

Appendix: Winch Telemetry to RigWatch

Recapitulation and adaptation of prior work in *Schlumberger Winch Data transfer to RigWatch (SLB to RW)*.

Operations Rationale

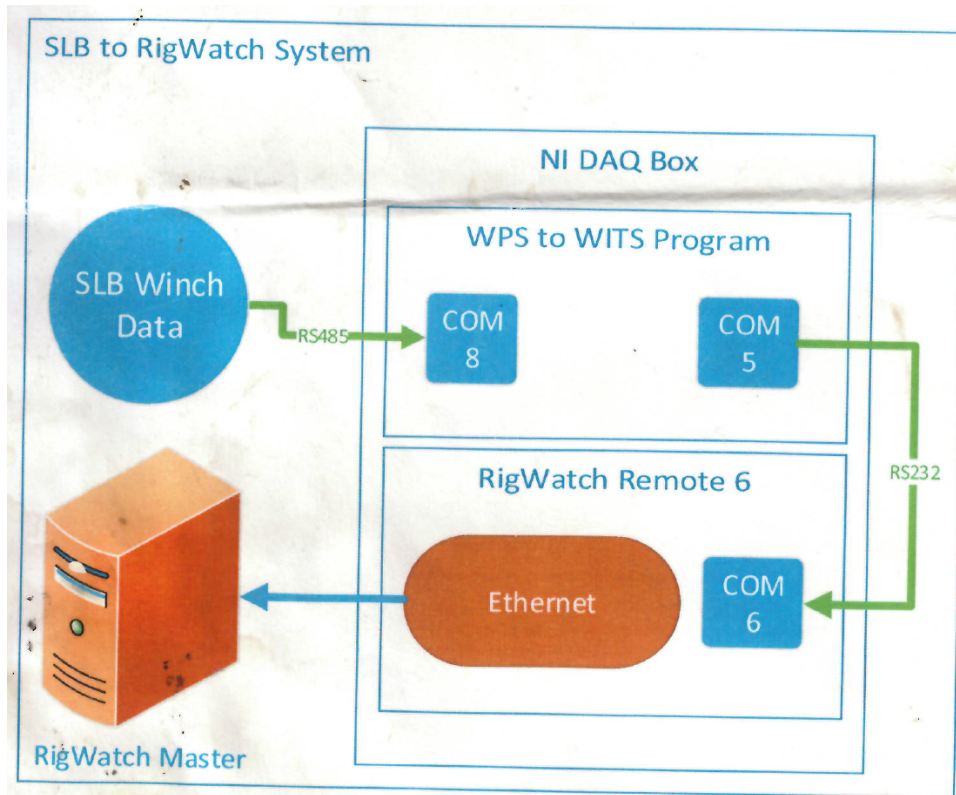
Tool string depth and tension are readily consumable gauges of the progress and status of logging operations. During logging operations, JRSO convention widely distributes Schlumberger winch position data via RigWatch displays.

System Overview

Schlumberger provides JRSO with winch position telemetry. Software systems have (repeatedly) been implemented to make that data more widely available. The present "SLB to RW system" converts Schlumberger winch position data to WITS. The WITS data is passed to RigWatch.

SLB to RW is presently hosted on a Dell laptop. The software system processes WPS telemetry through three serial devices:

- COM4 - Input. Winch position telemetry is received via an RS485 serial connection from the Schlumberger rack.
- COM3 - Output. "SLB to RW" formats incoming winch telemetry (COM4) to WITS. The WITS stream is forwarded to COM3 (RS232).
- COM1 - Output from "SLB to RW". Input WITS to RigWatch.
 - COM3 is physically connected in loopback fashion to COM1. The loopback cable is known as a null-modem or cross-over cable.
- During operations the RigWatch Master receives WITS-formatted winch telemetry--across the network--on device "COM1 - Remote 6".
Rephrased: the RigWatch instance on the Dell laptop (i.e. Remote 6 or 192.168.1.11) passes COM1 data to the RigWatch Master--only when the Master is configured to listen for that device.



This diagram accurately depicts the flow and handling of winch position telemetry through the present Dell laptop (and any subsequent system that may be put in place). The COM port numbers in the diagram are inaccurate and superseded by the text above.

Starting the Winch Telemetry Laptop

Cabling review

As-delivered, the COM-port wiring is placed as indicated

- COM4 is a National Instruments device RS-485 to USB.
 - The RS485 connector is tied to the Schlumberger WPS output.
 - The USB connector ties to the (only) left-hand-side (toward the back) USB connector.

- COM3 is provided via a Tripplite Keyspan USB to Serial RS232 device.
 - The USB cable ties to the (only) laptop backplane (right-hand-side) USB connector.
 - The 9-pin RS232 connector attaches to a null-modem cable.
 - The null-modem cable attaches to the only 9-pin RS232 connector on the laptop backplane (left-hand-side).

Power up

1. A single deliberate press of the laptop power button suffices (upper-right of the built-in keyboard).
2. Login as DAQ. The usual. Wait, up to a minute.
 - SLB to RW is run automatically.
 - RigWatch is run automatically.
 - Windows profile startup processing takes precedence. It may be 30 sec - 1 min before SLB2RW and RW start.
 - SLB2RW will typically launch faster than RW.
 - If the Schlumberger winch feed is on, SLB2RW will begin displaying winch telemetry within 3 seconds.

Because logging occurs at such wide intervals, it is best to catch the RigWatch startup before it defaults to the previously used hole. Choose "Start New Well", register to reconnect to the Master (192.168.1.7), and then continue RigWatch startup.

Notes

- The DAQ profile is configured for Windows Remote Desktop access via 192.168.1.11.
- SLB to RW is installed in C:\Users\daq\Documents\WPS to WITS - SLB to Rigwatch: SLB2RW.exe
- RigWatch is installed in C:\RWPRO. The executable resides at C:\RWPRO\Tools\interfaces\rw.exe.
- The shortcuts used to launch both programs reside in C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup\.

Glossary

WPS - Winch Positioning System. The collective hardware and software.

WPD - Winch Position Data. Core winch position data that may be retrieved from Schlumberger systems via specific communications protocols.

WITS - Wellsite Information Transfer Specification.

Configure the RigWatch Master to receive WITS telemetry from RigWatch COM1 - Remote 6

WITS setup - at Krakatoa, not at Remote 6

An "SLB WITS Setup" should already be saved in the current RW setup file. If so, skip to "Running SLB to RW".

1. Run RigWatch as usual.
2. Click Menu
3. Click the red "Setup" button
4. Click "Device Config" and enter the setup password
5. Click "Device Config" again
6. Scroll to "WITS Device 1" aka "wits1.dll" and select it
 - a. If there is no "WITS Device 1", click in a "(no tool)" box then click "Add +"
 - b. To add "WITS Device 1" scroll to near the bottom of the list and select "wits1.dll"
7. Set the following options
 - a. Set Communications Port to "COM1 - Remote 6"
 - b. Set Timeout Count to "10" (this is the default)
 - c. Set speed to "9600/N/8/1" (these are the defaults)

WITS variable setup - at Krakatoa, not at Remote 6

1. Run RigWatch as usual.
2. Click Menu
3. Click the red "Setup" button
4. Click "Variables" and enter the setup password
5. Click "Variables" again
6. Use the "Page Down" button to scroll to the SLB Winch Variables
7. Confirm or set the following for each SLB variable

Variable Number	Name	Input Device	Input Device Slot
28	SLB Cable Depth	WITS Device1	0921 [MWDm] <Spare 1>
29	SLB Cable Tension	WITS Device1	0922 [MWDm] <Spare 2>
30	SLB Cable Speed	WITS Device1	0923 [MWDm] <Spare 3>

8. To change "Input Device", click on "Input Device", then scroll to and select "WITS Device 1"

9. To change "Input Device Slot", click on "Input Device Slot" and scroll to and select the appropriate WITS tag number. The number is the important part, but if you want to edit the label portion of the tag, refer to the RigWatch Manual.

Manual startup of RigWatch, SLB2RW

RigWatch and SLB to RW software may be launched from desktop or task bar shortcuts. Look for the following.

SLB to RW

(If RigWatch is in the way, minimize it by clicking "Key Pad", then "Minimize Rigwatch".)

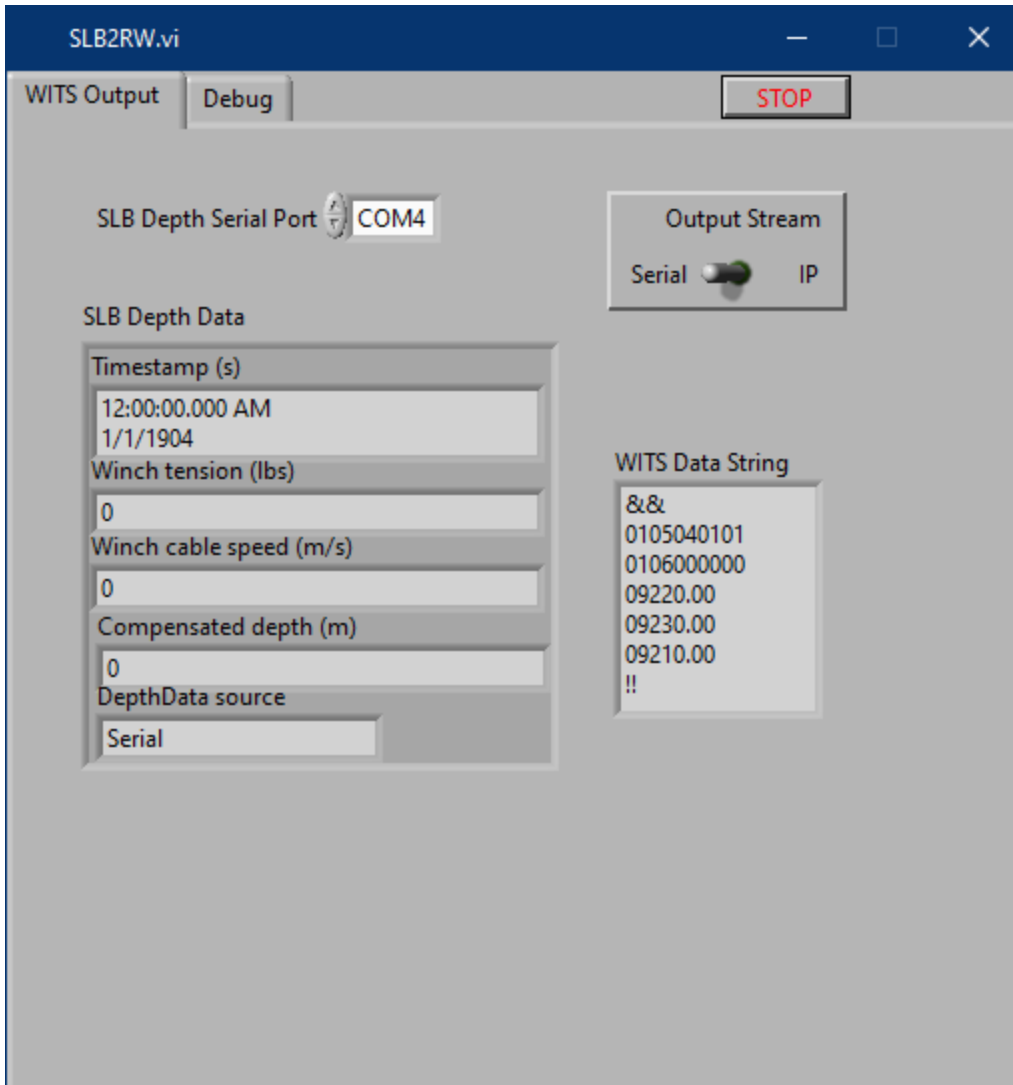
Find SLB2RW, click.



Reviewing the (right-click) Properties of the shortcut will indicate where the software is located on disk.

The SLB2RW main window looks like this.

- If the winch telemetry is enabled, the data will update approximately once per second.
- "SLB Depth Serial Port" is COM4.
(Changing it is not effective without also re-arranging the serial cables and recompiling the program.)
- "Output Stream" is Serial.
(The network IP address option is not implemented.)



The "WITS Data String" reflects the "SLB Depth Data" but with tags applied to adhere to the specification.

WITS Tag	Variable
0105	Date
0106	Time
0922	Winch Tension
0923	Cable Speed
0921	Depth

RigWatch

(Verify that RigWatch is not previously minimized by Alt-TABing through already opened programs.)

Find RigWatch, click.



Reviewing the (right-click) Properties of the shortcut indicates where the software is located on disk.

The RigWatch main screen will fill the entire display, e.g.

Rigwatch™ Location: N. Atlantic
Well ID: U1564A/SetUpFile:Winch position telemetry to Rig
Proposed Direction: 0.00
Total Correction: 0.00
Aug 3, 2021 10:18:49
Chat Drill Mode Mute Alarm Key Pad ? Click Lock

Pipe Counter 93 <small>stands</small>	Depth - Hole 2808.2 <small>mbrf</small>	Depth - Bit 2808.2 <small>mbrf</small>			On / Off Bottom Yes <small>on bott</small>
Depth - Bit 2808.2 <small>mbrf</small>	Hook Load 389 <small>klbs</small>	HEAVE (m) -.1 <small>m</small>	AD Hook Load 11 <small>WOB klb</small>	Pulling Speed -14 <small>M/Hr</small>	Slips Set No <small>on/off</small>
Depth Bit BSF 588.7 <small>mbrf</small>	Tracer ON/OFF CMD Off <small>on/off</small>	PITCH .3 <small>deg</small>	Bit On Bottom Hours 23:37:18 <small>hours</small>	ROLL .1 <small>deg</small>	
Depth Sea Floor 2219.5 <small>mbrf</small>	Mud pump #1-SPM 0 <small>spm</small>	Mud pump #2-SPM 65 <small>spm</small>	Mud Pump #2 Press. 1138 <small>psi</small>	Pumps Total 8 <small>bbbs</small>	TD RPM 62 <small>rpm</small>
SLB Cable Depth 0 <small>mbrf</small>	Mud pump #1 STKS 47403 <small>stks</small>	Mud pump #2 STKS 127685 <small>stks</small>		Stand Pipe Pressure 1100 <small>psi</small>	
SLB Cable Speed 0 <small>ft/hr</small>	SLB Cable Depth 0 <small>mbrf</small>	SLB Cable Tension 0 <small>lbf</small>	SLB Cable Speed 0 <small>ft/hr</small>	(v45) 2 <small>m3</small>	(v68) 9.9 <small>m3</small>
SLB Cable Tension 0 <small>lbf</small>		Depth Sea Floor 2219.5 <small>mbrf</small>	MUD Pmps TOT SPM 65 <small>spm</small>	MUD Pmps TOT STKS 4375 <small>stks</small>	
MENU F8					

Rigwatch™ Total Correction: 0.00
Toolface Offset: 0.00
Security: Lockdown
Contractor: SIEM
Operator: LODP-JRSO
Aug 3, 2021 10:17:39
Chat Drill Mode Mute Alarm Key Pad ? Click Lock

Pipe Counter 93 <small>stands</small>	Depth - Hole 2808.1 <small>mbrf</small>	Depth - Bit 2808.0 <small>mbrf</small>			On / Off Bottom Yes <small>on bott</small>
Depth - Bit 2808.0 <small>mbrf</small>	Hook Load 389 <small>klbs</small>	HEAVE (m) -.1 <small>m</small>	AD Hook Load 11 <small>WOB klb</small>	Pulling Speed -19 <small>M/Hr</small>	Slips Set No <small>on/off</small>
Depth Bit BSF 588.5 <small>mbrf</small>	Tracer ON/OFF CMD Off <small>on/off</small>	PITCH .5 <small>deg</small>	Bit On Bottom Hours 23:36:08 <small>hours</small>	ROLL -.1 <small>deg</small>	
Depth Sea Floor 2219.5 <small>mbrf</small>	Mud pump #1-SPM 0 <small>spm</small>	Mud pump #2-SPM 65 <small>spm</small>	Mud Pump #2 Press. 1143 <small>psi</small>	Pumps Total 8 <small>bbbs</small>	TD RPM 62 <small>rpm</small>
SLB Cable Depth 0 <small>mbrf</small>	Mud pump #1 STKS 47403 <small>stks</small>	Mud pump #2 STKS 127610 <small>stks</small>		Stand Pipe Pressure 1109 <small>psi</small>	
SLB Cable Speed 0 <small>ft/hr</small>	SLB Cable Depth 0 <small>mbrf</small>	SLB Cable Tension 0 <small>lbf</small>	SLB Cable Speed 0 <small>ft/hr</small>	(v45) 2 <small>m3</small>	(v68) 9.9 <small>m3</small>
SLB Cable Tension 0 <small>lbf</small>		Depth Sea Floor 2219.5 <small>mbrf</small>	MUD Pmps TOT SPM 65 <small>spm</small>	MUD Pmps TOT STKS 4300 <small>stks</small>	
MENU F8					

Source Code for SLB to RW

Local copy

- C:\Users\daq\Documents\WPS to WITS - SLB to Rigwatch\Vis 2015\ Double-click on SLB Winch Reader.lvproj to launch the LabVIEW 2020 development environment.

- The operational executable (when built from the local source code set) is placed here
C:\Users\daq\Documents\WPS to WITS - SLB to Rigwatch\SLB2RW.exe
with a few other files.

Network copy

- U: or \\CLEVELAND\VOL2, then \2-Engineering\2 RIS_RigWatch\RIS SOFTWARE\SLB to RW Software\SLB2RW-2021-CODE
- A copy of the executable is stored in the same location: SLB2RW-2021-EXE

Theory of Operation

The Read Depth Loop

The code reads a binary data stream from the serial port, finds the WPD packets, then extracts the interesting fields from them. The WPD packets are identified by transmission pauses before and after. Complete packets are 54 bytes long; packets which do not meet this gross integrity check are thrown out. Since the protocol is time critical, it is crucial for the code to check the serial port at regular intervals. This is achieved by using a Timed Loop which has high timing reliability (much better than a "plain" software loop). In VI Info (Ctrl+I from the LabVIEW development environment), the VI's Execution property is set to the highest possible for the operating system and hardware in use. Historically "time critical" was specified; with current hardware and software "high priority" is sufficient. The loop period is tuned to 50 ms, 1/4 of the expected 200 ms pause, so the pause can be clearly identified.

Algorithm

A state machine is used which looks for these 3 conditions.

- If the state machine spends more than 10 seconds in any of these states it will timeout, raising a warning.
- If the timed loop takes greater than 100 ms or less than 10 ms per iteration, then synchronization with the WPD timing may have been lost. If this occurs, the state machine goes straight to the *FindBreak* state. Any partially received packet is discarded.
- When a new packet is successfully read, the "New Data" flag is set "True" for 50 ms. This signals other VIs to take action. The new data is written to a global.

**** FindBreak **** - The starting state. Attempts to synchronize with the data stream.

- Wait for the pause in transmission which signals the gap between packets.
- Begin decoding a packet.
- Read and discard all bytes until one whole loop period (50 ms) expires without any bytes being received.
- Then processing is handed off to state *FindStart*.

****FindStart ****

- Wait for bytes at the serial port.
- Every 50 ms observe if any bytes have been received.
- When bytes are received, records the start time.
- Processing is handed off to state *Read*.

**** Read ****

- Accumulate bytes read until one whole 50 ms period passes with no bytes received. This signals the end of the packet.
- If there are 54 bytes in the packet, process it to extract Depth, Tension, etc. (data)
- The data is placed in a queue.
- Processing is handed off to state *FindBreak*.

The code could be upset if illusory pauses in reception are seen due to delays between when characters come in to the serial port and when VISA's "Bytes at Port" function registers them. The code might then identify these as packet delimiting pauses. The 54-byt length check will almost always catch this, but there is a slim chance that 2 delays exactly 250 ms apart might create a bogus packet read. Once possible cause might be a large FIFO buffer (>54 bytes) on the serial hardware. It is recommended to turn off the FIFO on the WPD serial port (Device Manager > Serial Ports > COM4: right-click Properties, Advanced...). This change requires a restart of the host to take effect.

While this algorithm has run stably for years, it is tuned to the rate at which the Schlumberger hardware presents data. If Schlumberger equipment changes, this code will likely need to change.

Recapitulation of the Schlumberger WPD protocol specification

Depth data is transmitted from Schlumberger to our system over an RS422 serial line using the WPD protocol. The protocol is documented in Schlumberger's SSD Maintenance Manual, Appendix C. This is an old standard. The version in use on the JR is slightly different than what is documented. A summary specification of the JR WPD protocol variant is as follows.

Data packets are

- 54 bytes long
- transmitted 4 times per second at 19200 baud, no parity, 8 data bits
- the packets are delimited by a pause in transmission of ~200 ms
"This timing critical protocol is a real pain to decode reliably."

The interesting parts of the data packet are

- Byte 0 - Flags. Bit 5 is units: 0=feet, 1=meters. JR usage has always been in feet.
- Bytes 3, 4, 5 - Depth. 24-bit 2's complement number. Specified in 0.1-inch units or an unspecified fraction of meters.
 - Byte 3 is LSB.
 - Byte 5 is MSB.
- Bytes 7, 8 - Speed. 16-bit unsigned integer. Units of ft/hr or m/hr, depending on byte 0.
- Bytes 12, 13 - Tension. 16-bit unsigned integer. Unit of pounds.

Software Dependencies

SLB to RW

SLB2RW is a compiled, 64-bit, LabVIEW 2020 executable.

- To communicate with the NI RS485 to USB device established at COM4, National Instruments (NI) Serial drivers are required.
- The SLB2RW executable requires the LabVIEW Runtime Environment (LVRTE) 2020 to operate.

The full LabVIEW development environment that is installed on this system is not required for operation of SLB2RW. The development environment is solely for the convenience of rebuilding the tool if the serial ports are changed.

RigWatch

The RigWatch installation is dependent on

- 1-Wire iButton drivers appropriate to the Windows environment. Stored in the JRSO software repository.
- NTVDM. A software layer to eke out more years of software compatibility for old Microsoft technologies: new technology virtual DOS machine.

RigWatch is not dependent on an external installation of Oracle's Java. It is better for RigWatch to run only with the components that its installer provides. This ensures that, for example, wits1.dll and other serial drivers will load and communicate effectively across the network with the Master station.

PuTTY

Provides the ability to communicate with serial ports. Provides an independent testing facility if operational tools are not working as expected.

Troubleshooting

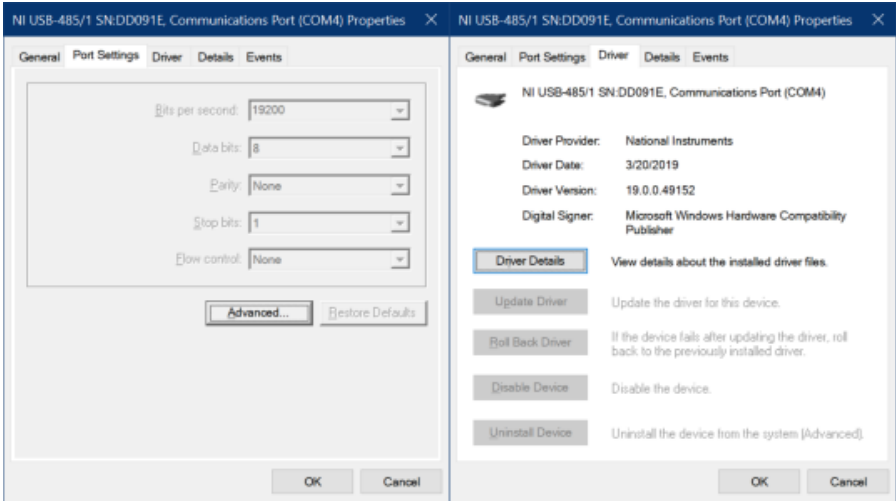
Normal operational state

Windows Device Manager normally shows these devices. This view was captured with "View > Show hidden devices" checked.

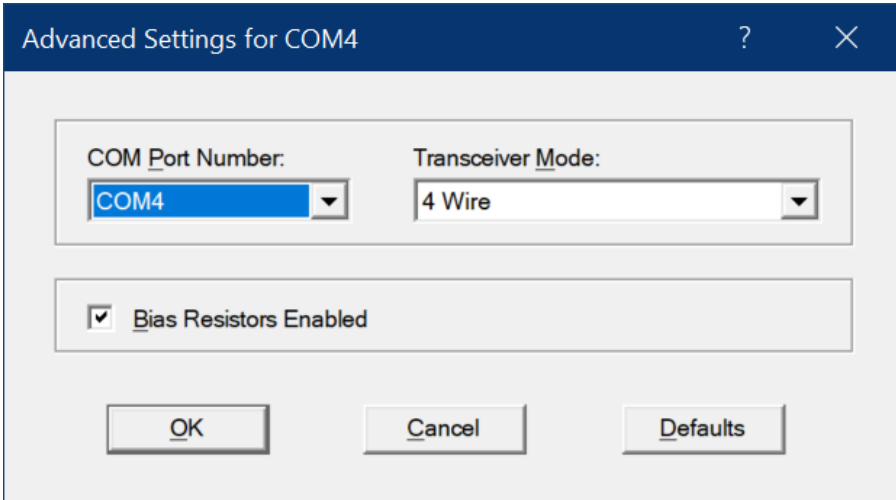


These are the device configurations as of this writing and implementation. Note that a different driver supports each serial port established.

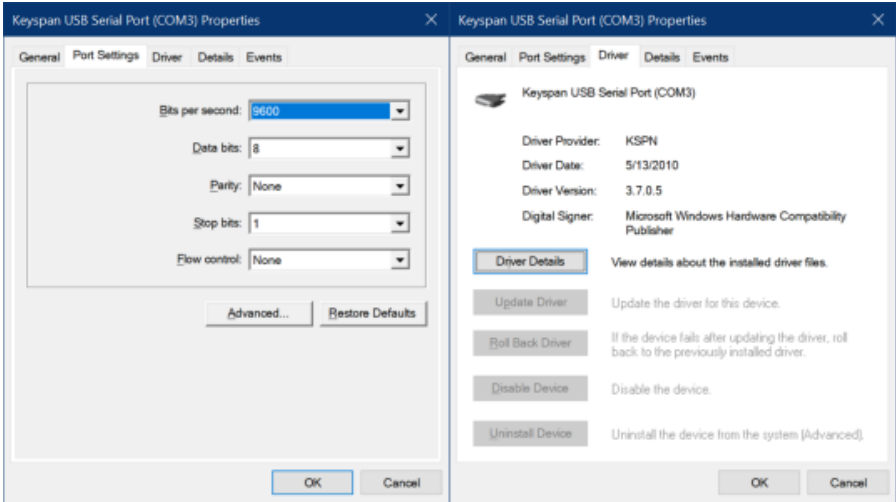
COM4 - Schlumberger winch position data - incoming



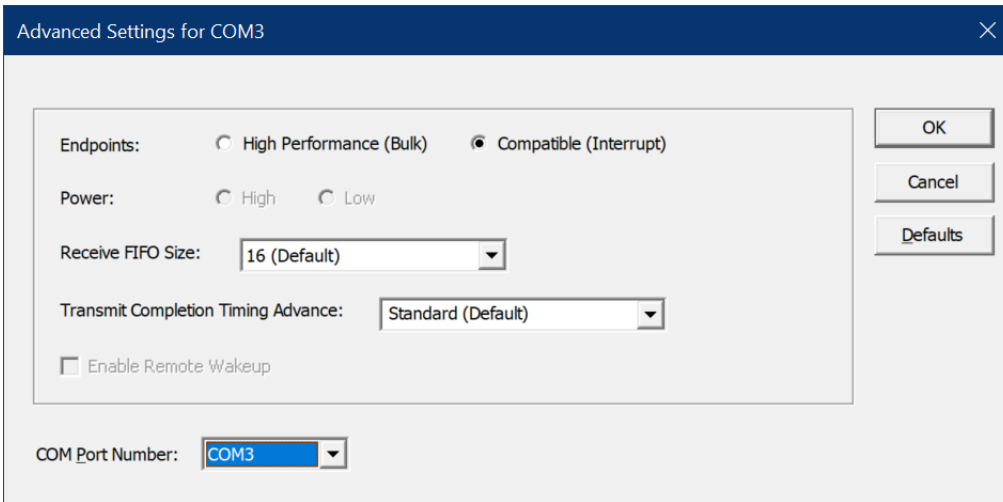
This dialog is obtained from the Advanced... button above. The settings reflect the defaults applied when the NI serial drivers were installed for the RS485 device. There is no need at this time to change them.



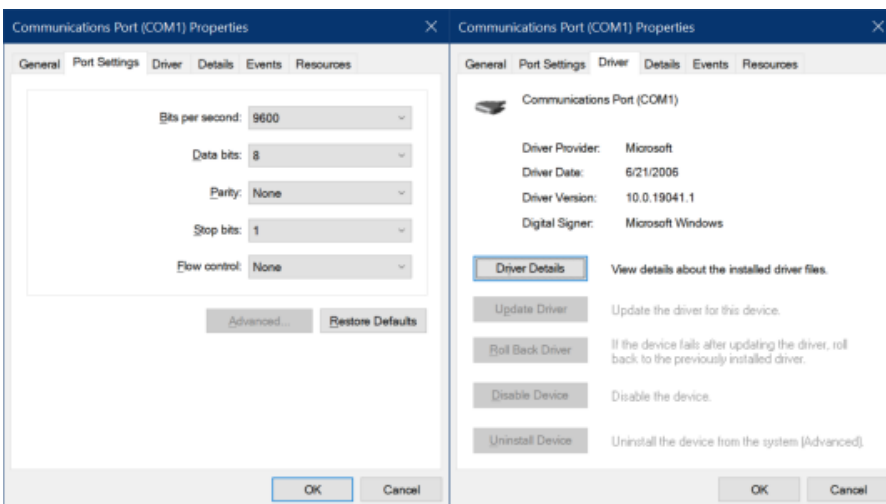
COM3 - SLB2RW program outputs WITS content to this serial port



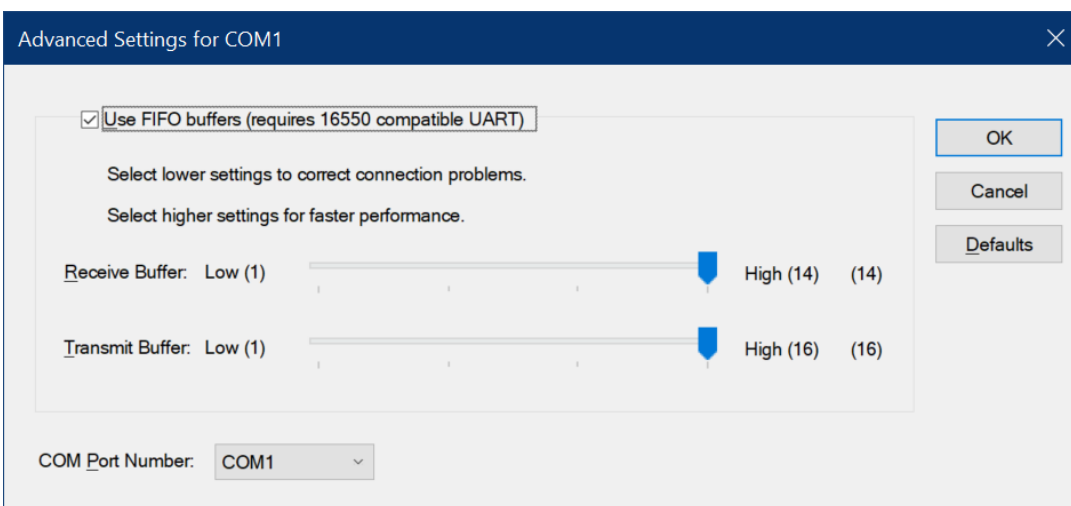
This dialog is obtained from the Advanced... button above. The settings reflect the defaults applied when the drivers were applied for the Triplite Keyspan USB-to-Serial. There is no need identified at this time to change them.



COM1 - device COM3 is physically looped back into the computer



This dialog is obtained from the Advanced... button above. The settings reflect the defaults applied by the Microsoft-provided serial driver. There is no need identified at this time to change these settings.



If there is a change in USB-to-serial equipment

Drivers must be changed. If the drivers are changed, the designated COM#'s may too.

- Be sure to find and apply the current USB-to-serial driver for the device.
- Modern Windows (10 and above) driver Properties present an Advanced... button in their interfaces.
 - Typically it presents FIFO queue configuration options AND a field to assign the COM#.
 - Recommend changing the COM# to those specified at the beginning of this document OR recompile SLB2RW to reflect the new ports, revise this document, and retrain both crews.

(As seen above) the "Advanced Settings" for COM ports allows an administrative account to change the assigned COM port number. Re-assign COM ports sparingly. Each change breaks communications for current operations, and requires a restart of the host computer.

PuTTY serial communications

PuTTY may be used to verify transmission of WITS data to COM1.

Nota bene: while PuTTY has a lock on the serial port, RigWatch Master cannot see or use device "COM1 - Remote 6".

If the NI RS485 to USB device is unplugged

The Windows Device Manager will disable the serial driver. This will interrupt the telemetry stream to RigWatch.

These recovery methods have not been tested. TBD. Each item is one step along the communication path.

- Best case. Plug the USB to RS485 (or other serial device) back in and it picks right up again.
- Middle of the road.
 - SLB2RW program must be shutdown and restarted to resynchronize with the WPS feed.
 - Possibly the instance of RigWatch at Remote 6 would need to be restarted.
- Worse case. The whole winch telemetry host must be shutdown and restarted.
- Worst case. Even if the winch telemetry host is working as expected after restart, it is possible that a "hang-up" and "re-connect" at the RigWatch Master may not work (seen once).

That "seen once" was corrected by a restart of RigWatch. Not a desirable activity during rig-floor operations and continuous data collection.

Change

20210804 df Smoke test. Last logging run of the expedition. TBD.

20210803 df Adapt *Schlumberger Winch Data transfer to RigWatch System (SLB to RW)* document to this form.

20210729 df WPS feed not working. See what can be done to make this more repeatable, stable.

- Last bit: restart RigWatch Master. It was insufficient to "hang-up" and reconnect the "COM1 - Remote 6" device. Full system observed to operated end-to-end. But once again only for the latter portion of a logging cycle.
- Run a repair install of NI Serial. Some event (unplug?) caused the driver to uninstall.
- Observed that COM1 and COM2 designations in software were swapped with respect to the physical configuration. Re-assigned the serial ports using the Device Manager > Properties > Advanced... settings.
- Conducted a variety of tests to gauge the effects of serial port driver removal, device disable/enable, and COM port renumbering. Every variant requires a host system restart.

20210720 df Delivered winch telemetry over the Dell 53043. Demonstrated workable for the remainder of that logging cycle.

20210718 df, nl Commit to rebuilding winch telemetry on new hardware. Selected the ruggedized Dell (53043), an NI RS485 USB-to-serial device, and a Triplite Keyspan USB-to-serial device.

- Repeated install and rollback attempts with the NI PXI-8106 embedded controller led to the realization that its on-board network controller is failing.
- The device has been carried forward as a one-off without spares for some time.