Connecting UniOP to Suconet K

This Technical Note contains all the information required to connect the UniOP panels to a Suconet K network with a Klockner Moeller PLC as the master and to take advantage from the advanced data access options supported by the UniOP communication driver.

***Important**: this Technical Note applies only to the Suconet K/K1 communication driver identified by the name 'Suconet K' associated to the Designer file UPLC101.DLL. To run this version **it is required** an operator panel withf hardware type –0045 and a communication module type TCM02. The communication driver contained in UPLC101.DLL is **not compatible** with the older hardware type –0043.*

Contents

1. Introduction 2
2. PS4-xxx-MM1 3
   2.1. Packing List 3
   2.2. Installation 3
   2.3. PLC Configuration 4
   2.4. Sample PLC Program 5
   2.5. Sample PLC Program with Sucosoft V2.11 7
   2.6. The Designer 9
3. PS416 10
   3.1. Packing List 10
   3.2. Installation 11
   3.3. PLC Configuration 13
   3.4. Sample PLC Program 16
   3.5. Sample PLC Program with Sucosoft V2.11 19
   3.6. The Designer 20
4. PS4-100 (Suconet K1) 21
   4.1. Packing List 21
   4.2. Installation 21
   4.3. PLC Configuration 21
   4.4. Sample PLC Program 22
   4.5. The Designer 24
5. PS 316 (Suconet K1) 26
   5.1. Packing List 26
   5.2. Installation 26
   5.3. PLC Configuration 26
   5.4. Sample PLC Program 27
   5.5. The Designer 29
Appendix 1 - Communication Error Codes 30
Appendix 2 - Technical Data and Connection Information 32

1. Introduction

This Technical Note describes how to connect a UniOP panel to the Klockner Moeller Suconet K network. UniOP panels can be connected to Suconet K as slave units; a KM PLC will act as master unit. UniOP will communicate directly with the master unit and not with other slave stations.
The KM PLCs that can be connected are the following:

- PS4-200-MM1, PS4-140-MM1, PS4-150-MM1
- PS416
- PS4-100
- PS306, PS316

The PS4-100, PS306 and PS316 are actually using Suconet K1

Suconet K communication requires the special communication module TCM02 which can be used in operator panels with hardware type –0045.

Suconet K communication with UniOP requires a special server program running on the master PLC. This server program can be easily integrated in the user's PLC application.

A disk containing sample PLC programs and PLC configuration files is available for a quick start with Suconet K applications.

2. PS4-xxx-MM1

This chapter describes how to connect a UniOP to a Suconet K network with a PS4-200 as the master.

2.1. Packing List

The sample program includes the following files:

EM_A0EF.CFG	device configuration file
PS4K_V03.P42	sample PLC program
PS4K_V03.Q42	
PS4K_V03.R42	
PS4K_V03.Z42	reference file
PS4K_V03.K42	configuration file

2.2. Installation

The original PS4 programming software does not support UniOP and, therefore, some modifications are necessary to support it. Assuming that original KM software is already installed, the device configuration file EM_A0EF.CFG must be copied to the CFG subdirectory of the programming software. The numbers in the filename indicate that it is configuration file for a non-KM device with variable communication data length. The content of the file is shown below.

```
[Version]
VersionNo=1.0
Date=05.12.95
Author=DT
[ModulOffline]
Box=3,16
Text0=1,1,SIS-TYP-UNI
[Modultype]
Type=EM
Ident=A0EF
[Network]
Type=SLAVE
MaxSlaveInput=78
MaxSlaveOutput=78
MaxSlaveIO=78
RunHalt=1
Net=K
[InputDiagnose]
Count=1,B
Access=XB
Offset=0
Attrib=R
Text0=Bit0
Text1=Halt
Text2=Input Length Error
Text3=Bit3
Text4=Hardware Error
Text5=Short Circuit
Text6=No Connection
Text7=Unexpected Type
[InputCommunication]
Count=0,B
```

```
Access=BW
Offset=1
Attrib=R
[OutputCommunication]
Count=0,B
Access=BW
Offset=0
Attrib=RW
[Image]
InImageLength=1
OutImageLength=0
[Device]
Parameter=KSlave
```

The meaning of most of the items is obvious. MaxSlaveInput and MaxSlaveOutput limits are due to the current communication memory limit of PS4-xxx-MM1. UniOP can send and receive telegrams up to 240 bytes long what is maximum by K and K1 protocol. So actual data length depends more upon the capabilities of the master device than upon UniOP. Communication memory of master device must be divided in separate parts if more than one slave device is connected. Minimal number of communication data for correct operation of UniOP is 7 bytes of Input data length and 6 bytes of Output data. That is from a point of view of master device, from UniOP side it means to be able to send 7 bytes and receive 6 bytes (Input data of master = Output data of slave).

***Note**:* *the content of the field Text, which is shown in the Device Configurator screen, has been changed from the previous Suconet K version and is now SIS-TYP-UNI (it was SIS-TYP-UNIOP in the previous version). Note also that the identification number (field Ident) has been changed from A0DF to the new value of A0EF.*

2.3. PLC Configuration

The PS4 PLC used as a master must be properly configured. Follows the steps shown below:

a) In DEVICE CONFIGURATOR select PLC as master device

b) Choose the Add station command

Figure 1 - Device Configurator screen

c) scroll down to the bottom of the list of available devices until SYS-TYPE-UNI is selected and confirm. The absolute position of device in configuration file defines its network address. The master is counted as address zero and the first device (being for example UniOP) has address of 1. The address of the slaves is defined by their position in the network in the network configuration itself.

d) edit parameters

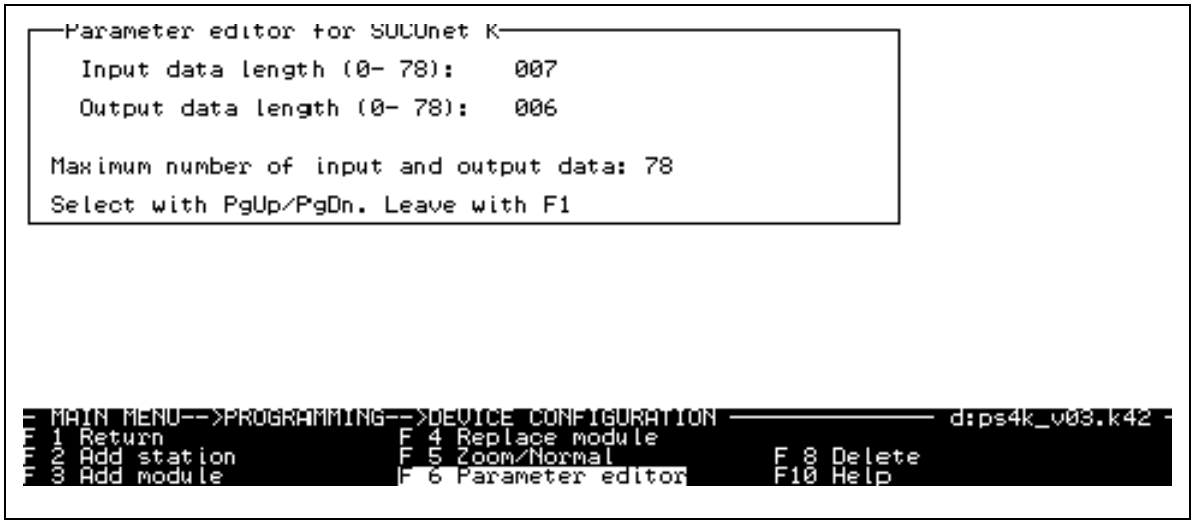

Figure 2 - Parameter editing for UniOP

The address is already defined by position and all what is left is to define Input and Output data len. That is done in displayed dialog box which is exactly the same as the dialog box for a PLC slave device. When configuring it please take into account total available communication area in the master, number of slave devices, their communication data lengths and minimal communication data length required by the UniOP station. Generally speaking, the larger the data size that UniOP can use, the faster will it operate. Minimal values are described in the previous chapter.

e) after slave configuration it is necessary to configure the master device using the same procedure (e.g. pressing F6). In the dialog box for the master unit, select baud rate and checksum type. UniOP supports 375 and 187.5 KBd and CRC16 set to ON or OFF. Baud rate depends upon other devices connected and protocol type (K1 supports only 187.5 KBd), but checksum type is independent for a device. It is recommended to set the checksum type to CRC16 ON for reasons of communication data reliability.

2.4. Sample PLC Program

This PLC program is a ready-to-run program supporting Suconet K communication. It can easily be included in any user application to add UniOP communication capability.
The program does not use any PLC memory register in the Merker area.

```
  " Printout of file d:ps4k_v03.q42   of the: 12. 7.97"

00000 START     "entry point of program - MASTER side
001             #include "ps4k_v03.k42"
002
003             L RDB1.1.0.0            job
004             CP SDB1.1.0.0           last job executed
005             BE  END
```

```
006
007             L RDB1.1.0.1
008             CP KHB 40
009             BNE  PENDING
010
011             L KB 4
012             = SDB1.1.0.1
013             JP  JOB
014
00001 PENDING   "pending request from UniOP
001             L& MB0
002             ADD RDW1.1.0.2
003             = ICPY1 SADR
004             = ICPY2 DADR
005
006             L RDB1.1.0.1
007             CP KHB 90
008             BE  WRITE_BY
009
010             L RDB1.1.0.1
011             CP KHB 10
012             BE  READ_BYT
013
014             L KHB 80
015             = SDB1.1.0.1
016             JP  JOB
017
00002 READ_BYT  "read bytes procedure
001
002             ICPY1
003           [ ] MOD:  K  1
004           [&] SADR:
005           [&] DADR: & SDB1.1.0.2
006           [b] NO:   RDB1.1.0.4
007           [b] ERR:
008
009             L ICPY1 ERR
010             BZ  READOK
011
012             L KHB 81
013             = SDB1.1.0.1
014             JP  JOB
015
00003 READOK    "
001             L KHB 01
002             = SDB1.1.0.1
003             JP  JOB
004
00004 WRITE_BY  "write bytes procedure
001
002             ICPY2
003           [ ] MOD:  K  1
004           [&] SADR: & RDB1.1.0.5
005           [&] DADR:
006           [b] NO:   RDB1.1.0.4
007           [b] ERR:
008
009             L ICPY2 ERR
010             BZ  WRITOK
011
012             L KHB 82
013             = SDB1.1.0.1
014             JP  JOB
015
00005 WRITOK    "
001             L KHB 02
002             = SDB1.1.0.1
003
004
00006 JOB       "
001             L RDB1.1.0.0            job
002             = SDB1.1.0.0            last job executed
003
```

```
00007 END        "end of communication part
001              EP
```

2.5. Sample PLC Program with Sucosoft V2.11

The fieldbus support disk provided with Designer contains also a sample PLC program created with Sucosoft V2.11.
Klockner Moeller developed a special function block called MI4K you could use in order to simplify the communication setup with UniOP.
In “Topology Configurator” use the Remote Expansion function to add the Suconet K slave.
The “Topology Configurator” dialog box will appear as shown in Figure 3.

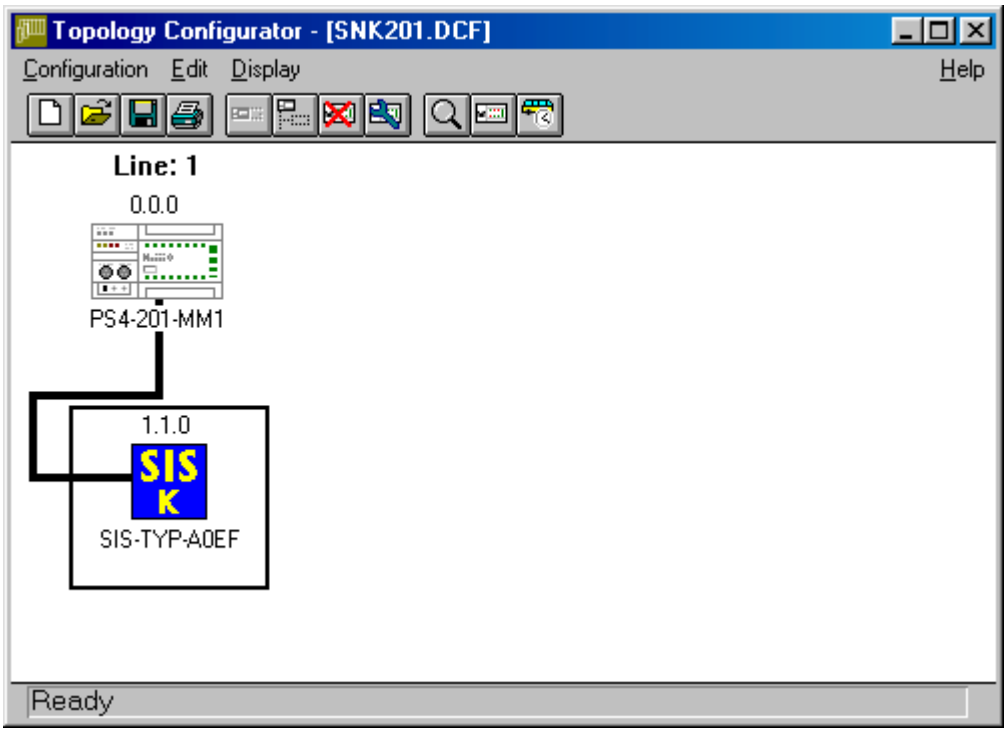

Figure 3 – Topology Configurator

The panel Configuration window will appear as shown in Figure 4 where you should properly set the buffer size.

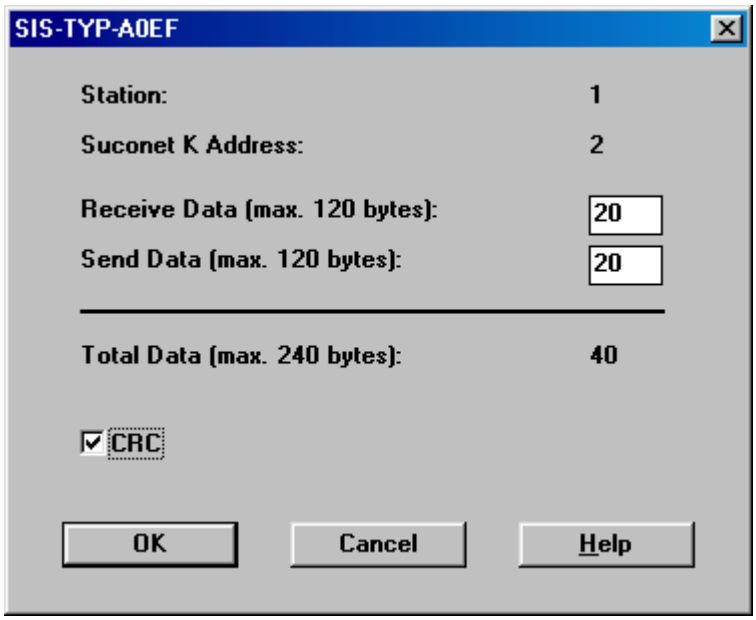

Figure 4 – UniOP configuration dialog box

The PLC program at this point is very simple. You should simply add the call to an instance of MI4K function block.
The declaration editor in the free form will be similar to the following example:

```
VAR_GLOBAL
Read_Data1      AT %RDB1.1.0.0  : ARRAY [1..20] OF BYTE;
Send_Data1      AT %SDB1.1.0.0  : ARRAY [1..20] OF BYTE;
Diagnose1                        : Byte;

DiagISB          AT %ISB1.1.0.0  : Byte;
outp                             : Byte;
outdia                           : Byte;
END_VAR

VAR
SuconetK    :MI4K;
END_VAR
```

Please note that the buffer size you choose in the Topology Configurator must match the array size you used in the declaration section.

The PLC program should be similar to following example:

```
CAL SuconetK(
RDB_Start :=Read_Data1,
SDB_Start :=Send_Data1
|
 Diagnose1:=Status
 )
 LD DiagISB
 ST outp
```

```
LD Diagnose1
ST outdia
```

Figure 5 shows the POU window with the simple program and the declaration section.

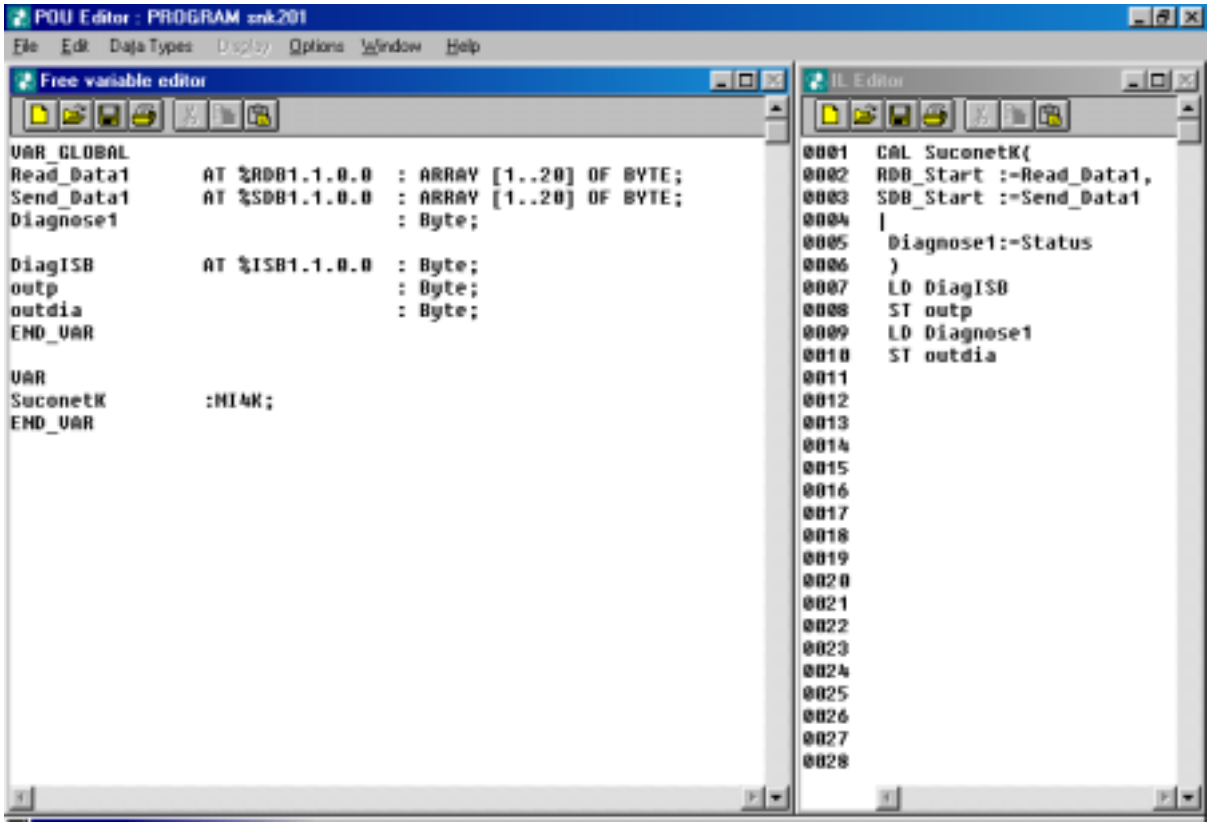

Figure 5- The POU Editor window

In case you have more than one panel connected to Suconet-K you should use one instance of MI4K function for each panel. The variables declaration section must match the new topology configuration.

2.6. The Designer

Data in PLC program configuration file must match the data specified in the Designer project file, otherwise the communication will not be established between UniOP and PLC. The data that must match are:

- UniOP node
- Input and Output data length must be exactly the same as in PLC configuration

3. PS416

This chapter describes how to connect a UniOP to a Suconet K network with a PS416 as the master.

3.1. Packing List

The delivery of the sample program includes the following files:

```
EM_A0EF.CFG          device configuration file

├─DATPARAM.VAR       sample PLC program and all associated files
├─USERTYPE.TYP
├─USERPROT.PTT
├─KOMPO.SYS
│    ├─PANEL_01.POE
│    ├─PANEL_01.DCF
│    ├─PANEL_01.INC
│    ├─PANEL_01.ICF
│    ├─PANEL_01.ICD
│    └─PANEL.CCF
└─PROJEKT.SYS
       S40_PROJ.DEF
       TOPOL.MAK
       OUTPUT.OUT
       DEVICE
          ├─DEV_1.MAK
          └─CONFIG
               ├─KON_1.MAK
               └─RESOURCE
                    ├─RES_1.MAK
                    ├─PANEL_01.OBJ
                    ├─PANEL_01.LOD
                    └─PANEL_01.PCD
```

The name of the sample program is PANEL_01; the directory structure shown above is consistent with the one required by SUCOSOFT.

3.2. Installation

The original PS416 programming software does not support UniOP and, therefore, some modifications are necessary to support it. Assuming that original KM software is already installed, the device configuration file EM_A0EF.CFG must be copied to the CFG subdirectory of the programming software. The numbers in the filename indicate that it is configuration file for a non-KM device with variable communication data length. The content of the file is shown below.

```
[Version]
VersionNo=1.0
Date=12.07.96
Author=M.Crestani
[ModulOffline]
Box=3,16
Text0=1,1,SYS-TYP-UNI
[Modultype]
Type=EM
MinSlot=0
MaxSlot=0
Ident=A0EF
[Network]
Type=SLAVE
MaxSlaveInput=20
MaxSlaveOutput=20
MaxSlaveIO=40
RunHalt=1
Net=K
[InputDiagnose]
Count=1,B
Access=XB
Offset=0
Attrib=R
Text0=Bit0
Text1=Halt
Text2=Input Length Error
Text3=Bit3
Text4=Hardware Error
Text5=Short Circuit
Text6=No Connection
Text7=Unexpected Type
[InputCommunication]
Count=0,B
Access=BW
Offset=1
Attrib=R
[OutputCommunication]
Count=0,B
Access=BW
Offset=0
Attrib=RW
[Image]
InImageLength=1
OutImageLength=0
[Device]
Parameter=KSlave
Structure=c:c:c
```

UniOP can send and receive telegrams up to 240 bytes long, which is the maximum for K and K1 protocols. The actual data length depends more upon the capabilities of the master device than upon UniOP. Communication memory of master device must be divided in separate parts if more than one slave device is connected. Minimal number of communication data for correct operation of UniOP is 7 bytes of Input data length and 6 bytes of Output data. The master device, from the UniOP, side must be able to send 7 bytes and receive 6 bytes (Send data of master = Receive data of slave).

Note: *the content of the field Text, which is shown in the Device Configurator screen, has been changed from the previous Suconet K version and is now SIS-TYP-UNI (it was SIS-TYP-UNIOP in the previous version).*

The sample program must be loaded using the SUCOSOFT Project Manager. Once loaded in the SUCOSOFT the program structure is shown as in the figure below which shows the window copied from an old version of the programming package.

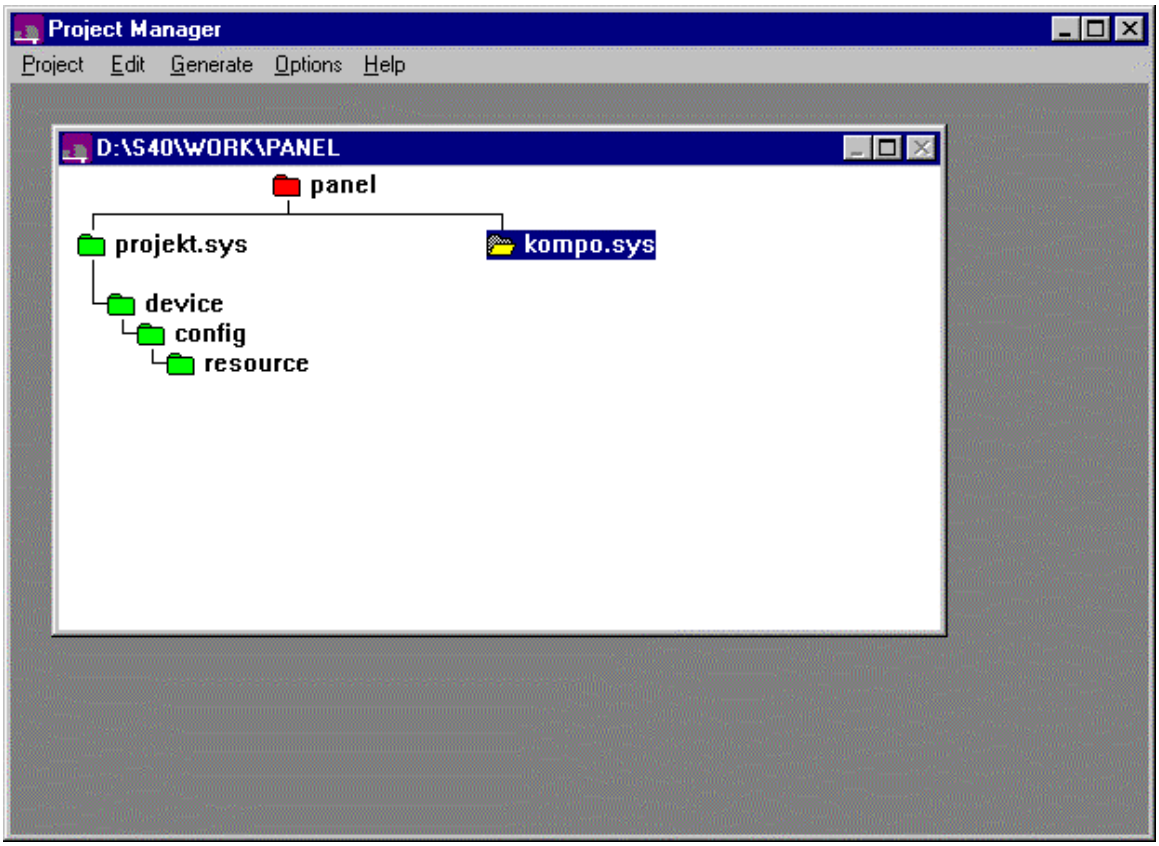

Figure 6 - Project Manager

3.3. PLC Configuration

The PS416 PLC used as a master must be properly configured. Follows the steps shown below:

a) In "Device Configurator" ("Topology Configurator" in Sucosoft V2.11) define the network configuration and select the PLC as the master device:

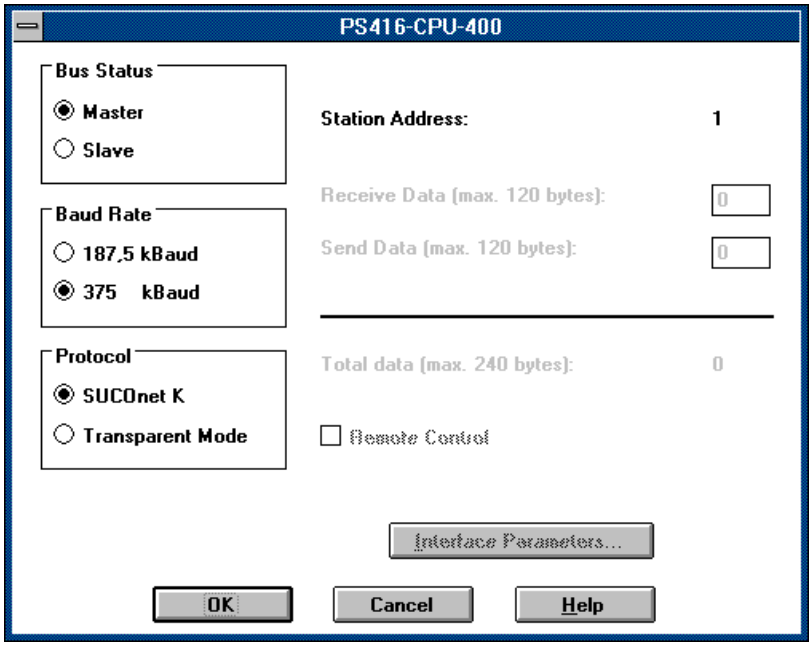

Figure 7 - Configuration of the master

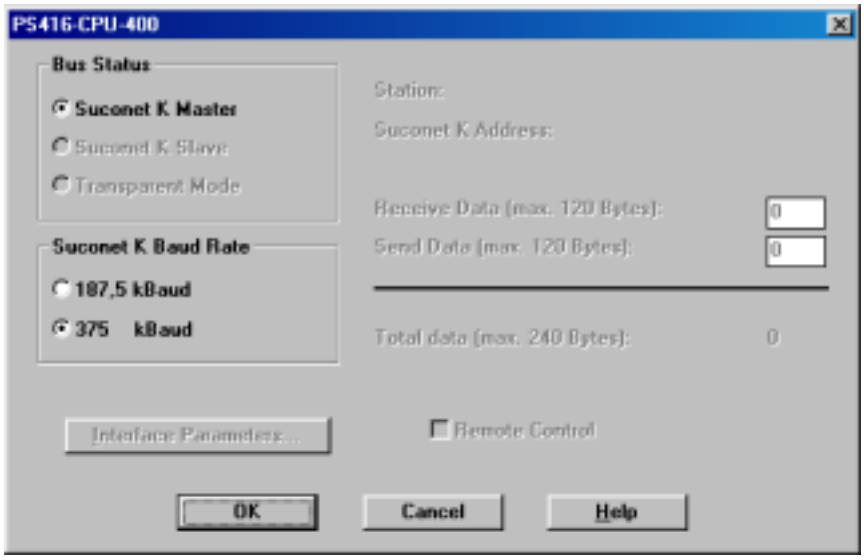

Figure 8 - Configuration of the master (Sucosoft V2.11)

In the dialog box for the master device, select the baud rate. UniOP supports 375 and 187.5 KBd. Baud rate depends upon other devices connected and protocol type (K1 supports only 187.5 KBd).

Figure 9 and Figure 10 show respectively the “Device Configurator” windows from old Sucosoft version and the “Topology Configurator” window from Sucosoft V2.11.

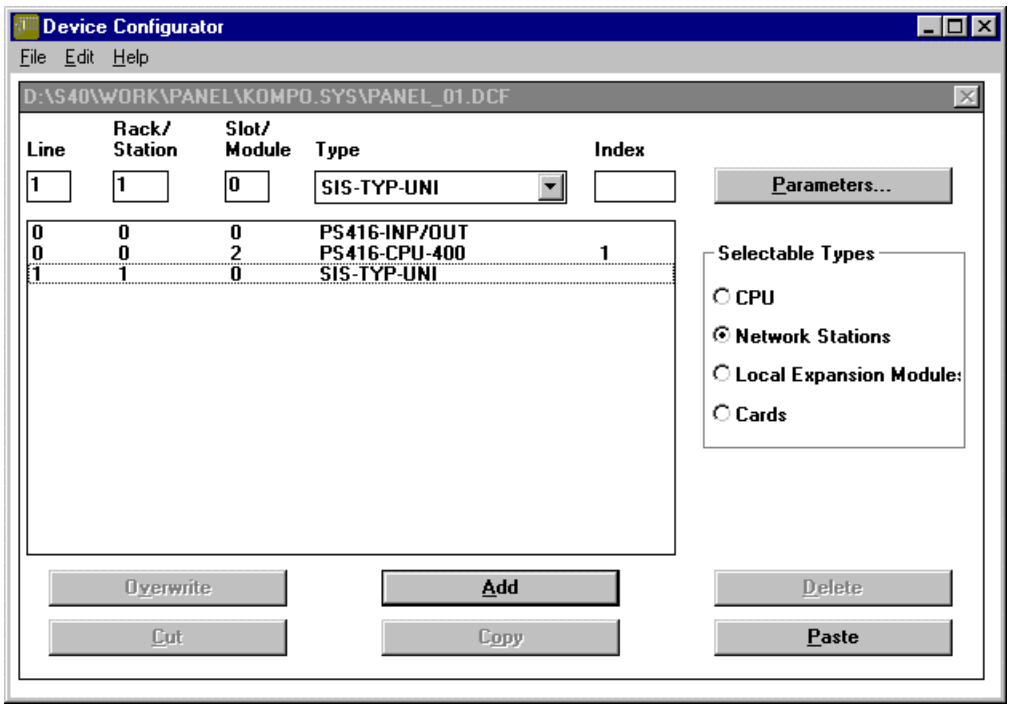

Figure 9 - Network configuration

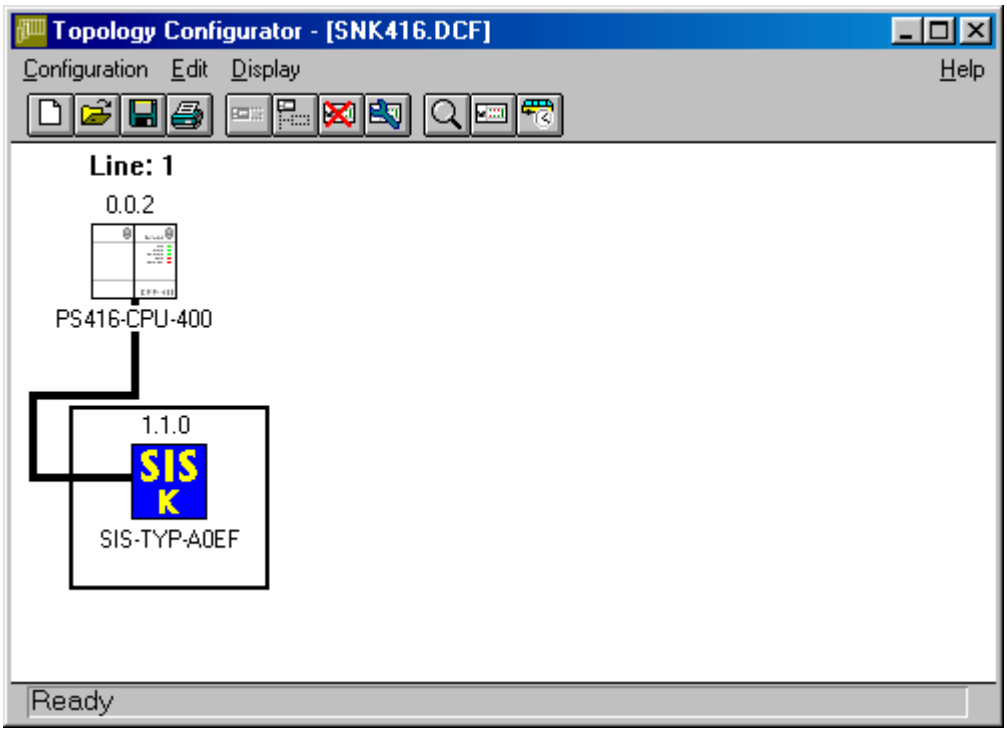

Figure 10 – Topology Configurator window with Sucosoft V2.11

b) Add a new station for UniOP

c) The absolute position of the device in the configuration defines its network address. The master is counted as address 1 and the first device (being for example UniOP) has a network address of 2. The address of the slaves is defined by their position in the network in the network configuration itself.

d) edit communication parameters for the slave station

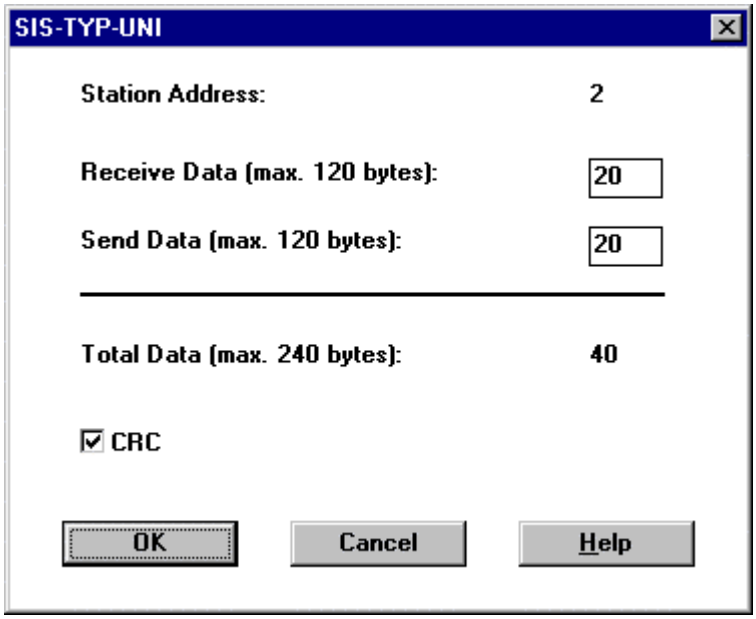

Figure 11 - Configuration of UniOP as a slave device

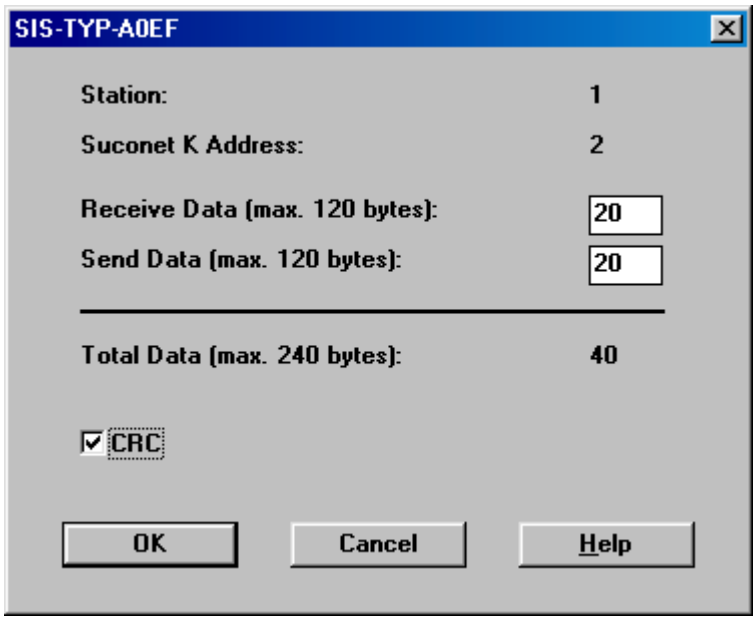

Figure 12- Configuration of UniOP as a slave device (Sucosoft V2.11)

The address is already defined by the position and only Receive and Send data size must be defined. That is done in the displayed dialog box which is exactly the same as the dialog box for a PLC slave device. When configuring it, please take into account total available communication area in the

master, number of slave devices, their communication data lengt has and minimal communication data length required by the UniOP station. Generally speaking, the larger the data size that UniOP can use, the faster will it operate. Minimal values are described in the previous chapter.

Important note:
the values entered in this dialog box must be the same which are assumed by the PLC program.

3.4. Sample PLC Program

This PLC program is a ready-to-run program supporting Suconet K comunication. It can easily be included in any user application to add UniOP communication capability.

Important note:
the two communication buffers must be defined large enough to hold the full size of the input and output data length which has been configured for the panel in the Suconet network configuration.

The listing of the sample program is shown below.

```
PROGRAM panel_01
(* Operator Panel - PS416 Program Interface - 27.06.96 by M. Crestani *)
(* The exchange mailbox length is set to 20 bytes, the Operator Panel network address is 1 *)

VAR_GLOBAL
        PANEL_AREA : ARRAY [0..1000] OF BYTE ;  (* Operator Panel Exchange Memory Area *)
END_VAR

VAR
        (* PANEL -> PLC Communication Buffer *)
        R_JOBNUMBER     AT %RDB1.1.0.0 : BYTE ;
        R_OPERATION     AT %RDB1.1.0.1 : BYTE ;
        R_OFFSET        AT %RDW1.1.0.2 : UINT ;
        R_LEN           AT %RDB1.1.0.4 : BYTE ;
        R_DATA00        AT %RDB1.1.0.5: BYTE ;
        R_DATA01        AT %RDB1.1.0.6: BYTE ;
        R_DATA02        AT %RDB1.1.0.7: BYTE ;
        R_DATA03        AT %RDB1.1.0.8: BYTE ;
        R_DATA04        AT %RDB1.1.0.9: BYTE ;
        R_DATA05        AT %RDB1.1.0.10: BYTE ;
        R_DATA06        AT %RDB1.1.0.11: BYTE ;
        R_DATA07        AT %RDB1.1.0.12: BYTE ;
        R_DATA08        AT %RDB1.1.0.13: BYTE ;
        R_DATA09        AT %RDB1.1.0.14: BYTE ;
        R_DATA10        AT %RDB1.1.0.15: BYTE ;
        R_DATA11        AT %RDB1.1.0.16: BYTE ;
        R_DATA12        AT %RDB1.1.0.17: BYTE ;
        R_DATA13        AT %RDB1.1.0.18: BYTE ;
        R_DATA14        AT %RDB1.1.0.19: BYTE ;

        (* PLC -> PANEL Communication Buffer *)
        S_JOBNUMBER     AT %SDB1.1.0.0 : BYTE ;
        S_STATUS        AT %SDB1.1.0.1 : BYTE ;
        S_DATA00        AT %SDB1.1.0.2 : BYTE ;
        S_DATA01        AT %SDB1.1.0.3 : BYTE ;
        S_DATA02        AT %SDB1.1.0.4 : BYTE ;
        S_DATA03        AT %SDB1.1.0.5 : BYTE ;
        S_DATA04        AT %SDB1.1.0.6 : BYTE ;
        S_DATA05        AT %SDB1.1.0.7 : BYTE ;
        S_DATA06        AT %SDB1.1.0.8 : BYTE ;
        S_DATA07        AT %SDB1.1.0.9 : BYTE ;
        S_DATA08        AT %SDB1.1.0.10 : BYTE ;
        S_DATA09        AT %SDB1.1.0.11 : BYTE ;
        S_DATA10        AT %SDB1.1.0.12 : BYTE ;
        S_DATA11        AT %SDB1.1.0.13 : BYTE ;
        S_DATA12        AT %SDB1.1.0.14 : BYTE ;
        S_DATA13        AT %SDB1.1.0.15 : BYTE ;
```

```
        S_DATA14        AT %SDB1.1.0.16 : BYTE ;
        S_DATA15        AT %SDB1.1.0.17 : BYTE ;
        S_DATA16        AT %SDB1.1.0.18 : BYTE ;
        S_DATA17        AT %SDB1.1.0.19 : BYTE ;

        (* Temporary variable used to copy PANEL Transmission Buffer <-> Global PANEL Memory
           Area *)
        TMPBUFFER : ARRAY [0..20] OF BYTE ;
        SRC_INDEX : UINT ;
        DST_INDEX : UINT ;
END_VAR

        (* PANEL: Command Interface *)

        LD      R_JOBNUMBER
        EQ      S_JOBNUMBER             (* New Message? *)
        JMPC    END

        LD      R_OPERATION
        NE      16#40                   (* Message is IDLE *)
        JMPC    PENDING
ACK:
        LD      04                      (* answer to IDLE *)
        ST      S_STATUS
        JMP     JOB
PENDING:
        (* New request incoming *)
        LD      R_OPERATION
        EQ      16#90                   (* Write data request *)
        JMPC    WRITE_BYT
        LD      R_OPERATION
        EQ      16#10                   (* Read data request *)
        JMPC    READ_BYT
        LD      16#80                   (* Unknown command request *)
        ST      S_STATUS
        JMP     JOB
WRITE_BYT:
        (* Write: Copy RD area to temporary buffer *)
        LD      R_DATA00
        ST      TMPBUFFER[0]
        LD      R_DATA01
        ST      TMPBUFFER[1]
        LD      R_DATA02
        ST      TMPBUFFER[2]
        LD      R_DATA03
        ST      TMPBUFFER[3]
        LD      R_DATA04
        ST      TMPBUFFER[4]
        LD      R_DATA05
        ST      TMPBUFFER[5]
        LD      R_DATA06
        ST      TMPBUFFER[6]
        LD      R_DATA07
        ST      TMPBUFFER[7]
        LD      R_DATA08
        ST      TMPBUFFER[8]
        LD      R_DATA09
        ST      TMPBUFFER[9]
        LD      R_DATA10
        ST      TMPBUFFER[10]
        LD      R_DATA11
        ST      TMPBUFFER[11]
        LD      R_DATA12
        ST      TMPBUFFER[12]
        LD      R_DATA13
        ST      TMPBUFFER[13]
        LD      R_DATA14
        ST      TMPBUFFER[14]
        (* Write: Copy from temporary buffer to PANEL_AREA*)
        LD      0
        ST      SRC_INDEX               (* Source index on temporary buffer *)
        LD      R_OFFSET
        ST      DST_INDEX               (* Destination index on PANEL_AREA *)
```

```
CPY1:
        LD      TMPBUFFER[SRC_INDEX]
        ST      PANEL_AREA[DST_INDEX]
        LD      DST_INDEX
        ADD     1
        ST      DST_INDEX
        LD      SRC_INDEX
        ADD     1
        ST      SRC_INDEX
        UINT_TO_BYTE
        NE      R_LEN
        JMPC    CPY1
        LD      02                      (* answer to WRITE *)
        ST      S_STATUS
        JMP     JOB
READ_BYT:
        (* Read: Copy from PANEL_AREA to temporary buffer *)
        LD      R_OFFSET
        ST      SRC_INDEX               (* Source index on PANEL_AREA *)
        LD      0
        ST      DST_INDEX               (* Destination index on temporary buffer *)
CPY2:
        LD      PANEL_AREA[SRC_INDEX]
        ST      TMPBUFFER[DST_INDEX]
        LD      SRC_INDEX
        ADD     1
        ST      SRC_INDEX
        LD      DST_INDEX
        ADD     1
        ST      DST_INDEX
        UINT_TO_BYTE
        NE      R_LEN
        JMPC    CPY2
        (* Read: Copy temporary buffer to SD area *)
        LD      TMPBUFFER[0]
        ST      S_DATA00
        LD      TMPBUFFER[1]
        ST      S_DATA01
        LD      TMPBUFFER[2]
        ST      S_DATA02
        LD      TMPBUFFER[3]
        ST      S_DATA03
        LD      TMPBUFFER[4]
        ST      S_DATA04
        LD      TMPBUFFER[5]
        ST      S_DATA05
        LD      TMPBUFFER[6]
        ST      S_DATA06
        LD      TMPBUFFER[7]
        ST      S_DATA07
        LD      TMPBUFFER[8]
        ST      S_DATA08
        LD      TMPBUFFER[9]
        ST      S_DATA09
        LD      TMPBUFFER[10]
        ST      S_DATA10
        LD      TMPBUFFER[11]
        ST      S_DATA11
        LD      TMPBUFFER[12]
        ST      S_DATA12
        LD      TMPBUFFER[13]
        ST      S_DATA13
        LD      TMPBUFFER[14]
        ST      S_DATA14
        LD      TMPBUFFER[15]
        ST      S_DATA15
        LD      TMPBUFFER[16]
        ST      S_DATA16
        LD      TMPBUFFER[17]
        ST      S_DATA17
        LD      01                      (* answer to READ *)
        ST      S_STATUS
        JMP     JOB
```

```
JOB:    (* Reply the Job Number *)
        LD      R_JOBNUMBER
        ST      S_JOBNUMBER
END:
END_PROGRAM
```

3.5. Sample PLC Program with Sucosoft V2.11

The fieldbus support disk provided with Designer contains also a sample PLC program created with Sucosoft V2.11.
Klockner Moeller developed a special function block called MI4netK you could use in order to simplify the communication setup with UniOP.
As previously shown for the old version of Sucosoft in “Topology Configurator” use the Remote Expansion function to add the Suconet K slave. The “Topology Configurator” dialog box will appear again as shown in Figure 10.
The panel Configuration window will appear as shown in Figure 12 where you should properly set the buffer size.

The PLC program at this point is very simple. You should simply add the call to an instance of MI4netK function block.
The declaration editor in the free form will be similar to the following example:

```
VAR_GLOBAL
Read_Data1      AT %RDB1.1.0.0  : ARRAY [1..20] OF BYTE;
Send_Data1      AT %SDB1.1.0.0  : ARRAY [1..20] OF BYTE;
Diagnose1                        : Byte;

DiagISB          AT %ISB1.1.0.0  : Byte;
outp                             : Byte;
outdia                           : Byte;
END_VAR

VAR
SuconetK   :MI4netK;
END_VAR
```

Please note that the buffer size you choose in the “Topology Configurator” must match the array size you used in the declaration section.

The PLC program should be similar to following example:

```
CAL SuconetK(
RDB_Start :=Read_Data1,
SDB_Start :=Send_Data1
|
 Diagnose1:=Status
 )
 LD DiagISB
 ST outp
 LD Diagnose1
```

```
ST outdia
```

Figure 5 shows the POU window with the simple program and the declaration section.

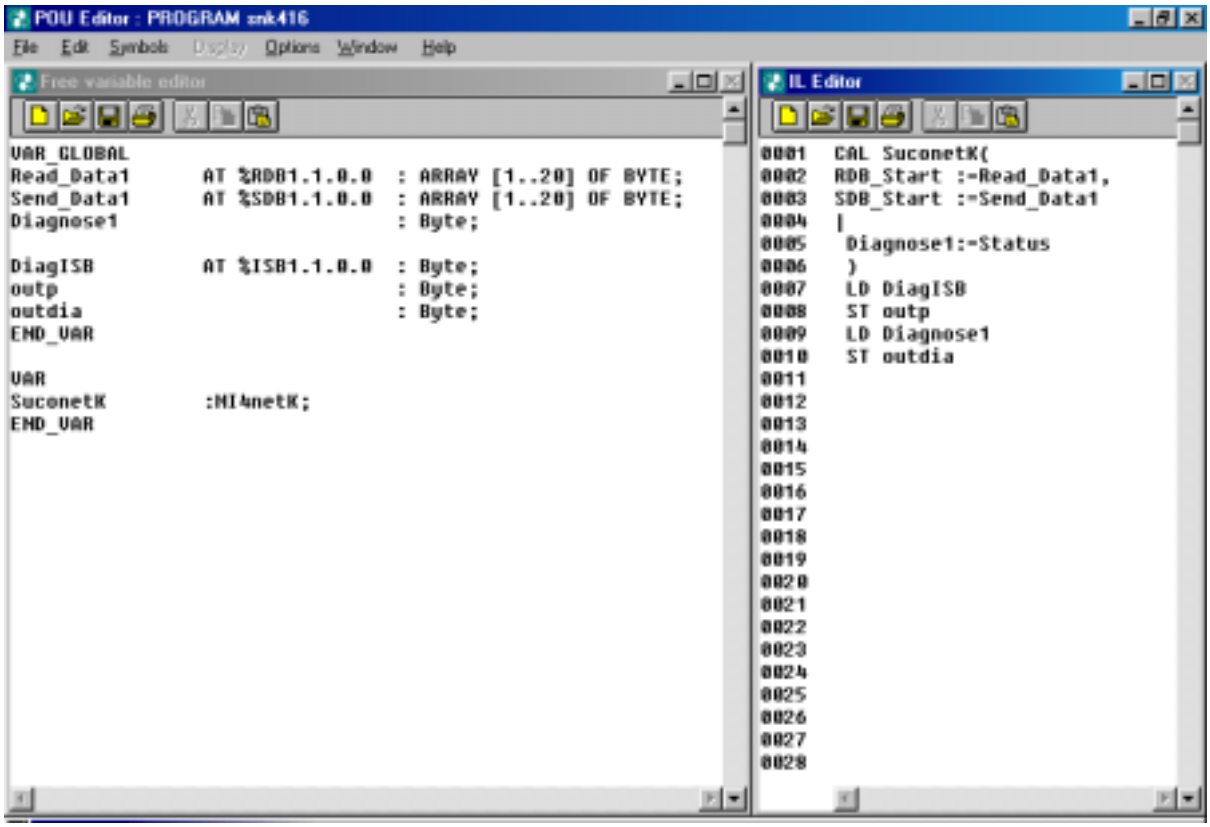

Figure 13- The POU Editor window

In case you have more than one panel connected to Suconet-K you should use one instance of MI4netK function for each panel. The variable declaration section must match the new topology configuration.

3.6. The Designer

Data in PLC program configuration file must match the data specified in the Designer project file, otherwise the communication will not be established between UniOP and PLC. The data that must match are:

- UniOP node
- Input and Output data length must be exactly the same as in PLC configuration

Important note:
When using a PS416 CPU, refer to the conversion table shown below to map between PS4 addresses and the PS416 data structures:

MB0	PANEL_AREA[0]
MB1	PANEL_AREA[1]
…..	…..

4. PS4-100 (Suconet K1)

This chapter describes how to connect a UniOP to a Suconet K1 network with a PS4-100 as the master.

4.1. Packing List

The sample program includes the following files:

SUC.DAT	device configuration file
PANELW.P3 PANELW.Q3 PANELW.Z3 PANELW.QLS	sample PLC program
SUCONK11.PRJ	sample project file

The sample program is a simple ready-to-run application, which includes also the project file for the panel. No additional PLC programming is required.

4.2. Installation

The original PS4 programming software does not includes UniOP in the list of known Suconet K1 peripherals and, therefore, some modifications are necessary to support it. Assuming that original K-M software is already installed, the device configuration file SUC.DAT must be edited as shown in the example below. An example of configuration file is included in the sample disk.

```
238     TYP-238 0       0       0       0       1130 152 1131 1132 152 1133 152;
;239    TYP-239 0       0       0       0       1130 152 1131 1132 152 1133 152;
239     PANEL   0       0       0       0       1130 152 1131 1132 152 1133 152;
240     TYP-??? 0       0       0       0       1130 152 1131 1132 152 1133 152 152;
```

4.3. PLC Configuration

No special configuration is required in the PLC to recognize the UniOP panel as a Suconet K slave.

The address map of the panel as a Suconet K1 slave is the following:

```
IB1.0       Panel 1st Inp Byte (JOB)
IB1.8       Panel 2nd Inp Byte (COMMAND)
IA1.0       Panel 3rd Inp Byte (ADDRESS)
IA1.1       Panel 4th Inp Byte (DATA LOW)
IA1.2       Panel 4th Inp Byte (DATA HIGH)

QB1.0       Panel 1st Out Byte (JOB)
QB1.8       Panel 2nd Out Byte (REPLY)
MWxx        Panel 5th and 6th Byte (DATA WORD)
```

MWxx is a PLC word choosen by programmer as preset for counter Cx.0 where x = 1, 2 or 3 is the slave number.

4.4. Sample PLC Program

This PLC program is a ready-to-run program supporting Suconet K1 communication. It can easily be included in any user application to add UniOP communication capability.

```
  " Stampa del file d:panelw.q3   del:  1. 8.97"
00000 L000      "Main program..."
001             L MB1.0                 [02=MB1.0] Counter...
002             ADD KB 1
003             = MB1.0                 [02=MB1.0] Counter...
004             L MB1.8                 [03=MB1.8]
005             ADD KB 2
006             = MB1.8                 [03=MB1.8]
007             L IB1.0                 Panel 1o Inp Byte (SYN)
008             SUB QB1.0               Panel 1o Out Byte (SYN)
009             BZ  END
010
00001           "Operator Panel Interface"
001             L IB1.8                 Panel 2o Inp Byte (CMD)
002             SUB KB 16
003             BZ  READ00
004             L IB1.8                 Panel 2o Inp Byte (CMD)
005             SUB KB 144
006             BZ  WRITE00
007             L IB1.8                 Panel 2o Inp Byte (CMD)
008             SUB KB 64
009             BZ  IDLE
010             L KB 128
011             = QB1.8                 Panel 2o Out Byte (REPLY)
012             JP  END
00002 READ00    "Read command (10H)
001             L IA1.0                 Panel 3o Inp Byte (ADDR)
002             SUB KB 0
003             BNZ  READ01
004             L MW0.0
005             = MW10.0
006             JP  END_OKR
00003 READ01    "Read command (10H)
001             L IA1.0                 Panel 3o Inp Byte (ADDR)
002             SUB KB 2
003             BNZ  READ02
004             L MW1.0
005             = MW10.0
006             JP  END_OKR
00004 READ02    "
001             L IA1.0                 Panel 3o Inp Byte (ADDR)
002             SUB KB 4
003             BNZ  READ03
004             L MW2.0
005             = MW10.0
006             JP  END_OKR
00005 READ03    "
001             L IA1.0                 Panel 3o Inp Byte (ADDR)
002             SUB KB 6
003             BNZ  READ04
004             L MW3.0
005             = MW10.0
006             JP  END_OKR
00006 READ04    "
001             L IA1.0                 Panel 3o Inp Byte (ADDR)
002             SUB KB 8
003             BNZ  READ05
004             L MW4.0
005             = MW10.0
006             JP  END_OKR
00007 READ05    "
```

```
001              L IA1.0                 Panel 3o Inp Byte (ADDR)
002              SUB KB 10
003              BNZ  END_ERR
004              L MW5.0
005              = MW10.0
006              JP  END_OKR
007
00008 END_OKR    "End Ok read"
001              L KB 1
002              = QB1.8                 Panel 2o Out Byte (REPLY)
003              L K  1
004              = M 11.0
005              JP  COUNT
006
00009 END_ERR    "read address error
001              L KB 129
002              = QB1.8                 Panel 2o Out Byte (REPLY)
003              JP  END
004
00010 WRITE00    "Write command (90H)
001              L IA1.0                 Panel 3o Inp Byte (ADDR)
002              SUB KB 0
003              BNZ  WRITE01
004              L IA1.1                 Panel 4o Inp Byte (DATA)
005              = MB0.0                 [00=MB0.0]
006              L IA1.2
007              = MB0.8                 [01=MB0.8]
008              JP  END_OKW
00011 WRITE01    "Write command (90H)
001              L IA1.0                 Panel 3o Inp Byte (ADDR)
002              SUB KB 2
003              BNZ  WRITE02
004              L IA1.1                 Panel 4o Inp Byte (DATA)
005              = MB1.0                 [02=MB1.0] Counter...
006              L IA1.2
007              = MB1.8                 [03=MB1.8]
008              JP  END_OKW
00012 WRITE02    "
001              L IA1.0                 Panel 3o Inp Byte (ADDR)
002              SUB KB 4
003              BNZ  WRITE03
004              L IA1.1                 Panel 4o Inp Byte (DATA)
005              = MB2.0                 [04=MB2.0]
006              L IA1.2
007              = MB2.8                 [05=MB2.8]
008              JP  END_OKW
00013 WRITE03    "
001              L IA1.0                 Panel 3o Inp Byte (ADDR)
002              SUB KB 6
003              BNZ  WRITE04
004              L IA1.1                 Panel 4o Inp Byte (DATA)
005              = MB3.0
006              L IA1.2
007              = MB3.8
008              JP  END_OKW
00014 WRITE04    "
001              L IA1.0                 Panel 3o Inp Byte (ADDR)
002              SUB KB 8
003              BNZ  WRITE05
004              L IA1.1                 Panel 4o Inp Byte (DATA)
005              = MB4.0
006              L IA1.2
007              = MB4.8
008              JP  END_OKW
00015 WRITE05    "
001              L IA1.0                 Panel 3o Inp Byte (ADDR)
002              SUB KB 10
003              BNZ  END_ERW
004              L IA1.1                 Panel 4o Inp Byte (DATA)
005              = MB5.0
006              L IA1.2
007              = MB5.8
008              JP  END_OKW
```

```
009
00016 END_OKW    "End Ok write
001              L KB 2
002              = QB1.8                 Panel 2o Out Byte (REPLY)
003              JP  END
004
00017 END_ERW    "write address error
001              L KB 130
002              = QB1.8                 Panel 2o Out Byte (REPLY)
003              JP  END
004
00018 IDLE       "idle rensponse
001              L KB 4
002              = QB1.8                 Panel 2o Out Byte (REPLY)
003              JP  END
004
00019 END        "End communication program
001              L K  0
002              = M 11.0
003
00020 COUNT      "
001              L IB1.0                 Panel 1o Inp Byte (SYN)
002              = QB1.0                 Panel 1o Out Byte (SYN)
003
004
005              C1.0
006            [ ] S:    M 11.0
007            [w] I:    MW10.0
008            [w] Q:
009
00021 EXIT       "
001              EP
```

The project file is a simple 1-page application with just a few data fields.
Note that only the first word of the Keyboard Updated Area of the RDA is enabled. The rest of the RDA is disabled.

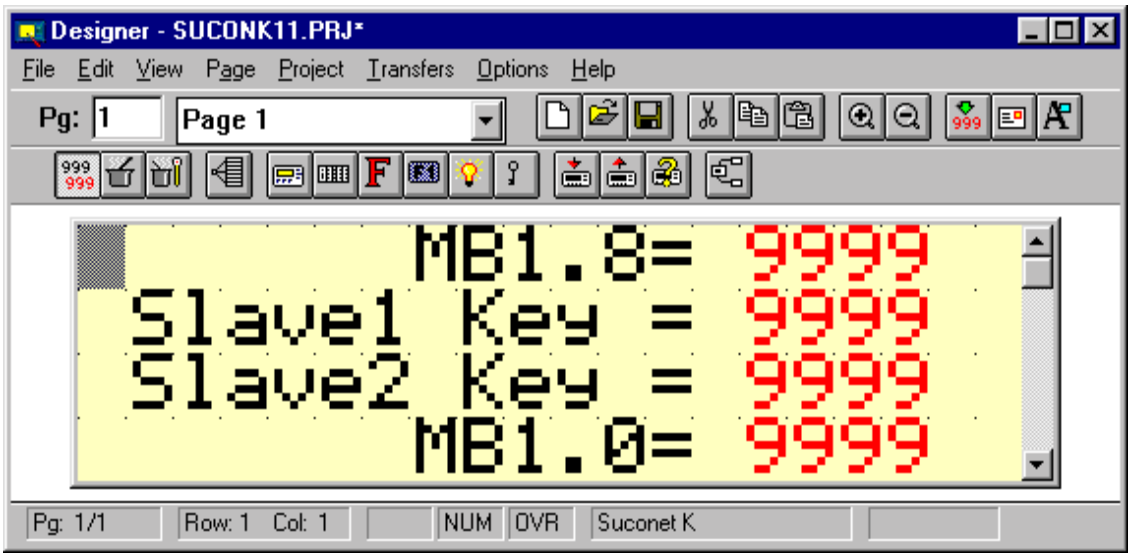

Figure 14 - Sample project page

4.5. The Designer

The data included in the project file by the Designer must match the data specified in the PLC program, otherwise the communication will not be properly established. The data that must match are:

- the PLC model (choose PS4-111-DR1 from the list in Controller Setup dialog box)
- the node number of the panel in the network
- interpretation of the addresses in the project file and in the PLC program must be consistent.

Even if PS4-100 is selected, the Controller Setup dialog box will present the information on the size of communication buffers. This information is not used in Suconet K1 where all buffers have fixed size.

5. PS 316 (Suconet K1)

This chapter describes how to connect a UniOP to a Suconet K1 network with a PS 316 as the master.

5.1. Packing List

The sample program includes the following files:

SUC.DAT	device configuration file
TESTSK.P6W TESTSK.Q6W TESTSK.Z6W TESTSK.QLS	sample PLC program

5.2. Installation

The original PS 316 programming software does not includes UniOP in the list of known Suconet K1 peripherals and, therefore, some modifications are necessary to support it. Assuming that original K-M software is already installed, the device configuration file SUC.DAT must be edited as shown in the example below. An example of configuration file is included in the sample disk.

```
238     TYP-238 0       0       0       0       1130 152 1131 1132 152 1133 152;
;239    TYP-239 0       0       0       0       1130 152 1131 1132 152 1133 152;
239     PANEL   0       0       0       0       1130 152 1131 1132 152 1133 152;
240     TYP-??? 0       0       0       0       1130 152 1131 1132 152 1133 152 152;
```

5.3. PLC Configuration

A special configuration is required in the PLC to recognize the UniOP panel as a Suconet K1 slave. The sample program works with the EBE295.1 SBI board that needs to be initialized by the program. In our sample program the board is configured to works in slot #4 with 1 slave.

In this condition the address map of the panel as a Suconet K1 slave is the following:

```
PP192B0.0  Panel 1st Inp Byte (JOB)
PP192B0.8  Panel 2nd Inp Byte (COMMAND)
PP192B1.0  Panel 3rd Inp Byte (ADDRESS LOW)
PP192B2.0  Panel 4th Inp Byte (ADDRESS HIGH)
PP192B3.0  Panel 5th Inp Byte (DATA LENGTH)
PP192B4.0  Panel 6th Inp Byte (DATA 0)
PP192B5.0  Panel 7th Inp Byte (DATA 1)

PP192B8.0  Panel 1st Out Byte (JOB)
PP192B8.8  Panel 2nd Out Byte (REPLY)
PP192B9.0  Panel 3rd Out Byte (DATA 0)
PP192B10.0 Panel 4th Out Byte (DATA 1)
PP192B11.0 Panel 5th Out Byte (DATA 2)
PP192B11.8 Panel 6th Out Byte (DATA 3)
```

Selecting a different slot for SBI board or managing more slaves means to access different addresses: please refer to K-M documentation in those cases.

The SBI board needs also a special management for synchronization with the program. Before reading the new data the program has to wait the SBI board to prepare them. When the execution time of the PLC program is slower than the SBI polling cycle time it is necessary to repeat the synchronization at the end of the program.
The sample program uses three words of Merkers as temporary memory during data transfer operations, MW2170, MW2171, MW2172. These words cannot be accessed by the UniOP project.

5.4. Sample PLC Program

This PLC program is a ready-to-run program supporting Suconet K1 communication. It can easily be included in any user application to add UniOP communication capability.

```
  " Printout of file d:testsk.q6w   Printed:  6. 3.98"
00000           "Sample program for UniOP interface
001
002             "Important note:
003             "this program uses memory locations
004             "MW2170, MW2171, MW2172 for indirect memory addressing
005             "these locations are not accessible by UniOP and an
006             "error 11 will be generated if accessed.
007
008             "EBE295.1 board in position 4; 1 slave enabled
009             #DEFINE "r0s4 sbi0:1:1"
010
011
00001 INIT      "activate operation with the telegram consistency
001             L IFC                   sbi-stb1
002             JCN  CON
003
004             L KHB C0
005             = PP192B126.0           sbi-stb1
006
00002 CON       "wait until the SBI has terminated
001             L PP192B126.0           sbi-stb1
002             BB 7 ENDSUCO
003
004
00003           "check job number
001             L PP192B0.0             received job number
002             = MB2172.0
003             L PP192B8.0             reply job number
004             CP MB2172.0
005             "if same job ignore it
006             BE  ENABL
007
008             "analyse request
009             L PP192B0.8             request
010             CP KHB 40
011             BNE  PENDING
012
013             "no request, return idle
014             L KB 4
015             = PP192B8.8             reply code
016             JP  JOB
017
018
00004 PENDING   "new request from slave
001             "prepare address
002             L PP192B1.0             address low
003             = MB2172.0
004             L PP192B2.0             address high
005             = MB2172.8
006
```

```
007             "check the type of memory accessed
008             L MW2172.0
009             CP KW 250
010             BLT  LOWMEM
011
012             "if highmem, prepare base address and correct offset
013             L& MB125.0
014             = ICPY0 SADR
015             = ICPY1 DADR
016
017             L MW2172.0
018             SUB KW 250
019             = MW2172.0
020             JP  OPERAT
021
00005 LOWMEM    "operation in low memory, prepare base address
001             L& MB0.0
002             = ICPY0 SADR
003             = ICPY1 DADR
004
00006 OPERAT    "check the operation
001             L PP192B0.8             request
002             CP KHB 90
003             BE  WRITE
004
005             L PP192B0.8             request
006             CP KHB 10
007             BE  READ
008
009             "the request code is unknown
010             L KHB 80
011             = PP192B8.8             reply code
012             JP  JOB
013
014
00007 READ      "read operation, move bytes to temporary memory
001             ICPY0
002           [ ] MOD:  K  1
003           [&] SADR:
004           [w] SOFF: MW2172.0
005           [&] DADR: & MB2170.0
006           [w] DOFF: KW 0
007           [b] NO:   PP192B3.0       length
008           [b] ERR:
009
010             "check if errors
011             L ICPY0 ERR
012             BZ  READOK
013
014             "errors reading
015             L KHB 81
016             = PP192B8.8             reply code
017             JP  JOB
018
019
00008 READOK    "transfer data to telegram mailbox (max. 4 bytes)
001             L MB2170.0
002             = PP192B9.0
003             L MB2170.8
004             = PP192B10.0
005             L MB2171.0
006             = PP192B11.0
007             L MB2171.8
008             = PP192B11.8
009
010             L KHB 1
011             = PP192B8.8             reply code
012             JP  JOB
013
014
00009 WRITE     "write operation
001             "read data from telegram (max. 2 bytes)
002             L PP192B4.0             data0
```

```
003             = MB2170.0
004             L PP192B5.0
005             = MB2170.8
006
007             "transfer to memory
008             ICPY1
009           [ ] MOD:  K  1
010           [&] SADR: & MB2170.0
011           [w] SOFF: KW 0
012           [&] DADR:
013           [w] DOFF: MW2172.0
014           [b] NO:   PP192B3.0       length
015           [b] ERR:
016
017             "check if errors
018             L ICPY1 ERR
019             BZ  WRITEOK
020
021             "errors in write
022             L KHB 82
023             = PP192B8.8             reply code
024             JP  JOB
025
026
00010 WRITEOK   "
001             L KHB 2
002             = PP192B8.8             reply code
003             JP  JOB
004
00011 JOB       "copy job number
001             L PP192B0.0             received job number
002             = PP192B8.0             reply job number
003
00012 ENABL     "enable sbi again
001             L KHB C0
002             = PP192B126.0           sbi-stb1
003
00013 ENDSUCO   "user program
001             "…………………………………
002
003              EP
004
```

5.5. The Designer

The data included in the project file by the Designer must match the data specified in the PLC program, otherwise the communication will not be properly established. The data that must match are:

- the PLC model (choose PS316 from the list in Controller Setup dialog box)
- the node number of the panel in the network
- interpretation of the addresses in the project file and in the PLC program must be consistent.

Even if PS316 is selected, the Controller Setup dialog box will present the information on the size of communication buffers. This information is not used in Suconet K1 where all buffers have fixed size.

Appendix 1 - Communication Error Codes

UniOP panels provide some diagnostic information on the status of the communication in the System Menu; these codes are especially useful in all cases in which proper communication cannot be established between UniOP and the PLC. The Suconet K communication driver is using several special codes which are described below:

00 **No errors**
Normal communication.

01 **Request not accepted by PLC**
Mailbox command not recognized by the PLC server function.

04 **Mailbox not acknowledged - different size**
UniOP has found different length of PLC reply mailbox.

05 **Time-out error on mailbox**
The communication is active, but mailbox refresh from PLC is missing. Probably the PLC program cannot process the data for the UniOP panel.

06 **Response error - incorrect answer from PLC**
UniOP found unexpected answer in PLC reply mailbox.

07 **Communication error - PLC not connected**
Communication timeout. The PLC is not connected or switched off. Normally the PLC polls the slave.

10 **Master PLC in STOP**
The PLC status signals that the PLC is in STOP and UniOP requests cannot be processed.

11 **Read address not defined on PLC**
The requested variable is not defined on PLC. The UniOP project is not correct

12 **Write address not defined on PLC**
UniOP project error requested variable is not defined on PLC.

The following error code are associated to hardware malfunctions in the communication between the UniOP CPU and the TCM02 communication coprocessor.

20 **Time-out on TCM02 communication coprocessor**
The TCM02 coprocessor is not responding. Missing or bad module.

21 **TCM02 communication coprocessor firmware is invalid**
After start-up UniOP checks the coprocessor firmware. If an incorrect state is found, this error is raised.

22 **TCM02 communication coprocessor bootstrap error - retry**
Downloaded firmware execution by coprocessor is not correct. The coprocessor must signal program start to UniOP CPU.

23 **TCM02 communication coprocessor answer error - restarted**
An error occurred during normal communication between the UniOP CPU and the TCM02 coprocessor.

24 **TCM02 communication coprocessor hardware error**
After start-up UniOP reads the hardware signature and release number from the TCM02 module and check compatibility with current version of the communication protocol. If find wrong type or missing TCM02 module, this error is raised.

Appendix 2 - Technical Data and Connection Information

The main technical information on the UniOP Suconet K/K1 Slave interface using TCM02 is shown in the table below:

Protocols	Suconet K / Suconet K1
Baudrate	187.5/375 Kb
Buffer size	Up to 78 bytes input, 78 bytes output
Slave address	Software configurable
Optical insulation	Yes
Suconet Connector	Standard, 9 pin female sub-D type

UniOP is equipped with a standard Suconet connector. Suconet communication is always performed through the Aux Port of the UniOP. A simple point to point connection can be estabilished with the following cables:

PLC Model	**Cable**
PS4-140, PS4-150, PS4-200	CA105
PS416	CA137
PS4-100	CA105
PS316	CA137