
Peng Xin

# IEC 61850 Testing and Documentation

Information Technology
2010

# Foreword

This thesis would not be possible without the help of many people such as Dr. Smail Menani from VAMK OY who inspired me with the project idea and technical support. Many thanks to Mr. Olavi Vahamaki from VAMP OY, who assisted me whenever I face difficulties with the VAMP relay. Not to forget mentioning my teachers of the telecommunication module, Dr. Gao Chao, Mr. Virtanen Antti and Mr. Jukka Matila for their valuable instructions during my studies. Special thanks to Vaasa University of Applied Sciences for giving me the opportunity and possibilities to succeed in my studies.

At last, thanks to my family and teammates who gave a lot of support and encouragement during the critical moments.

## ABBREVIATIONS

| | |
|---|---|
| DS | Data Set |
| GOOSE | Generic Object Oriented Substation Events |
| GCB | GOOSE Control Block |
| IED | Intelligent Electronic Device |
| LD | Logical Device |
| LN | Logical Node |
| MMS | Manufacturing Message Specification |
| ASN.1 | Abstract Syntax |
| BER | Basic Encoding Rule |
| PDU | Protocol Data Unit |
| APDU | Application Protocol Data Unit |
| FC | Functional Constraint |
| FCDA | Functional Constraint Data Attribute |

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Degree Program of Information Technology

**ABSTRACT**

Author: Peng Xin
Title: GOOSE Transmission Laboratory Exercises and GOOSE Messages Decoding
Year: 2010
Language: English
Pages: 43
Supervisor: Menani Smail

---

The purpose of this thesis is to develop laboratory experiments for a data transmission course. The laboratory experiments deals with the generation of GOOSE messages using VAMP relays on the one hand and the decoding of the generated messages on the other hand. The decoding scheme includes theoretical overview, instructions and practical description related to the laboratory exercises.

The methodology of achieving the result is to intercept proper data matches to the structure of GOOSE through a series of physical connection and software configuration. The set up of the laboratory exercise is based on the connection between a Vamp Relay, a Laptop and a set of programs such as Vampset, IEC 61850 Simple Tester, GOOSE Sender and Wireshark. These programs are used to configure and detect GOOSE messages. Through physical connection and proper configuration of the devices and tools, the data traffic has to be detected and decoded.

The project is divided into 3 tasks: pre-exercise for telecommunication course, laboratory exercise for telecommunication course and decoding GOOSE messages for telecommunication course. In this thesis, the task is to filter GOOSE messages and decode them according to GOOSE structure.

Laboratory exercises documentation is not included in this document but delivered separately to the concerned teacher to be used in the laboratory course.

---

Keyword GOOSE, Telecommunication, Decode, Target, Analyze, Communication

CONTENTS

# 1. Laboratory Exercises Description

There are 3 telecommunication laboratory exercises in this project. The first exercise is to activate the digital inputs using RJ45 cable as a transmission medium. This is the basic experiment to find the method to detect required messages. The second exercise utilizes the switchgear to adjust the connection and test for GOOSE transmission between the Vamp Relay and the laptop. The last exercise realizes the Trip and Block functions of the protection Relay by applying the function generator which can change the values of frequency and amplitude. These exercises not only present the outcome from the researched procedure, but also provide different kinds of methods to detect GOOSE messages.

## 1.1. Laboratory Distribution

Lab 1: Activate to detect message between Relay and Laptop

1. Get familiar to the basic configuration on relays
2. RJ45 Cable connection between logical nodes in rear panel

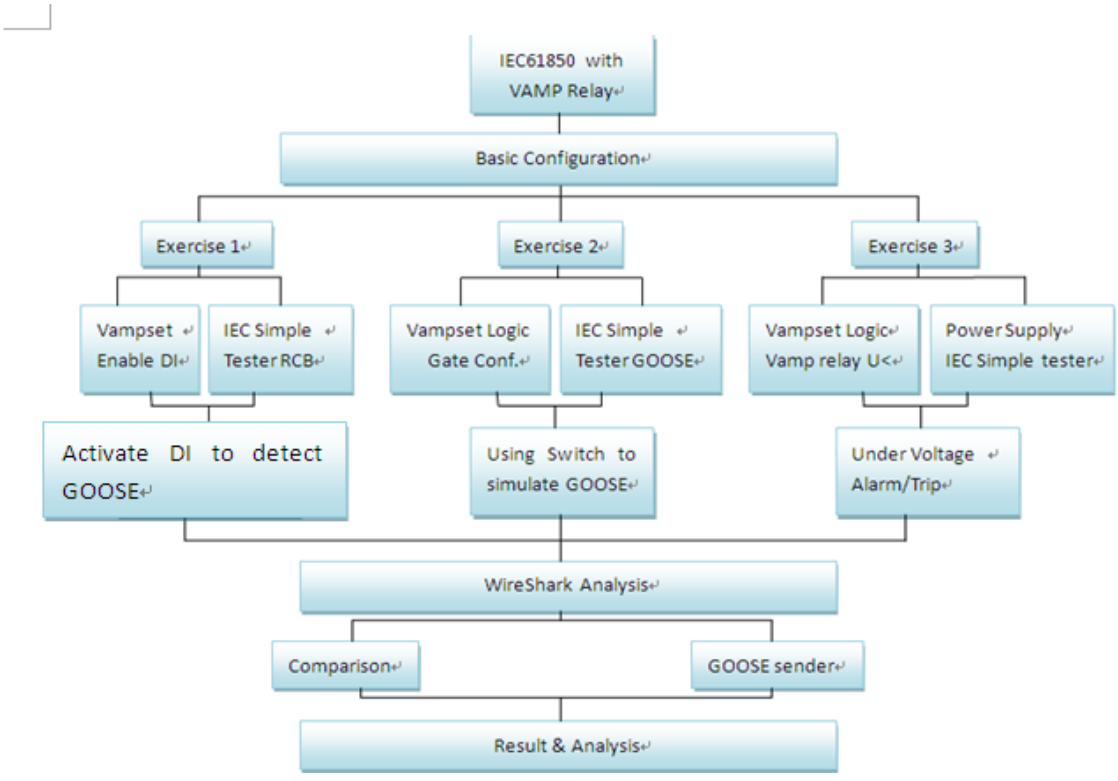
Lab 2: Switchgear to simulate GOOSE message

1. Connect a switch to Vamp Relay.
2. Set the right configuration in software.
3. Intercept GOOSE messages from Wireshark and analyze the messages. Give interrelated analysis from this exercise.

Lab 3: Trip/Block function

1. Get familiar to datasheet of the relay.
2. Connect the power supply to the relay as described in the instructions.
3. Give correct configuration in Vampset and compare the result to the principle.

1.2. Technical Laboratory Architecture



## 1.3. Devices & Software

**Devices:**

Vamp 257 Feeder Manager

Vamp 257 Feeder Manager is the protection relay used for dangerous prevention in substation. There are series of interrelated choice to set the address, values of voltage, status, and mode and so on. The overview of current information represents whether the settings are requisite in the function. The values in Vamp can be changed by inputting the secret number. Otherwise, it keeps the default information with the initial configuration and cannot be altered without the password. Physical connection and digital inputs can be enabled on the rear panel of the Vamp 257.

Vamp257 Feeder Manager（reference:www.vamp.fi）

Function Generator

The function generator produces various patterns of voltage at a variety of frequencies and amplitude. It is used for testing safety of the circuit to input signals.

**Software:**

## VAMPSET

VAMPSET comprises all of the attributes of GOOSE Control Block. GOOSE Enable, GOOSE name, Data Sets included in the attributes can be applied in this program. There is a string of menu which gives detailed information of each attribute and application. From these settings, the Vamp Relay is controlled by Vampset. This program shows the same parameters as in the Vamp Relay. Logical connection is created in tag logic to achieve the aim of GOOSE transmission between the devices.

## IEC61850 Simple Tester

This simple program is used to apply the Logic Nodes, Object and Attributes. It is divided into 5 parts: Host address, Read, Control, Report and GOOSE. The value shows the result monitored from the communication between the switchgear device and the Vamp Relay. The Report Control Block reports whether there's emergency happens if some status changes.

## GOOSE Sender

GOOSE Sender is a convenient program for GOOSE transmission. It simulates a GOOSE message directly from the Vamp Relay to other devices. Through the interrelated configuration in the devices, GOOSE Sender makes the communication with GOOSE messages. The request and response are formed by this simulated transmission.

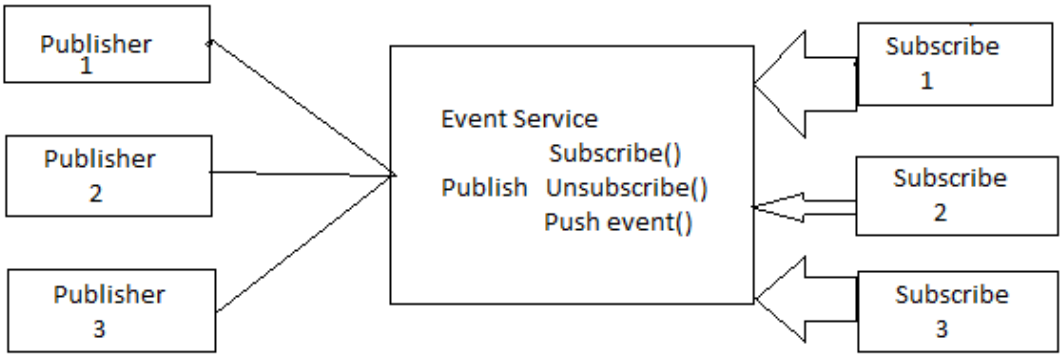
## Wireshark

Wireshark is an analyzer of network transmission. It filters useful protocol for users to display the packets needed. Updating the new versions of this program, more functions and details gives more information to the user. This software helps to monitor every frame with the exact time, address and path. Even the structures of some infrequent protocols can be excavated by Wireshark. For saving time in filtering useful messages, it has the function to capture the protocol needed from a large number of frames.

# 2. GOOSE messages Specification

## 2.1. Transmission principle of GOOSE messages

GOOSE (Generic Object Oriented Substation Event) is a transmission of exchanging a wide range of data in a dataset. With a fast connection and less communication service which is used for the transfer of time-critical data, the communication network updates the content of messages as soon as the values changed. In order to more applications, GOOSE message exchange is based on the multicast application association. This routing technique provides to deliver messages from one publisher to one or many subscribers. The publisher is referred to as a sender who requests with a group of members in a dataset. These requests are sent to the transmission buffer of the publisher, afterwards this buffer is updated with the publish service and the values are transmitted by a GOOSE message. On the receiver, the reception buffer in the subscriber receives the new values and it is updated by mapping services.




Multicast Application

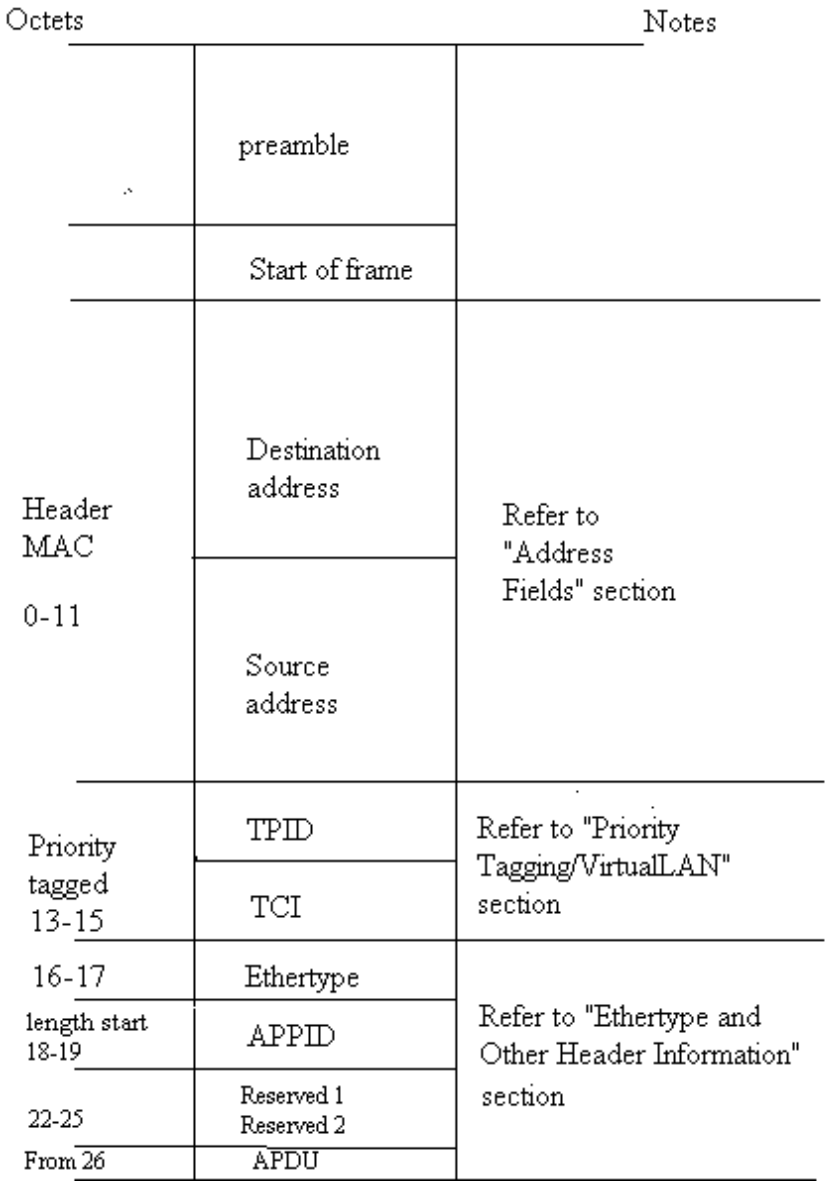
(Reference: International Standard IEC61850-5)

Publish/Subscribe systems are useful: first of all, while unreliable delivery mechanisms are present, they distribute large quantities of time-critical information well and quickly. Then it can handle very complex data flow patterns. Finally many-to-many model is very efficient in both bandwidth and latency.

## 2.2. GOOSE Data Structure

The following figure is an overview of ISO/IEC 8802-3 frame structure for GSE management and GOOSE. In a GOOSE message, the header MAC comprises 12 bytes, first 6 bytes are destination address and the last 6 bytes are source address. Next 14 bytes are Priority tags which present Ethertype, Application ID, message length and reserved number. Besides the 26 bytes enumerate the fundamental information of the GOOSE message, from byte 27 to the unlimited number of data

are decoded with APDU (Application Protocol Data Unit). The detailed explanation of GOOSE messages intercepted from the outcome in the laboratories will be revealed in the subsequent chapter.




ISO/IEC 8802-3 frame format

(Reference: International Standard IEC61850-5)

In APDU structure, there are 13 labels which display the specific content in the configuration of the experiments. The labels stand for the attributes of GOOSE Control Block.

```
Tag 0:   gocbRef (GOOSE Control Block Reference, type: visible-string)
Tag 1:   timeAllowedtoLive (type: integer)
Tag 2:   datSet (Dataset, type: visible-sting)
Tag 3:   goID (GOOSE Identifier, type: visible-string optional)
Tag 4:   t (time&quality, type: UtcTime)
Tag 5:   stNum (Status Number, type: integer)
Tag 6:   sqNum (Sequence Number, type:integer)
Tag 7:   test (type: Boolean default false)
Tag 8:   confRev (Configuration Revision, type: integer)
Tag 9:   ndsCom (needs Commission, type: Boolean value default false)
Tag 10:  numdatSetEntries (Number of Dataset Entries, type: integer)
Tag 11:  allData (sequence of data)
Tag 12:  security (any optional, reserved for digital signature)
```

APDU Structure

This APDU structure is referred from INTERNATIONAL STANDARD 61850-8-1.

# 3. Procedure of 3 Laboratories for Detecting GOOSE messages

**Equipments used in exercises:**

Vamp257 Feeder Manager

Laptop

RJ45 Cable

Ethernet with Optional to baset

Switchgear

Function Generator

Cables for Connection

**Physical Connection:**

Use a RJ-45 cable to connect the Ethernet port of your Laptop and the VAMP 257. Then run Program Vampset on the Laptop.

**Fundamental Settings on Vamp Relay:**

For making connections, you should start with setting the IP address of the Vamp Relay to 192.168.65.2 by using the keypad on the obverse of the Vamp Relay. As a prerequisite to set the IP address, ensure you have entered the password. Press the Down arrow key to find the option ‘Bus’. Then keep clicking the Right arrow

key until you reach Ethernet Port. Press ‘Enter’ to go in the menu. Enter the password ‘0002’ for the configuration in the Vamp Relay. Select IP option with the Down key. Make sure the IP addresses of the Relay and the laptop are in the same range. In this case, set the IP address of the laptop to ‘192.168.65.20’.

## 3.1. Laboratory 1 Activate DI’s to Detect messages

Based on the previous connection, start Vampset to control the Vamp Relay. Firstly, on the left menu of the program choose ‘Data Map (2)’ in ‘IEC61850 main config’ to ensure ‘digital input 1’ is set to ‘yes’. Then Select the sequenced tag ‘Data Map (3)’. The digital inputs from pin2 to pin 7 are shown on the right of Vampset. Set them to ‘yes’. These pins mapped in map (3) in this software correspond to the logic nodes from LN1 to LN6. At last remember to save the configuration and write the changes to the device. These configurations are clearly shown in Figure 3.1 and Figure 3.2.

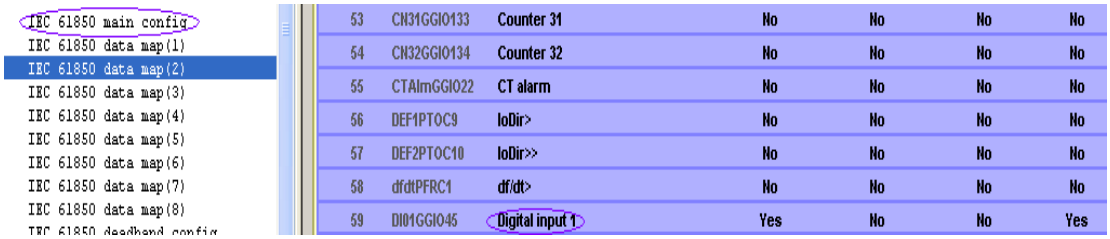



Figure 3.1 DI1 in use

| Index | LN | Description | Dataset 1 | Dataset 2 | Dataset 3 | In use |
|---|---|---|---|---|---|---|
| 60 | DI02GGIO46 | Digital input 2 | Yes | No | No | Yes |
| 61 | DI03GGIO47 | Digital input 3 | Yes | No | No | Yes |
| 62 | DI04GGIO48 | Digital input 4 | Yes | No | No | Yes |
| 63 | DI05GGIO49 | Digital input 5 | Yes | No | No | Yes |
| 64 | DI06GGIO50 | Digital input 6 | Yes | No | No | Yes |
| 65 | DI07GGIO51 | Digital input 7 | Yes | No | No | Yes |

Figure 3.2 DI2 to DI7 in use

### 3.1.1. Setting in Read Menu

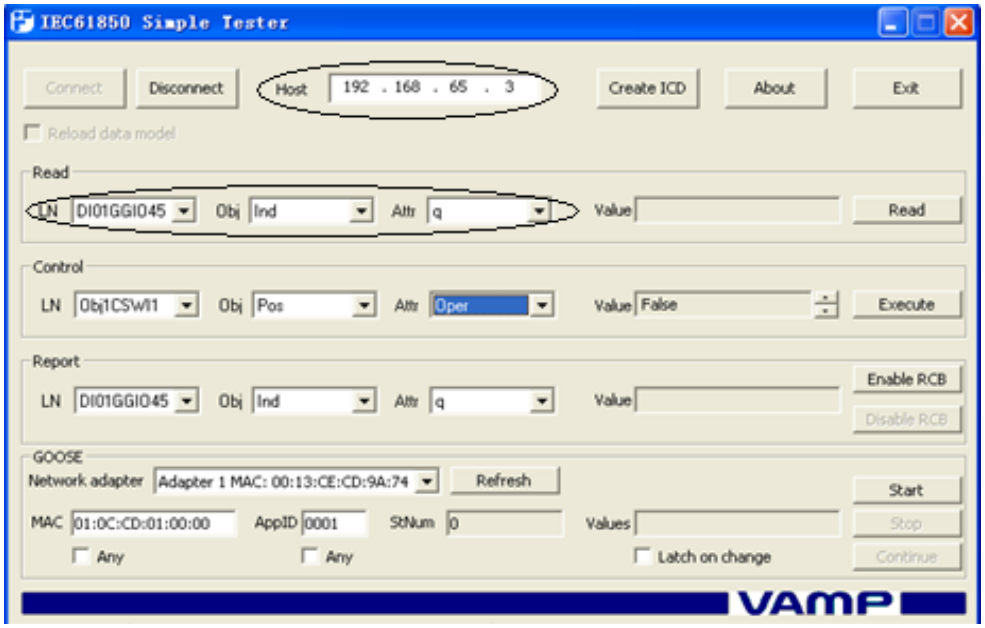



Figure 3.3 Setting in IEC Simple Tester

After setting the digital inputs in Vampset, open IEC Simple Tester. Figure 3.3 indicates how to set the address in hot and the elements in Read menu. Set the host address to the IP address of the Vamp Relay. In read position, select the digital input enabled in Vampset. The name of LN 'DI01GGIO45' can be selected here when digital input 1 is used in Vampset. Set the Object to 'Ind' (indication) and Attribute to 'StVal' (state value).

Remember to write all changes after adding every new LN. And then enable the RJ45 cable connected the pins' connectors on the rear slot. Pin 1 is digital input1, it must be always connected to one side of the cable. Connect the other side of the cable to other digital inputs one by one (from pin 2 to pin 7). Insert the cable to a different pin

every time and set the corresponded LN name in ‘read’ menu. When insert the cable to a pin and press ’read’ button, the value becomes ‘true’ at the moment. Take out the side of cable which is not connected to pin 1 and press ‘read’ again, the value is affected to ‘false’. From this test, every LN name includes the number which is calculated by adding 1 to the corresponded pin number. For instance, the LN name of Pin2 is ‘DI01...’; pin 3 is ‘DI02...’; pin 4 is ‘DI03...’...

### 3.1.2. Setting in Control Menu

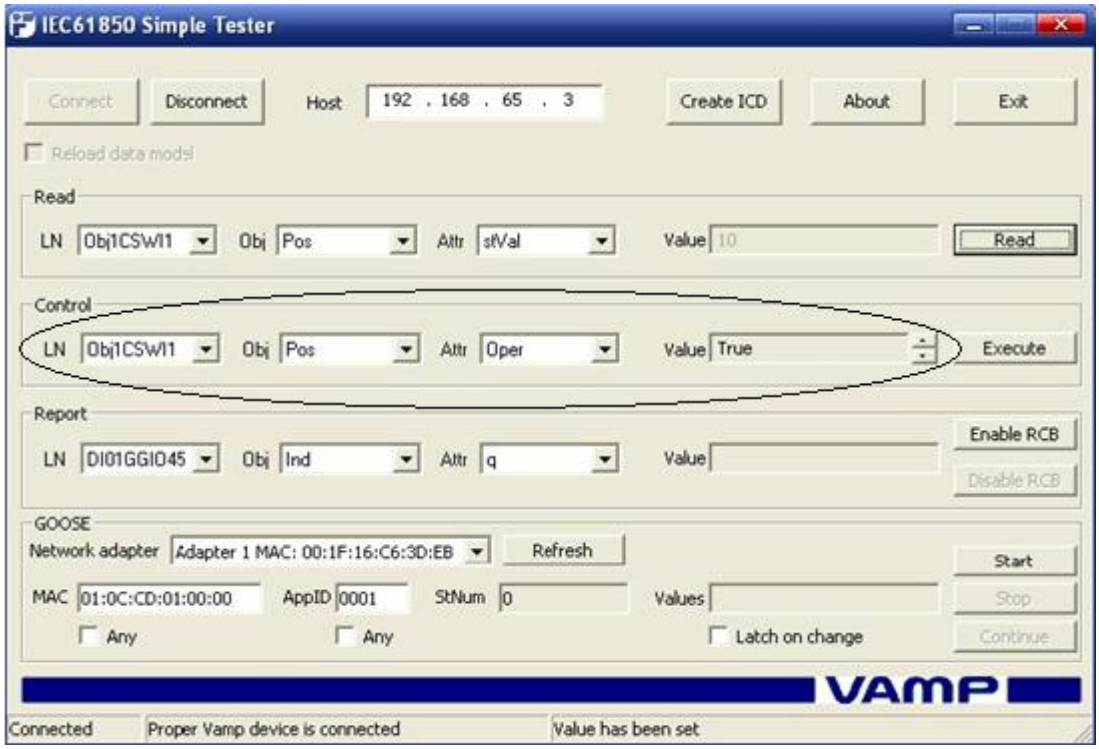



Figure 3.4 Settings in Control Area

Figure 3.4 shows the selections of elements in Control menu. In the Control Menu, set object to ‘pos’ and attribute to ‘Oper’. In this case, ‘Obj1CSWI1’ is the unique option in LN. This LN is used to control the circuit breaker. Choose the value to ‘false’ and press “Execute”. At the same time a blip can be heard clearly from VAMP

Feeder Manager and the switch of the circuit shown on the relay is turning off. This operation affects the value in read area which changes to”01”. In opposite way when set the value in Control menu to true, the value in Read changes to”10”. It shows whatever the value in CONTROL changes during the configuration, the value in READ never goes to”00” or”11”. That is because two connectors must be different values when it is not known that the power is on or off. In this case there should be a setting to a safety mode.

### 3.1.3. Settings in Report Menu

After controlling step, set object to ‘Ind’ and attribute to ‘StVal’ in Report menu. Click ‘enable RCB’. On the rear of Vamp Relay, one side of the RJ45 cable is fixed to the pin1. Connect the unfixed side of the cable to pin 2 and select the corresponded name ‘DI01GGIO45’ in ‘LN’. Then the value in the Report menu becomes ‘true’. Take out the unfixed side of the cable. The value changes to ‘false’. This result presents when the emergency happens, the report part reacts the change in the value as soon as possible.

## 3.2. Switchgear to simulate GOOSE messages

In this exercise, connect the vamp relay to a switch. With the configuration in Vampset, GOOSE Sender and IEC Simple Tester, the reaction in Simple Tester and data in Wireshark will correspond to each other by adjusting the switch.

There are 3 wires connected to the switch. Distinguish the three wires from different colors: red, yellow and black. They are connected to different logic nodes. The wires are connected to Pin 1, 2, 3 at the beginning.

### 3.2.1. GOOSE Sender Configuration for GOOSE

GOOSE message is a timely message. It detects and operates as soon as the

mistake is sending. This notation will also be mentioned in the next laboratory exercise. Our target in this part is sending GOOSE message and intercepted it when there are changes. At first, open GOOSE sender to configure. The configuration is shown as Figure 3.5. The Network adapter must be chosen to the second MAC address because another is for wireless.

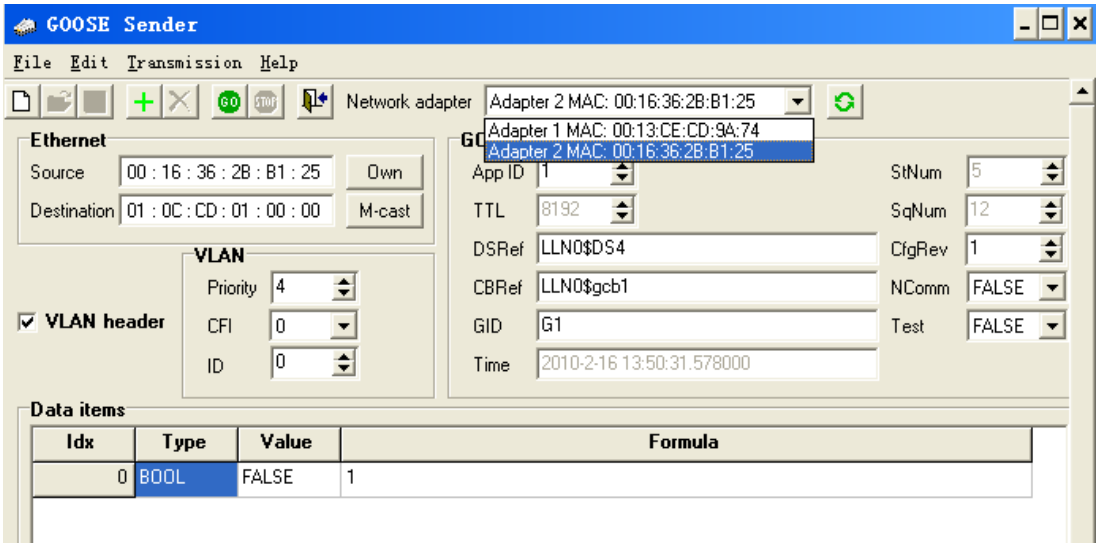



Figure 3.5 Configurations in GOOSE Sender

Then control the switch. The switch is connected to pin 1 to pin 3 on Relay A. Control the switch, LEDA of Relay B lights on. After that reclose the switch, you'll find LED A on relay B is turned off. Relay B reacts as soon as the switch connected to Relay A is adjusted.

### 3.2.2. Vampset Configuration for GOOSE

Open Vampset, click connect button, input the code number 0002 to

complete the connection. After few minutes, it finishes connection and shows the menu details on the left as in figure 3.6. Click 'GOOSE configuration'. The right side shows Publisher parameters, Publisher Configuration GCB 1 and Publisher configuration GCB 2.

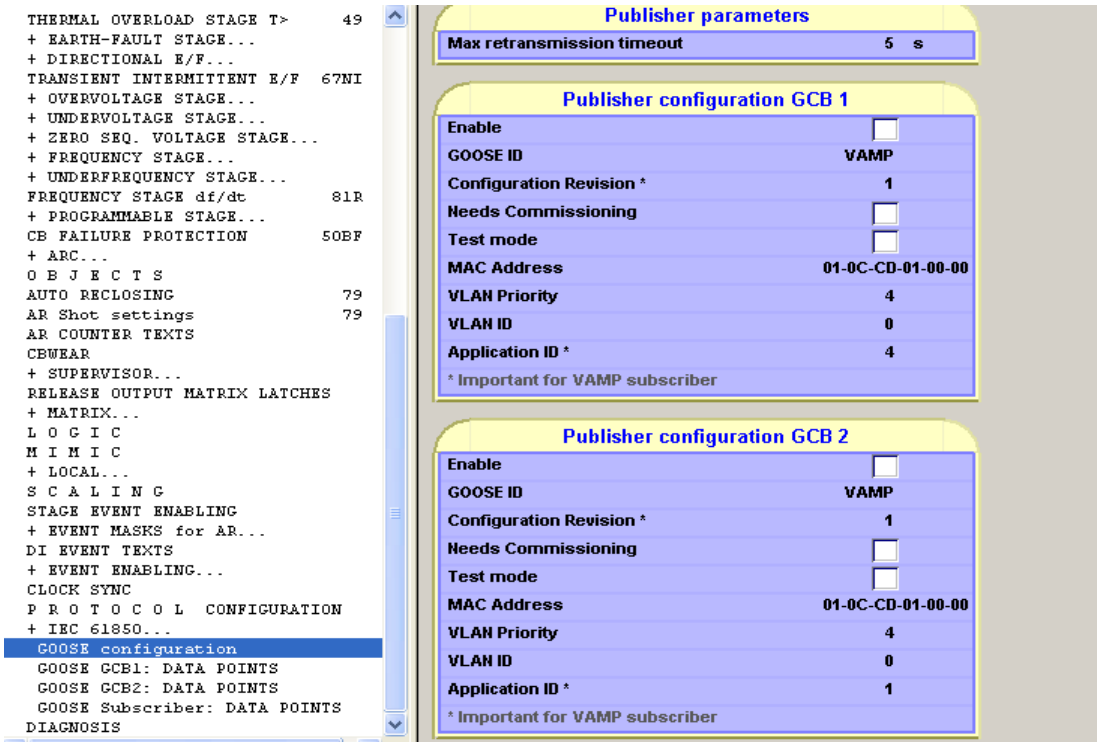



Figure 3.6 GOOSE configuration

Make sure to enable Publisher Configuration GCB 1. The Mac address must be 01-0C-CD-01-00-00 and fill the Application ID with 4. Figure 3.7 shows the configuration of enabling GOOSE Control Block in Vampset.

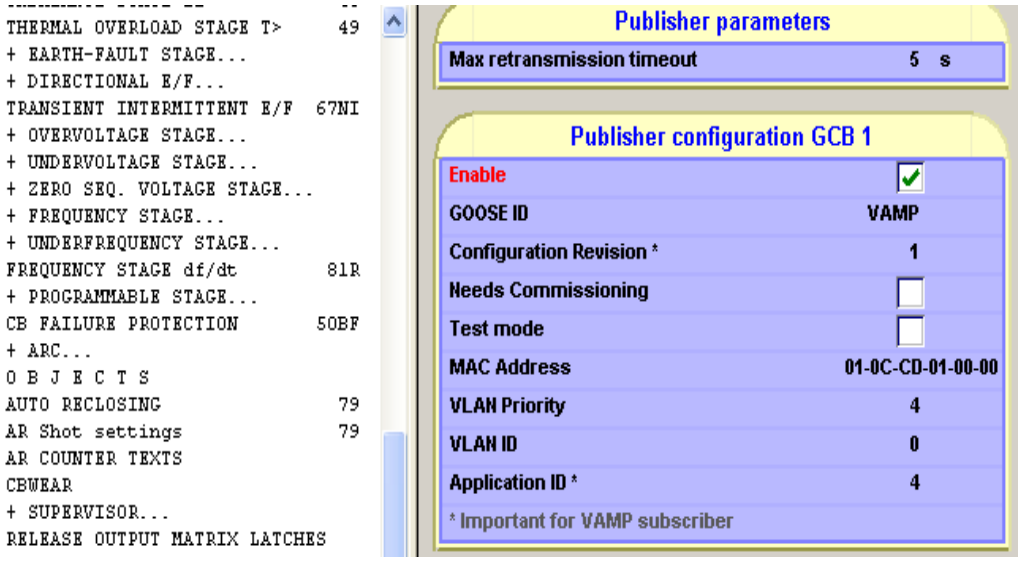



Figure 3.7 GCB enable

Then select GOOSE GCB 1: DATA POINTS on the left side menu in Figure 3.8. Change Signal to Vl1 as the settings shown in Figure 3.9 and the IEC-61850 variable changes automatically.

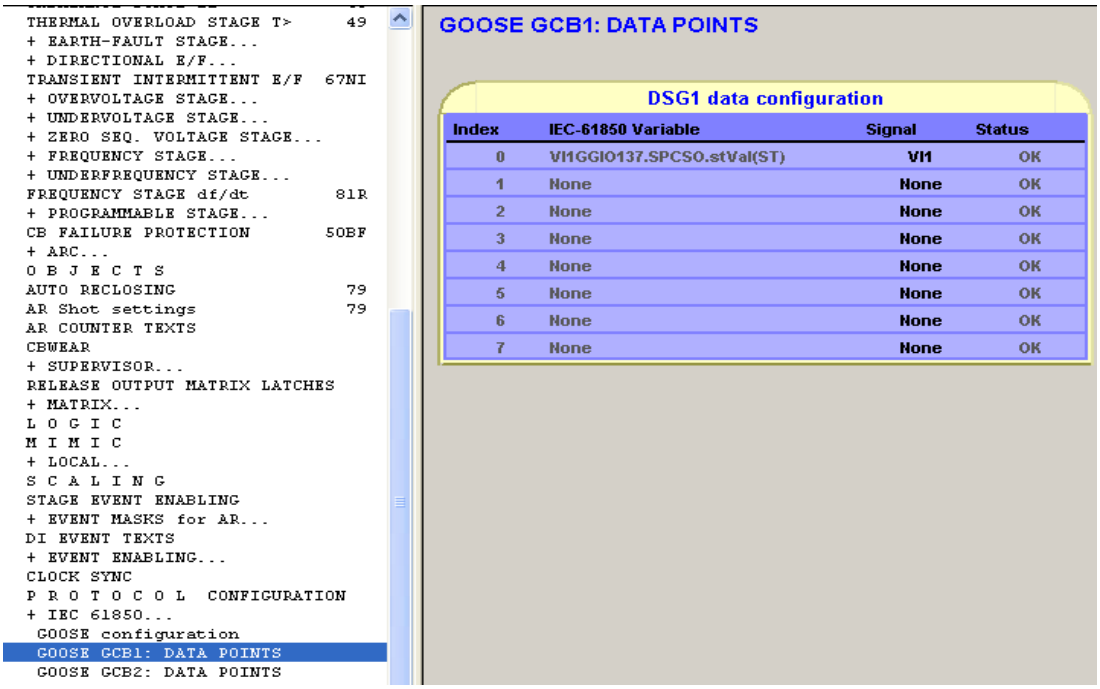



Figure 3.8 VI1 in use

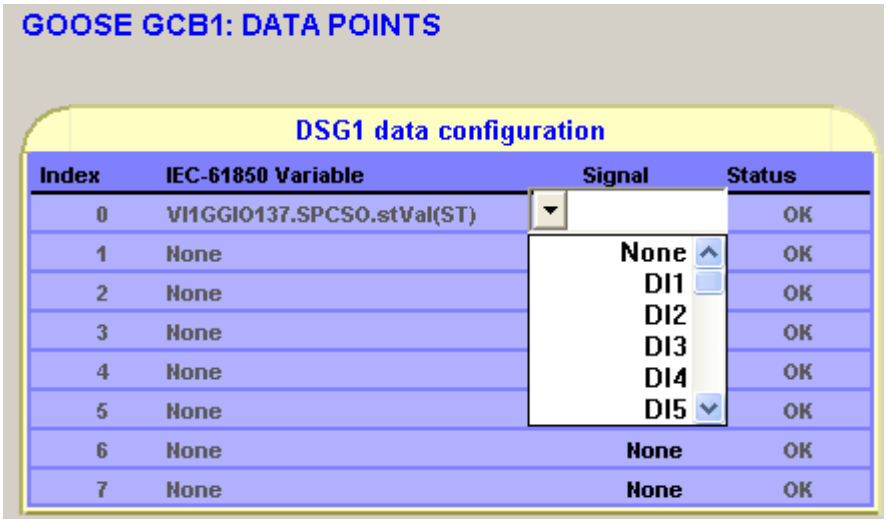



Figure 3.9 Select Signal

There are 3 LED lights on the relay: A, B and C. These three lights represent in the Logic menu. Figure 3.10 illustrates the logic map created in the

program. You can choose which LED light by setting in Vampset. Following Logic figure shows the logic connection for LED. LA after the AND logic gate stands for LED A. You can add any LED from A to C in the logic menu. In this case it works successfully in our testing process.

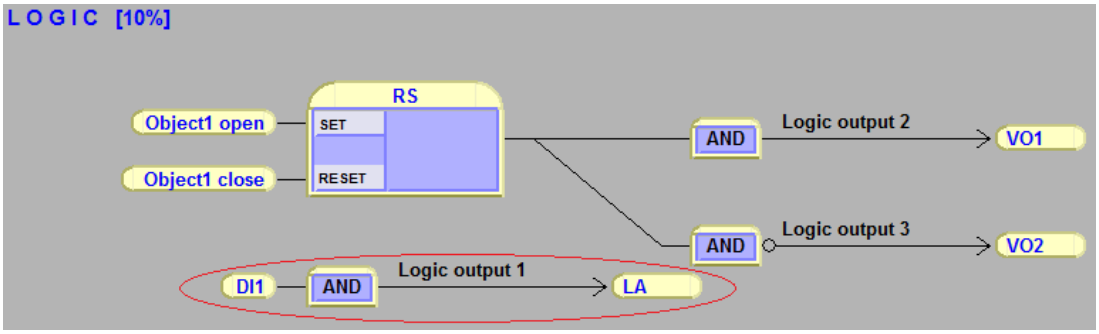



Figure 3.10 Add LA in Logic

After the configuration, adjust the switch, you can turn on and turn off the LED light.

### 3.2.3. GOOSE in Wireshark

Open Wireshark and click the first button to list the available capture interfaces. Figure 3.11 shows the interface selecting button.

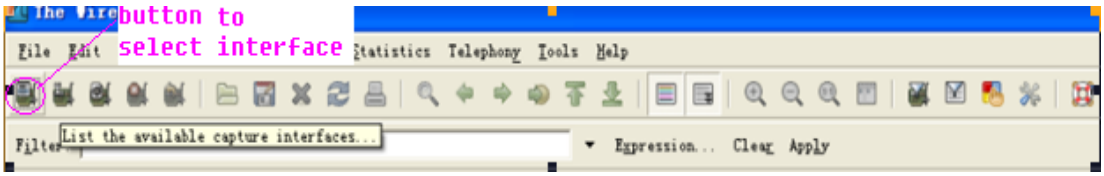



Figure 3.11 Select Interface

In the selective capture interfaces, start the second one (Ethernet).

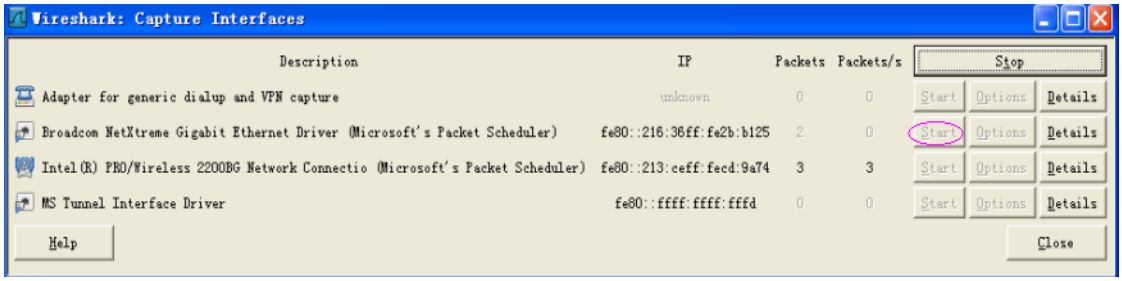



Figure 3.12 Start to detect messages

Following picture 3.13 points out the GOOSE message intercepted when the switch reclose and LED A is on.

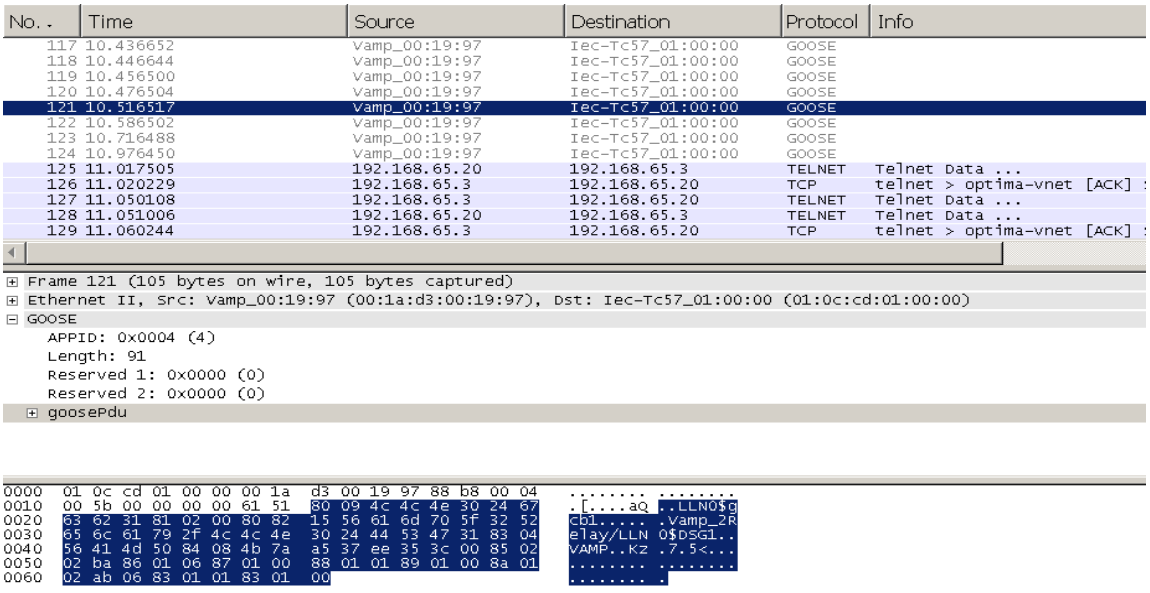



Figure 3.13 GOOSE message in Wireshark

This exercise is like a Test-Equipment simulation. Relay A is the protection Relay. The protection Relay A issues a trip signal to the switchgear-Relay B indicating that the relay has picked up on receiving a fault current. When Relay B receives GOOSE messages form Relay A, it trips and turns on the

LED light to react the trip. Then it sends a GOOSE message which contains the status of the circuit breaker and switches to tell our laptop. Each time when a new GOOSE message arrives at the laptop, the publisher details that GOOSE messages are displayed together with the response in relay to the GOOSE message. For receiving at the Relay B which includes the GOOSE subscription and notification packets as well as the GOOSE message. The size of GOOSE subscription and notification packets are measured as 74 bits whereas the size of the GOOSE as 244 bits.

Relay B receives a total number of 8 GOOSE messages. Only two of them are the event-driven GOOSE message. The remaining six messages are the retransmissions of the original ones. The 2 event-driven GOOSE messages are the original ones. Hence, three transmissions take place for every original message.

## 3.3. Trip/Block Functional Simulation for GOOSE

The functions of Tripping and Blocking on Vamp Relay are used for protection. In the protection simulation in this exercise, the configuration of GOOSE Control block in Vampset and the voltage boundary is set in Vamp Relay. The communication is between vamp relay and function generator.

### 3.3.1. Physical Connection

In this exercise connect a Function Generator to the Vamp Relay. The lights of Vamp Relay are affected by controlling the generator with Frequency and

Amplitude. Pin 11 and 12 on the rear panel of the relay is connected to the Positive and negative power of the Function Generator. For this lab, only one Relay is used to connect to the laptop.

### 3.3.2. Software Configuration& Test Procedure

Configuration of Vamp Relay

| Status: Blocked |
|---|
| f < 48Hz |
| t < 3.00s |
| LVBlk 40%Un |

Status setting

| CONF | |
|---|---|
| Un | 200V |
| Usec | 50V |
| Uose | 100.000V |
| Umode | 2LL+U0 |

Voltage configuration

| Mode Auto | |
|---|---|
| fAdop | 50.0Hz |

Mode setting

| U> | U>> | U>>> |
|---|---|---|
| 300V | 300V | 300V |
| 150%Un | 150%Un | 150%Un |
| t>0.20s | t>>0.10s | t>>>0.10s<br>Hyster 3.0% |

Configuration of U>,U>>,U>>>

| Status: Trip | | |
|---|---|---|
| U< | U<< | U<<< |
| 60V | 140V | 140V |
| 30%Un | 70%Un | 70%Un |
| t<1.00s | t<<2.00s | T<<<1.00s |
| LVBlk 12%Un | LVBlk 80%Un | LVBlk 72%Un |
| RlsDly 0.06s | | |
| Hyster 3.0% | | |
| PreFlt 5.0%Un | PreFlt 1.9%Un | PreFlt 7.9%Un |
| Edly 100% | Edly 100% | Edly 100% |

Trip Status Setting

Configuration in Software Vampset

Firstly, go to IEC Data Map (7) to make “U<”,”U<<”,”U<<<”in use. It like the figure 3.14 shows below. This configuration creates the corresponding values of the three factors in Vamp Relay settings “in use”.


Figure 3.14 Use U<, U<<, U<<<

Add a new logical map in this Logic tag. “VO6” is virtual output 6. This factor value isn’t the same as the name used in the previous logic map. It is an important logical connection for representing GOOSE messages during the trip.

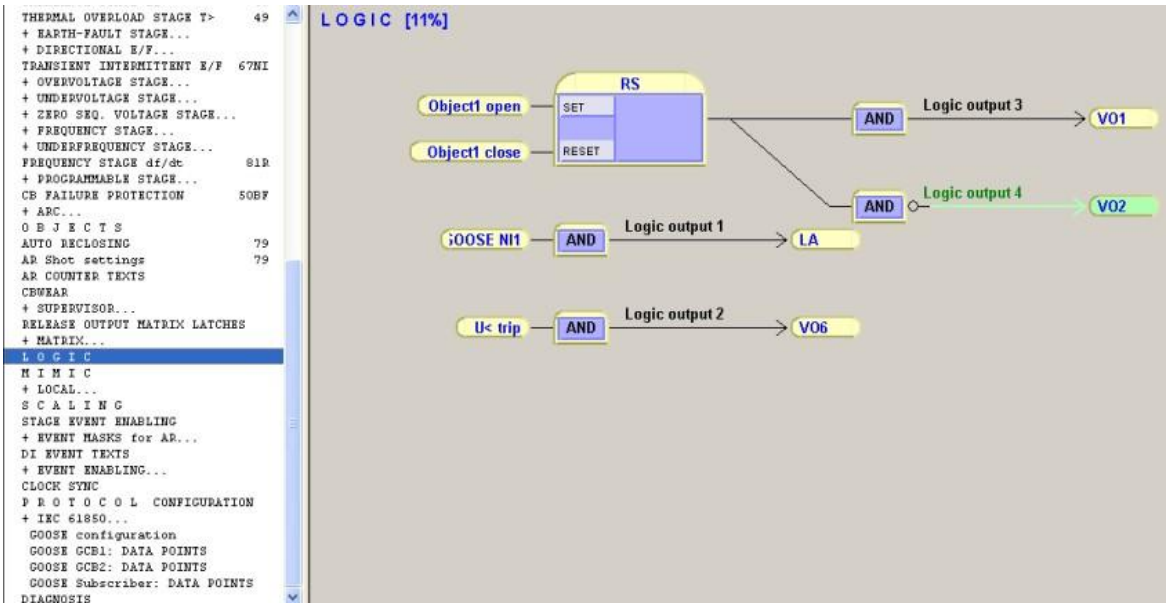



Figure 3.15 Apply the signals

Because many new digital inputs and outputs are used in the logic part, be sure to make them variable in “GOOSE GCB1”. Then add Signals such as digital input 1, virtual input 1 and virtual output 6 on Figure 3.16. The logic map goes through AND gate makes sense to transmit with GOOSE messages.

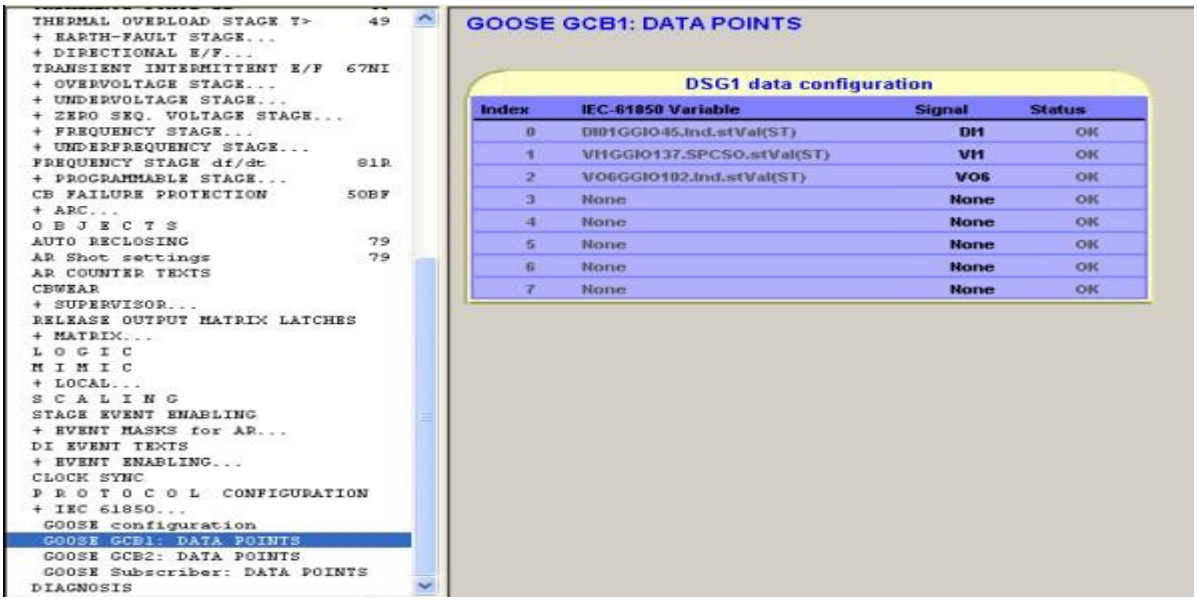



Figure 3.16 Settings in DSG1

Figure 3.16 shows data points in GOOSE Control Block1. Digital input 1, virtual input 1 and virtual output 6 are the signals used in logic menu to detect GOOSE message. Then apply the three signals and create a new logical circuit in this part. These two parts react to each other to transmit through GOOSE messages. GOOSE Control Block name is unique within the Logical Node 0, for instance, the Logical Device. Different applications within this station must have unique application ID value.

Following figure (Figure 3.17) illustrates the result in IEC 61850 Simple Tester. This result also shows GOOSE monitoring panel. In Simple Tester, set the Network adapter as our own laptop address. Then click on Start, the Value displays “F, F, and T”. These three Boolean values stand for the digital input 1, virtual input 1 and virtual output 6 which have been added in

the GOOSE Control Block. This result also gives detailed explanation in Wireshark by receiving GOOSE message.

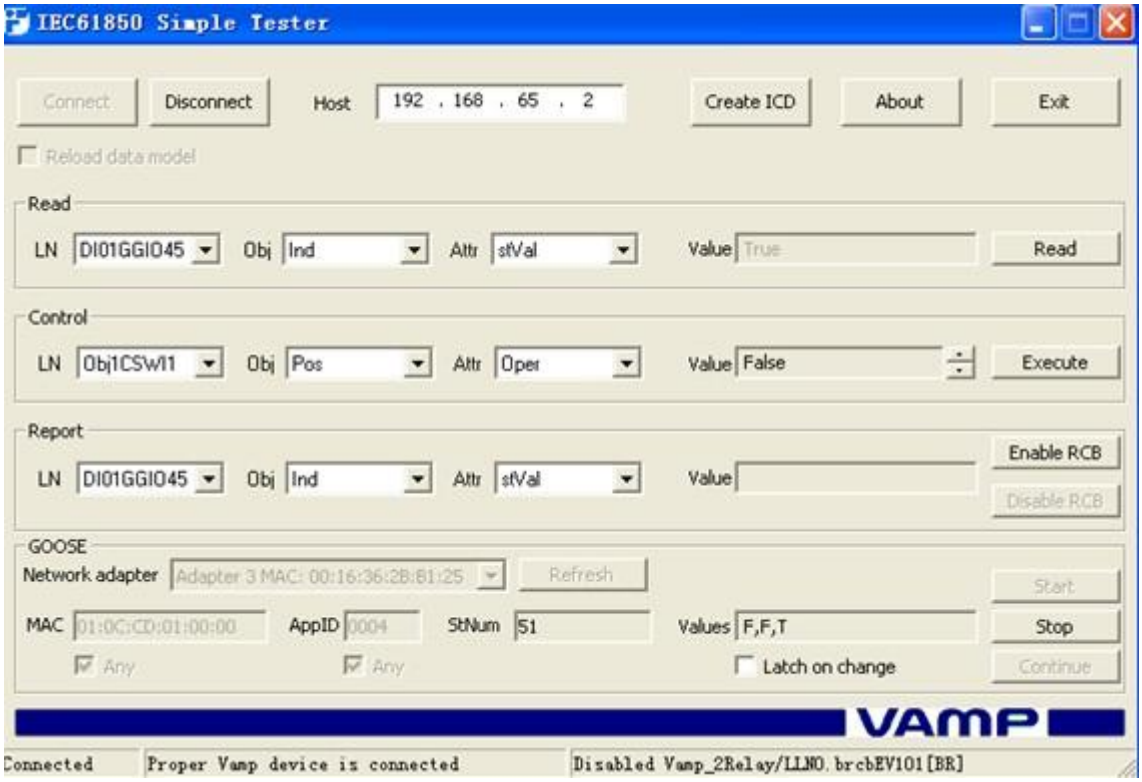



Figure 3.17 Reaction shown in Simple Tester

Figure 3.17 shows the AppID, StNum and Values in GOOSE menu. AppID (application identifier) shows the setting of AppID in Vampset. (StNume (Status Number) records the accounts of changing the status during the configuration. The values correspond to the activated LNs in Vampset. The status of any LN changes will reload the values and the status number.

Open Wireshark to look for the GOOSE transmission.

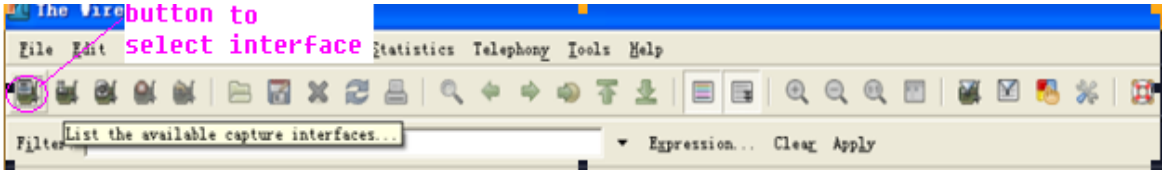



Figure 3.18 Select Interface

In the menu shown on figure 3.18, click the button capture interfaces, start the Ethernet interface

Following Figure list the frames of GOOSE in the trip transmission.

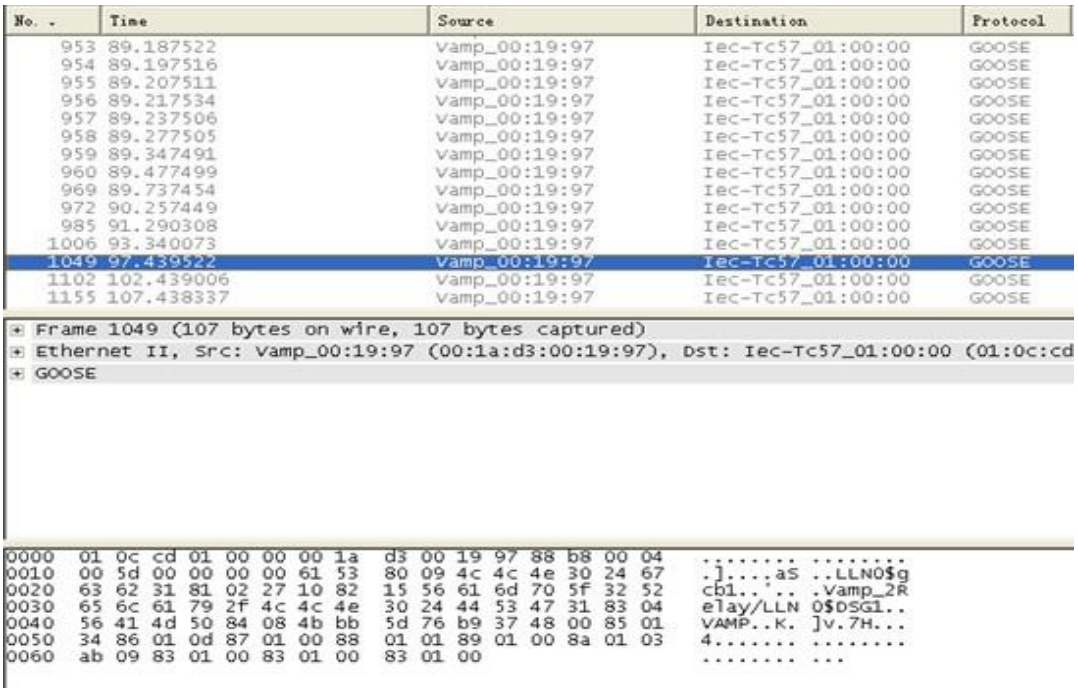



Figure 3.19 Frames in Wireshark

Figure 3.19 illustrates the data in Wireshark. In this picture, GOOSE messages keep

sending during the transmission. The destination address “01-0c-cd-01-00-00”, “01-0c-cd”is assigned by IEEE; the forth byte “00” shows it is transmitted with GOOSE; the last two bytes “00 00” are the starting address number in GOOSE message.

## 4. GOOSE messages Analysis

GOOSE message is reflected from application layer to data link layer. It is not across the transmission layer and network layer. The messages contain information that allows the receiving device to know the changed status and the time when it changed. The changed time allows receiving devices to set the relating local time to the receiving events. Even if there’s no change of the status value, GOOSE messages keep sending message in a cycle time. This ensures the activated devices know the current status values of the peer devices. From the data in our exercises, there are many attributes included in GOOSE Control Block: GOOSE Control Block name, GOOSE Control Block reference, Application ID, Data Set, Configuration Revision, Needs Commission and Services. The attributes are mentioned in the analysis. Every GOOSE message contains attributes they need. All of GOOSE messages use the same GOOSE structure. But the time to live and Boolean values are different among the messages of these laboratories because of we enable different numbers of digital nodes. Figure 4.1, 4.2 and the specification following will explain more details in GOOSE examples.

## 4.1. GOOSE message in Exercise 2

| No. | GOOSE message | Description |
|---|---|---|
| 1. | 01 0c cd 01 00 00 | Destination Address for GOOSE |
| 2. | 00 1a d3 00 19 97 | Source Address for Relay |
| 3. | 88 b8 | Ether type Values (GOOSE) |
| 4. | 00 04 | Application ID 4 |
| 5. | 00 59 | Length 89 |
| 6. | 00 00 | Reserved 1 |
| 7. | 00 00 | Reserved 2 |
| 8. | 61 4f | Implicit Sequence, APDU length 79 bytes |
| 9. | 80 09 | GOOSE Control Block Reference, 9 bytes |
| 10. | 4c 4c 4e 30 24 67 63 62 31 | LLN0$gcbl (Logic node0 GOOSE control block) |
| 11. | 81 01 | Time to Live, 1 byte |
| 12. | 14 | 20 ms |
| 13. | 82 15 | Data Set, 21 bytes |
| 14. | 56 61 6d 70 5f 32 52 65 6c 61 79 | Vamp_Relay |
| 15. | 2f 4c 4c 4e 30 24 44 53 47 31 | /LLN0$DSG1 |
| 16. | 83 04 | GOOSE ID, 4 bytes |
| 17. | 56 41 4d 50 | VAMP |
| 18. | 84 08 | Time and Quality, 8 bytes |
| 19. | 4b bb 5d 76 b9 37 48 00 | date: 03, 03, 2010, time: 18: 43: 33.1644999981<br>Quality: 00 |
| 20. | 85 01 | Status Number, 1 byte |
| 21. | 0e | 14 |
| 22. | 86 01 | Sequence Number, 1 byte |
| 23. | 06 | 6 |
| 24. | 87 01 | test, 1byte |
| 25. | 00 | 0 |
| 26. | 88 01 | Configuration Revision, 1 byte |
| 27. | 01 | 1 |
| 28. | 89 01 | needs Commission, 1 byte |
| 29. | 00 | 0 |
| 30. | 8a 01 | Number of Data Set Entries, 1 byte |
| 31. | 02 | 2 |
| 32. | ab 06 | Boolean Value, 6 bytes |
| 33. | 83 01 | |
| 34. | 00 | Boolean Value=0 |
| 35. | 83 01 | |
| 36. | 00 | Boolean Value=0 |

Figure 4.1 GOOSE message detected from exercise 2

1. Destination Address (01 0c cd 01 00 00); Source Address (01 1a d3 00 19 97); Ethertype (88 b8):

In Destination address, “01 0c cd” is assigned by IEEE, “01” means this message is transmitted with GOOSE, “00 00” is the starting address number of the GOOSE message, the end address numbered with “01 ff”. Source MAC address is the MAC

address for the laptop which connected to VAMP Feeder Manager. “88 b8” is the Ethertype values for GOOSE.

2. Application ID (00 04); Length (00 59):

Application ID is the Application Identify used for distinguishing the GOOSE message in subscriber. In this GOOSE message application ID is 4.The length of this GOOSE message is 89 bytes.

3. APDU length (61 4f):

The 23th byte “61” is signed by APDU. “4f” is the length of APDU command data. The command data length in this message is 79 bytes.

4. GoRef (80 09 4c 4c 4e 30 24 67 63 62 31):

GOOSE Control Block Reference is Logic Node 0 from GOOSE Control Block. This is the route of this message and the reference must be the unique path-name of a GOOSE Control Block within the Logic Node 0.

5. TimeAllowedtoLive (81 01 14):

The time allowed to live is 20ms. Because of the importance of GOOSE messages even if status of device is not changing, the messages should be sent repetitive. This parameter point out to subscriber that the maximum waiting time for next GOOSE message.

6. DataSet (82 15 56 61 6d 70 5f 32 52 65 6c 61 79 2f 4c 4c 4e 30 24 44 53 47 31):

Dataset may have several members called MemberOffset. Each member shall have a Member Reference referencing the data attribute with a specific functional constraint. In this dataset, “Vamp-2Relay” is the logic device, Logic node is logic node 0 and dataset group 1 is the functional constraint.

7. GoID (83 04 56 41 4D 50); Utc time & quality (84 08 4b 83 f6 9e 2a 1c ac 00):

The GOOSE ID is “VAMP” and Utc time is the time when this message is transmitted. For the data which represent time, the time tag is “4b 83 f6 9e 2a 1c ac”. The first 4 bytes stand for the amount of time in seconds from the date 01,01,1970 at time 00:00:00 to the time the data is detected. Therefore the date should be calculated by the user. The last 3 bytes calculated for the decimal part of second. Firstly, we change the hex value “4b 83 f6 9e” to decimal value “1266939550” and the unit is “second”. For calculating the exact date, we converse the unit to year, month, day, hour and minute.

The calculation of the time is shown below.

1266939550s/60s/60m/24h/365d=40.17…years

(40.17…-40)*12month=2.09… months

(2.09…-2)*30day=2.78…days

(2.78…-2)*24hour=18.72…hours

(18.72…-18)*60=43.56…minutes

(43.56…-43)*60=33.69…seconds

The last 3 bytes “2a 1c ac” was decimal point value: （0x2a 1c ac）/(2^24)= 0.1644999981second.

To the addition of the value that compute to the original date and time, and then calculate the date and time we in the experiment is：

Year： 1970+40=2010

Month： 01+02=03

Day： 01+02=03

Hour: 00+18=18

Minute: 00+43=43

Second: 00+33.2759852=33. 1644999981

For all the calculation above, the exact date is: 03,03,2010

Time is: 18:43:33.1644999981

8. StNum (85 01 0e); SeqNum (86 04 06):

Status Number shows how many times status changes. Sequence number of this message is 6. There's no testing during this configuration.

9. ConfRev (88 01 01):

Configuration Revision presents a count of configuration of the dataset has been changed. The count should not include a restart of the IED.

10. NdsCom (89 01 00):

"Needs Commission" have a value of true if the attribute dataset has a value of "null". In this message because the attribute dataset is not null, this value is false.

11. numdatSetEntries (8a 01 02); Boolean Values (ab 06 83 01 00 83 01 00):

The number of dataset entries is 1. The two Boolean values are the values of the inputs applied in exercise 2. The first one is the value of digital input 1 and the other value is for Virtual input 1. So as the result of true setting in exercise 2, this GOOSE message shows the false value which presents in IEC Simple Tester as well.

For the tag Boolean Value, 01 means True, 00 means false. And the corresponded Boolean Value also reacts in IEC Simple Tester. If the value is true with the value "1 0", GOOSE Sender corresponds to be "T, F". If the value was false as "0 0", GOOSE Sender corresponds to be "F, F".

## 4.2. GOOSE message in Exercise 3

| No. | GOOSE message | Description |
|---|---|---|
| 1. | 01 0c cd 01 00 00 | Destination Address for GOOSE |
| 2. | 00 1a d3 00 19 97 | Source Address for Relay |
| 3. | 88 b8 | Ether type Values (GOOSE) |
| 4. | 00 04 | Application ID 4 |
| 5. | 00 59 | Length 89 |
| 6. | 00 00 | Reserved 1 |
| 7. | 00 00 | Reserved 2 |
| 8. | 61 53 | Implicit Sequence, APDU length 83 bytes |
| 9. | 80 09 | GOOSE Control Block Reference, 9 bytes |
| 10. | 4c 4c 4e 30 24 67 63 62 31 | LLN0$gcbl (Logic node0 GOOSE control block) |
| 11. | 81 01 | Time to Live, 1 byte |
| 12. | 14 | 20 ms |
| 13. | 82 15 | Data Set, 21 bytes |
| 14. | 56 61 6d 70 5f 32 52 65 6c 61 79 | Vamp_Relay |
| 15. | 2f 4c 4c 4e 30 24 44 53 47 31 | /LLN0$DSG1 |
| 16. | 83 04 | GOOSE ID, 4 bytes |
| 17. | 56 41 4d 50 | VAMP |
| 18. | 84 08 | Time and Quality, 8 bytes |
| 19. | 4b bb 5d 76 b9 37 48 00 | date: 14, 03, 2010, time: 05: 28: 04.41099977493<br>Quality: 00 |
| 20. | 85 01 | Status Number, 1 byte |
| 21. | 34 | 52 |
| 22. | 86 01 | Sequence Number, 1 byte |
| 23. | 0d | 13 |
| 24. | 87 01 | test, 1byte |
| 25. | 00 | 0 |
| 26. | 88 01 | Configuration Revision, 1 byte |
| 27. | 01 | 1 |
| 28. | 89 01 | needs Commission, 1 byte |
| 29. | 00 | 0 |
| 30. | 8a 01 | Number of Data Set Entries, 1 byte |
| 31. | 03 | 3 |
| 32. | ab 09 | Boolean Values, 9 bytes |
| 33. | 83 01 | |
| 34. | 00 | Boolean Value=0 |
| 35. | 83 01 | |
| 36. | 00 | Boolean Value=0 |
| 37. | 83 01 | |
| 38. | 00 | Boolean Value=0 |

Figure 4.2 GOOSE message detected from exercise3

Figure 4.2 illustrates the GOOSE message and description. This message is intercepted in Wireshark in exercise 3. The destination address (01 0c cd 01 00 00), source address (00 1a d3 00 19 97) and ethertype value (88 b8) are the same as the message in exercise 2. The differences between the two messages in 4.1 and 4.2 are the length of the whole message, the length of APDU command, Utc time, Status number, Sequence number and Boolean values. The specification below presents element content which are different from exercise 3.

Line 5(00 59) and Line 8 (61 53) in Figure 4.2 show the length of this message is 89 bytes and the length of APDU is 83 bytes.

1. TimeAllowedtoLive (81 01 14):

Because of the importance of GOOSE message, GOOSE must keep sending even though the state is not changed. Time allowed to live is the longest time for subscribers waiting for the next message. When the waiting time is longer than the time to live and the subscriber still doesn't receive the message. Subscriber thinks the communication is missing. The value should be replaced by the default value defined before. The time allowed to live here is 20 ms.

2. Utc time & quality (84 08 4b bb 5d 76 69 37 48 00)

The method of calculating the Utc time is introduced in last exercise. In this message, the date is 14,03,2010 at the time 05：28：04. Quality is satisfied.

3. StNum (85 01 34):

Status number decides whether this message is a new event. Just check if the state number is the same as the value of GOOSE message. If they are the same, this message is not a new event. So in this case, the amount of testing is 52 which is much more than the number in last message. It shows there are several tests during the two exercises.

4. SeqNum (86 01 0d):

The sequence Number of a new event must be 1. But the sequence number of

message retransmitted is steadily increasing.

5. Boolean Values (ab 09 83 01 00 83 01 00 83 01 00):

There are 3 Boolean Values in this message. They are the values of Digital Input 1, Virtual Input 1 and Virtual Output 6. The values react in the result shown in the IEC Simple Tester of exercise 3. In the result, the three values are false which correspond to the Boolean value ‘00’ ‘00’ ‘00’ in the message.

# 5. Conclusion

This project has focus on creating telecommunication Laboratory exercises about using IEC61850. The three telecommunication Laboratories are designed according to the principle where by the uses are familiar with the idea of the IEC standard, and secondly, with the Vamp Relay. Searching the GOOSE structure and decoding GOOSE message cost a lot of time during the whole procedure.

When the team of three was facing the difficulties typical to similar new technical problems, we encouraged ourselves and never lost faith. We kept looking for interrelated references and generated precious suggestions. These are effective manners to solve the problems.

In the project, we have written pre-laboratories, laboratory instructions to help students and teachers with researching the IEC61850 exercises. Each report presents the purpose, method and request for students who will do the laboratory exercises to understand and figure out problems for themselves. Hopefully, the precious instructions will support and guide through the teachers during the telecommunication Laboratory courses. Maybe our project is still unfinished, due to the limited scope of the final year research project. However, the three of us would eagerly welcome an opportunity to spend more time on further research for this interesting protocol.

REFERENCES:

[1] IEC61850 Interface in Vamp Relays Configuration Guide. Author: Wojciech Kozlowski. Date: 25-11-2009(initial version).

[2]Configuration of the 61850 Interface of Vamp 50/257/259 Version 1.4.

[3]The Application of IEC61850 to Replace Auxiliary Devices Including Lockout Relays. Author: Robert Brantley; Kevin Donahoe P.E; Jacobus Theron; Eric Udron.

[4]MMS and ASN.1 Encodings; Description: Simple Examples and Explanations on How to Crack an MMS PDU. Author: Herbert Falk (SISCO) AND Dr. Martin Burns (Hyportek)

[5]International Standard IEC61850-5; Communication Networks and Systems in Substations. Part 5: Communication Requirements for Functions and Device Models. Date: 07-20003(first edition). Reference No.: IEC61850-5, 2003(E).

[6] International Standard IEC61850-6; Communication Networks and Systems in Substations. Part 6: Configuration Description Language for Communication in Electrical Substations related to IEDs. Date: 03-2004(first edition) Reference No.: IEC61850-6, 2004(E).

[7] International Standard IEC61850-7-1; Communication Networks and Systems in Substations. Part 7-1: Basic Communication Structure for Substation and Feeder Equipment-Principles and Models. Date: 07-2003(first edition) Reference No.: IEC61850-7-1, 20003(E).

[8] International Standard IEC61850-7-2; Communication

Networks and Systems in Substations. Part 7-2: Basic Communication Structure for Substation and Feeder Equipment-Abstract Communication Service Interface (ASCI). Date: 05-2003(first edition) Reference No.: IEC61850-7-2, 2003(E).

[9] International Standard IEC61850-7-3; Communication Networks and Systems in Substations. Part 7-3: Basic Communication Structure for Substation and Feeder Equipment-Common Data Classes. Date: 05-2003(first edition) Reference No.: IEC61850-7-3, 2003(E).

[10] International Standard IEC61850-7-4; Communication Networks and Systems in Substations. Part 7-4: Basic Communication Structure for Substation and Feeder Equipment-Compatible Logical Node Classes and Data Classes. Date: 05-2003(first edition) Reference No.: IEC61850-7-4, 2003(E).

[11] International Standard IEC61850-8-1; Communication Networks and Systems in Substations. Part 8-1: Specific Communication Service Mapping (SCSM)-Mappings to MMS and to ISO/IEC 8802-3. Date: 05-2004(first edition) Reference No.: IEC61850-8-1, 2004(E).

[12]Manufacturing Message Specification. Author: Jan Tore Sorensen, Martin Gilje Jaatun.

[13]A Compact Abstract Syntax for MMS. Author: Lausanne, Switzerland.