

Windows Standard
Serial Communications
Library for Visual FoxPro
Programmer's Manual

(WSC4FP)

Version 7.0

September 10, 2019.

*This software is provided as-is.
There are no warranties, expressed or implied.*

Copyright (C) 2019
All rights reserved

MarshallSoft Computing, Inc.
Post Office Box 4543
Huntsville AL 35815

Email: info@marshallsoft.com
Web: www.marshallsoft.com

MARSHALLSOFT is a registered trademark of MarshallSoft Computing.

TABLE OF CONTENTS

1	Introduction	Page 3
1.1	Features	Page 4
1.2	Documentation Set	Page 6
1.3	Example Program	Page 6
1.4	Installation	Page 7
1.5	Uninstalling	Page 7
1.6	Pricing	Page 7
1.7	Updates	Page 7
2	Library Issues	Page 8
2.1	Dynamic Link Libraries	Page 8
2.2	Key Code	Page 8
2.3	INCLUDE Files	Page 9
2.4	Limitations on COM Ports	Page 9
2.5	Dynamic Strings	Page 9
2.6	Waiting for New Serial Data	Page 10
2.7	SioEvent Logic	Page 10
2.8	FoxPro Forms	Page 10
2.9	Error Display	Page 10
2.10	Virtual Serial Ports	Page 10
2.11	Using 16-bit FoxPro for Windows	Page 10
2.12	64-bit FoxPro	Page 10
2.13	Adding WSC to a VFP Program	Page 10
3	Compiling Issues	Page 11
3.1	Compiling and Linking Programs	Page 11
3.2	Compiling to an Executable	Page 11
4	Example Programs	Page 12
4.1	WSCVER	Page 12
4.2	SIMPLE	Page 12
4.3	SIMPLE2	Page 12
4.4	XMS and XMS	Page 12
4.5	YMS and YMR	Page 12
4.6	FINDER	Page 12
4.7	LISTER	Page 12
4.8	DIALER	Page 13
4.9	DEVICE	Page 13
4.10	SELFTEST	Page 13
4.11	AT_OK	Page 13
4.12	Scale	Page 13
4.13	ReadGPS	Page 13
5	Revision History	Page 14

1 Introduction

The **Windows Standard Serial Communications Library for Visual FoxPro (WSC4FP)** is a toolkit that allows software developers to quickly develop serial communication applications in Visual FoxPro.

The **Windows Standard Serial Communications Library (WSC)** is a component DLL library used to create serial communications programs that access data from a serial port using RS232 or multi-drop RS422 or RS485 ports. **WSC** also supports virtual serial ports using Bluetooth serial and USB to serial converters. The **WSC** component library uses the Windows API for all communication and can be used to easily write applications to control serial devices such as barcode scanners, modems, lab instruments, medical devices, USB serial devices, scales, GPS navigation, etc.

The **Windows Serial Communications Library for Visual FoxPro (WSC4FP)** component library supports all 32-bit versions of Microsoft Visual FoxPro. **WSC4FP** runs under Windows XP through Windows 10 and includes numerous example programs that demonstrate serial port communications functions.

The **Windows Standard Communications Library** DLLs can also be used from any development environment (C/C++, Delphi, Visual Basic, PowerBASIC, Visual FoxPro, Visual dBase, XBase++, COBOL etc.) capable of calling the Windows API.

When comparing the **Windows Standard Serial Communications Library** against our competition, note that:

1. **WSC** is a standard Windows DLL (NOT an OCX or ActiveX control) and is much smaller than a comparable OCX or ActiveX control.
2. **WSC** is implemented as a Win32 DLL.
3. **WSC** does NOT depend on ActiveX or Microsoft Foundation Class (MFC) libraries or similar "support" libraries.
4. The Win32/Win64 versions of **WSC** are fully thread safe.
5. The **WSC** functions can be called from applications not capable of using controls.

MarshallSoft also has versions of the **Windows Standard Serial Communications Library** for Visual C/C++ (**WSC4C**), Delphi (**WSC4D**), PowerBASIC (**WSC4PB**), Visual Basic (**WSC4VB**), Visual dBASE (**WSC4DB**), and Xbase++ (**WSC4XB**). All versions of **WSC** use the same DLLs (**WSC32.DLL** and **WSC64.DLL**). However, the examples provided for each version are written for the specified computer programming language.

The latest versions of the **Windows Standard Serial Communications Library (WSC)** can be downloaded from our web site at

<http://www.marshallsoft.com/serial-communication-library.htm>

Our goal is to provide a robust serial communication library component that you and your customers can depend upon. Contact us if you have any questions.

1.1 Features

Some of the many features of the **Windows Serial Communications Library** are:

- Works on 32-bit and 64-bit Windows through Windows 10.
- Can control any serial device (scale, barcode reader, etc) connected to the serial port.
- Can control up to 256 ports simultaneously.
- Can be used with virtual serial ports using Bluetooth serial or a USB to serial converter.
- Includes 52 functions plus modem control.
- Comes with ANSI emulation and ASCII, XMODEM and YMODEM.
- Supports RS23 and multidrop RS422, and RS485 serial.
- Supports hardware and software flow control.
- Supports any baud rate.
- Ability to specify the parity, word size, and number of stop bits.
- Supports binary and text data transfer.
- State driven Xmodem and Ymodem on multiple ports simultaneously.
- Supports character peek (**SioEventChar**).
- Port re-entrant.
- Is fully threadable.
- Supports transmit and receive timeouts.
- Can send Windows messages on completion of events (incoming character, etc.)
- **Free** technical support for one year.
- License covers all programming languages.
- Royalty free distribution with a compiled application. There are no run time fees.
- Evaluation versions are fully functional. No unlock code is required.
- Supports Windows XP through Windows 10.
- Implemented as a **standard** Windows DLL, which will work with all versions of Windows.
- Is native Windows code but can also be called from managed code.
- Will run on machines with or without .NET installed
- Supports all versions of 32-bit Visual FoxPro through VFP 9.0.
- Does **not** depend on support libraries. Makes calls to core Windows API functions only.
- Can be used with any program (in any language) capable of calling Windows API functions such as Visual C++, Visual C++ .NET, Visual Basic, VB.NET, Delphi, Xbase++, dBASE, COBOL, Access and Excel.
- Can be purchased with or without ANSI C source code to the WSC DLLs.
- Purchase a developer license for WSC4FP and use the DLLs with any other development environment (C++, Visual Basic, etc).
- Updates are **free** for one year (updates to source code are separate).
- Documentation online as well as in printable format.

A good selection of Visual FoxPro example programs with full source code is included. Refer to Section 4.0 for more details on each of the example programs.

[PROGRAM]	[DESCRIPTION]
WSCVER	: Program that displays the WSC version number.
SIMPLE	: A simple terminal emulator.
SIMPLE2	: A simple terminal emulator (alternate version)
FINDER	: Finds a modem connected to one of your serial ports.
LISTER	: Lists all serial ports.
XMS , XMR	: XMODEM programs.
YMS , YMR	: YMODEM programs.
DEVICE	: Program sends string to serial device.
SELFTTEST	: Performs port functionality testing.
AT_OK	: Form sends "AT" and waits for response.
SCALE	: Reads data from a digital scale.
ReadGPS	: Reads NMEA 183 GPS data.

WSC4FP contains 52 functions and modem control. All functions return a negative number if an error condition is detected. For more details, consult the WSC Reference Manual ([WSC_REF](#)) and the RS232/422/485 Serial Communications User's Manual ([SERIAL](#)).

1.2 Documentation Set

The complete set of documentation consists of four manuals in Adobe PDF format. This is the first manual (WSC_FP) in the set.

- [WSC_4FP Programmer's Manual](#) (WSC_4FP.PDF)
- [WSC User's Manual](#) (WSC_USR.PDF)
- [WSC Reference Manual](#) (WSC_REF.PDF)
- [SERIAL User's Manual](#) (SERIAL.PDF)

The WSC_4FP Programmer's Manual is the language specific (Visual FoxPro) manual and provides information needed to compile your programs in a Visual FoxPro programming environment.

The WSC User's Manual ([WSC_USR](#)) discusses language independent serial communications programming issues including modem control. Purchasing and license information is also provided.

The WSC Reference Manual ([WSC_REF](#)) contains details on each individual WSC function.

The Serial Communications Manual ([SERIAL](#)) contains background information on serial port hardware.

The documentation is also provided on our web site at

<http://www.marshallsoft.com/wsc4fp.htm>

1.3 Example Program

The following example program segment sets DTR (Data Terminal Ready):

```
#INCLUDE C:\WSC4FP\APPS\KEYCODE.FOX
#INCLUDE C:\WSC4FP\APPS\WSC32CON.FOX

DECLARE INTEGER SioKeyCode in WSC32.DLL LONG KeyCode
DECLARE INTEGER SioDTR in WSC32.DLL INTEGER Port, INTEGER Cmd
DECLARE INTEGER SioClose in WSC32.DLL INTEGER Port
DECLARE INTEGER SioOpen in WSC32.DLL INTEGER Port,

* pass the key code
if SioKeyCode(WSC_KEY_CODE) < 0
    ? "ERROR: Bad Key Code!"
    return
endif
* open port COM1
Code = SioOpen(COM1)
If Code < 0
    ?"ERROR: Cannot open port"
    return
endif
* set DTR
Code = SioDTR(COM1, ASC("S"))
* close port
Code = SioClose (COM1)
```

Refer to section 4.0 for complete examples with source.

1.4 Installation

(1) Before installation of WSC4FP, your Visual FoxPro compiler should already be installed on your system and tested.

(2) Unzip WSC4FP70.ZIP (evaluation version) or WSCxxxxx.ZIP (for registered users where xxxxx is your Customer ID) using any Windows unzip program.

(3) Run the installation program SETUP.EXE which will install all WSC4FP files, including copying WSC32.DLL to the Windows directory.

Note that no DLL registration is required.

1.5 Uninstalling

Uninstalling WSC4FP is very easy.

First, run UNINSTAL.BAT that will delete WSC32.DLL from your Windows directory, which is normally C:\WINDOWS.

Second, delete the WSC project directory created when installing WSC4FP.

1.6 Pricing

A developer license for WSC4FP can be registered for \$119 (or \$199 with ANSI C source code to the library DLL's). Purchasing details can be found in the WSC User's Manual (WSC_USR), Section 1.3, "How to Purchase" (http://www.marshallsoft.com/wsc_usr.pdf) . Also see INVOICE.TXT provided with the evaluation person or order directly on our web site at

<http://www.marshallsoft.com/order.htm>

1.7 Updates

When a developer license is purchased, the developer receives a set of registered DLLs plus a license file (WSCxxxx.LIC). The license file is needed to download updates to the registered DLL's for a period of one year from purchase. Updates can be downloaded from

<http://www.marshallsoft.com/update.htm>

After one year, your license must be updated if you want to be able to download updates. Your license can be updated for:

- \$33 if the update is ordered within one year of the original purchase (or previous update).
- \$55 if the update is ordered between one and three years of the original purchase.
- \$77 if the update is ordered after three years of the original purchase.

If source code was previously purchased, updates to the source code can be purchased for \$40 along with the DLL update.

Note that the registered DLL's do not expire.

2 Library Issues

The **Windows Standard Serial Communications Library (WSC)** has been tested on multiple computers running Windows XP through (32-bit and 64-bit) Windows 10.

The WSC4FP library works with all versions of Visual FoxPro (VFP 3.0 through VFP 9.0).

The SETUP installation program will copy the DLLs to the Windows directory. Refer to Section 1.4 "Installation". After SETUP is run, the WSC4FP files are copied to the directory specified (default \WSC4FP). Three sub-directories are created, as follows:

```
DOCS - All documentation files
APPS - All example code
DLLS - All DLL's
```

2.1 Dynamic Link Libraries

The WSC4FP serial communication library component includes a Win32 dynamic link library (DLL). A DLL is characterized by the fact that it need not be loaded until required by an application program and that only one copy of the DLL is necessary regardless of the number of application programs that use it. Contrast this to the traditional static library that is bound to each and every application that uses it at link time.

An important advantage that DLLs have over other "popular" library formats such as VBX or OCX is that DLLs are callable by all Windows applications. Since DLLs are the building blocks of the Windows Operating System, they will not be replaced by a "newer technology".

The following files can be found in the DLL sub-directory when SETUP is run:

```
wsc32.dll - Win32 version of WSC
```

2.2 Keycode

Each WSC DLL has a keycode encoded within it. The keycode is a 9 or 10-digit decimal number (unless it is 0), and will be found in the file KEYCODE.FOX. The keycode for the evaluation (shareware) version is 0. The developer will receive a new keycode and a set of new DLL's after purchasing a license. The KEYCODE is passed to **SioKeyCode**.

If you get an error message (value -108) when calling **SioKeyCode**, it means that the keycode in your application does not match the keycode in the DLL. After registering, it is best to remove the evaluation version of the WSC32.DLL from the Windows search path or delete them.

2.3 INCLUDE Files

All example programs include two files: KEYCODE.FOX and WSC32CON.FOX. The file, KEYCODE.FOX, FCE32CON.FOX contains the license keycode as discussed in the previous section. The file, WSC32CON.FOX, contains all the necessary constants for WSC4FP. Similarly, some example programs will contain MIO32CON.FOX and XY32CON.FOX.

Since function declarations cannot be in an INCLUDE file (at least through VFP version 5.0), they are listed in each program following the INCLUDE file or files. The complete list of WSC function declarations is also in file WSC32FUN.FOX.

Due to the behavior of Visual FoxPro regarding INCLUDE files, we recommend handling INCLUDE files in either of the following ways:

- (1) Edit each INCLUDE file path in each program with the physical location of the INCLUDE file.
- (2) Replace each INCLUDE file in each program with the contents (i.e., copy and paste contents) of the INCLUDE file.

2.4 Limitations on COM Ports

WSC4FP works with all real and virtual RS232 / RS422 / RS485 serial ports.

2.5 Dynamic Strings

The Visual FoxPro language use a technique known as "garbage collection" to manage string space at runtime, and may be called internally at any time by the FoxPro runtime, asynchronous to what you may be doing in your code.

When passing a string buffer to a DLL function into which text will be copied, it is strongly recommended that the local string be allocated immediately before use. For example:

```
Code = SioOpen(Port)
if Code < 0
  ? "ERROR " + Str(Code)
  * allocate 'Buffer' immediately before use
  Buffer = SPACE(128)
  Code = SioWinError(@Buffer, 128)
  if Code > 0
    ? Left(Buffer,Code)
  endif
endif
```

This technique is not necessary for passing a string to a DLL function, only when passing a buffer to a DLL into which data is to be placed by the DLL function.

2.6 Waiting for New Serial Data

All serial data is moved from the UART's buffer to the receive queue in memory (by the Windows serial port driver) under interrupt control. Similarly, all out going serial data is moved to the transmit queue in memory.

SioGetc and **SioGets** can be called directly. Note that if there is no input data available, **SioGetc** returns WSC_NO_DATA while **SioGets** returns zero.

2.7 SioEvent Logic

SioEvent, **SioEventChar**, and **SioEventWait** will block (the application will sleep) until the specified event or timeout (**SioEventWait** only) occurs. Descriptions of these functions can be found in the WSC Reference Manual ([WSC REF](#)).

Also see the SIMPLE2.PRG example program that demonstrates the use of the **SioEventWait** function.

2.8 FoxPro Forms

WSC functions can be called from any Visual FoxPro code module, such as programs, classes, and forms. See the AT_OK.SCX example form.

2.9 Error Display

All WSC functions return an integer code. Error conditions always have negative return codes as listed in WSC32CON.FOX (in the APPS sub-directory) and wscErrors.txt (in the DOCS sub-directory). Each sample program contains examples of error processing.

2.10 Virtual Serial Ports

A “virtual” serial port is COM port that appears to be a real RS232 serial port to the Windows API (and thus to WSC), but is in reality a COM port emulator.

The two most common virtual ports are those created for USB/serial port converters and Blue Tooth. WSC will work with most USB-to-serial port converters and with Bluetooth serial.

More information about Virtual serial ports can be found in Section 2.12 of the WSC User's Manual (WSC_USR). (http://www.marshallsoft.com/wsc_usr.pdf).

2.11 Using 16-bit FoxPro for Windows

Support for Win16 was dropped beginning with version 5.2. Version 5.1 is still available (free when purchasing the current version) for those wanting Win16 support.

2.12 64-bit FoxPro

WSC4FP will support 64-bit (Win64) FoxPro if it is released by Microsoft. The current version of FoxPro (9.0) creates 32-bit applications, not 64-bit applications. Of course, you can run 32-bit FoxPro (using WSC32.DLL) on 64-bit machines.

2.13 Adding WSC to a VFP Program

- (1) - Add the WSC constants (found in WSC32CON.FOX) that you will use in your program.
- (2) - Add the WSC function declarations (found in WSC32FUN.FOX) that you will call to your program.

3 Compiling Issues

3.1 Compiling and Linking Programs

The example programs are compiled from the Visual FoxPro development environment. Before running the example programs edit each example program with your serial port parameters as shown in the example program in Section 1.3 above.

3.2 Compiling to an Executable

FoxPro programs end in ".PRG". They can be compiled to an executable using the FoxPro BUILD command.

For example, to create WSCVER.EXE from WSCVER.PRG in the C:\WSC4FP directory, type the following in the FoxPro command window:

```
BUILD PROJECT C:\WSC4FP\APPS\WSCVER FROM C:\WSC4FP\APPS\WSCVER
BUILD EXE C:\WSC4FP\APPS\WSCVER FROM C:\WSC4FP\APPS\WSCVER
```

FoxPro executables require VFP500.DLL and VFP5ENU.DLL (ENglish User), and may have to be copied from the VFP CDROM. If you are using an earlier or later version of FoxPro than version 5.0, substitute the appropriate DLL's for the above.

The FoxPro output display window will disappear as soon as your executable completes. In order to allow the user to control when the display window disappears, add the following code to your application, just before the final return.

```
? " Type any key to exit..."
X = InKey(0)
```

4 Example Programs

Each of the following example programs is written for 32-bit FoxPro.

4.1 WSCVER

The WSCVER ("WSC Version") example program displays the WSC version number. This is the first program to compile and build since it verifies that WSC32.DLL is installed properly.

4.2 SIMPLE

SIMPLE is a very simple communications program using WSC4FP. Everything that is typed on the keyboard is sent to the serial port, and everything incoming from the serial port is displayed on the screen.

The easiest way to test SIMPLE is to connect to a modem. Typing 'AT' should result in an 'OK' being displayed.

A null-modem cable can also be used to connect two computers together with their serial ports. Run SIMPLE on both machines. Whatever is typed on one machine will be displayed on the other.

4.3 SIMPLE2

SIMPLE2 is the same program as SIMPLE, except that it calls **SioPuts** rather than **SioPutc**, and **SioGets** rather than **SioGetc**. **SioPuts** and **SioGets** send and receive strings instead of characters.

4.4 XMS and XMR

XMS (XMODEM Send) and XMR (XMODEM Receive) are programs that send and receive files using the XMODEM protocol.

4.5 YMS and YMR

YMS (YMODEM Send) and YMR (YMODEM Receive) are programs that send and receive files using the YMODEM protocol.

4.6 FINDER

The FINDER program is a program that searches for a connected modem. Your modem must be connected to one of COM1 through COM4, and the modem must be turned on. Note that FINDER uses the MIO module.

4.7 LISTER

The LISTER program lists all serial ports.

4.8 DEVICE

The DEVICE program is designed for talking to an arbitrary serial device. Use this program as a guide when communicating with all serial devices other than modems and other computers.

4.9 SELFTEST

SELFTEST performs basic serial port I/O functionality testing using a loopback adapter. Refer to LOOPBACK.TXT for an explanation of how to make a loopback adapter (without tools!).

4.10 AT_OK

The AT_OK example form sends the string "AT" (followed by a carriage return) to COM1 and waits a short time for a response.

From the VFP menu (File/Open), open the form AT_OK.SCX (with "File of Type: Form"). When the "Form" menu tab appears on the VFP menu bar, choose "Run Form". This form can also be opened from the VFP command window by typing "modify form \wsc4fp\apps\at_ok.scx".

4.11 SCALE

The Scale example program transmits an ASCII command string to a scale, then reads incoming (response) data until a 100 ms data pause (Quiet) is detected.

4.12 ReadGPS

The ReadGPS console mode program reads lines from a device that is outputting NMEA 183 GPS sentences, although this program will read complete lines from any serial device that outputs such lines.

5 Revision History

The WSC DLL (WSC32.DLL) is written in ANSI C. All language versions of WSC (C/C++, Delphi, Visual Basic, PowerBASIC, Visual dBase, Visual FoxPro, Xbase++, and COBOL) use the same identical DLL.

Version 3.0: August 14, 2000.

- The initial release of the Visual FoxPro version of WSC.

Version 3.1: May 24, 2001.

- RESETDEV Win API call not called (allows USB/serial converters).
- SioPutc and SioPuts return immediately (optionally).
- XYM (XMODEM/YMODEM) allows local upload/download directory to be specified.

Version 3.2: August 20, 2002.

- Added SELFTEST example program.
- Default for RESETDEV is "not called". SioDebug(ASC("R")) to enable.
- SioGetc & SioGets zero unused bits (DataBits 5,6,7).
- Corrected problem with SioBaud(-1, BaudRateCode).
- SioDebug returns -1 if no match.
- Added SioDebug(ASC("W")) toggle SioPuts wait for I/O completion.
- Added code to detect active threads & to close thread handles.
- Added USE_THREADS, so can compile version of WSC32.C without threads.
- Comm handle not saved in SioReset unless it is good.
- SioEvent returns mask that caused the event.
- Added SioInfo(ASC("B")) to get build number.

Version 4.0: January 7, 2004.

- Can now order either with or without source code to the DLLs.
- Added SioSetInteger function to set port specific integer parameters.
- Added SioKeyCode function to pass the key code to the DLL.
- Added SioGetReg function to return the registration string.
- Added "Burst Size" parameter for setting the TX burst size.
- Added ability to signal blocked thread that was blocked by SioEvent.

Version 4.1: August 12, 2004

- Fixed problem with SioTxClear.
- Added overlapped I/O (for non-Win95) so can signal threads to exit w/o killing them.
- Increased default burst size to 256.
- SioFlow returns WSC_RANGE if cannot recognize parameter.
- Adjusted XModem/YModem timing for faster transfers.

Version 4.2: March 29, 2006.

- SioFlow returns 1 if OK.
- SioSetInteger(Port, 'S', 1) always forces SioEvent to unblock.
- Event mutex code added to EventThread() to prevent race conditions.
- Message box displays error if SioWinError(Buffer, 0) called.
- Major change in overlapped I/O
- Fixed problem: SioEvent returning wrong code.
- SioRxClear clears byte saved by SioUnGet.
- Number of supported ports increased to a maximum of 256.
- Added SioEventChar() and SioEventWait() functions.

Version 4.3: October 5, 2007.

- Fixed problem with SioTxQue returning wrong values.
- Changed SioParms so it checks the range of passed arguments.
- Port is verified in SioEventChar.
- SioStatus returns -1 if port is not functioning (USB/serial port disconnected).
- Added SioByteToShort and SioShortToByte (WSC32 only).

Version 4.4: January 30, 2009

- Added SioTimeout() function (sets TX and RX time-outs).
- Provide documentation files in Adobe PDF format.

Version 5.0: December 2, 2009

- Supports 64-bits
- Added SioHexView function
- Added SioSleep function

Version 5.1: September 8, 2011

- Added SioRxWait function
- Added LISTER example program

Version 5.2: July 10, 2012

- Added function SioQuiet()
- Added function SioWaitFor()
- Added example program ProXR.prg

Version 5.3: November 6, 2013

- Added SioLRC() that computes the "longitudinal redundancy check" per ISO 1155.
- SioQuiet() and SioWaitFo() verify the passed port number.
- SioWaitFor() verifies that the passed baud rate is > 0.
- SioSetInteger() no longer requires an open port for global (all ports) parameters.
- Modified SioReset() to make it more tolerant opening slow virtual ports.

Version 5.4: August 26, 2015

- Added SioCRC16() that computes 16-bit CCITT CRC (polynomial 1021 hex).
- Added SioCRC32() that computes 32-bit CCITT CRC (polynomial 04C11DB7 hex).

Version 6.0: March 22, 2017

- Added additional error codes: WSC_BUFFER_RANGE, WSC_BUFLLEN_RANGE, WSC_BAD_CMD
- Added additional error codes: WSC_BAD_PARITY, WSC_BAD_STOPBIT, WSC_BAD_WORDLEN
- Added SioErrorText() that returns text associated with specified error codes.
- Added SioPortInfo() that returns baud in BPS (bits per sec) and the theoretical port CPS (char per sec).
- Added SioGetsC() that receives an entire line through the stop (end-of-line) character (usually CR).
- Added ReadGPS example program.

Version 7.0: September 10, 2019

- Fixed: SioErrorText() now returns text length from call to SioWinError()
- Fixed: SioGets() would never timeouts when overlapped I/O was enabled.
- Added: function SioOpen - same as SioReset(Port, 1280, 1280)
- Added: function SioClose - same as SioDone.
- Added: function SioGetsQ - reads port until no incoming data for specified "quiet" time.
- Added Scale.prg program..