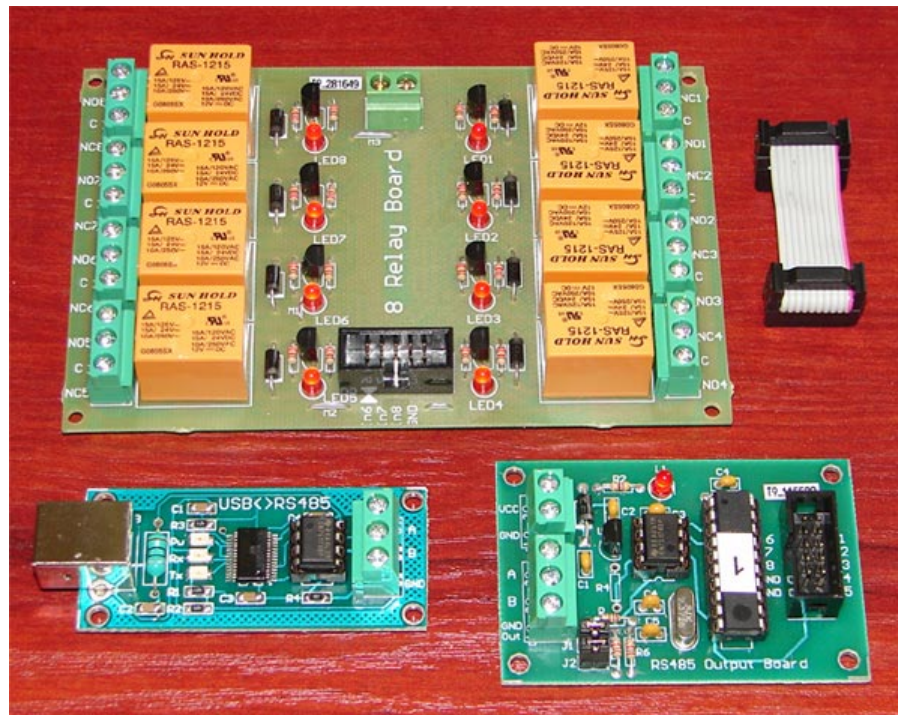


USB > RS485 > 8 Channel Relay Controller



Description :

The general purpose of RS-485 Relay Controller. RS-485 BUS can span relatively large distances (up to 4000 feet or just over 1200 meters). Control up to 8 devices using your PC. The required power is 12V DC / minimum 500 mA. Each relay has an LED to indicate when it is operated.

For 12V/24V DC 15A or 120V/220V AC at 10A max.

The complete includes:

- 1x USB to RS485 FTDI interface Board
- 1x RS485 to 8 Digitally Outputs controller, RS485 BUS ID: 01, Relays: 1-8 (12V)
- 1x 8 RELAY board (12V)
- 1x 10PIN flat cable approximate 5cm length

Specifications :

Relays has Normally Open (NO) and Normally Closed (NC) Contacts rated for voltages:

- 12VDC/15A

- 24VDC/15A
- 125VAC/15A
- 250VAC/10A

Relay's datasheet :

(relays used : RAS-1215 - 1C)

<http://www.sigma-shop.com/information/Relay/RAS.pdf>

Windows Server 2008 R2, Windows 7, Windows 7 x64, Windows Server 2008, Windows Server 2008 x64, Windows Vista, Windows Vista x64, Windows XP, Windows XP x64, Windows 2000, Windows Server 2003, Windows Server 2003 x64, Windows ME, Windows 98, Linux, Mac OS X, Mac OS 9, Mac OS 8, Windows CE.NET (Version 4.2 and greater)

Required power:

12VDC / minimum 500 mA.

Dimensions:

Relay board : 120mm / 80mm / 20mm

RS485 controller : 40 mm / 60 mm

Communication Parameters:

8 Data, 1 Stop, No Parity

Baud rate : 9600

Commands:

FIRST chanel commands:

OFF command : FF 01 00 (HEX) or 255 1 0 (DEC)

ON command : FF 01 01 (HEX) or 255 1 1 (DEC)

...

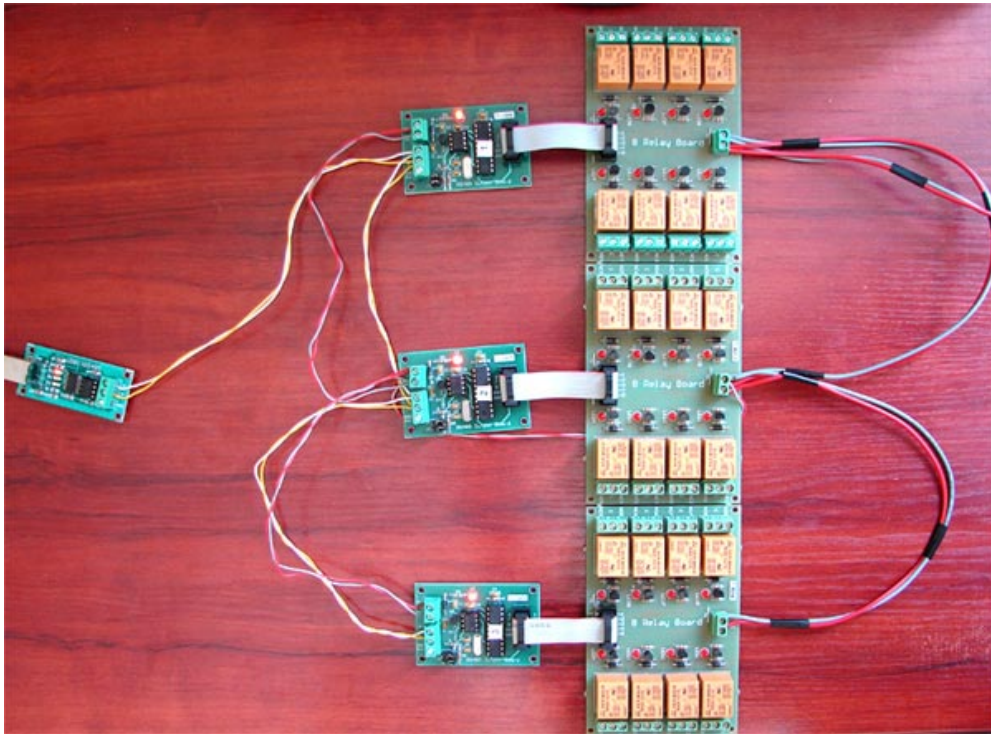
EIGHT chanel commands:

OFF command : FF 08 00 (HEX) or 255 8 0 (DEC)

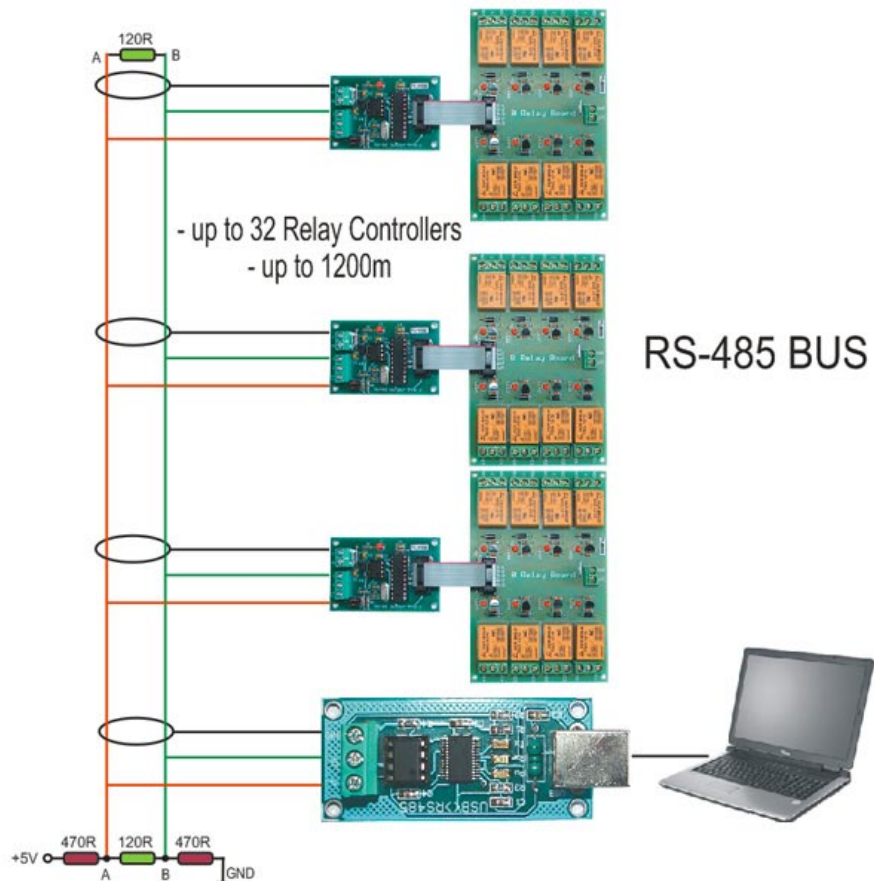
ON command : FF 08 01 (HEX) or 255 8 1 (DEC)

EXAMPLES:

3x 8RELAY controllers connected to PC via RS485



RS485 BUS example

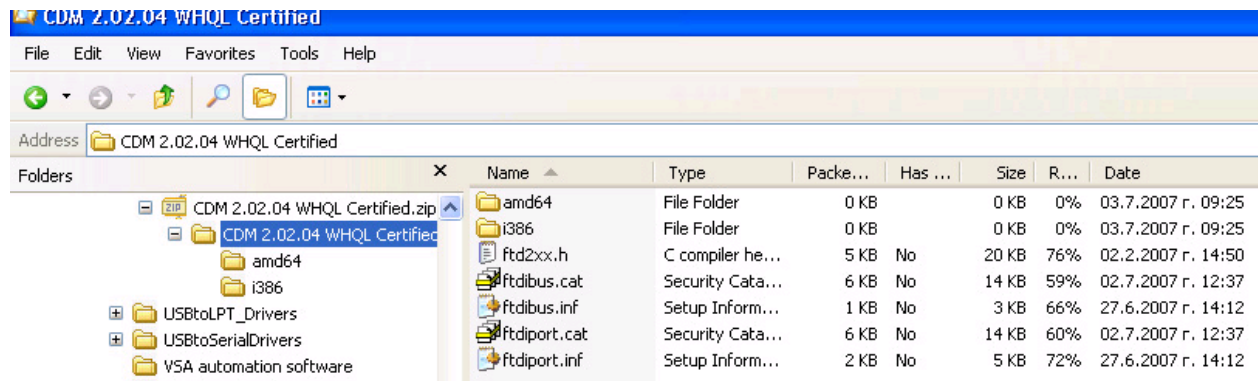


You have to download the drivers from FTDI website :

<http://www.ftdichip.com/Drivers/VCP.htm>

<http://www.ftdichip.com/Drivers/CDM/CDM%202.04.06%20WHQL%20Certified.zip>

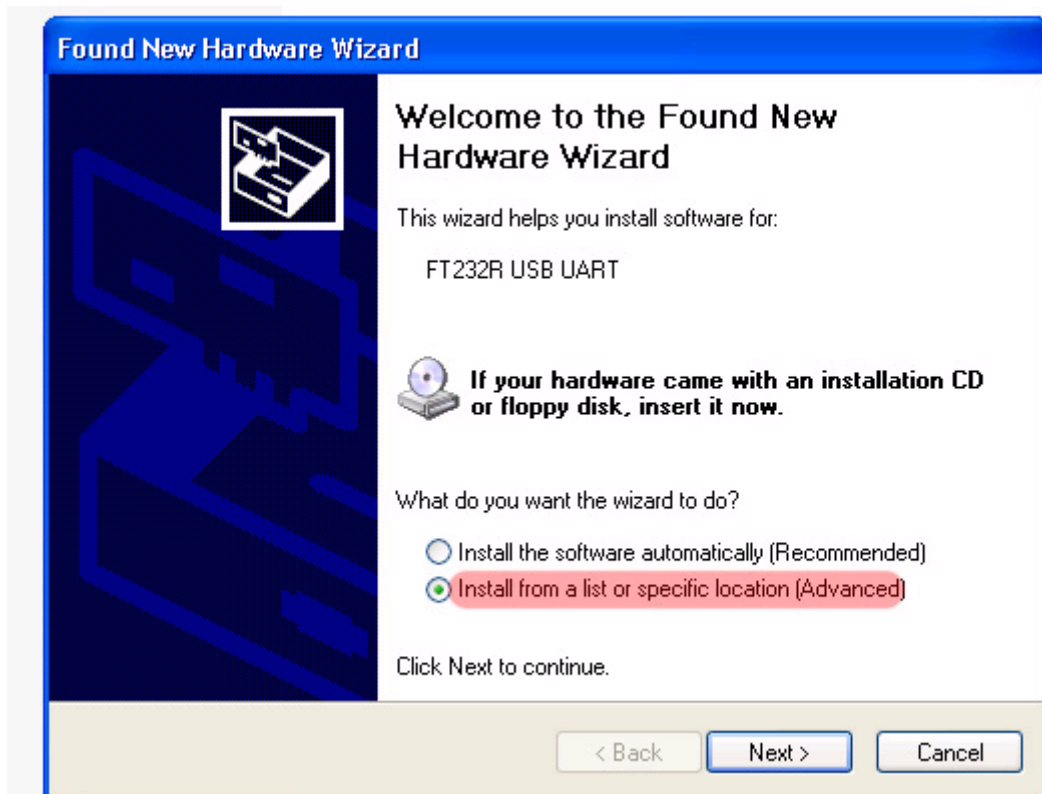
Unzip it to your own folder

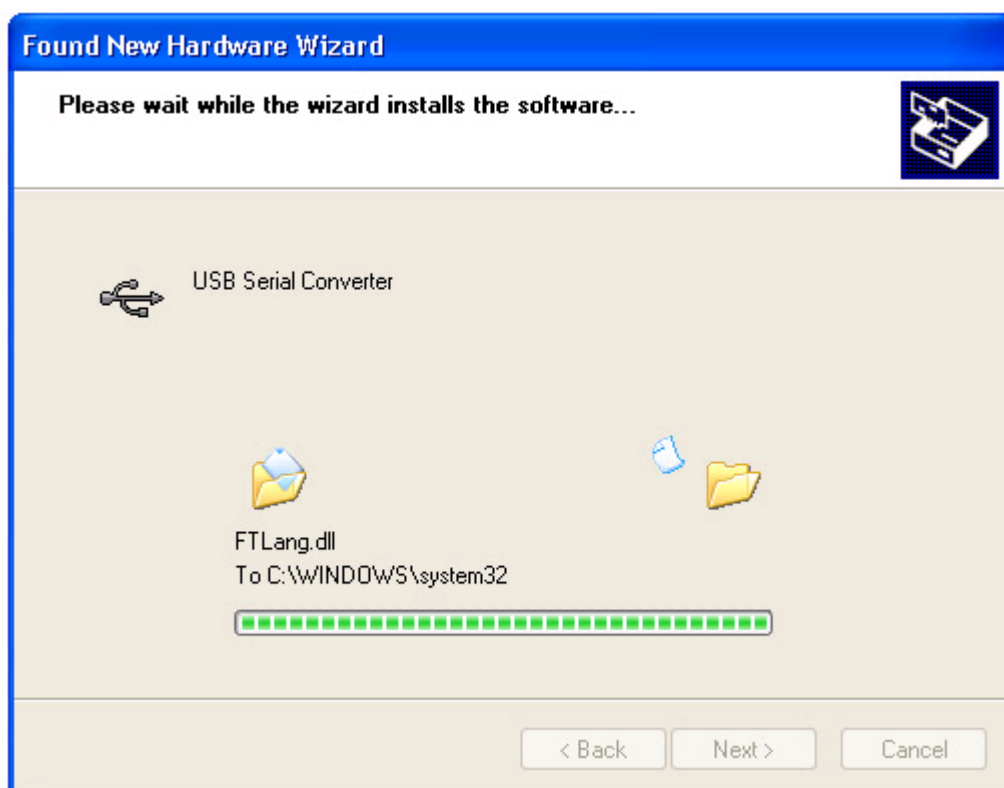
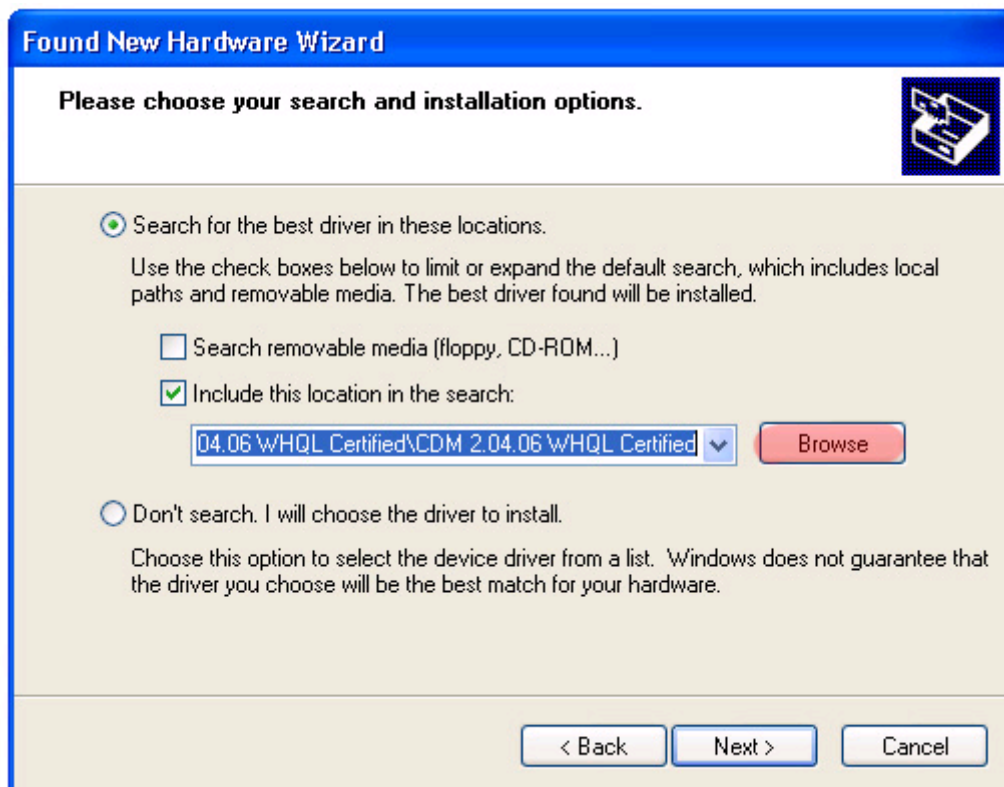


When you connect the USB board, the following message appears: FOUND NEW HARDWARE and drives need to be installed.

When you connect the USB board to the PC the red and green LED flashes 2,3 times and turn off. When the red LED and the green LED flashes means that there is a communication between the programmer and PC.

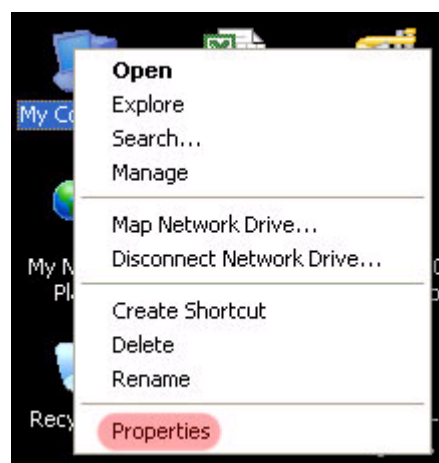
1. Drivers' installation:

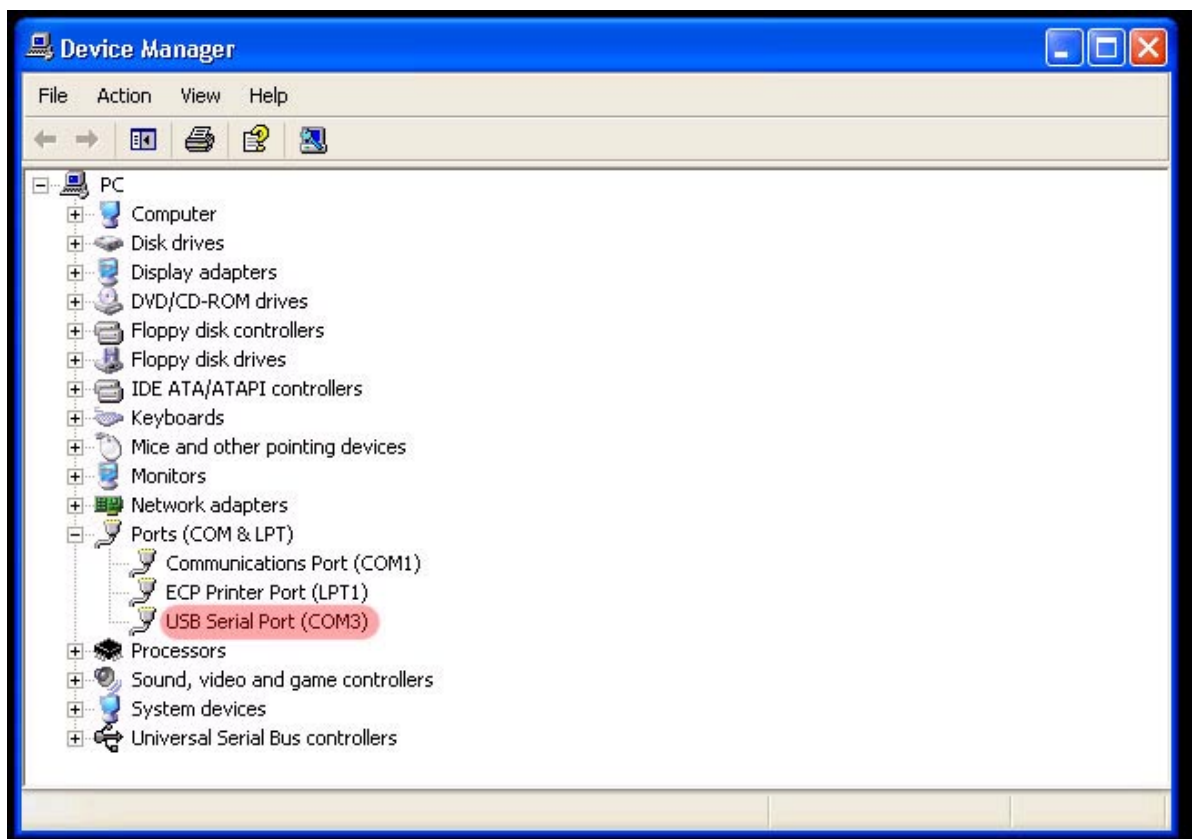
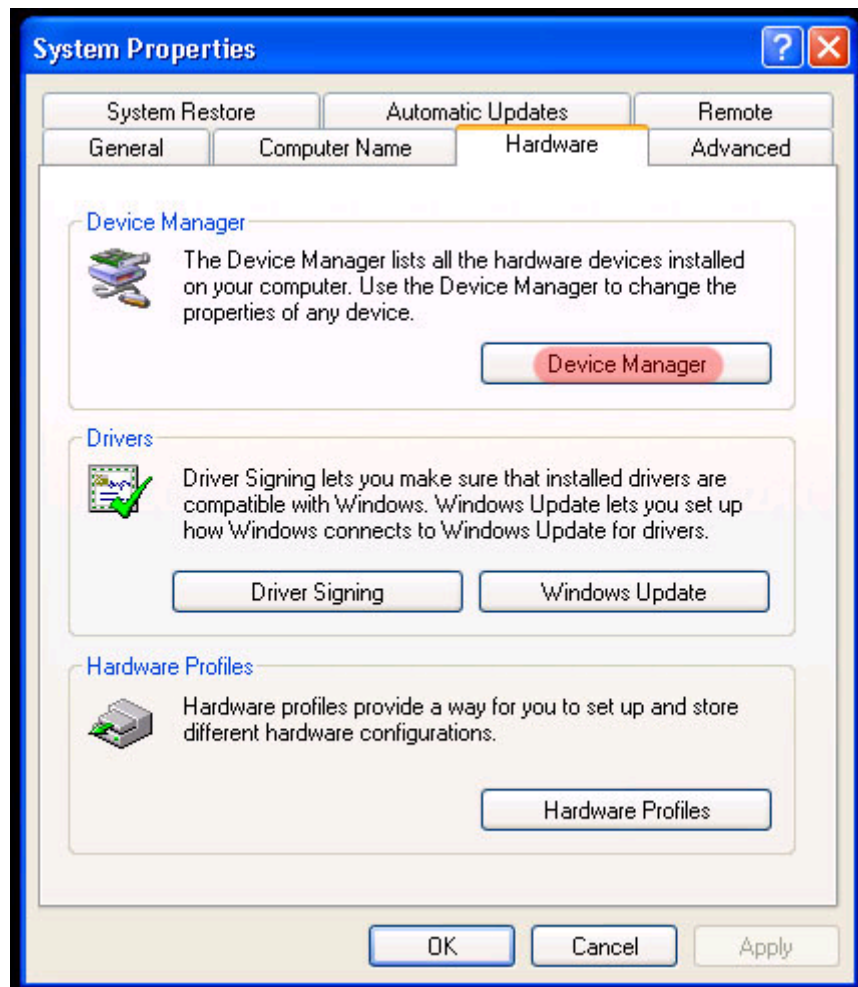






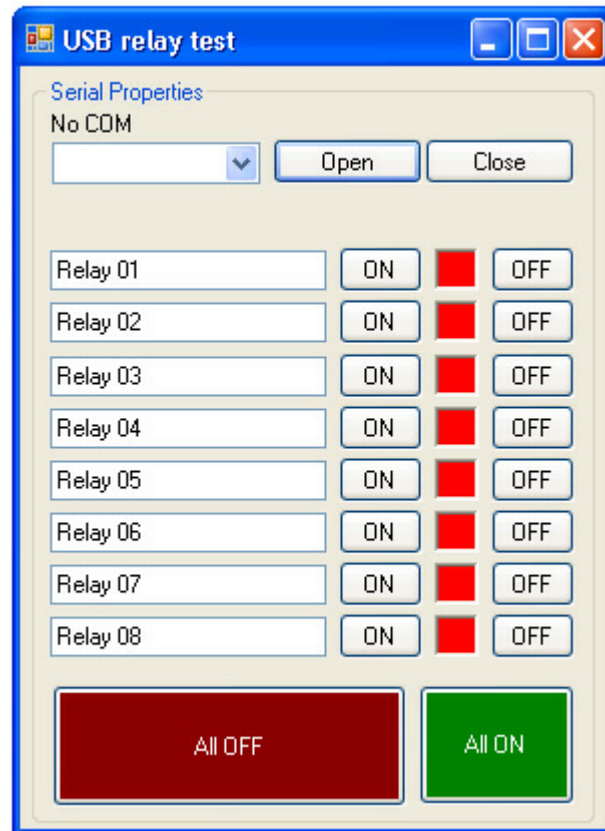
After the drivers' installed, an additional COM port appears in the Device Manager of Windows- usually it is COM3:



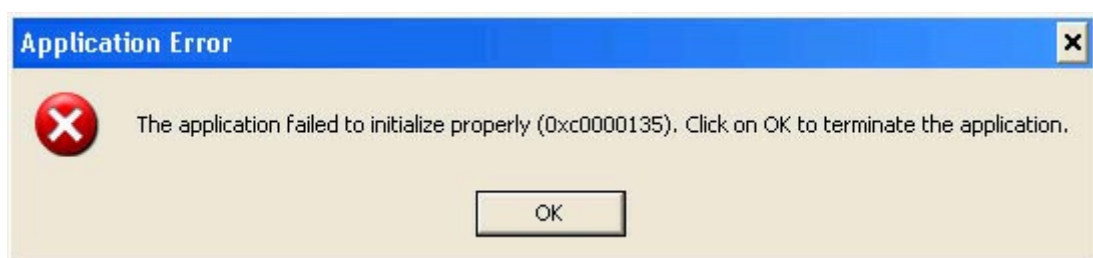


Testing :

1. Download "USBRelay_test.exe" from :
http://www.sigma-shop.com/information/USB_Relay/USBRelay_test.exe
2. Connect board to USB and run software.
If everything is ok this should appear :



If you get this message you see the following error :



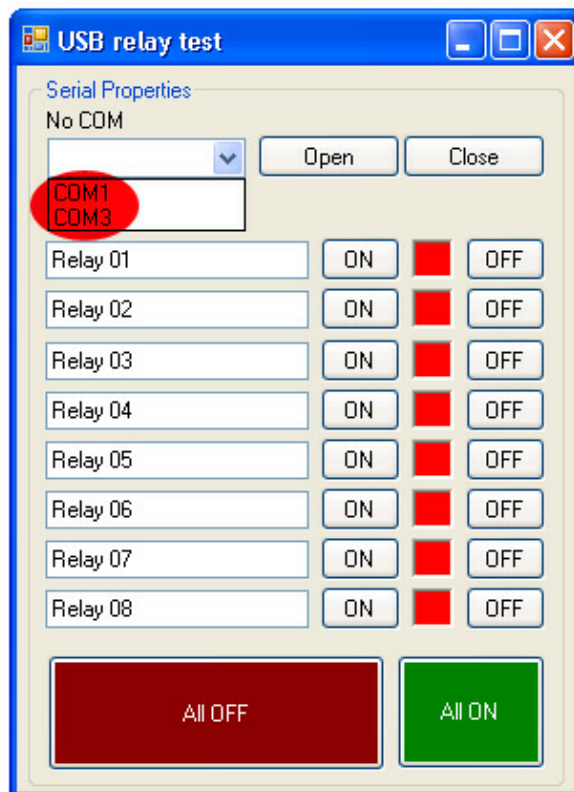
that means that you do not have the latest version of Microsoft .NET Framework installed on your computer.

The program has to be installed on your computer in order to use the software.

You can download this from [there](#)

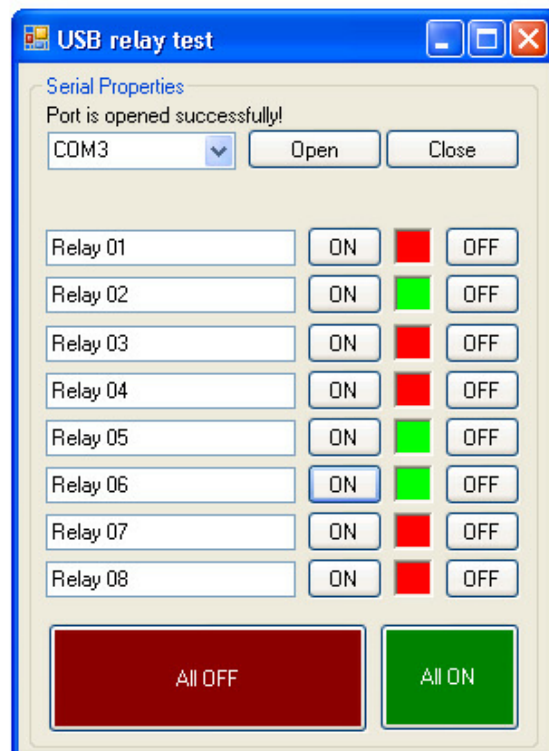
<http://www.microsoft.com/downloads/details.aspx?FamilyId=333325FD-AE52-4E35-B531-508D977D32A6&displaylang=en>

3. Software will detect all COM ports you have and there must be one more (USB Serial COM port) :

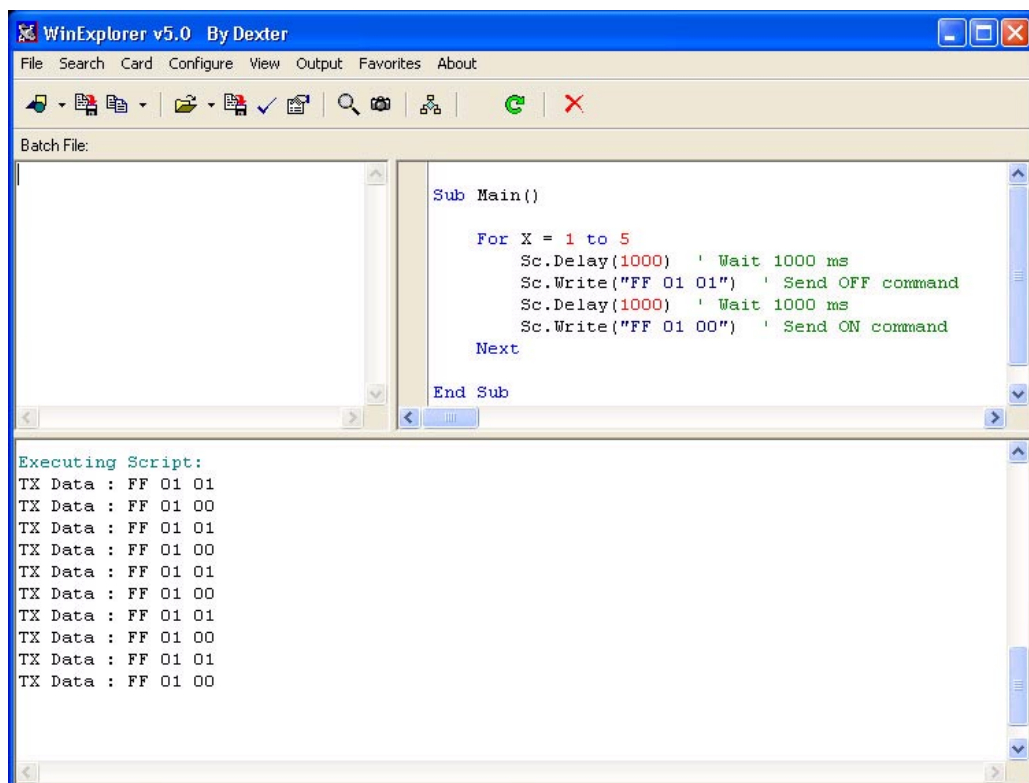
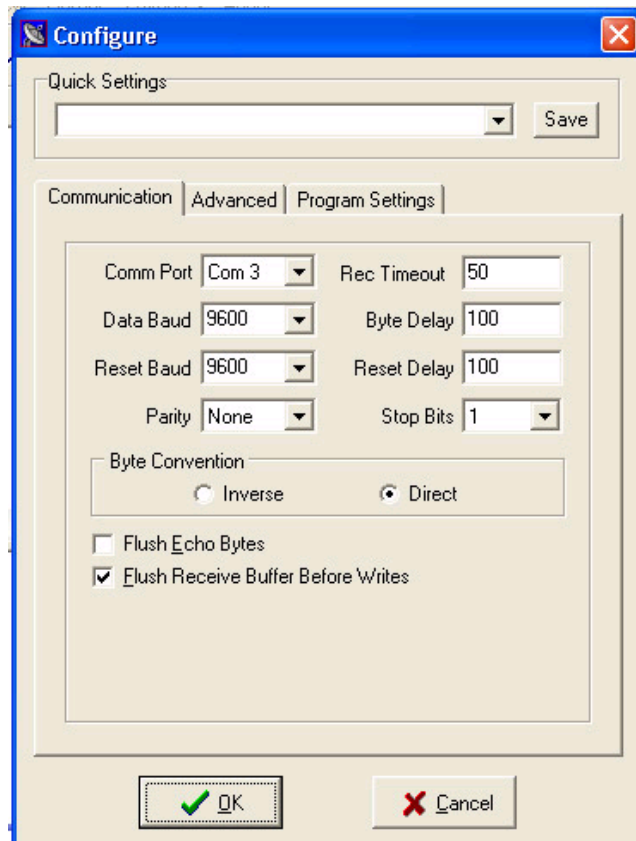


4. Choose COM port. Click on "OPEN" button and using "ON" and "OFF" buttons you can test relays.

NOTE : USB one relay no support "All ON" and "All OFF" buttons



WinExplorer :



http://www.sigma-shop.com/information/USB_Relay/WinExp.zip

http://www.sigma-shop.com/information/USB_Relay/USB_8Relay_test.Xvb

Sample VB6 code :

```
Private Sub cmdOff_Click()  
With MSComm1  
'make sure the serial port is open  
If .PortOpen = False Then .PortOpen = True  
'send the data  
.Output = Chr$(255)  
.Output = Chr$(1)  
.Output = Chr$(0)  
End With 'MSComm1  
End Sub
```

```
Private Sub cmdOn_Click()  
With MSComm1  
'make sure the serial port is open  
If .PortOpen = False Then .PortOpen = True  
'send the data  
.Output = Chr$(255)  
.Output = Chr$(1)  
.Output = Chr$(1)  
End With 'MSComm1  
End Sub
```

Sample C# code :

```
private void button1_ON_Click(object sender, EventArgs e)  
{  
    serialPort1.Write(new byte[] { 0xFF, 0x01, 0x01 }, 0, 3);  
}  
  
private void button1_OFF_Click(object sender, EventArgs e)  
{  
    serialPort1.Write(new byte[] { 0xFF, 0x01, 0x00 }, 0, 3);  
}
```

Linux :

Instructions to install new driver: You may require the sources matching the current kernel to be installed on your system (and built).

To install the ftdi_sio driver use the following steps:

1. Create a temporary folder in your linux machine.
2. Extract the files from ftdi_sio.tar.gz file to your temporary folder
"gunzip ftdi_sio.tar.gz"
"tar -xvf ftdi_sio.tar"
3. Build the driver
"make"
4. Plug in your ftdi device
5. Check to see if default driver was loaded
"lsmod" - you will see ftdi_sio if a driver is loaded
6. Remove the default installed driver
"rmmod ftdi_sio"
7. Install the newly built driver
"insmod ftdi_sio.o"

NOTES:

- 1.This driver was adapted from the 2.4.32 kernel to support both the 2232C and 232R chip
- 2.There is no need to follow this procedure if you want 232R chip support. The 232BM driver will be sufficient.Changes made to the driver for the 232R chip are purely cosmetic (plug/unplug will appear as a 232R chip in the kernel log).

In this case the driver is correctly installed and with the bash's command :

The old kernel's "echo" commands:

```
echo $'\xff\x01\x01' > ttyUSB0  
the relay 1 go ON.
```

```
echo $'\xff\x01\x00' > ttyUSB0  
the relay 1 go OFF.
```

The new kernel's "echo" commands:

```
echo -e '\xff\x01\x01' > /dev/ttyUSB0  
the relay 1 go ON.
```

```
echo -e '\xff\x01\x00' > /dev/ttyUSB0  
the relay 1 go OFF.
```
