# 3DLevelScanner II

## Different Ways of Connecting to 3DLevelScanner II

A.P.M Automation Solutions LTD.
www.apm-solutions.com

Version 3.0

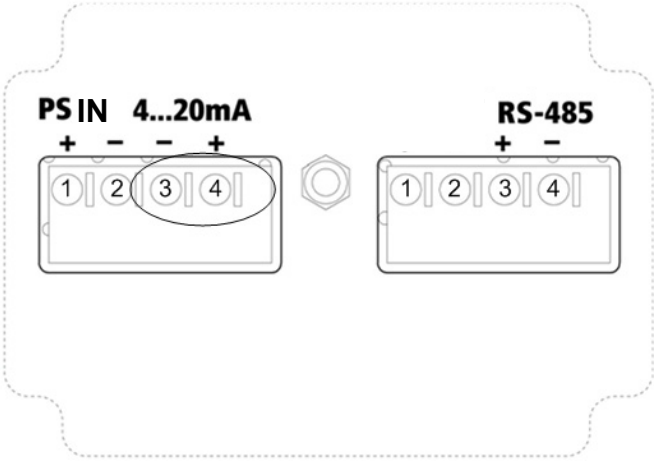
## Table of Contents

**Abstract**

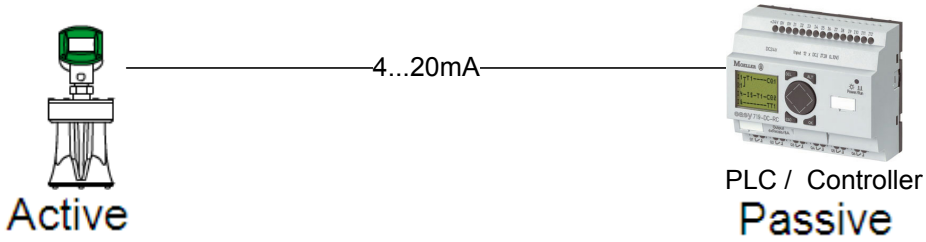
This document describes different methods for connecting the 3DLevelScanner II (types S/M/MV and MVL) in variety of communication ways.

Note: All methods of connections are for recommendations only and the user may use more or other solutions to get connected to the 3DLevelScanner II.

1. **Connection via 4…20mA.**
   a. The 4…20mA current output is available through ports 3 and 4 of the left green connector (as shown in the drawing below). Ports 3 and 4 are the negative and positive poles, respectively.




   b. The 4…20mA line goes directly from the scanner mounted on the vessel to the PLC/device (as shown below).

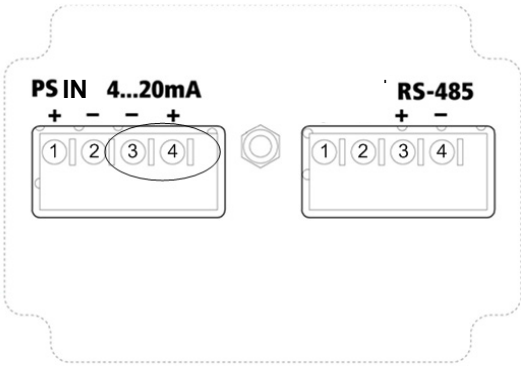



Note:

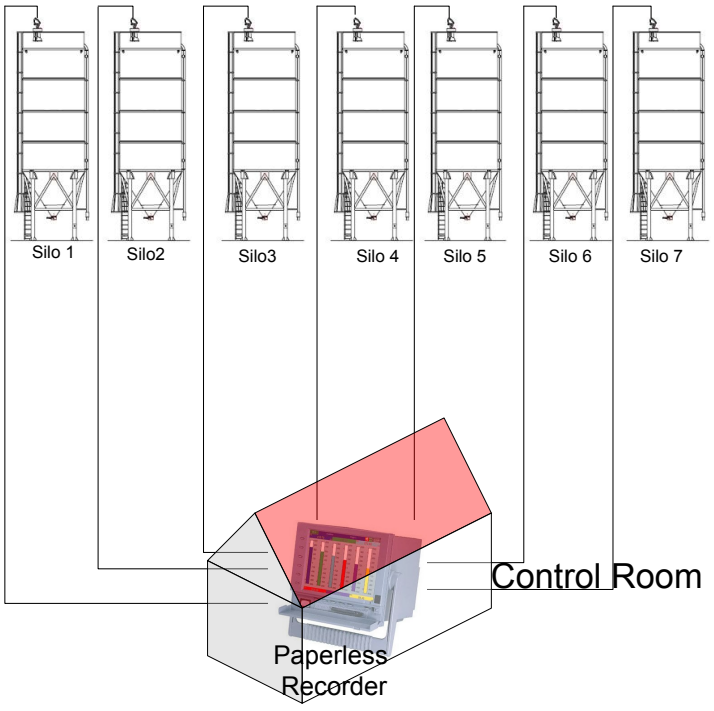
- <u>This type of connection is active and not passive,</u> hence the 3DLevelScanner II is the active module and the PLC should be the passive module.
- This connection does not require the APM 3DLevel Manager Software tool.

## 2. Connection via Paperless Recorder (based on 4…20mA)

a. Each mounted scanner is connected directly to the recorder via ports 3 and 4 of the left green connector as shown below. Ports 3 and 4 are the negative and positive poles, respectively.




b. The paperless recorder gathers all the 4…20mA inputs and displays the measurement on its screen (an example of connection of 7 scanners mounted on 7 silos is shown below):

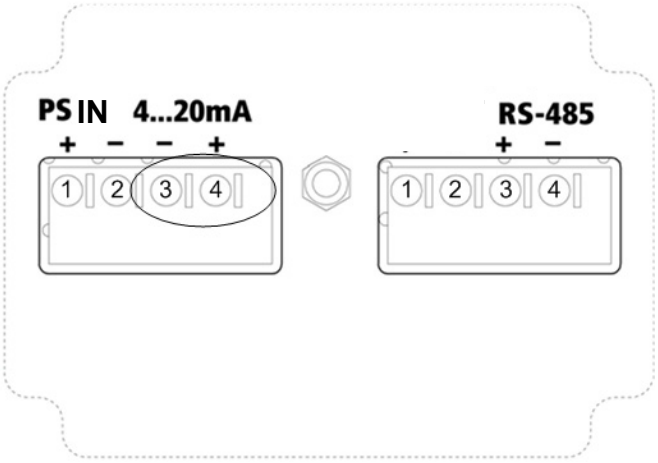



Note:

- This connection does not require the APM 3DLevel Manager Software tool.
- This type of connection is active and not passive, hence the 3DLevelScanner II is the active module and the PLC should be the passive module.

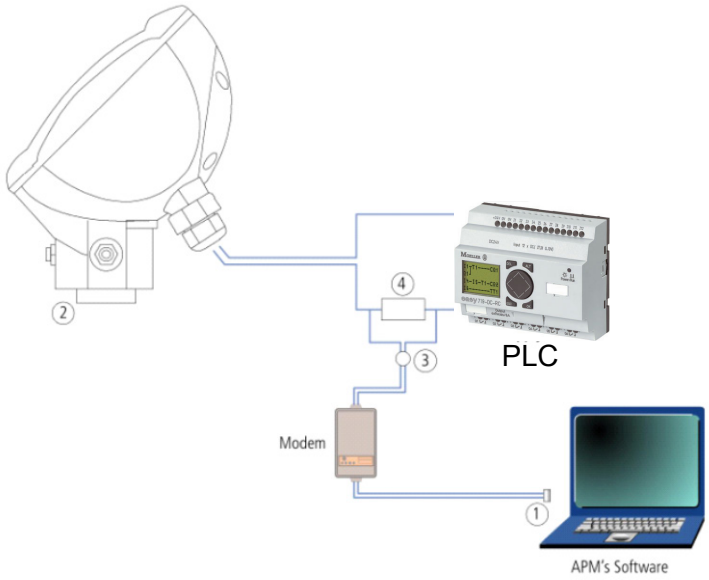
## 3. Connection via HART

a. The HART connection is via ports 3,4 of the left green connector as shown below.




Note:

1. The HART connection has no polarity (it is allowed to switch between ports 3 and 4).
2. The diagram below shows a combined connection of HART and 4…20mA.



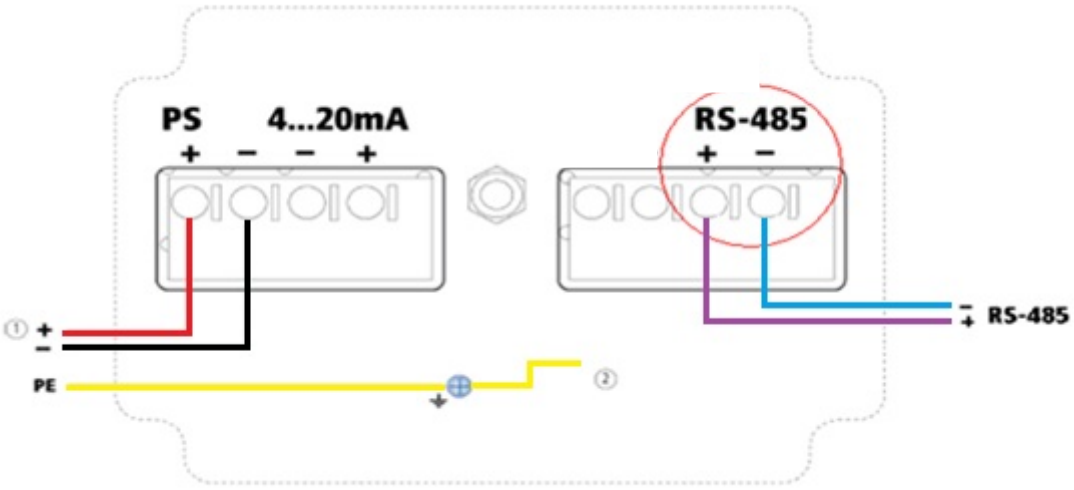

1. RS-232/USB connection
2. 3DLevelScanner II
3. HART Adapter cable
4. Resistor of 250 ohm

## 4. Multidrop (“daisy chain”) Connection

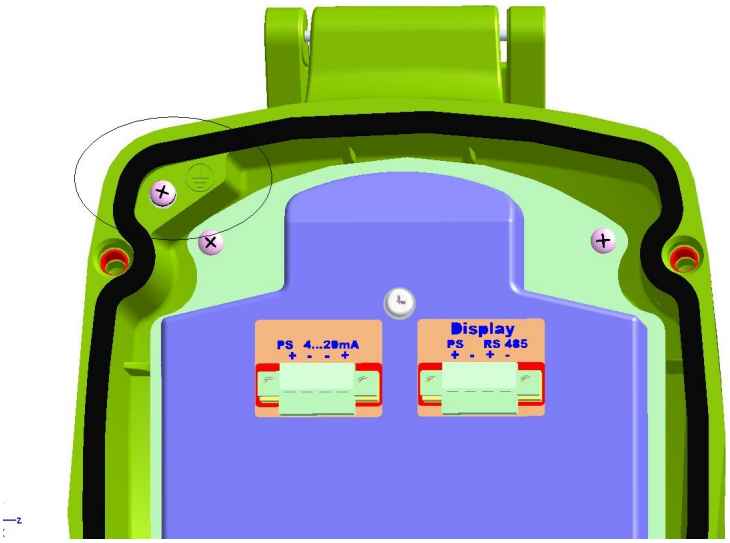
a. Physical Connections:

1. All 3DLevelScanners should be connected in parallel mode in the RS485 ports. Hence, all the ‘+’ positive ports of the RS485 should commonly be connected and all the ‘-’ negative ports of the RS485 should commonly be connected.

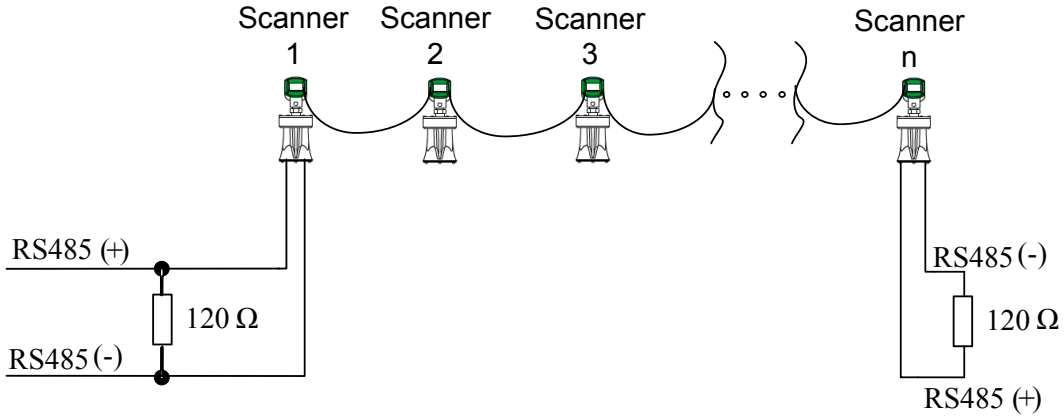
Note: The multidrop (“daisy chain”) connection is possible via RS485 interface only.




2. The RS485 cable should be shielded and twisted pair. The shield of the cable should be connected to the ground screw on the back of the head of the 3DLevelScanner (as below):



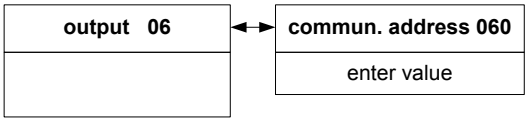
b. At each end of the connection, a 120 Ohm termination resistor should be connected between the '+' and the '-' RS485 ports.




n= 1…63

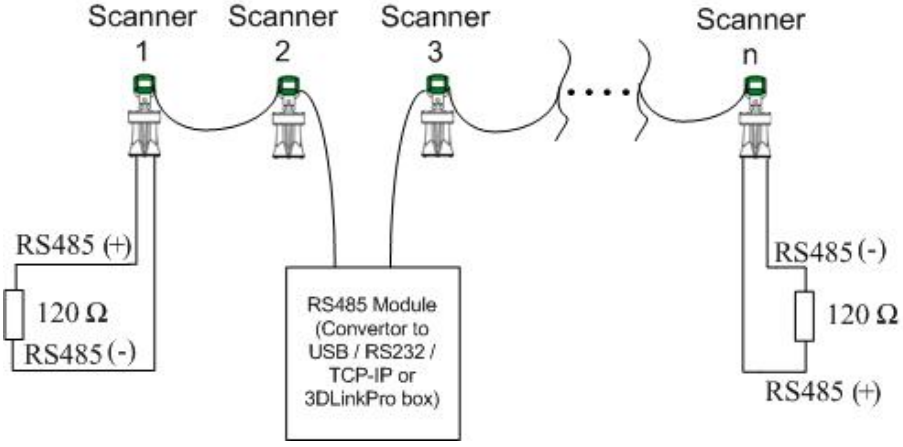
c. Each Scanner should be configured with a different polling address. The configuration of the polling address is set in the LCD panel (as shown below).

1. In the main menu of the LCD click on 'E' and scroll down to the Output submenu.
2. Click on 'E'.
3. Configure the polling address number – each scanner should have a different polling address (legal values 00…63).




**IMPORTANT NOTE:** When connecting a 3DLinkPro, one of the scanners must have a polling address 00 (does not matter which 3DLevelScanner in the bus)

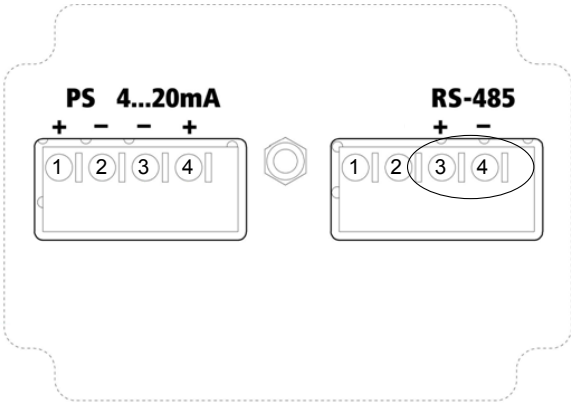
Another example of a multiple / daisy chain connection:



## 5. Connection via GSM Cellular Modem

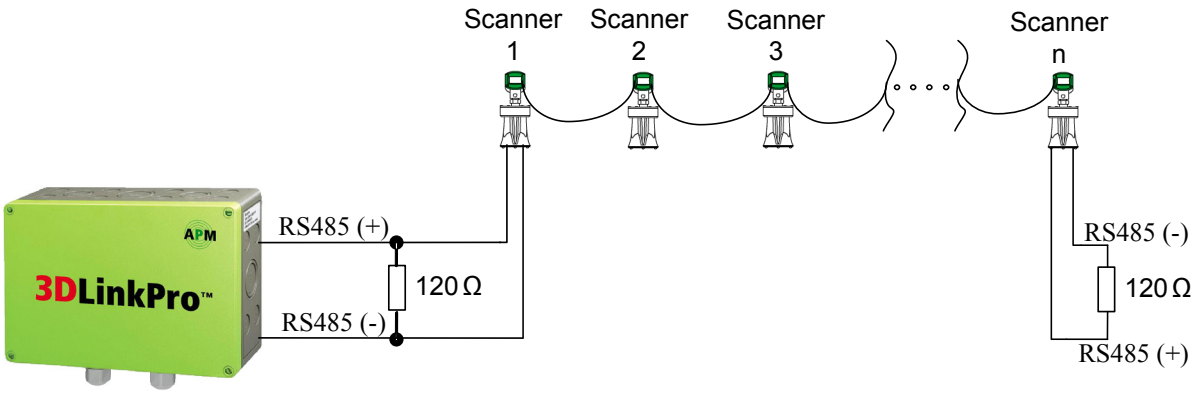
a. In this method a GSM modem is connected to a PC and the 3DLinkPro is connected to the scanner (as shown below):


b. The connection between the scanner and the 3DLinkPRo is via the RS485 ports in the electronic card, using ports 3 and 4 of the right connector (as shown below):




c. The RS485 layer requires a parallel connection (all '+' connected together and all '-' connected together)

Note (1): The RS485 allows a multi drop connection of up to 64 scanners (as shown below):




Note (2): There should be termination resistors of 120ohms between the RS485 lines inside the 3DLinkPro and at the end of the line in the last 3DLevelScanner (in the right scanner at the image above); hence, placed between the positive (+) and negative (-) lines of the RS485 (in the 3DLinkPro ports 11 and 12).

## 6. Connection via GPRS

a. The physical connection of the scanners and the 3DLinkPro is done exactly the same as in GSM method (see clause 4)

b. On the PC side, the communication to the scanner will be done via a local internet network.

c. Pre steps – Local Network Preparations:

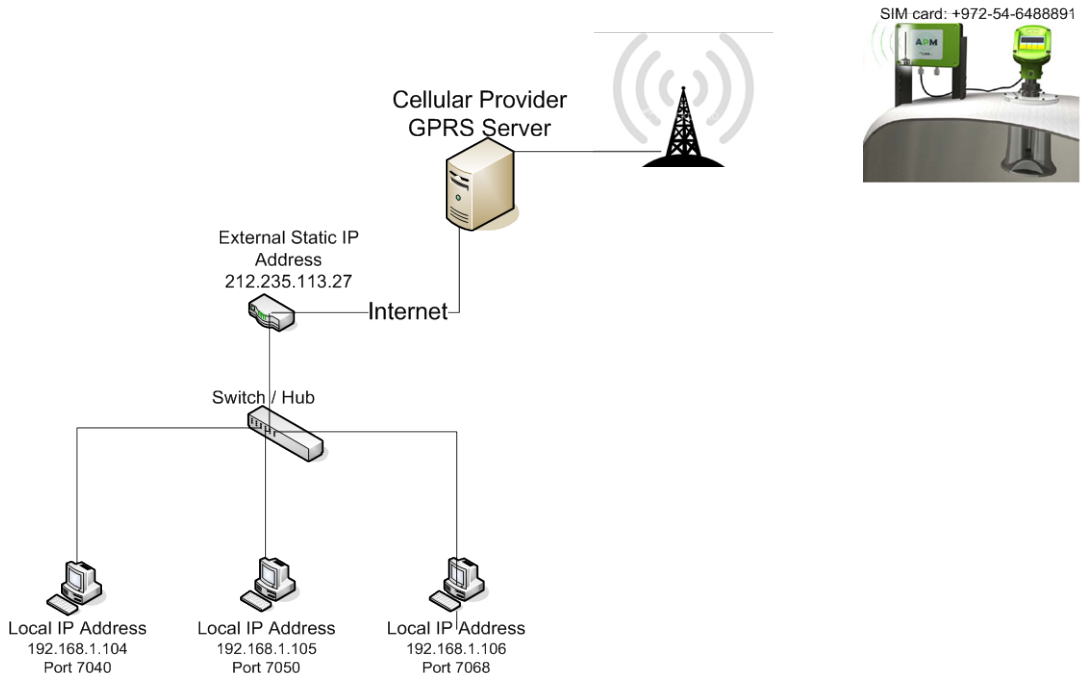
   i. In order to get connected from a local PC in a shared network, it is required to do the following configurations:

      1. Acquire the external IP of the shared network (e.g. browse to www.whatismyip.com)
      2. Find the IP address of the local PC running the APM 3DLevel Manager
      3. Set a port number which will be used for connection on the local PC.
      4. Configure the network's gateway for port forwarding ("*NAT*") so that any connection to the port chosen on clause 3 will be forwarded to the IP address of the local machine (from clause 2). It is advised that this task is done by the network administrator.
      5. Contact the cellular provider (the provider of the SIM card in the 3DLinkPro) and get the APN (Access Point Name) word for GPRS connection.

d. Below is a configuration example. In this example we have:

   1.External IP is 212.235.113.27

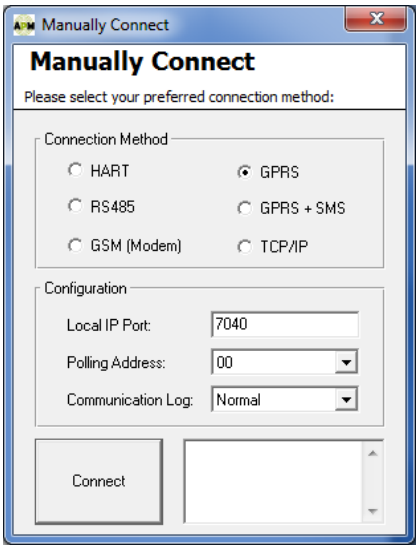
2. There are 3 local machines with IP addresses:
    a. 192.168.1.104
    b. 192.168.1.105
    c. 192.168.1.106
3. The gateway is configured so that port 7040 is forwarded to IP address 192.168.1.104; port 7050 is forwarded to IP address 192.168.1.105; port 7068 is forwarded to IP address 192.168.1.106.
4. The APN word of the cellular provider is "internet.t-mobile".
5. The number of the SIM card in the 3DLinkPro is: +972-54-6488891




Network Connections Scheme

6. Operation of the APM 3DLevel Manager (PC side) in this example
    a. Run the APM 3DLevel Manager and go to Device→ Manually Connect menu.
    b. Select in the Manually Connect window the GPRS radio button (see image below).
    c. In order to connect to PC with the local IP address 192.168.1.104, set the "Local IP Port" field to 7040.
    d. Set the polling address of the scanner you wish to get connected to.

e. Then press the *Connect* button.

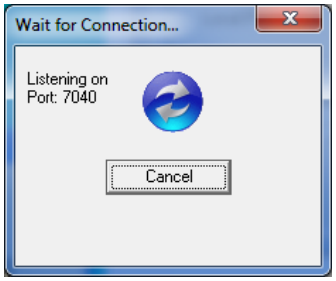



Manually Connect Window

**Note**: Since this method involves TCP/IP communication, a Windows message might appear (see image below), in that case, select the unblock option.


Window Security Message

f. The 'Wait for Connection' window will pop up:




3DLevel Manager ready for connection

g. The local PC is now ready to be connected via GPRS.

h. Send the following SMS message to the number of SIM card in the 3DLinkPro:

*CALLAPM,212.235.27.113,7040,internet.t-mobile,*

**Important (1)**: the SMS message should be in the exact same format as above (case sensitive, no spaces and comma at the end).

**(2):** There should be a termination resistor of 120ohms between the RS485 lines; hence, placed between the positive (+) and negative (-) lines of the RS485 in both sides of the RS-485 bus.

**(3):** For a remote hardware modem reset (the modem in the 3DLinkPro) send the text message (case sensitive):
*RESET MODEM*

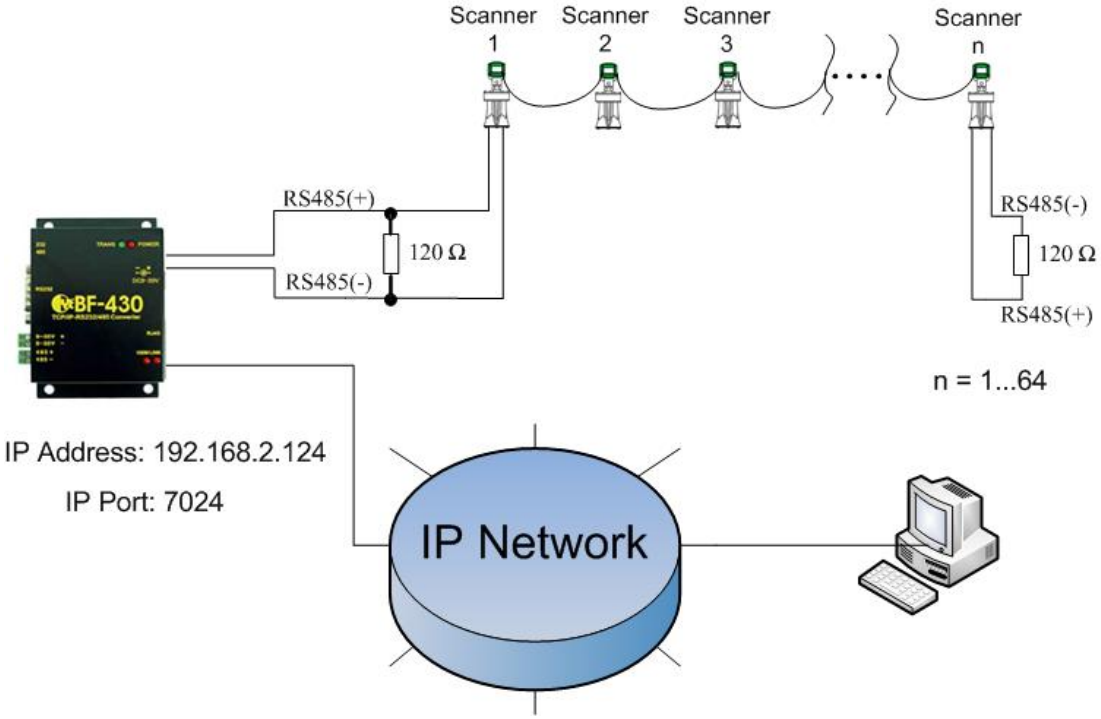
## 7. Connection via TCP/IP (using TCP/IP Converter)

b. The Communication between the scanner and the converter is via RS485.

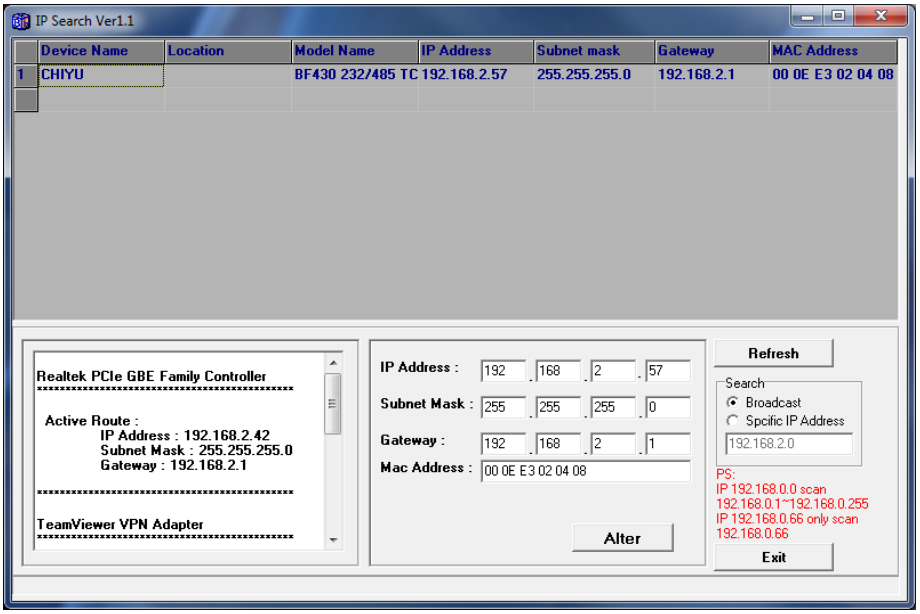
c. The Communication between the Converter and the PC / Laptop is via the TCP/IP networking.

d. **Important:** A 120 ohm resistor must be connected in parallel to the RS485 line (as shown below).

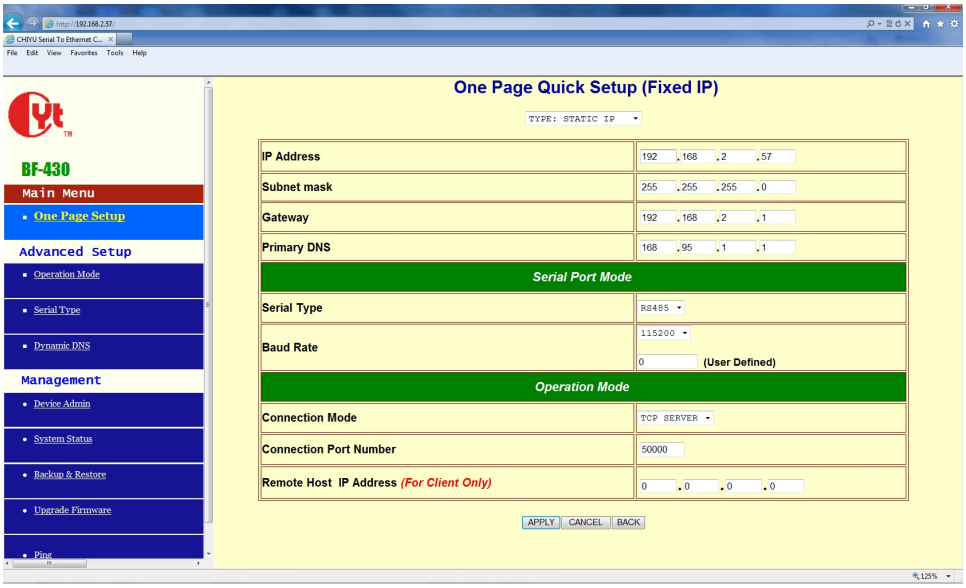
**Note:** APM is recommending the CHIYU BF-430 as the RS485 to TCP/IP converter (as shown below)




e. For initial setup, the converter must be found in the network. Use the "*IP Search Software Tool*" (comes in the CD supplied with the converter) as shown below. After running the software click on the Refresh button to locate the convertor.




f. Once the convertor is found, double click on the line where the convertor was found, will run an html (as shown below) with all the parameters to configure:




g. Operation of the APM 3DLevel Manager (PC side)

i. Run the APM 3DLevel Manager and go to Device→ Manually Connect menu.

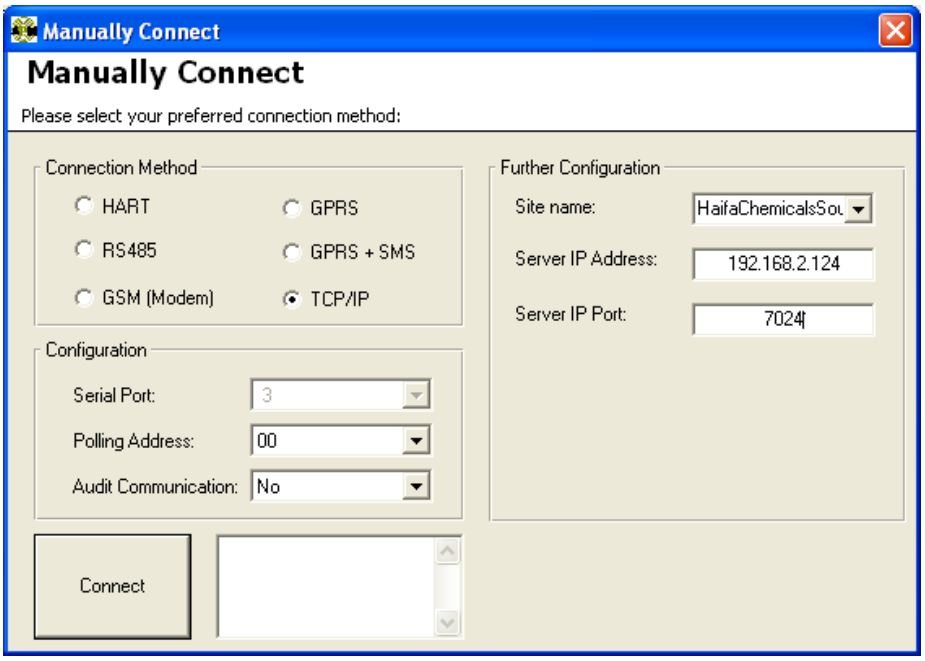
ii. Select in the Manually Connect window the TCP/IP radio button (see image below).

iii. Set the IP address of the CHIYU Converter (e.g. 192.168.2.124) in the "Server IP Address" field.

iv. Set the IP port of the CHIYU Converter (e.g. 7024) in the "Server IP Port" field.

v. Set the polling address of the scanner you wish to get connected to (or select *Multiple*).

vi. Then press the *Connect* button.



## 8. Connection via Modbus

a. Initial Settings:
   i. Comport settings
      1. Baud Rage = 115200
      2. Data Bits = 8
      3. Parity = None
      4. Stop Bits = 1

b. In order to get the values from the Scanner, a Modbus RTU command with function code 3 should be executed.

c. The address of the device equals its Polling Address + 1

d. Registers - The relevant parameters are stored in registers 40001-40026 (see table).

| No. | Register Number | Parameter | Units | Type |
|---|---|---|---|---|
| 1 | 40001 & 40002 | Avg. Distance | m | Floating Point |
| 2 | 40003 & 40004 | Min Distance | m | Floating Point |
| 3 | 40005 & 40006 | Max Distance | m | Floating Point |
| 4 | 40007 & 40008 | Volume | % (*1000) | Floating Point |
| 5 | 40009 & 40010 | 4-20mA | mA | Floating Point |
| 6 | 40011 & 40012 | SNR | dB | Floating Point |
| 7 | 40013 & 40014 | Temperature | °C | Floating Point |
| 8 | 40015 & 40016 | Temperature | °F | Floating Point |
| 9 | 40017 & 40018 | Avg. Level | m | Floating Point |
| 10 | 40019 & 40020 | Bulk Density | g/cm^3 | Floating Point |
| 11 | 40021 & 40022 | Mass | customer unit | Floating Point |
| 12 | 40023 & 40024 | S/N | number | Long Integer |
| 13 | 40025 & 40026 | Tag Name | First 4 chars | 4 Characters Integer |

e. All parameters are stored as IEEE754 floating point; hence each parameter is stored in two consecutive registers. The byte sending order is from high to low.

f. Communication examples for a device with polling address 0:
   i. Example A:
      1. To retrieve the average distance, we send: 01 03 00 00 00 02 C4 0B
      2. A sample reply would be: 01 03 04 3D 85 1E 20 EE 0E.
         This means that Distance = 0x3D851E20, which represents a float value of "0.065".
   ii. Example B:
      1. to acquire all relevant parameters, we send: 01 03 00 00 00 1A C4 01

**Note:** Mass value is in the chosen configured units and relevant only if the scanner linearization is configured to mass.

## 9. Connection via RS485

a. The physical layer is RS-485 as follows:

1. Baud Rage = 115200
2. Data Bits = 8
3. Parity = None
4. Stop Bits = 1

b. The available parameters and their units are as follows:

| Parameter | Units |
|---|---|
| Avg. Distance | m |
| Min Distance | m |
| Max Distance | m |
| Volume | % |
| 4-20mA | mA |
| SNR | dB |
| Temperature | °C |
| Temperature | °F |
| Avg. Level | m |
| Bulk Density | g/cm^3 |
| Mass | Customer units |
| S/N | number |
| Tag Name | First 4 chars |

To retrieve all thirteen parameters, the controller needs to transmit a single command. The Packet format of the request is as follows:

| 1 BYTE | 1 BYTE | 4 BYTES | 2 BYTES |
|---|---|---|---|
| Polling Address +1 | 03 | 00 00 00 1A | CRC-16 |

- The above values are all in hexadecimal
- For CRC-16, see reference implementation in Appendix B
- The below table indicates the request string for retrieving data from 15 3DLevelScanners (polling addresses 00 to 15):

| Polling Address | Request |
|---|---|
| 00 | 01 03 00 00 00 1A C4 01 |
| 01 | 02 03 00 00 00 1A C4 32 |
| 02 | 03 03 00 00 00 1A C5 E3 |
| 03 | 04 03 00 00 00 1A C4 54 |
| 04 | 05 03 00 00 00 1A C5 85 |
| 05 | 06 03 00 00 00 1A C5 B6 |
| 06 | 07 03 00 00 00 1A C4 67 |
| 07 | 08 03 00 00 00 1A C4 98 |

| Polling Address | Request |
|---|---|
| 08 | 09 03 00 00 00 1A C5 49 |
| 09 | 0A 03 00 00 00 1A C5 7A |
| 10 | 0B 03 00 00 00 1A C4 AB |
| 11 | 0C 03 00 00 00 1A C5 1C |
| 12 | 0D 03 00 00 00 1A C4 CD |
| 13 | 0E 03 00 00 00 1A C4 FE |
| 14 | 0F 03 00 00 00 1A C5 2F |
| 15 | 10 03 00 00 00 1A C7 40 |

c. Then, the packet format of the reply is as follows:

| 1 BYTE | 1 BYTE | 1 BYTES | 4 BYTES | 4 BYTES | 4 BYTES |
|---|---|---|---|---|---|
| Polling Address +1 | 03 | 34 | Avg. Distance | Min Distance | Max Distance |

| 4 BYTES | 4 BYTES | 4 BYTES | 4 BYTES | 4 BYTES | 4 BYTES |
|---|---|---|---|---|---|
| Volume | 4-20mA | SNR | Temp (°C) | Temp (°F) | Avg. Level |

| 4 BYTES | 4 BYTES | 4 BYTES | 4 BYTES | 2 BYTES |
|---|---|---|---|---|
| Bulk Density | Mass | S/N | Tag Name | CRC-16 |

- Each of the eight 4-byte block represents a floating point value according to the IEEE-754 standard.
  We refer the reader to:
  http://en.wikipedia.org/wiki/IEEE_754-1985
- Example:
  i. Suppose the reply starts as follows (in this example we show only the first 4 bytes of payload):

| Byte Number | Value |
|---|---|
| 00 | 01 |
| 01 | 03 |
| 02 | 34 |
| 03 | 3D |
| 04 | 85 |
| 05 | 1E |
| 06 | 20 |
| … | … |

  ii. Then the average distance is represented by a 4-bytes-block of 0x3D851E20 which corresponds to floating point value of 0.065m.

To verify, enter "3d851e20" in the *hexadecimal representation* text box at the following webpage:

http://babbage.cs.qc.cuny.edu/IEEE-754/32bit.html

To Calculate CRC-16, it is possible to use the following website:

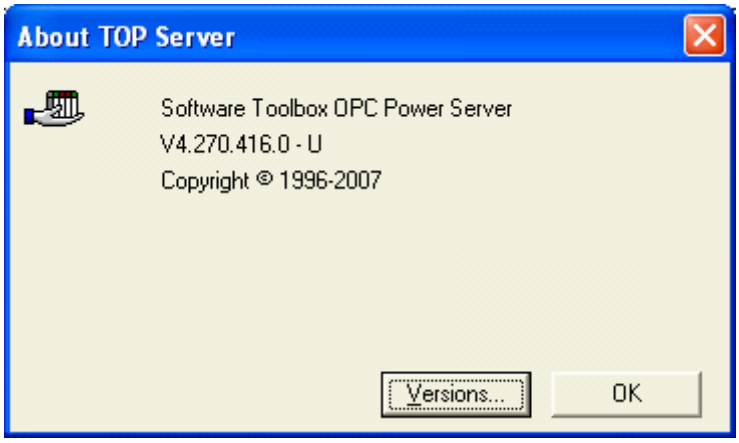
http://www.lammertbies.nl/comm/info/crc-calculation.html

Select input type HEX, type in the string to convert and press the 'Calculate CRC' button, look for the 'CRC-16 (Modbus)' value in the table.

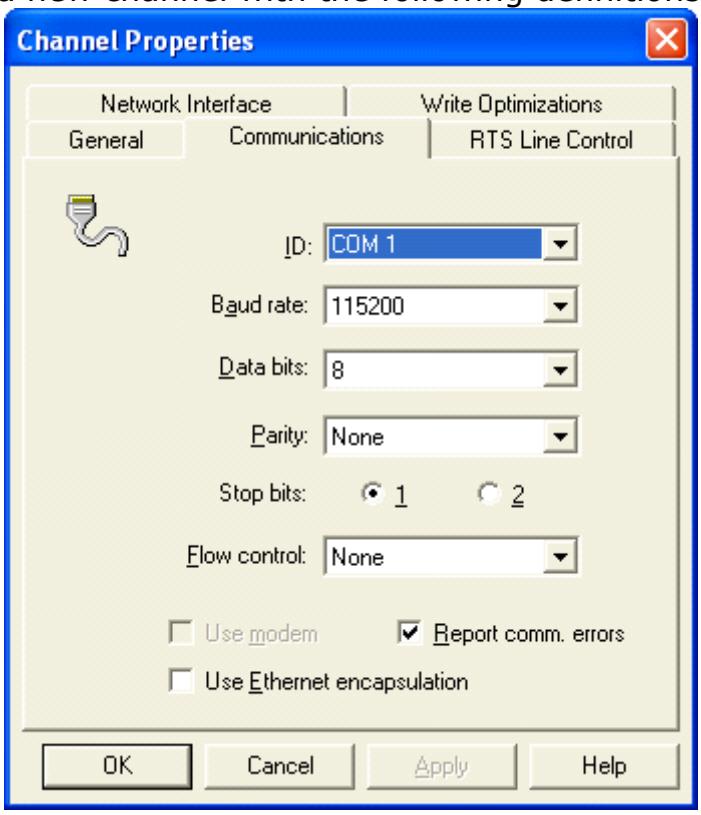
Output result is flipped, for example, if the result is 0x01C4, the two bytes to be transmitted are C401.

## 10.Appendix A – Example using InTouch System

1. This example uses TOP Server with version:

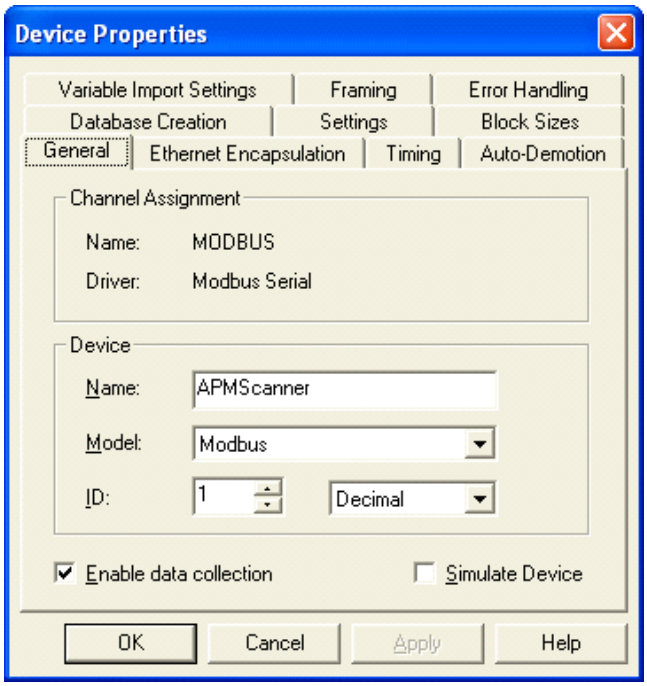



2. Create a new file.
3. Create a new channel with the following definitions:

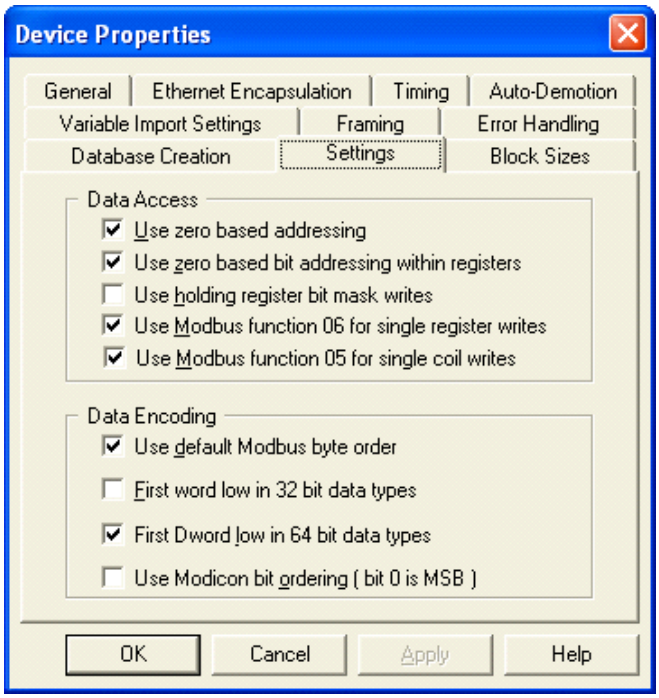


3. Add a new device (MODBUS serial) and set the following properties:

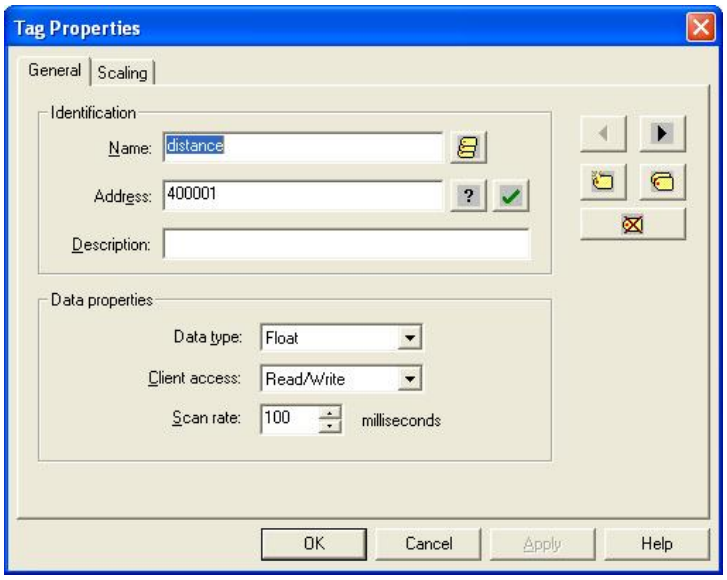
   3.1 ID = polling address +1



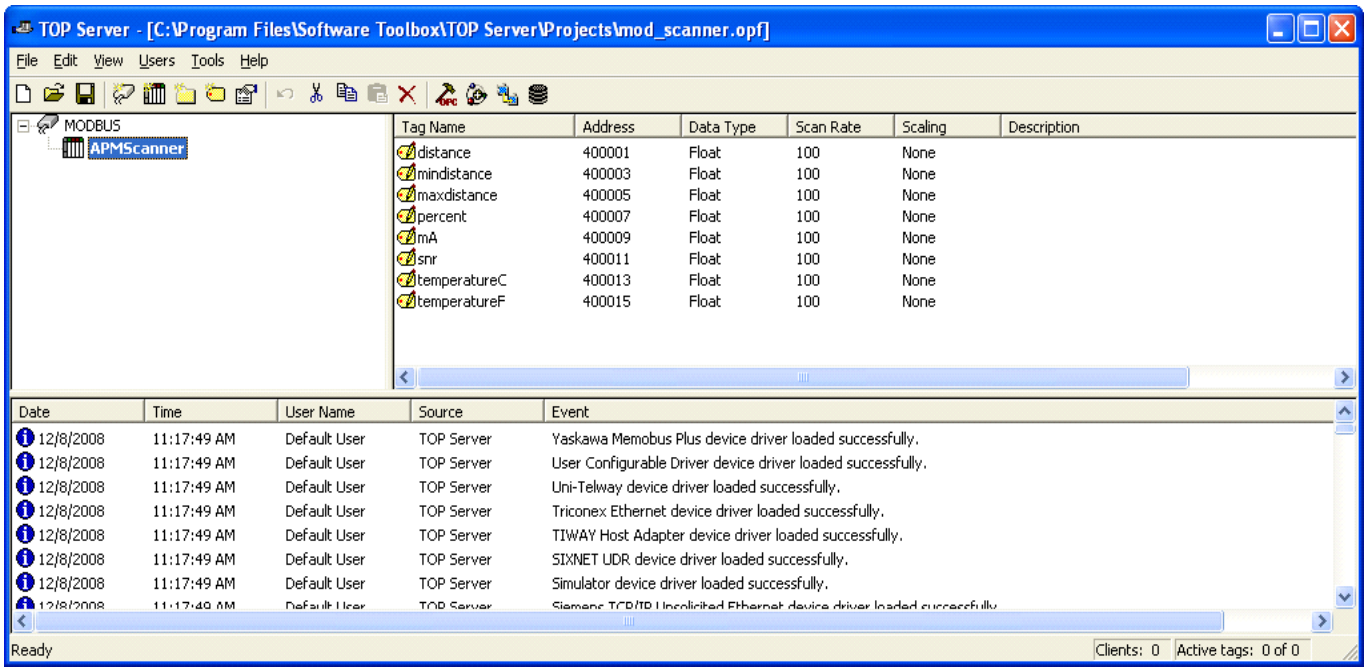

   3.2



4. Create tag Properties according to the table in page 2 / clause 4

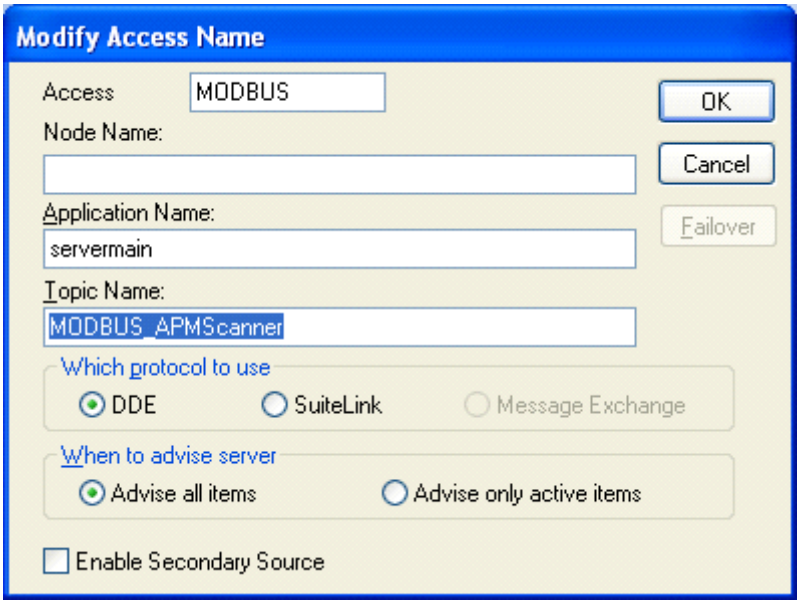



5. The TOP Server will then present all the registers as follows:

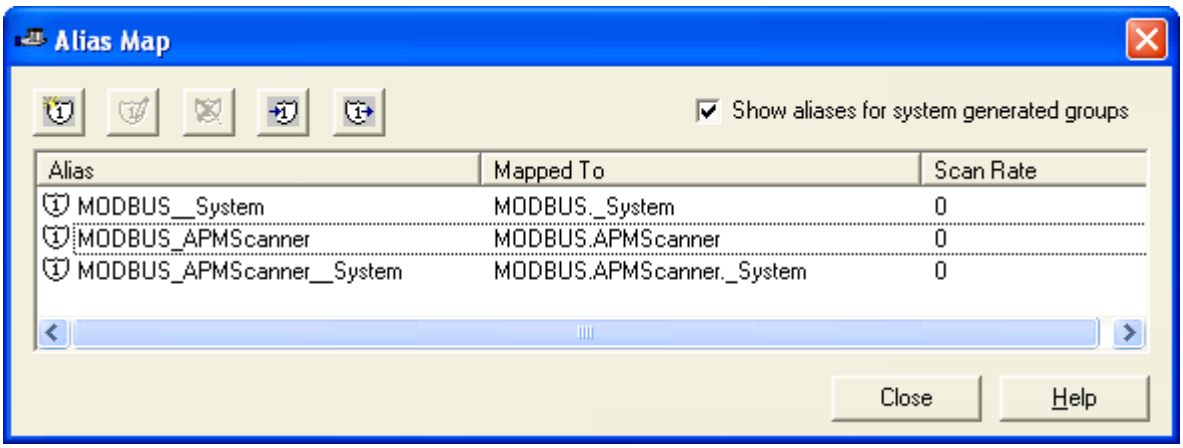


## In the InTouch configure the following:

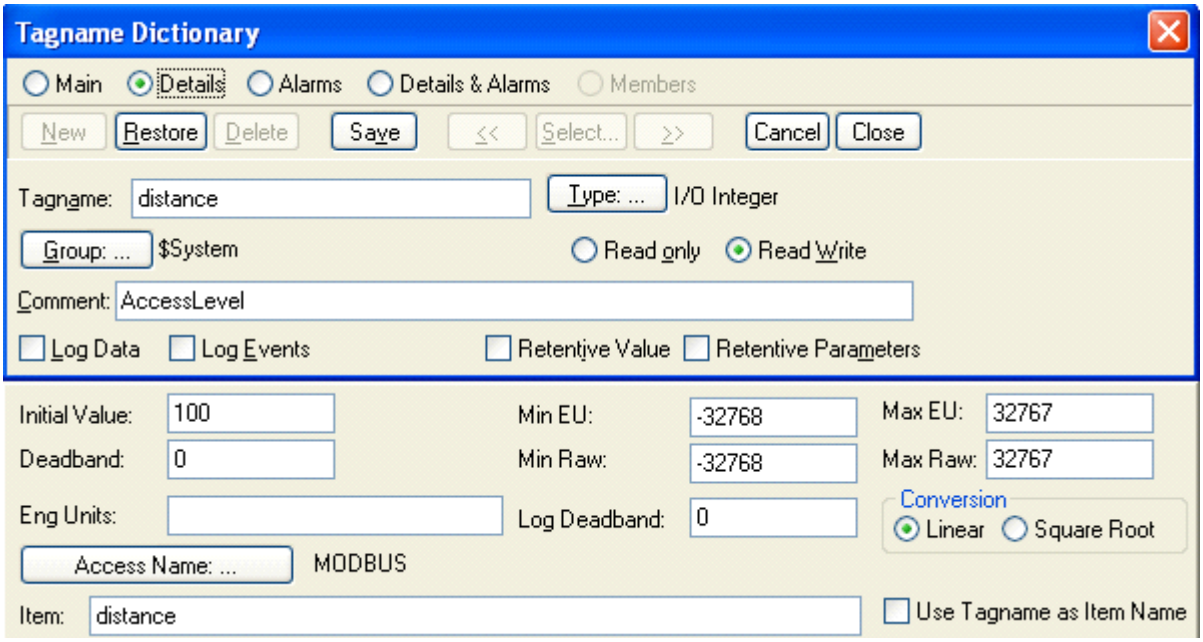
1. Create a new access name with topic name = alias name from TOP Server:



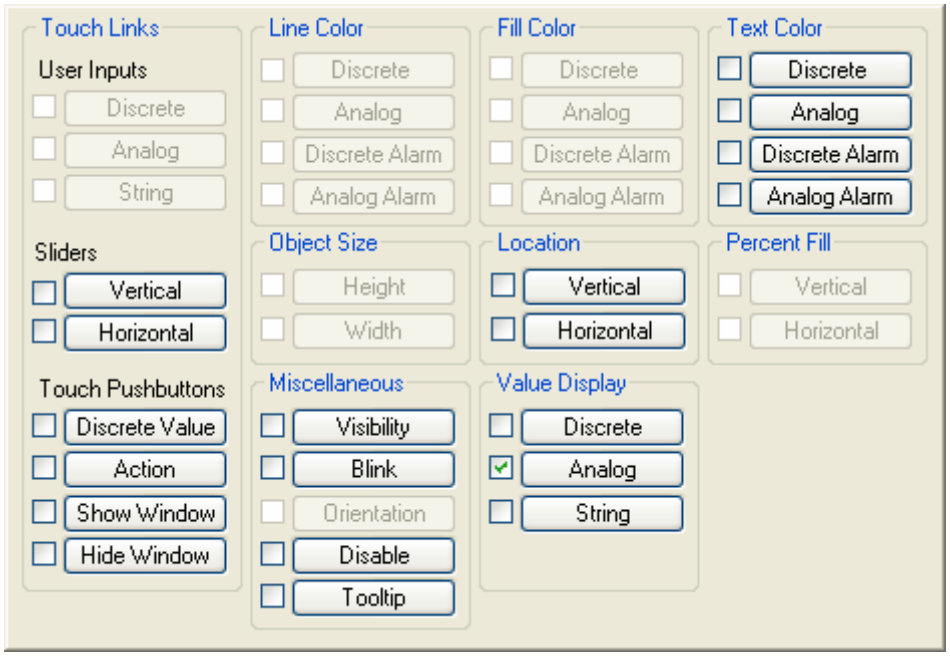

2. Topic name should be the same as the alias name in TOP Server (in this example connecting to *APMScanner* device in MODBUS channel, this means the access name should be *MODBUS_APMScanner*) see image below:



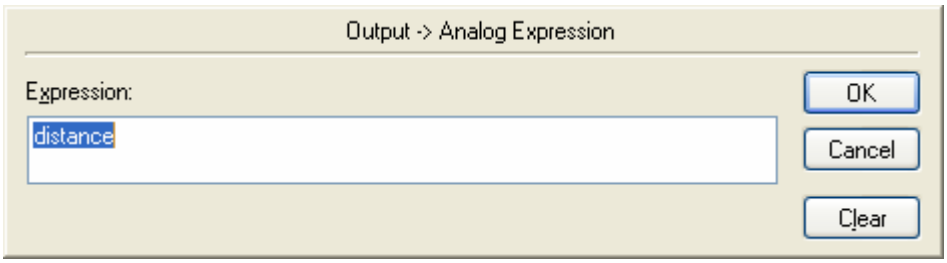
3. Define tags according to your tag names definitions in TOP Server. Put your created access name



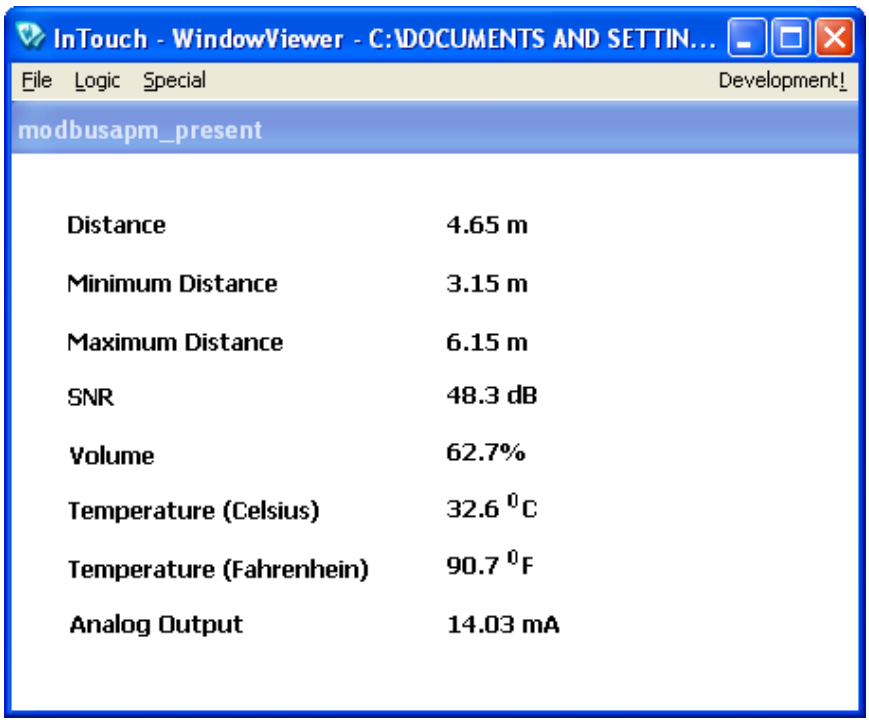

4. Create a new screen with some texts that will display analog value:



5. In the expression put the tag name:




6. The WindowViewer of InTouch will present the outcome:



## 11.Appendix B – CRC-16 Reference implementation

```
//
// code for computing crc-16
// The polynomial is
// X^16+X^15+X^2+X^0 (8005)
/* Table of CRC values for high-order byte */
unsigned char crc16tableHI[256] = {
0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,
0x40,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,
0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,
0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,
0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,
0x40,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,
0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,
0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,
0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,
0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x01,0xC0,
0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,
0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,
0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,
0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x01,0xC0,
0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,
0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,
0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,
0x40
} ;
/* Table of CRC values for low-order byte */
unsigned char crc16tableLO[256] = {
0x00,0xC0,0xC1,0x01,0xC3,0x03,0x02,0xC2,0xC6,0x06,0x07,0xC7,0x05,0xC5,0xC4,
0x04,0xCC,0x0C,0x0D,0xCD,0x0F,0xCF,0xCE,0x0E,0x0A,0xCA,0xCB,0x0B,0xC9,0x09,
0x08,0xC8,0xD8,0x18,0x19,0xD9,0x1B,0xDB,0xDA,0x1A,0x1E,0xDE,0xDF,0x1F,0xDD,
0x1D,0x1C,0xDC,0x14,0xD4,0xD5,0x15,0xD7,0x17,0x16,0xD6,0xD2,0x12,0x13,0xD3,
0x11,0xD1,0xD0,0x10,0xF0,0x30,0x31,0xF1,0x33,0xF3,0xF2,0x32,0x36,0xF6,0xF7,
0x37,0xF5,0x35,0x34,0xF4,0x3C,0xFC,0xFD,0x3D,0xFF,0x3F,0x3E,0xFE,0xFA,0x3A,
0x3B,0xFB,0x39,0xF9,0xF8,0x38,0x28,0xE8,0xE9,0x29,0xEB,0x2B,0x2A,0xEA,0xEE,
0x2E,0x2F,0xEF,0x2D,0xED,0xEC,0x2C,0xE4,0x24,0x25,0xE5,0x27,0xE7,0xE6,0x26,
0x22,0xE2,0xE3,0x23,0xE1,0x21,0x20,0xE0,0xA0,0x60,0x61,0xA1,0x63,0xA3,0xA2,
0x62,0x66,0xA6,0xA7,0x67,0xA5,0x65,0x64,0xA4,0x6C,0xAC,0xAD,0x6D,0xAF,0x6F,
0x6E,0xAE,0xAA,0x6A,0x6B,0xAB,0x69,0xA9,0xA8,0x68,0x78,0xB8,0xB9,0x79,0xBB,
0x7B,0x7A,0xBA,0xBE,0x7E,0x7F,0xBF,0x7D,0xBD,0xBC,0x7C,0xB4,0x74,0x75,0xB5,
0x77,0xB7,0xB6,0x76,0x72,0xB2,0xB3,0x73,0xB1,0x71,0x70,0xB0,0x50,0x90,0x91,
0x51,0x93,0x53,0x52,0x92,0x96,0x56,0x57,0x97,0x55,0x95,0x94,0x54,0x9C,0x5C,
0x5D,0x9D,0x5F,0x9F,0x9E,0x5E,0x5A,0x9A,0x9B,0x5B,0x99,0x59,0x58,0x98,0x88,
0x48,0x49,0x89,0x4B,0x8B,0x8A,0x4A,0x4E,0x8E,0x8F,0x4F,0x8D,0x4D,0x4C,0x8C,
0x44,0x84,0x85,0x45,0x87,0x47,0x46,0x86,0x82,0x42,0x43,0x83,0x41,0x81,0x80,
0x40
};
unsigned short DoCrc16Block( unsigned char *s, unsigned short length )
{
    unsigned char HI = 0xFF ; /* high byte of CRC initialized  */
    unsigned char LO = 0xFF ; /* low byte of CRC initialized  */
    unsigned i ;                  /* index into CRC lookup table  */
    while (length--)
    {
          i = LO ^ *s++;
          LO = crc16tableHI[i] ^ HI;
          HI = crc16tableLO[i];
    }
    return (HI << 8 | LO) ;
}
```