Hi)-5	02
Data(Low)-5	FF
Data(Hi)-6	00
Data(Low)-6	01

練習 (Read Holding Registers)

Query

定義	通信內容(Hex)
Slave Address	
Function Code	
Data Address(Hi)	
Data Address(Low)	
Number(Hi)	
Number(Low)	

Response

定義	通信內容(Hex)
Slave Address	
Function Code	
Number	
Data(Hi)	
Data(Low)	
Data(Hi)	
Data(Low)	

Query: 向由Device Address 88
讀取Register Address 41012 – 41110 的A0 資料

練習 (Read Holding Registers) -A

Query

定義	通信內容(Hex)
Slave Address	58
Function Code	03
Data Address(Hi)	03
Data Address(Low)	F3
Number(Hi)	00
Number(Low)	63

Query: 向由Device Address 88($58)
讀取Register Address 41012 – 41110 的A0 資料，
➔Start Address 為1011($03F3)
➔讀取點數99, 需要198 bytes (C6)。

Response

定義	通信內容(Hex)
Slave Address	58
Function Code	03
Number	C6
Data(Hi)-1	00
Data(Low)-1	63
Data(Hi)-2	30
Data(Low)-2	30
Data(Hi)-3	FA
Data(Low)-3	77
Data(Hi)-4	03
Data(Low)-4	15
Data(Hi)-5	02
Data(Low)-5	FF
Data(Hi)-6	00
Data(Low)-6	01

練習 (Write single coil)

Query:

定義	通信內容(Hex)
Slave Address	
Function Code	
Data Address(Hi)	
Data Address(Low)	
Status(Hi)	
Status(Low)	

Response:

定義	通信內容(Hex)
Slave Address	
Function Code	
Data Address(Hi)	
Data Address(Low)	
Status(Hi)	
Status(Low)	

向Device Address 10 寫入Relay Address 10012 的D0 資料，
Start Address 為11($0B)。
設定狀態為ON

練習 (Write single coil)-A

Query:

定義	通信內容(Hex)
Slave Address	0A
Function Code	05
Data Address(Hi)	00
Data Address(Low)	0B
Status(Hi)	FF
Status(Low)	00

Response:

定義	通信內容(Hex)
Slave Address	0A
Function Code	05
Data Address(Hi)	00
Data Address(Low)	0B
Status(Hi)	FF
Status(Low)	00

向Device Address 10 寫入Relay Address 10012 的D0 資料，
Start Address 為11($0B)。
設定狀態為ON

練習寫到ModSim中

練習 (Write single coil)

Query:

定義	通信內容(Hex)
Slave Address	
Function Code	
Data Address(Hi)	
Data Address(Low)	
Status(Hi)	
Status(Low)	

Response:

定義	通信內容(Hex)
Slave Address	
Function Code	
Data Address(Hi)	
Data Address(Low)	
Status(Hi)	
Status(Low)	

向Device Address 36寫入Relay Address 10188 的D0 資料。
設定狀態為ON (FF00)

練習 (Write single coil)-A

Query:

定義	通信內容(Hex)
Slave Address	24
Function Code	05
Data Address(Hi)	00
Data Address(Low)	BB
Status(Hi)	FF
Status(Low)	00

Response:

定義	通信內容(Hex)
Slave Address	24
Function Code	05
Data Address(Hi)	00
Data Address(Low)	BB
Status(Hi)	FF
Status(Low)	00

向Device Address 36寫入Relay Address 10188 的D0 資料，
Start Address 為187($BB)。
設定狀態為ON (FF00)

練習寫到ModSim中

練習 (Write single Register)

Query

定義	通信內容(Hex)
Slave Address	
Function Code	
Data Address(Hi)	
Data Address(Low)	
Value(Hi)	
Value(Low)	

Response

定義	通信內容(Hex)
Slave Address	
Function Code	
Data Address(Hi)	
Data Address(Low)	
Value(Hi)	
Value(Low)	

向Device Address 13 寫入Register Address 40112 的A0 資料，
Start Address 為111($006F)。
設定整數值999 (0x03E7)。

練習 (Write single Register)-A

Query

定義	通信內容(Hex)
Slave Address	0D
Function Code	06
Data Address(Hi)	00
Data Address(Low)	6F
Value(Hi)	03
Value(Low)	E7

Response

定義	通信內容(Hex)
Slave Address	0D
Function Code	06
Data Address(Hi)	00
Data Address(Low)	6F
Value(Hi)	03
Value(Low)	E7

向Device Address 13 寫入Register Address 40112 的A0 資料，
Start Address 為111($006F)。
設定整數值999 (0x03E7)。

練習寫到ModSim中

練習 (Write single Register)

Query

定義	通信內容(Hex)
Slave Address	
Function Code	
Data Address(Hi)	
Data Address(Low)	
Value(Hi)	
Value(Low)	

Response

定義	通信內容(Hex)
Slave Address	
Function Code	
Data Address(Hi)	
Data Address(Low)	
Value(Hi)	
Value(Low)	

向Device Address 155 寫入Register Address 40692 的A0 資料。
設定整數值1576。

練習 (Write single Register)-A

Query

定義	通信內容(Hex)
Slave Address	9B
Function Code	06
Data Address(Hi)	02
Data Address(Low)	B3
Value(Hi)	06
Value(Low)	27

Response

定義	通信內容(Hex)
Slave Address	9B
Function Code	06
Data Address(Hi)	02
Data Address(Low)	B3
Value(Hi)	06
Value(Low)	27

向Device Address 155($9B) 寫入Register Address 40692 的A0 資料，
Start Address 為691($02B3)。
設定整數值1576(0x0627)。

練習寫到ModSim中

(Lab)
工控網路封包擷取實驗與工控命令分析解讀

監控ICS網路實驗

- 啟動Wireshark
- 啟動並確認ModScan與ModSim連線
- 修改ModScan封包內容(Coli Status / Holding Register)
- 觀察ModSim是否隨之改變
- 以Wireshark觀察傳送封包內容

練習準備

- 需開啟兩個VM 或者在不同電腦執行
  C:\Users\root\Desktop\modsim64\ModSim64.exe

- ModScan
  模擬執行SCADA的HMI介面軟體，可下Modbus request 命令更改或讀取PLC (ModSim)記憶體位址之值

- ModSim
  模擬執行PLC，接受Modbus指令，更改設定值，並回應回傳值

- VM1打開 Wireshark，紀錄練習過程之封包

A. 練習 (Read Coils)

Query:

定義	通信內容(Hex)
Slave Address	11
Function Code	01
Data Address(Hi)	00
Data Address(Low)	23
Number(Hi)	00
Number(Low)	0F

Response:

定義	通信內容(Hex)
Slave Address	11
Function Code	01
Number	02
Data(#27~#20)	D3
Data(#35~#28)	17

向Device Address 17 讀取Relay Address 20 – 32 的DI 資料，
Start Address 為31($23)，讀取點數15 ($0F) 。

練習寫到ModSim中

B. 練習 (Read Holding Registers)

Query

定義	通信內容(Hex)
Slave Address	29
Function Code	03
Data Address(Hi)	02
Data Address(Low)	CC
Number(Hi)	00
Number(Low)	05

Response

定義	通信內容(Hex)
Slave Address	29
Function Code	03
Number	0C
Data(Hi)-1	00
Data(Low)-1	63
Data(Hi)-2	FA
Data(Low)-2	77
Data(Hi)-3	03
Data(Low)-3	15
Data(Hi)-4	02
Data(Low)-4	FF
Data(Hi)-5	00
Data(Low)-5	01

Query: 向由Device Address 41
讀取Register Address 40717 – 40722 的A0 資料，
Start Address 為716($02CC)，讀取點數5。

練習寫到ModSim中

C. 練習 (Write single coil)

Query:

定義	通信內容(Hex)
Slave Address	0A
Function Code	05
Data Address(Hi)	00
Data Address(Low)	0D
Status(Hi)	FF
Status(Low)	00

Response:

定義	通信內容(Hex)
Slave Address	0A
Function Code	05
Data Address(Hi)	00
Data Address(Low)	0D
Status(Hi)	FF
Status(Low)	00

向Device Address 10 寫入Relay Address 10014 的D0 資料，
Start Address 為13($0D)。
設定狀態為ON

練習寫到ModSim中

D. 練習 (Write single coil)

Query:

定義	通信內容(Hex)
Slave Address	24
Function Code	05
Data Address(Hi)	00
Data Address(Low)	AA
Status(Hi)	FF
Status(Low)	00

Response:

定義	通信內容(Hex)
Slave Address	24
Function Code	05
Data Address(Hi)	00
Data Address(Low)	AA
Status(Hi)	FF
Status(Low)	00

向Device Address 36寫入Relay Address 10171 的D0 資料，
Start Address 為170($AA)。
設定狀態為ON (FF00)

練習寫到ModSim中

F. 練習 (Write single Register)

Query

定義	通信內容(Hex)
Slave Address	0D
Function Code	06
Data Address(Hi)	00
Data Address(Low)	AF
Value(Hi)	03
Value(Low)	E7

Response

定義	通信內容(Hex)
Slave Address	0D
Function Code	06
Data Address(Hi)	00
Data Address(Low)	AF
Value(Hi)	03
Value(Low)	E7

向Device Address 13 寫入Register Address 40176 的A0 資料，
Start Address 為175($00AF)。
設定整數值999 (0x03E7)。

練習寫到ModSim中

G. 練習 (Write single Register)

Query

定義	通信內容(Hex)
Slave Address	9B
Function Code	06
Data Address(Hi)	01
Data Address(Low)	B3
Value(Hi)	06
Value(Low)	38

Response

定義	通信內容(Hex)
Slave Address	9B
Function Code	06
Data Address(Hi)	01
Data Address(Low)	B3
Value(Hi)	06
Value(Low)	38

向Device Address 155($9B) 寫入Register Address 40436 的A0 資料，
Start Address 為435($01B3)。
設定整數值1592(0x0638)。

練習寫到ModSim中

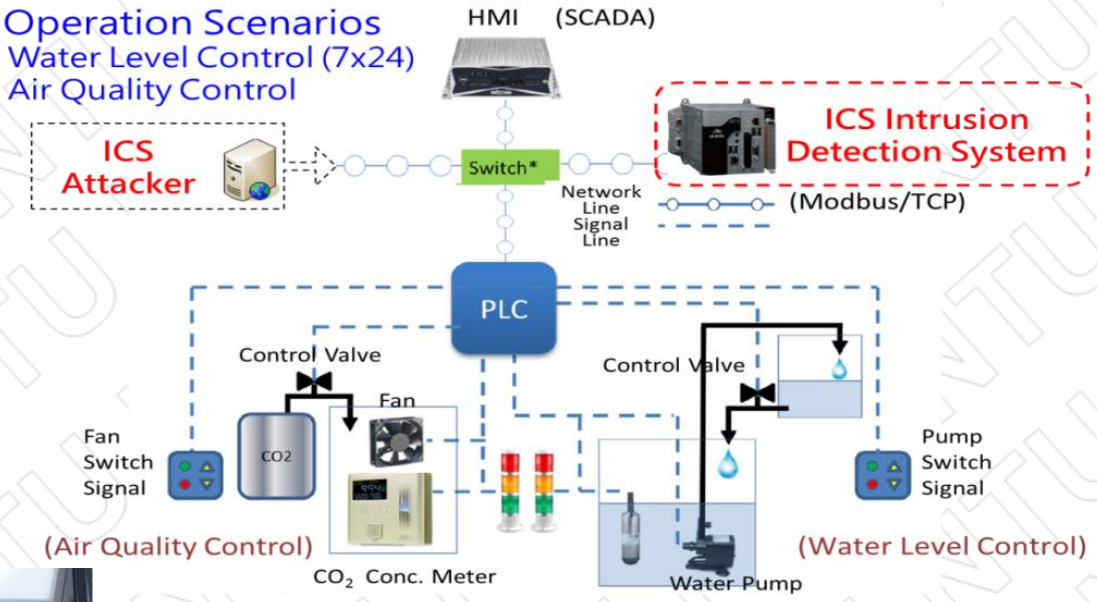
作業 1-A

1. 對於練習A~F，截圖列出下列結果
   1) ModSim 對應位址及值內容
   2) Wireshark中，找出該筆封包內容，指出該筆封包應用層內容

(Lab)
工控系統網路封包分析

III ICS Test Bed

- 2工控運轉程序，水處理水位控制及空汙自動排氣
- SCADA HMI & PLC架構
- Modbus/Tcp、OPC/UA工業標準控制協定



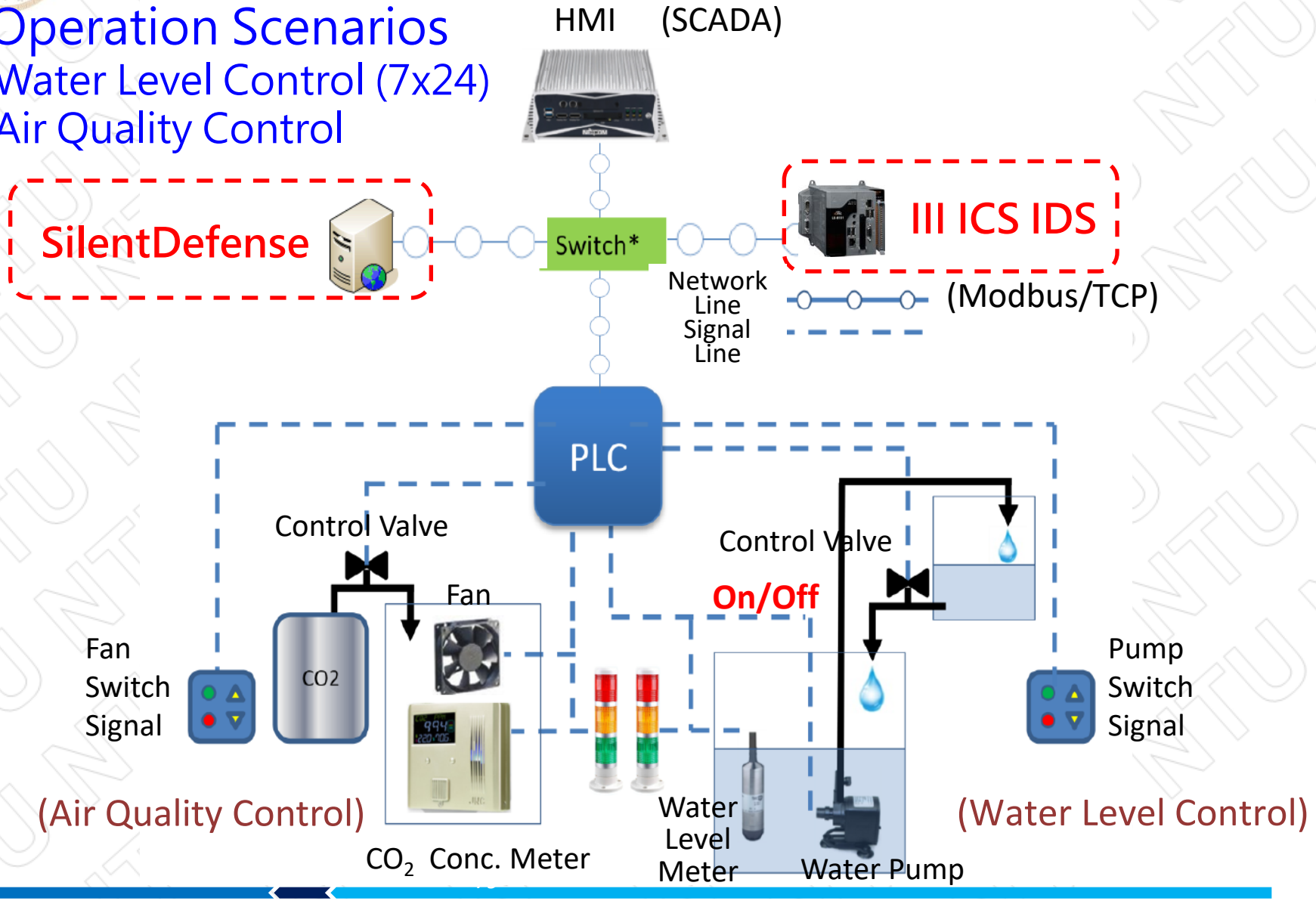
ICS TestBed

Video: https://youtu.be/5nfufZwuvsw

III Testbed

2 Operation Scenarios

- Water Level Control (7x24)
- Air Quality Control



工控系統網路封包分析實驗

- 啟動Wireshark
- 讀取normal_1hr.pcapng

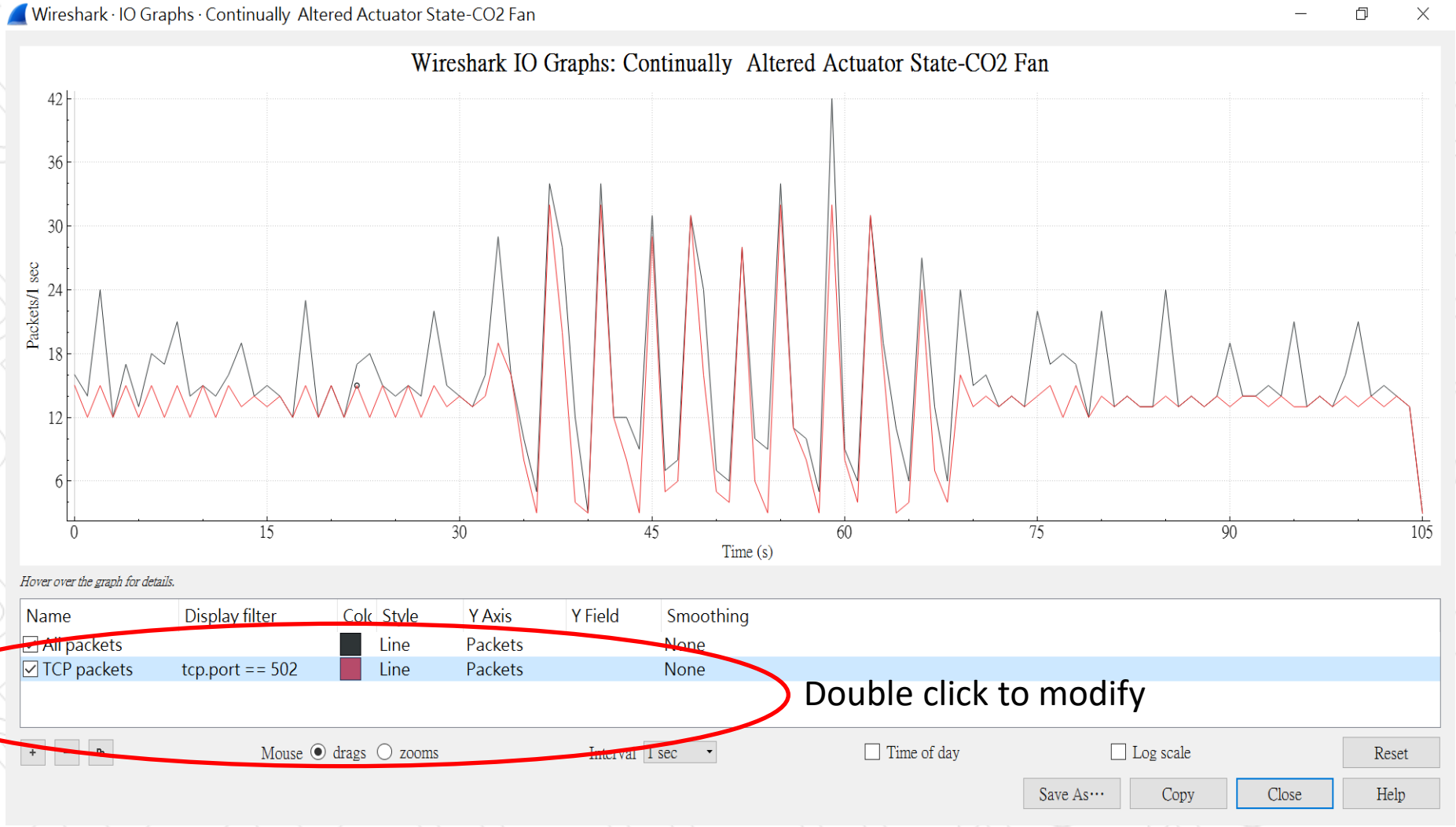
實驗內容

資料側錄來源：判斷設備位址與控制點 (Test Bed)

- 執行步驟：
  1. 判斷Master與Slave的IP位址
  2. 檢查使用過那些Function code
  3. 檢查讀取的設定點以及資料類型
  4. 檢查寫入的設定點以及資料類型
- 作業：

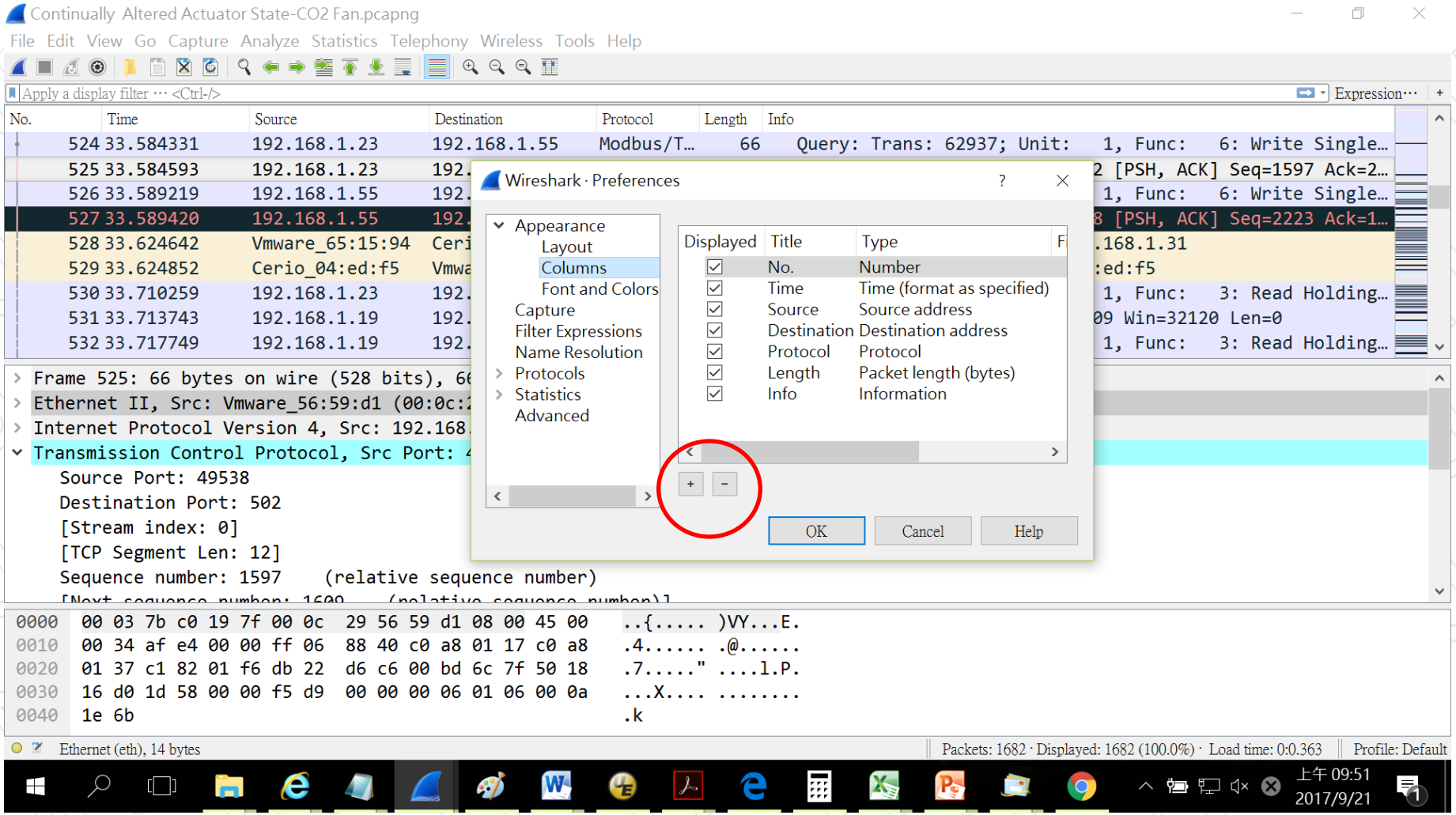
Network Spike 分析

Statistics/ I/O Graph:



Finding Interesting Packets

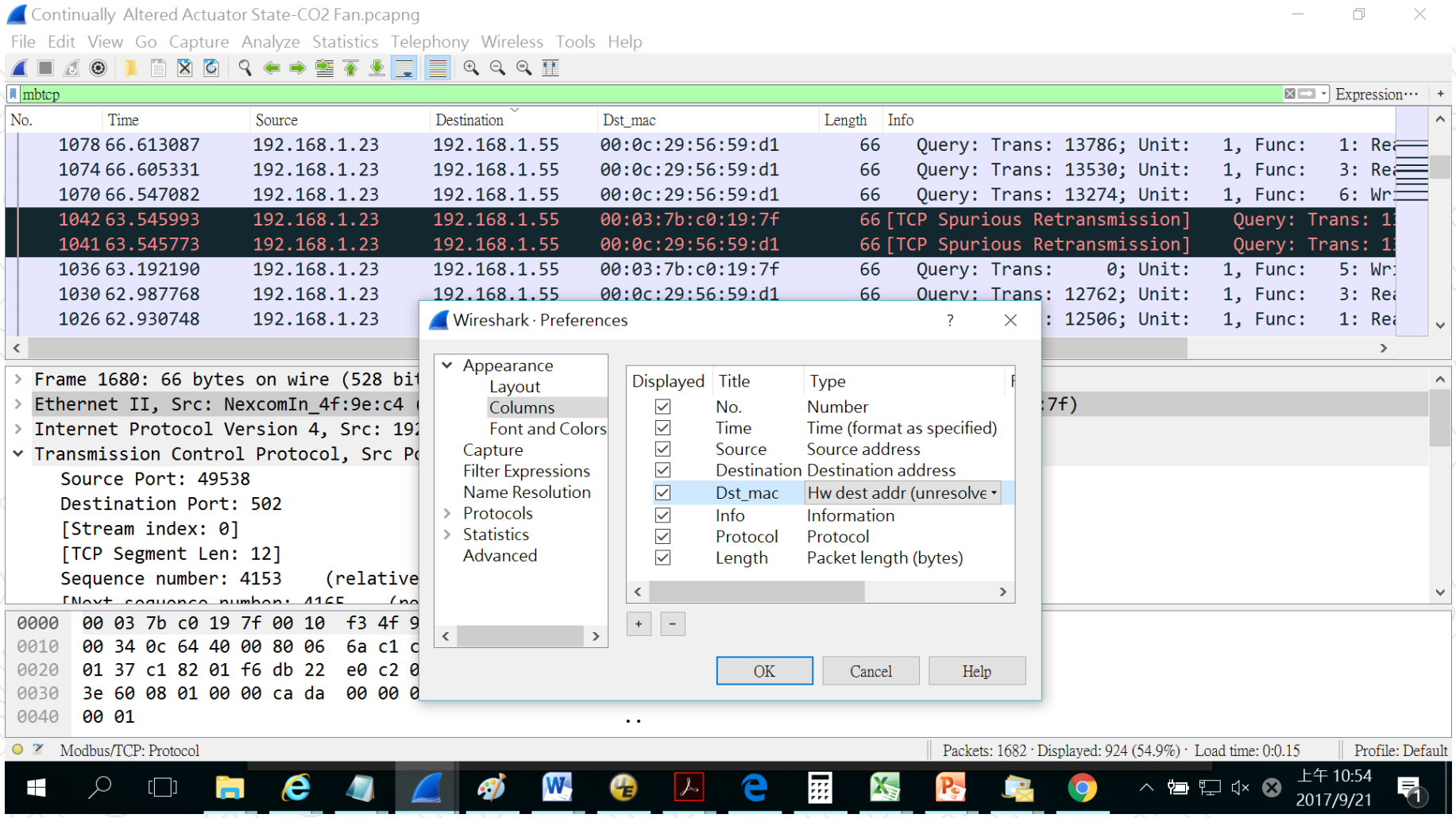
Edit /Preferences …:



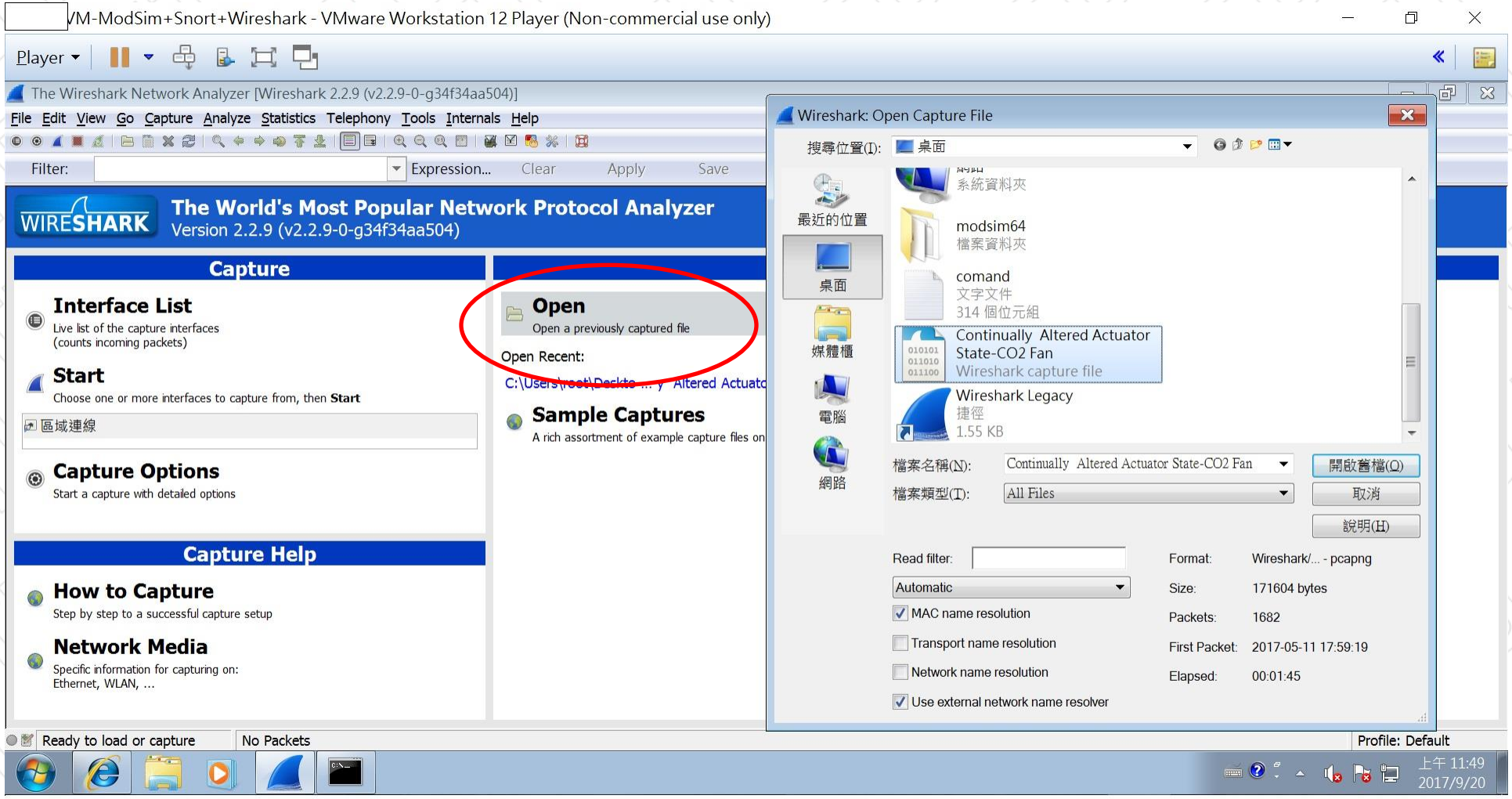
Adding Useful Sort Columns

Edit /Preferences …:

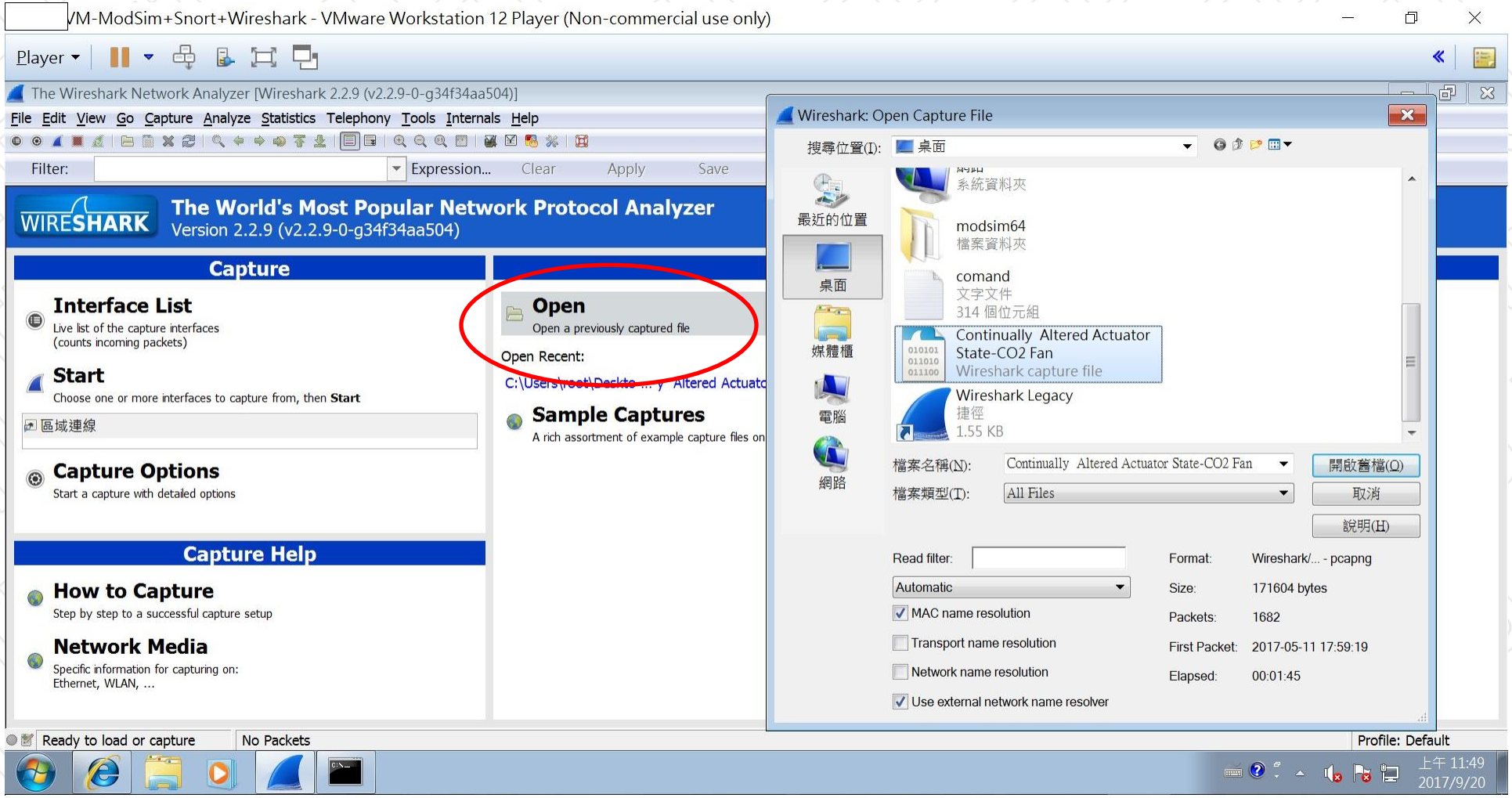
- ✓ Add a Dst_mac column with Hw dest addr,
- ✓ Sorting by Dst_Mac, add a filter with mbtcp, sort



Wireshark – Open a Capture File



Wireshark – Open a Capture File



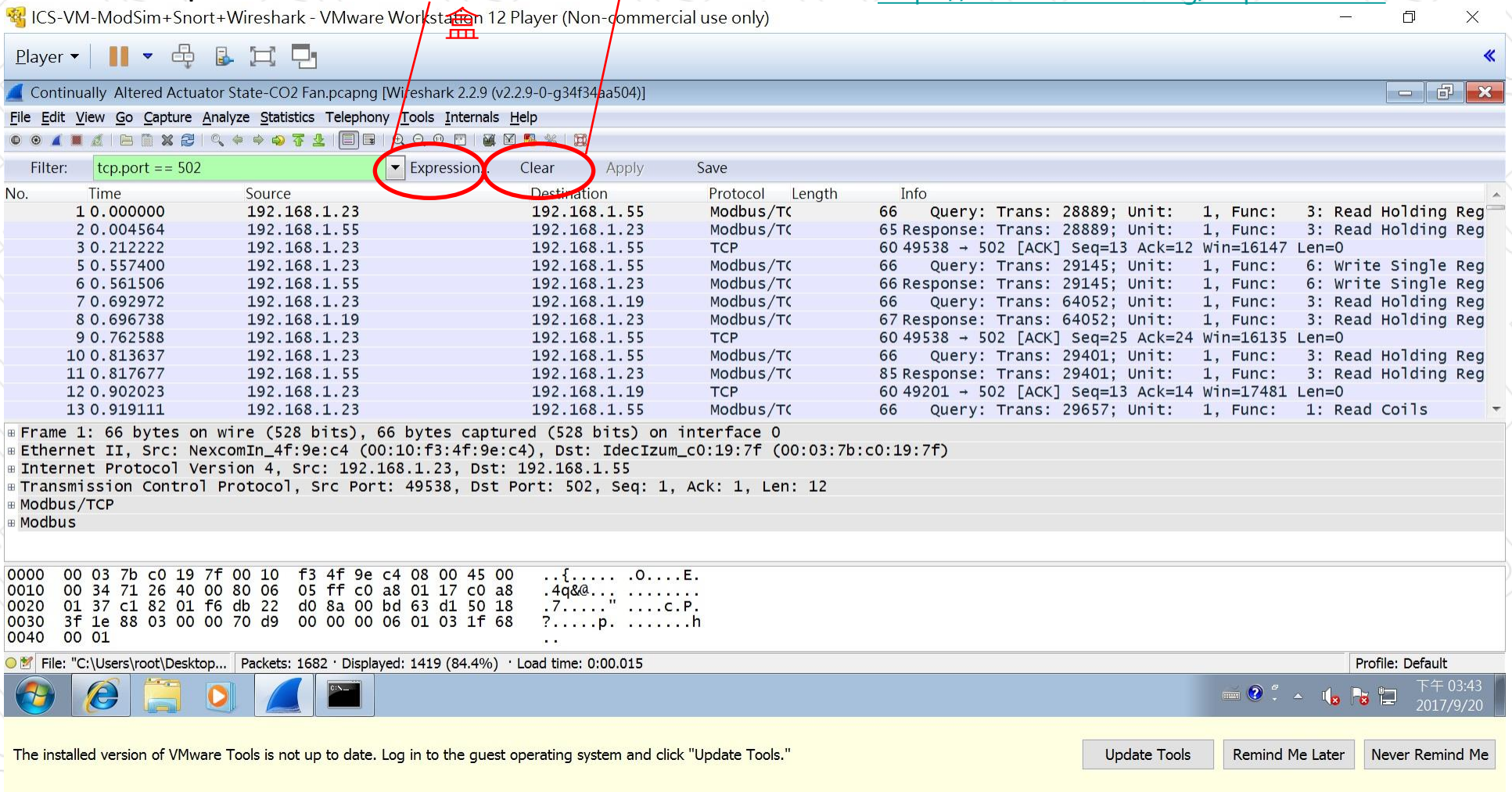
Wireshark – Filtering for Modbus

- ✓ tcp.port==502
- ✓ mbtcp

Filter 設計工具

取消 Filter

- ✓ https://www.wireshark.org/docs/wsug_html_chunked/index.html
- ✓ https://wiki.wireshark.org/CaptureFilters

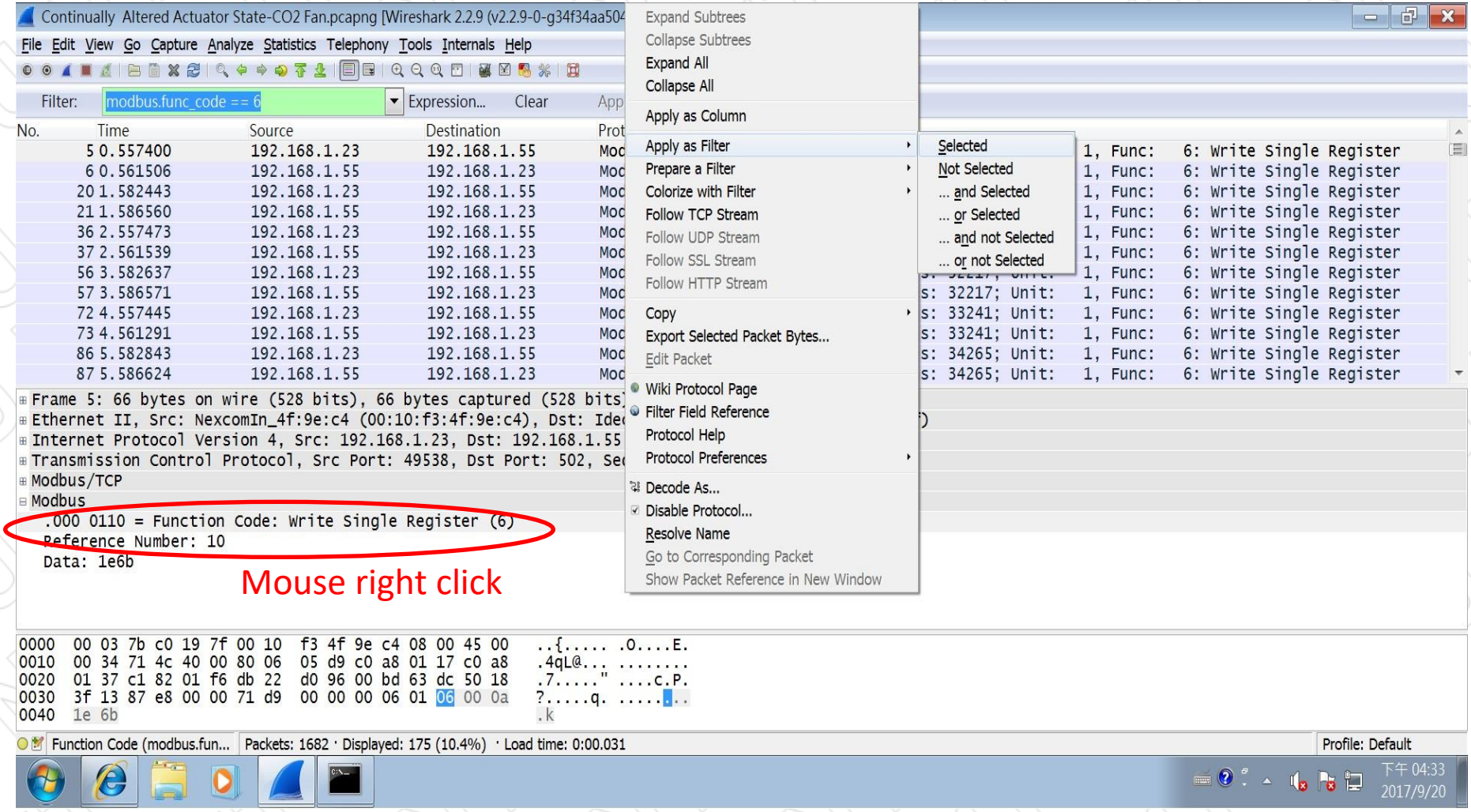


Wireshark – Filtering for Modbus

Master 有下指令給Slave嗎?

modbus.func_code == 6 (5,6,15,16)

只挑出 function code == 6 之封包



Wireshark – Filtering Example

eq == Equal
ex: ip.src==192.168.2.89

ne != Not equal
ex: ip.src!= 192.168.2.89

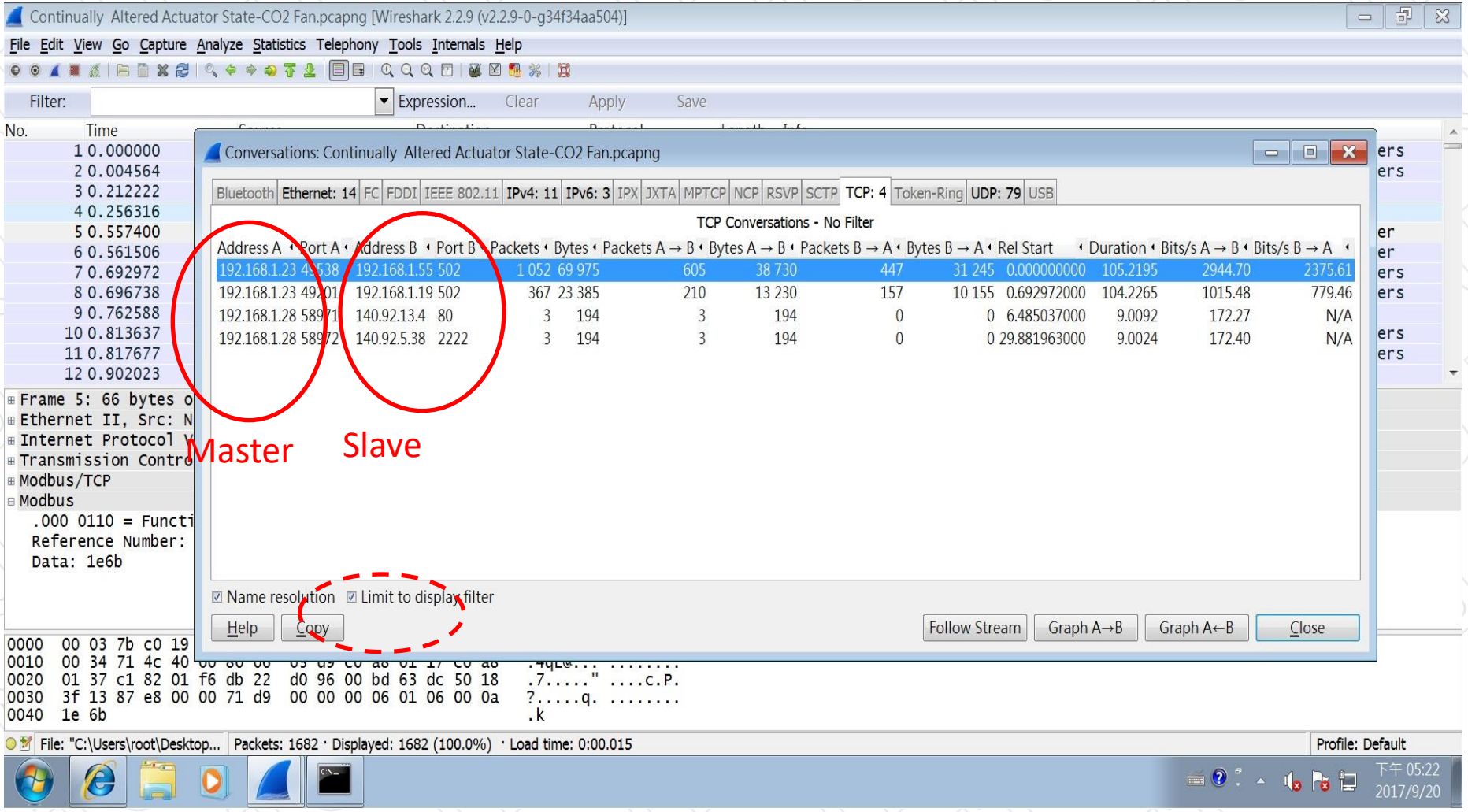
gt > Greater than
ex: frame.len > 10

lt <>= Greater than or equal to
ex: frame.len ge 0x100

le <= Less than or equal to
ex: frame.len <= 0x20

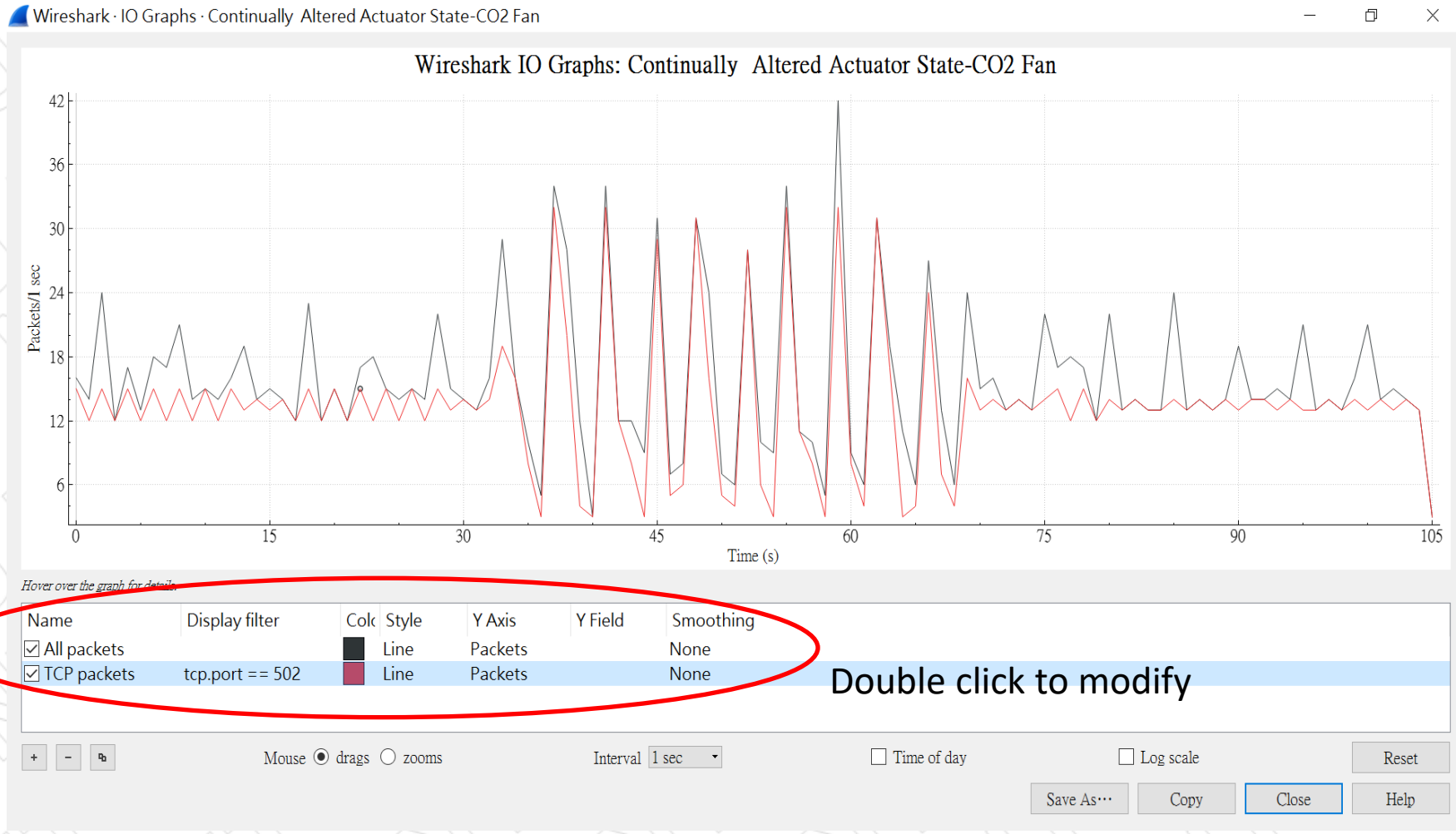
TCP Conversation for Modbus

Statistics/Conversations:



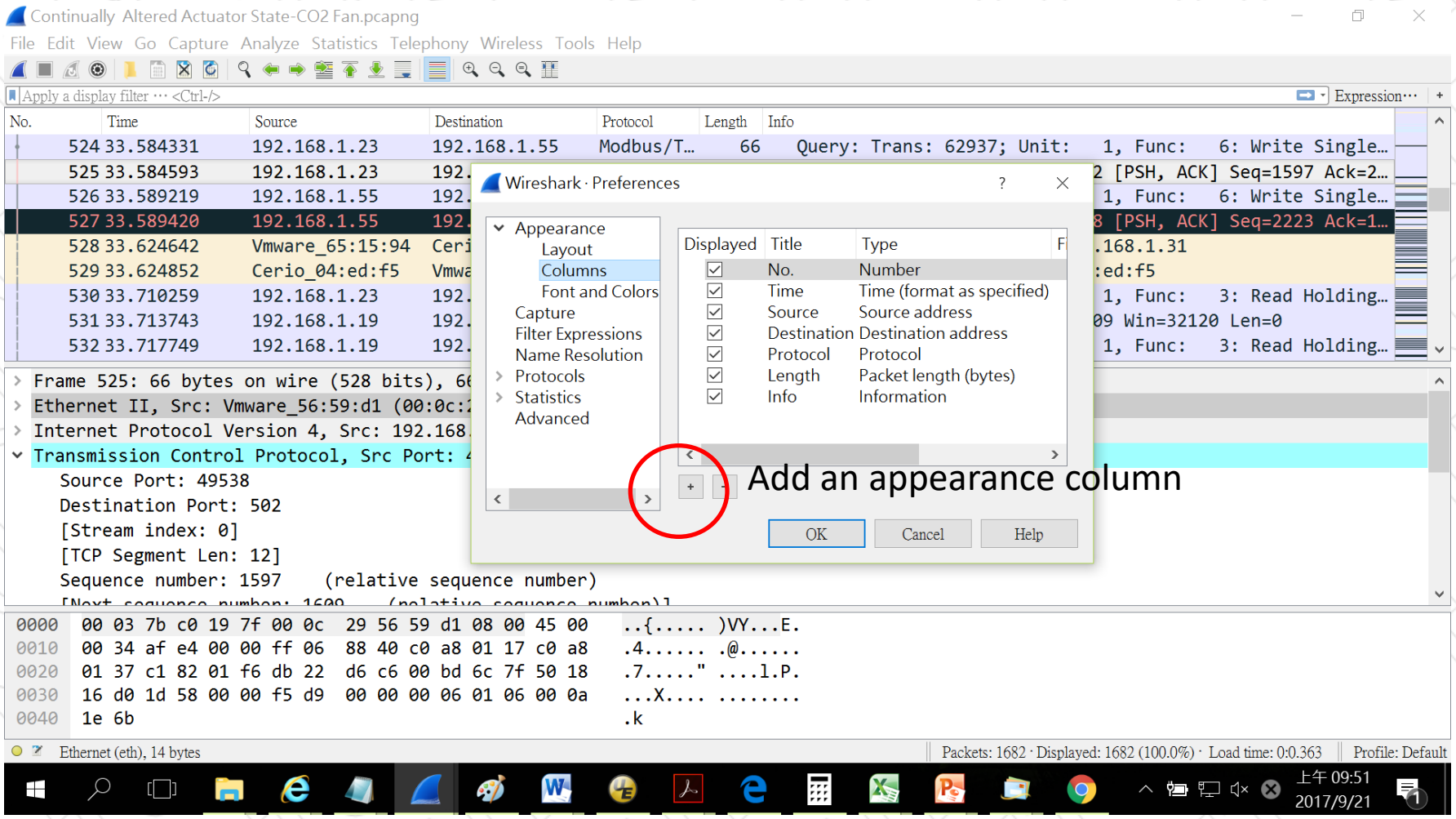
Network Spike 分析

Statistics/ I/O Graph:



Finding Interesting Packets

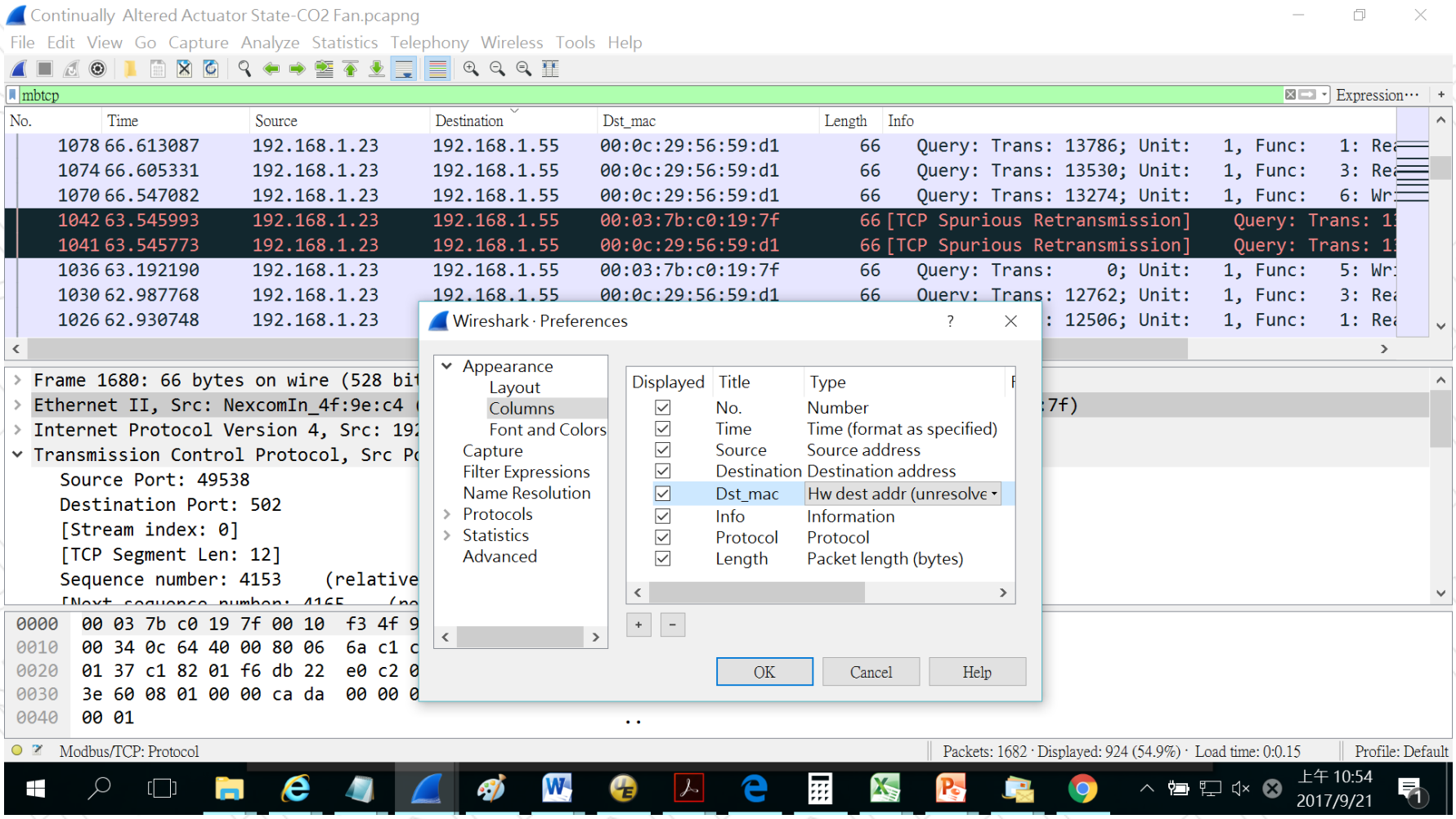
Edit /Preferences …:



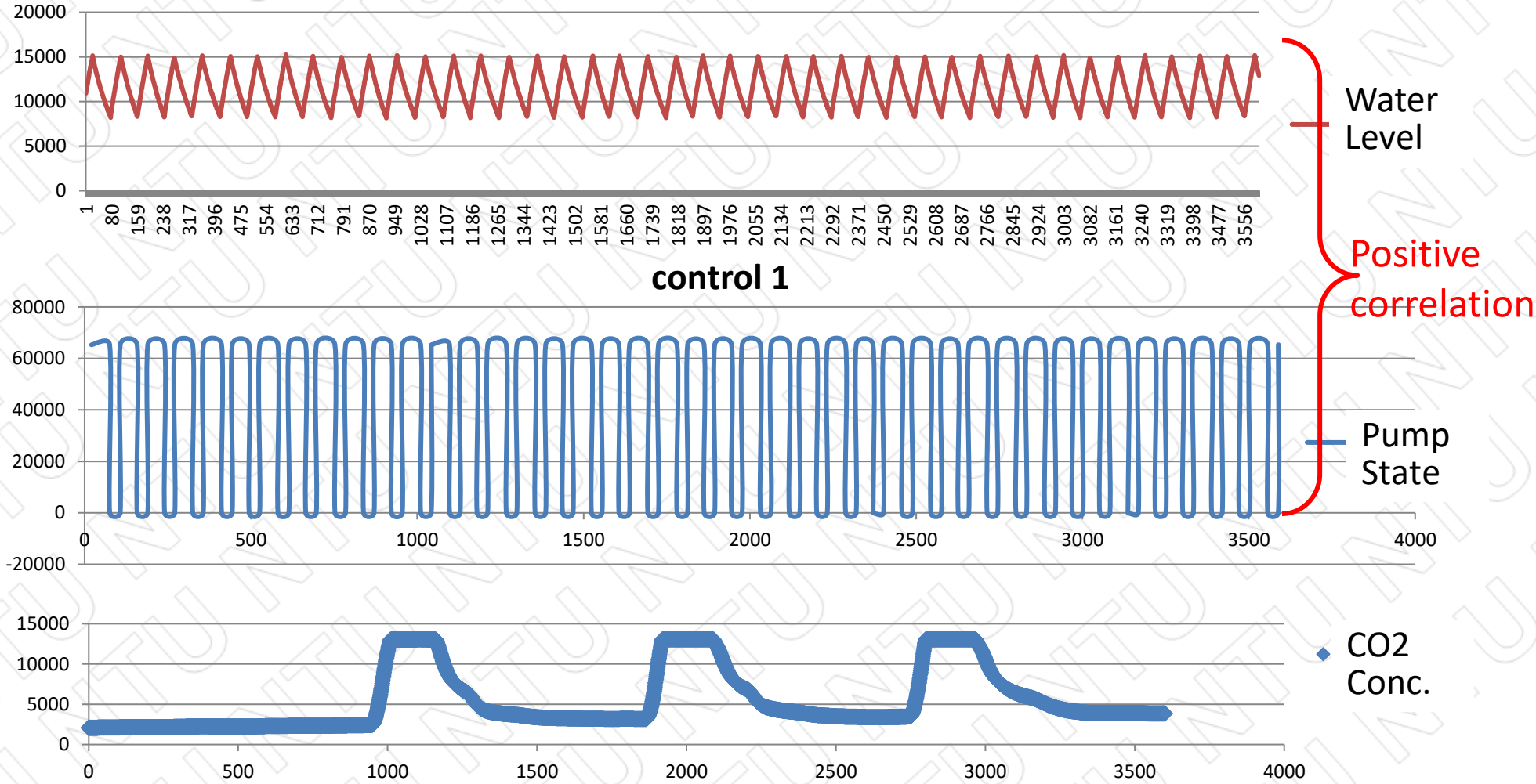
Adding Useful Sort Columns

Edit /Preferences …:

- ✓ Add a Dst_mac column with Hw dest addr,
- ✓ Sorting by Dst_Mac, add a filter with mbtcp,



數值與開關趨勢圖


20000
15000
10000
5000
0
1 80 159 238 317 396 475 554 633 712 791 870 949 1028 1107 1186 1265 1344 1423 1502 1581 1660 1739 1818 1897 1976 2055 2134 2213 2292 2371 2450 2529 2608 2687 2766 2845 2924 3003 3082 3161 3240 3319 3398 3477 3556
Water Level
control 1
80000
60000
40000
20000
0
-20000
0 500 1000 1500 2000 2500 3000 3500 4000
Pump State
Positive correlation
15000
10000
5000
0
0 500 1000 1500 2000 2500 3000 3500 4000
CO2 Conc.

作業 1-B

1. Wireshark各式分析統計圖
2. SCADA系統數值資訊：
   - Which IP address is the master on ?
   - How many slaves is the master talking to ?
   - Master 對 Slave 的那些位址做了那些func. Code?
   - Slave回應了哪些位址的那些數據內容(前2筆)?
3. 下列操作數值資訊(以192.168.1.55(PLC)為主)
   - 找出馬達開關位址(持續on-off-on-off切換)
   - 找出風扇開關位址(只有三次on-off切換)
   - 找出手自動切換開關位址(只操作一次on，然後off)

THANK YOU