

Using Excel 97 as a Low Cost HMI

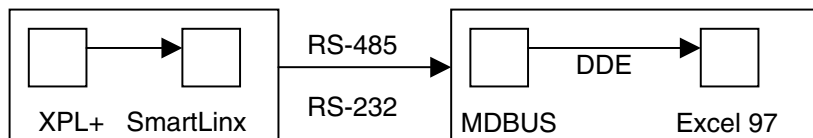
For an XPL+ with SmartLinx Modbus RTU and MDBUS by Calta computer systems.

This application guide describes how to set up a system to pass information from Milltronics XPL+, to a computer running Excel 97 and MDBUS driver. This is a low cost method of getting level information remotely (a low cost HMI (Human Machine Interface)). This setup would work largely the same way with any other Milltronics product equipped with Modbus.

The computer used to develop this application was a Pentium running Window 95, Excel 97, and MDBUS Version 3.24 (made by Calta computer systems, www.calta.com). The AirRanger used was an XPL+ (software version 5.00) with a SmartLinx Modbus RTU card installed. Three types of connection were tested and described here: RS-232 direct cable, RS-485 cable with RS232-RS485 converter at the computer side, and dial-up modems (3com PCM/CIA modem at the computer side and a 56K Sportster modem at the XPL+ side).

MDBUS is a low cost DDE (Dynamic Data Exchange) Modbus driver designed to interface with Excel using VBA (Visual BASIC for Applications) macros. VBA comes as part of Excel 97.

Level information passes from the XPL+ through the Modbus holding registers (4xxxx) on the SmartLinx module, across the serial connection, through the MDBUS driver (using DDE), and into the Excel spreadsheet.



Accompanying this application brief is an example Excel file complete with the macros described here.

Figure 1 below shows a typical application.

Keywords:

SmartLinx Modbus RTU, Excel, DDE, remote telemetry, RS-232, RS-485, MDBUS, PC, Modbus Driver, Cheap HMI, modem

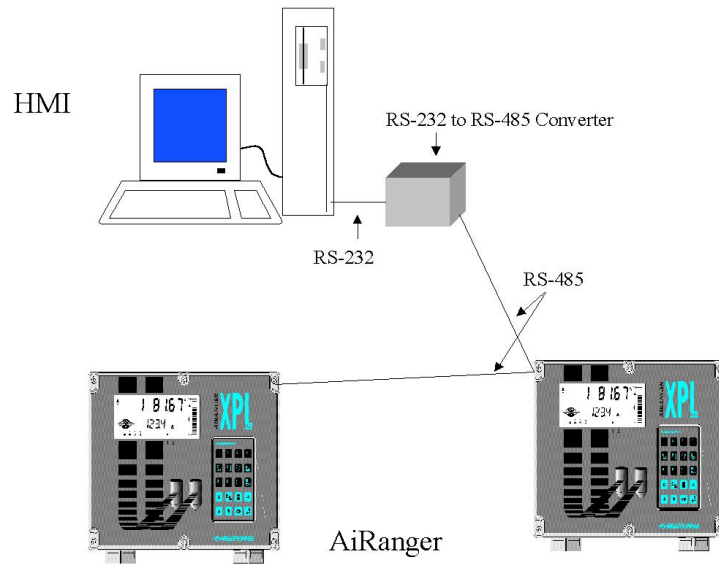


Figure 1 – Typical Configuration

Description:

Programming the XPL+

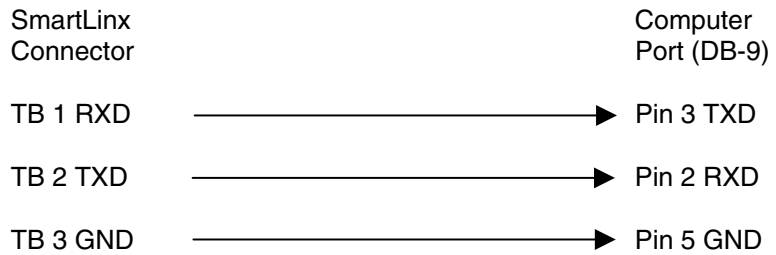
After programming the XPL+ for the application, only three parameters need to be set to configure the SmartLinX Modbus RTU module. They are:

- P751 = 3 (for 9600 baud)
- P752 = 0 (no parity)
- P753 = 1 (for station 1)
- P758 = 15 (Interframe spacing – for modem use only)

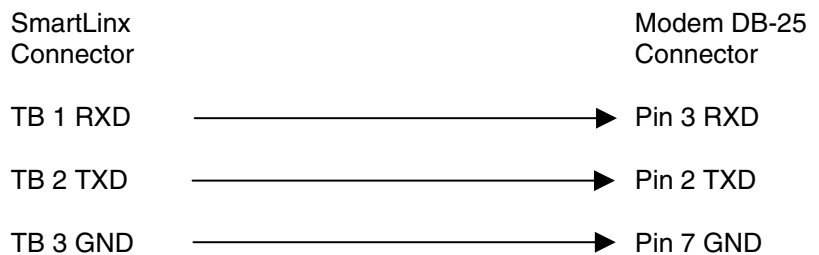
Please note that these parameters are not accessible if the SmartLinX Modbus RTU module is not installed. Also, if a direct connection using RS-232 or RS-485 is used, then the interframe spacing, P758, should remain at the default of 0.

There is also one dip switch on the SmartLinX board that needs to be set to determine if RS-232 or RS485 are used. SW1 of 8 is off for RS-232 and on for RS-485 (See SmartLinX manual for more details). The setting of this switch depended on the cable used. If a modem or RS-232 direct connection was used, then the setting for RS-232 was used. If a RS-485 connection was used, then the RS-485 setting was used.

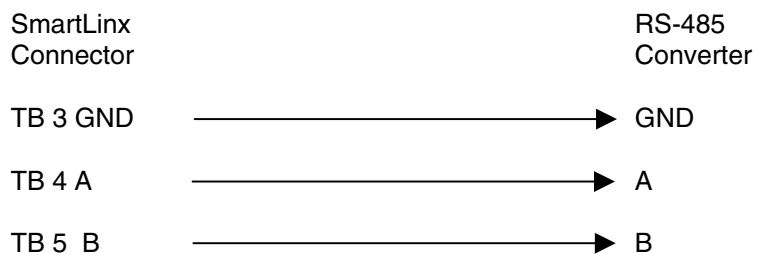
For RS-232 direct connection:



For Modem connection:



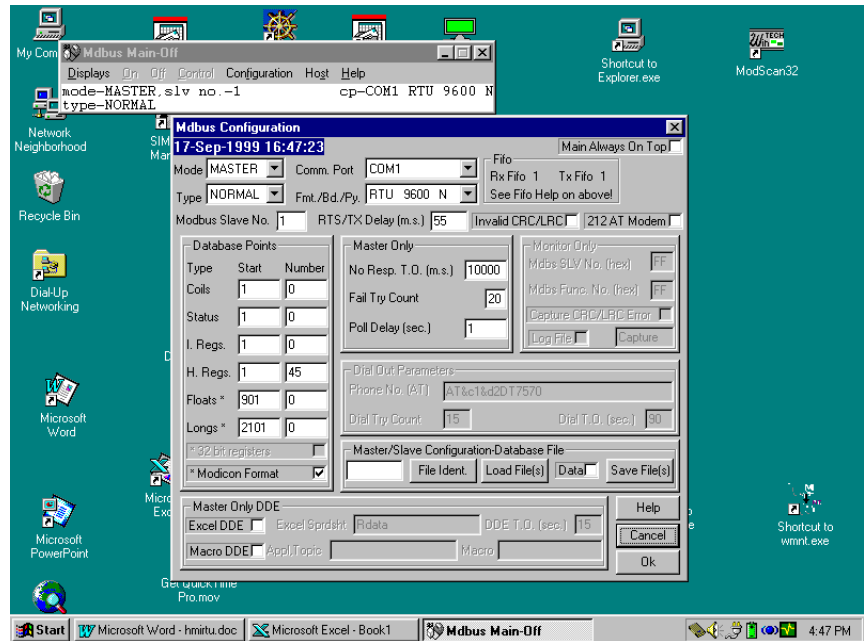
For RS-485 connection:



The RS-232 to RS-485 converter used is a port-powered converter made by B&B Electronics (Milltronics Part# 20150159). The device is wired as above then plugged into the RS-232 port on your computer.

Configuring MDBUS:

After installing MDBUS, run MDBUS from the start menu by clicking Start/Programs/Modbus 32 Simulator/mdbus. To configure MDBUS, click on configure in the top selection and you will get a screen like:



The settings are:

- Mode: Master
- Type: Normal
- Comm. Port: Com1 (*depends on application and setup of computer*)
- Fmt/Bd/Py: RTU 9600 N
- Modbus Slave No.:1 (*address of XPL+, see P753*)
- RTS/TX Delay (ms.): 55
- Invalid CRC/LRC: *not checked*
- 212 AT Modem: (*checked for modem application, unchecked for others*)
- Coil Start: 1
- Coil Number: 0
- Status Start: 1
- Status Number: 0
- I Regs. Start: 1
- I Regs. Number: 0
- H Regs. Start: 1
- H Regs. Number: 45
- Floats Start: 1
- Floats Number: 0
- Longs Start: 1
- Longs Number: 0
- Modicon format: *checked*
- No. Resp. T.O. (m.s.): 1000
- Fail Try Count: 20
- Poll Delay (sec.): 1
- Excel DDE: *not checked*
- Macro DDE: *not checked*

For Modem applications, the 212 AT Modem box is checked and the following setting done:

Phone No AT: AT&C1&D2DTaaaa where aaaa= phone number
Dial Try Count: 15
Dial T.O. (sec.): 90

The Excel example given here is using the advanced features of MDBUS to get level, volume, and temperature information from the XPL+. However, if the user only wants the level information, then change the following in the MDBUS setup:

Excel DDE: checked
Excel Spdsht: sheet2
DDE T.O. (sec.): 15

This will have the Holding registers 1 to 45 sent Excel and stored on sheet 2 starting in column 4 row 4. The levels would then be stored starting at column 4 row 5 to row 15. The levels are in % of span and set to two decimal places (i.e. 4500 would mean 45.00 %). With the Excel DDE box checked, the level information will be updated automatically every second.

Excel Setup

Excel is set up with three sheets named Display, Rdata, and Def.

The Display sheet is used to display the data and call the macros used to get the data.

The Rdata sheet is used to place the data from the DDE transfer.

The Def sheet is used to store some values used by the macros for DDE commands.

There are a total of 4 macros: Loadmdbus, Runmdbus, Update1, UpdateC. These macros are shown below for completeness.

Loadmdbus loads the driver program called mdbus:

```
Sub Loadmdbus()  
  
    'This command loads mdbus driver  
    Shell "C:\mdbus\mdbus.exe", vbNormalFocus  
  
End Sub
```

Note: you must change the directory given above if you did not accept the MDBUS defaults.

Runmdbus runs the MDBUS driver. This will use the last configuration set in MDBUS. Therefore this routine is dependent on the configuration of MDBUS being done first and not being changed. This program could be modified to set up the configuration itself using the DDE link.

```
Sub Runmdbus()  
  
    'run the driver which has already been loaded
```

```
channel = DDEInitiate("mdbus", "poke")
DDEPoke channel, "state", "Def!r1c1"
```

```
'we want to wait 2 seconds before proceeding to give
'time for communications to start
```

```
start = Timer
```

```
Do While Timer < start + 2
Loop
```

```
End Sub
```

Update1 uses the multiple parameter access method (MPA) to get the level information in units, volume, and temperature of the 10 tanks. For more detail on how MPA works please see the SmartLinx users manual.

```
Sub Update1()
```

```
Dim channel
Dim start
Dim datamg
Dim MPAPar As Integer
```

```
'set up MPA
```

```
DDEPoke channel, "HREG 035", "Def!r5c1"
DDEPoke channel, "HREG 034", "Def!r4c1"
DDEPoke channel, "HREG 033", "Def!r3c1"
```

```
'start a loop to go through readings, volume and temperature
```

```
For I = 1 To 3
```

```
    If I = 1 Then
        Worksheets("Def").Cells(2, 1).Value = 921
    End If
```

```
    If I = 2 Then
        Worksheets("Def").Cells(2, 1).Value = 924
    End If
```

```
    If I = 3 Then
        Worksheets("Def").Cells(2, 1).Value = 664
    End If
```

```
'Select Parameter for MPA
DDEPoke channel, "HREG 032", "Def!r2c1"
```

```
' Wait for the MPA to get to the AirRanger
start = Timer
Do While Timer < start + 1
    DoEvents
Loop
```

```
'get reg. table
datamg = DDERequest(channel, "hreg 1 45")
```

```
' Wait for dderequest to be done
start = Timer
Do While Timer < start + 2
    DoEvents
Loop

'put reg. table in excel
For j = LBound(datamg) To UBound(datamg)
    Worksheets("Rdata").Cells(j + 3, 4).Formula = datamg(j)
Next j

' move %span into other cells and scale them for display
For j = 1 To 10
Worksheets("Rdata").Cells(j + 85, 1).Value= Worksheets("Rdata").Cells(j + 4,
4).Value / 10000
Next j

' get the MPA parameter number so we know which ' value
we are getting
MPApar = Worksheets("Rdata").Cells(35, 4).Value

' Depending on what Parameter we requested
'determines where we put the values

If MPApar = 921 Then
    'move the MPA for P921 (Level measurment in units)
    For j = 1 To 10
Worksheets("Rdata").Cells(j + 50, 1).Value =
Worksheets("Rdata").Cells(j + 24, 4).Value
    Next j
End If

If MPApar = 924 Then
    ' move the MPA for P924 (Volume measurement in units)
    For j = 1 To 10
Worksheets("Rdata").Cells(j + 62, 1).Value =
Worksheets("Rdata").Cells(j + 24, 4).Value
    Next j
End If

If MPApar = 664 Then
    'move the MPA for P664 (Temperature in deg. C)
    For j = 1 To 10
Worksheets("Rdata").Cells(j + 74, 1).Value =
Worksheets("Rdata").Cells(j + 24, 4).Value
    Next j
End If

Next I

End Sub
```

UpdateC is the same as Update 1, except there is an infinite loop in it so that the program will continuously go out and get the data.

```
Sub UpdateC()
```

```
Dim channel  
Dim start  
Dim datamg  
Dim MPAPar As Integer
```

```
'set up MPA
```

```
DDEPoke channel, "HREG 035", "Def!r5c1"  
DDEPoke channel, "HREG 034", "Def!r4c1"  
DDEPoke channel, "HREG 033", "Def!r3c1"
```

```
'Loop to get data
```

```
For K = 1 To 10
```

```
' want to loop forever so we reset K  
K = 1
```

```
'start a loop to go through readings, volume and temperature
```

```
For I = 1 To 3
```

```
    If I = 1 Then  
        Worksheets("Def").Cells(2, 1).Value = 921  
    End If
```

```
    If I = 2 Then  
        Worksheets("Def").Cells(2, 1).Value = 924  
    End If
```

```
    If I = 3 Then  
        Worksheets("Def").Cells(2, 1).Value = 664  
    End If
```

```
'Select Parameter for MPA  
DDEPoke channel, "HREG 032", "Def!r2c1"
```

```
' Wait for the MPA to get to the AirRanger  
start = Timer  
Do While Timer < start + 1  
    DoEvents  
Loop
```

```
datamg = DDERequest(channel, "hreg 1 45") 'get reg. table
```

```
' Wait for dderequest to be done  
start = Timer  
Do While Timer < start + 2  
    DoEvents  
Loop
```

```
'put reg. table in excel  
For j = LBound(datamg) To UBound(datamg)  
    Worksheets("Rdata").Cells(j + 3, 4).Formula = datamg(j)  
Next j
```

```
' move %span into other cells and scale them for display
For j = 1 To 10
Worksheets("Rdata").Cells(j + 85, 1).Value =
Worksheets("Rdata").Cells(j + 4, 4).Value / 10000
Next j

' get the MPA parameter number so we know which value
we are getting
MPApar = Worksheets("Rdata").Cells(35, 4).Value

' Depending on what Parameter we requested determines
' where we put the values

If MPApar = 921 Then
' move the MPA for P921 (Level measurement in units)
For j = 1 To 10
Worksheets("Rdata").Cells(j + 50, 1).Value =
Worksheets("Rdata").Cells(j + 24, 4).Value
Next j
End If

If MPApar = 924 Then
' move the MPA for P924 (Volume measurement in units)
For j = 1 To 10
Worksheets("Rdata").Cells(j + 62, 1).Value =
Worksheets("Rdata").Cells(j + 24, 4).Value
Next j
End If

If MPApar = 664 Then
' move the MPA for P664 (Temperature in deg. C)
For j = 1 To 10
Worksheets("Rdata").Cells(j + 74, 1).Value =
Worksheets("Rdata").Cells(j + 24, 4).Value
Next j
End If

Next I

Next K

End Sub
```

Note: The information in this document is intended as a “guide” only. Milltronics assumes no responsibility for its application.