DIGITAL 24-CHANNEL 'HYDRA'

MODULAR RAKE SYSTEM

User Manual

WARNING

Read this document before using the product.

This system is not certified for use on aircraft.

Contents

Version Control

Version	Date	Summary of changes	Software release
1.0	11-2025	New document	1.0
1.1	12-2025	Updated data conversion	1.1

1 INTRODUCTION

Principle of operation

This rake system includes a 24-element pressure-based directional velocity rake. The rake system also contains a six-component inertial measurement unit and one or more fluid temperature sensors (which may be built into the rake sting). The rake system can be configured to stream data over a serial data line once powered on, but may also be used together with a computer using the on-board USB terminal.

System description

Integrated modular digital multi-function rake system.

System components

1x Modular rake system assembly	Modular system comprised of 1 (or more) sting assembly, sensor package, retainer (which may be built into the sting assembly) and gasket.
1x USB cable	A low-profile USB micro->A connector is normally bundled with the system.

Please ensure that all the system components listed above have been supplied, and that there is no apparent damage from shipping.

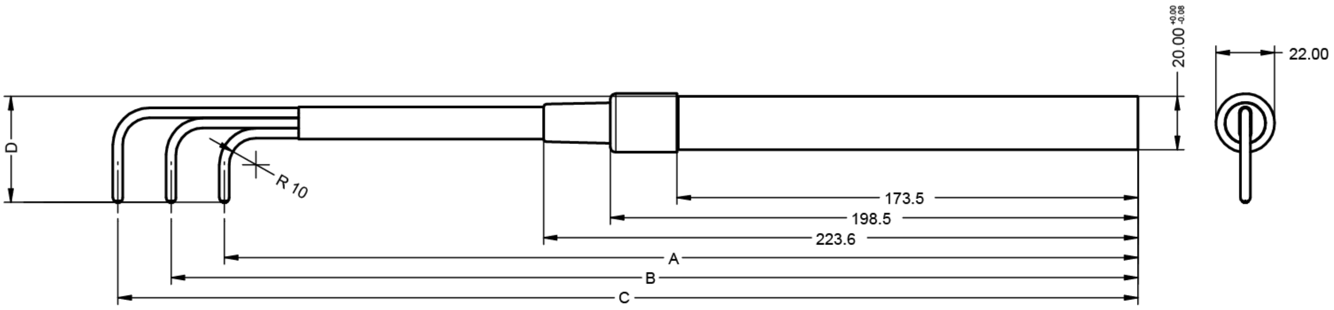
DETAILED SPECIFICATION

Standard sensor specifications (custom available on request)		
Product code	MD24HP-6K9	
Standard pressure range	6.9 kPa FS	
Maximum overpressure	34 kPa	
Sensor repeatability[1]	± 0.024 % FS	
Sensor accuracy[2], inc. calibration	± 0.25 % FS	
Total error band after auto-zero[3]	± 1.25 % FS	
Operational mode	24 channels true differential, with common reference	
Operating temperature range	-40° to +85° C non-condensing	
Compensated temperature range	0° to +50° C non-condensing	
Storage temperature range	-40° to +85° C	
Vibration	Sensors rated to 10 g, 10 Hz to 2 Hz	
Shock	50 g, 6 ms duration	
Maximum relative humidity	95 %	
Fluid temperature sensor[4]	-40°C - 85°C ± 0.5°C	
Relative ambient humidity sensor specification	0 % to 100 % RH, +/- 3%	
Ambient temperature sensor specification[5]	0°C - 65°C ± 0.5°C	
Ambient absolute pressure sensor specification	30-110 kPa FS, +/- 0.1kPa	
IMU specification	3 axis gyro, 125 °/s FS, ± 3.9 x10-3 °/s 3 axis accelerometer, 2g FS, ± 0.061 mg	
Communications	USB	UART
Voltage	Via USB	Regulated 5 Vdc
Power	490 mW	
Data acquisition rate	200 Hz Typ., simultaneous sampling	
Digital resolution	24-bit	
System specification[6]	2.5 GHz quad core Intel i5 with 8 Gb RAM	

[1] Includes errors due to pressure non-linearity, pressure hysteresis and non-repeatability.
[2] Includes errors due to pressure non-linearity, pressure hysteresis, non-repeatability and calibration uncertainty.
[3] Total residual error after auto-zero, excluding residual temperature sensitivity.
[4] Temperature is measured from rake body. Temperature readings may depend on thermal management.
[5] Temperature is recorded on the PCB and will be affected by waste heat generated.
[6] Minimum requirement for real-time data conversion. Offline post-processing required on slower systems.

Dimensions

MD24HP-6K9-3X7



Typical 3x seven-hole probe configuration
Dimensions A, B, C & D can be customer-specified
Dimension A - min 380 mm
Dimension C - max 240 mm
Dimension D - min 36 mm

2 SERIAL COMMUNICATIONS

When the rake system is powered on, it will undergo a brief system diagnostic test; if the test is passed, then a green LED will illuminate at the rear of the rake near the cable connector. Data will then begin streaming serial data on the UART serial lines if streaming on power-up has been enabled. The system will stream data continuously while powered on, until commanded to stop.

The sensors are factory-calibrated, and the calibration data is held in on-board memory. Data are converted into SI units by the firmware and streamed as single-precision floats.

Serial stream

Each data packet consists of 162 bytes, beginning with an unsigned-integer frame character ("#") and terminating with a checksum (both inclusive); the UART configuration is the typical 8-N-1. The data order is shown below, and is the same for both UART and USB.

byte index	Description	Type	Unit
0	Frame character '#'	uint8	-
1	P0	float32	Pa
2			
3			
4			
5	P1	float32	Pa
6			
7			
8			

9	P2	float32	Pa
10			
11			
12			
13	P3	float32	Pa
14			
15			
16			
17	P4	float32	Pa
18			
19			
20			
21	P5	float32	Pa
22			
23			
24			
25	P6	float32	Pa
26			
27			
28			
29	P7	float33	Pa
30			
31			
32			
33	P8	float32	Pa
34			
35			
36			
37	P9	float32	Pa
38			
39			
40			
41	P10	float32	Pa
42			
43			
44			
45	P11	float32	Pa
46			
47			
48			
49	P12	float32	Pa
50			
51			
52			
53	P13	float32	Pa
54			
55			
56			
57	P14	float32	Pa
58			
59			
60			

61	P15	float32	Pa
62			
63			
64			
65	P16	float32	Pa
66			
67			
68			
69	P17	float32	Pa
70			
71			
72			
73	P18	float32	Pa
74			
75			
76			
77	P19	float32	Pa
78			
79			
80			
81	P20	float32	Pa
82			
83			
84			
85	P21	float32	Pa
86			
87			
88			
89	P22	float32	Pa
90			
91			
92			
93	P23	float32	Pa
94			
95			
96			
97	External temperature	float32	°C
98			
99			
100			
101	Board temperature	float32	°C
102			
103			
104			
105	Atmospheric pressure	float32	Pa
106			
107			
108			
109	Relative humidity	float32	%
110			
111			
112			

113	Accelerometer x component	float32	g
114			
115			
116			
117	Accelerometer y component	float32	g
118			
119			
120			
121	Accelerometer z component	float32	g
122			
123			
124			
125	Gyroscope x component	float32	deg / s
126			
127			
128			
129	Gyroscope y component	float32	deg / s
130			
131			
132			
133	Gyroscope z component	float32	deg / s
134			
135			
136			
137	P0 status bits	uint8	-
138	P1 status bits	uint8	-
139	P2 status bits	uint8	-
140	P3 status bits	uint8	-
141	P4 status bits	uint8	-
142	P5 status bits	uint8	-
143	P6 status bits	uint8	-
144	P7 status bits	uint8	-
145	P8 status bits	uint8	-
146	P9 status bits	uint8	-
147	P10 status bits	uint8	-
148	P11 status bits	uint8	-
149	P12 status bits	uint8	-
150	P13 status bits	uint8	-
151	P14 status bits	uint8	-
152	P15 status bits	uint8	-
153	P16 status bits	uint8	-
154	P17 status bits	uint8	-
155	P18 status bits	uint8	-
156	P19 status bits	uint8	-
157	P20 status bits	uint8	-
158	P21 status bits	uint8	-
159	P22 status bits	uint8	-
160	P23 status bits	uint8	-
161	CRC16-CCITT	uint16	-
162			

IMPORTANT NOTE: Data are transmitted using the little-endian convention, so that the first byte transmitted for each quantity is the least significant.

Checksum & data corruption warning

A CRC-16 checksum word (uint16) is included at the end of each data packet to provide a warning of data loss or corruption in transmission. Example C++ code and DLL files to compute the CRC-16 checksum are available upon request. If the computed and transmitted checksums do not match, the entire data packet should be discarded.

Note that additional details about the system communications, including a summary of CRC-16-CCITT implementation, are appended to the end of this document.

3 PHYSICAL CONNECTIONS

The rake system is supplied either as a single integrated unit or as a modular system containing 1 or more sting assemblies with a single sensor package. Note that the on-board IMU will enable the roll-alignment of the rake to be matched to calibration conditions.

WARNING: Do not apply local stresses to the rake casing, or the casing may crack. The rake should be held in place with a friction collet, or other circular clamping arrangement.

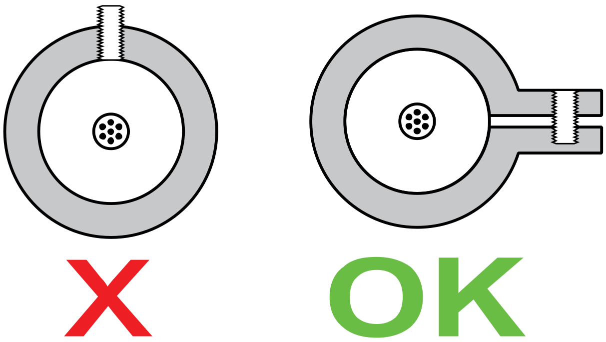


Figure 1: Correct probe mounting

Sting assembly

The sting assembly will consist of some number of probes bundled together, with a pressure coupling to join it to the sensor body. The sensor system can accommodate up to 24 independent pressure measurements; if the rake's particular sensor arrangement does not require 24 channels, those unused channels will remain internally unconnected.

For the case of the standard 3x7-hole probe rake arrangement, the holes are numbered as illustrated in figure 2. Note that the roll alignment is arbitrary.

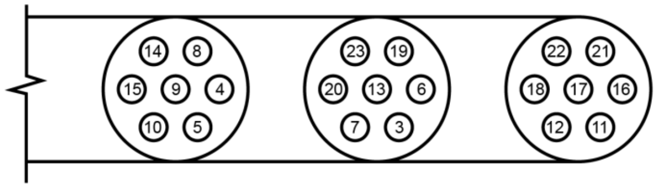


Figure 2: Typical hole numbering convention for 3x7-hole configuration, looking at probe tips from upstream, with the rake body located to the left. Channels 0, 1 & 2 unused. Note that hole numbering may vary from rake to rake.

Sensor package

There are two user-accessible ports on the sensor package- a micro-USB port and a serial comms port.

Micro-USB port: This allows the user to access the sensing and diagnostic functions of the rake system using a PC (with the appropriate drivers and software installed).

Comms port: This is the UART serial communications connection. There are four pins: GND (1), Tx (2), Rx (3) and V+ (4), where pin 1 is on the left when the rake is oriented such that the serial port is above the micro-USB port (see figure 3). Note that pin 1 on the matching plug is marked with a small recessed triangle.

IMPORTANT: UART Rx and Tx pins operate on 3.3V level, but are 5V tolerant. V+ must be well-regulated 5 Vdc, absolute maximum 5.5 Vdc

The maximum length of the data cables will depend on local EMI environment and shielding provisions. Powered repeaters should be used for USB cables longer than 3m.

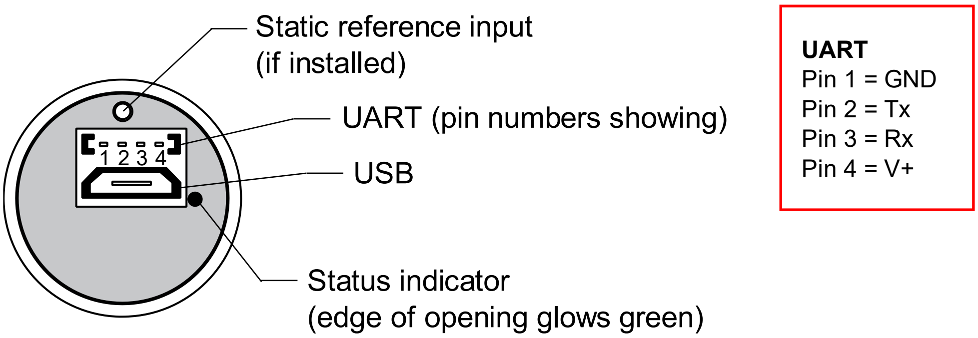


Figure 3: connector interface and pin assignment

IMPORTANT: The Tx terminal is the transmit line for the sensor package. This should be connected to the Rx terminal on the receiving unit.

Note that the pin numbering shown here is for the hardware end. Cable assemblies are not cross-linked: if a mating cable assembly is used, the pin order will be reversed on the cable end.

Modular system components

The modular rake system consists of a sensor package, gasket, sting assembly and retainer as shown in figure 4.

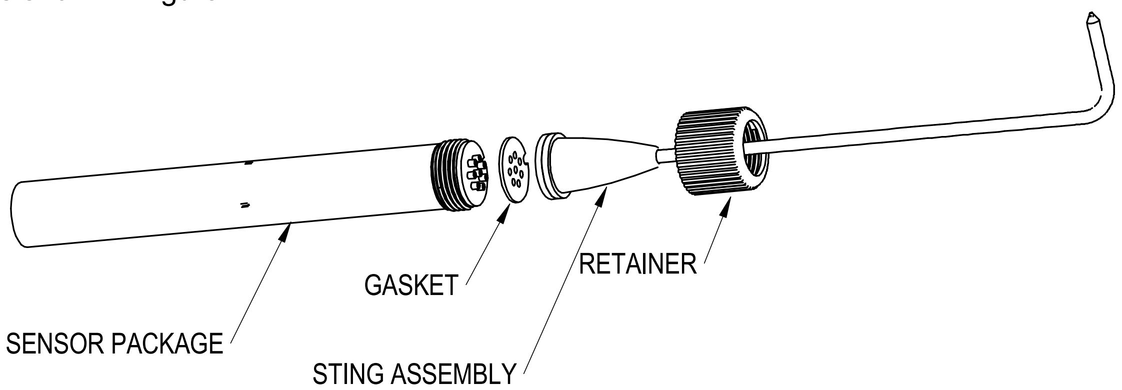


Figure 4: Schematic of modular rake system components

To assemble the rake, ensure the gasket and sting assembly are both fitted onto the barbs on the sensor package. Be sure to align the marks on the components to ensure correct assembly. Failure to do this will not prevent assembly and effective sealing of the rake components, but it will result in disagreement between pressure sensor channel numbers and hole numbering.

To secure the rake assembly, slide the retainer over the sting and screw by hand onto the sensor package. Note that the system does not require significant force to tighten and seal, and may be damaged by overtightening.

4 CARE AND HANDLING

WARNING: Do not allow any conductive materials to come into contact with the system, or it may be permanently damaged.

- Place protective covers over the probe tips when not in use, and handle the rake with care.
- Protect the sensor package from moisture and dust, and store in an ESD-safe sleeve when not in service.
- The sensor package enclosure includes components made from acrylic, and may be susceptible to solvents. The rake sting is not electrically connected to the system ground. Sensible ESD precautions should be taken before handling.
- Ensure that no dust, dirt or liquid enters the probes. This will alter the system performance. Any foreign material introduced into the sensors themselves may permanently damage the sensors.
- Do not use the rake in wet or condensing conditions. In the event of clean water ingress, the rake can be dried by heating to 40°C.
- The rake sting can be removed and cleared with compressed air if necessary. Ensure that the sting is dry before reassembling.
- Care must be taken to avoid damage to the rake sting or the probe tips. If the damage-evident coating is chipped, it is likely that the calibration of the probes have changed.
- Probes may be cleaned using a mild detergent solution and a fibre-free cloth. Do not use solvents or alcohol on the probe, and take care not to get any cleaning solution in the holes.
- Do not apply any bending moments on the rake sting.

Connect cables to the rake with care, as the socket mountings are fragile.

WARNING: Ensure that appropriate strain relief is used: cable strain can cause erroneous sensor readings and may cause catastrophic damage to the rake.

5 SOFTWARE AND DRIVERS

Although this rake system can function independently of a PC by continuously streaming data (when configured to do so), it is possible to interface with a PC via the included USB port for data logging, conversion, diagnostics and visualization.

Drivers

There are two external drivers which must be downloaded and installed on the computer in order for the PC to be able to interface with the rake system, in addition to the specific system driver for your rake.

- National Instruments LabView Run Time Engine (LVRTE)
- National Instruments VISA Run-Time Engine (NIVISA)

These drivers are freely available for download from National Instruments. Ensure that the 64-bit version of the NI LabVIEW Run Time Engine is selected (note this is not the default option), and restart the computer following each installation. Correct download settings for each driver are shown in figures 6 and 7 below.

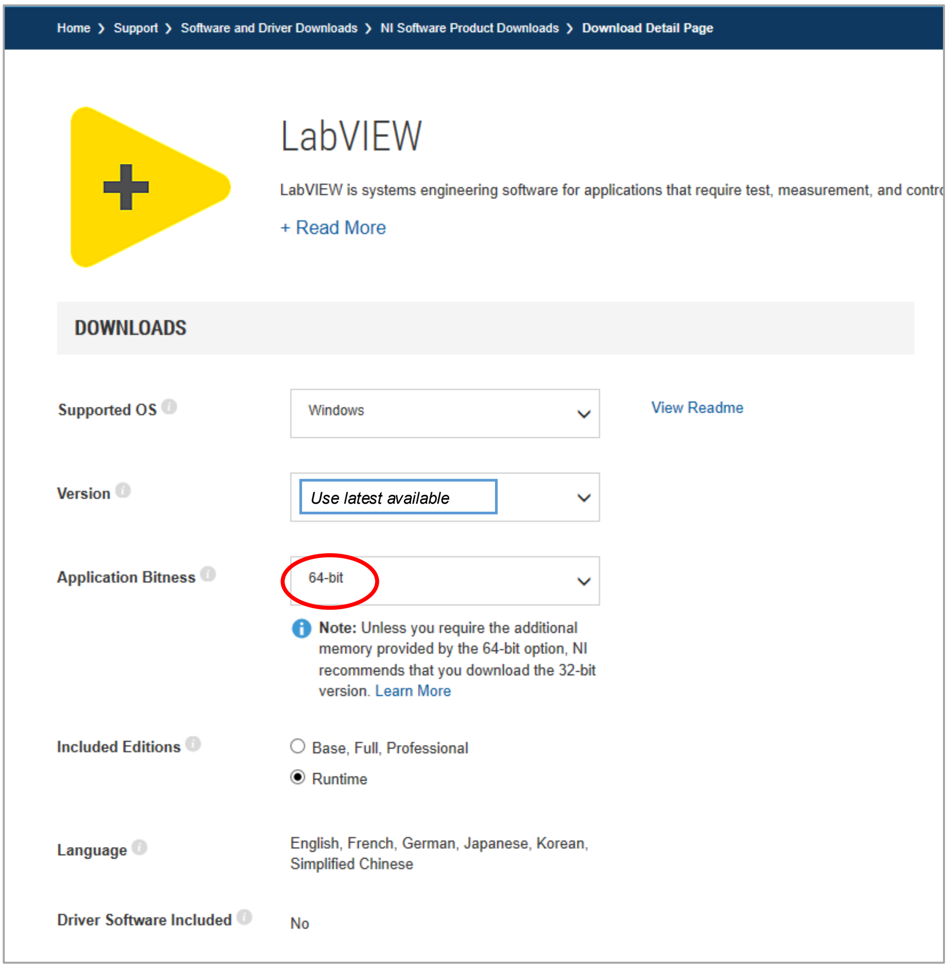


Figure 5: Download settings for NI LabVIEW Runtime Engine

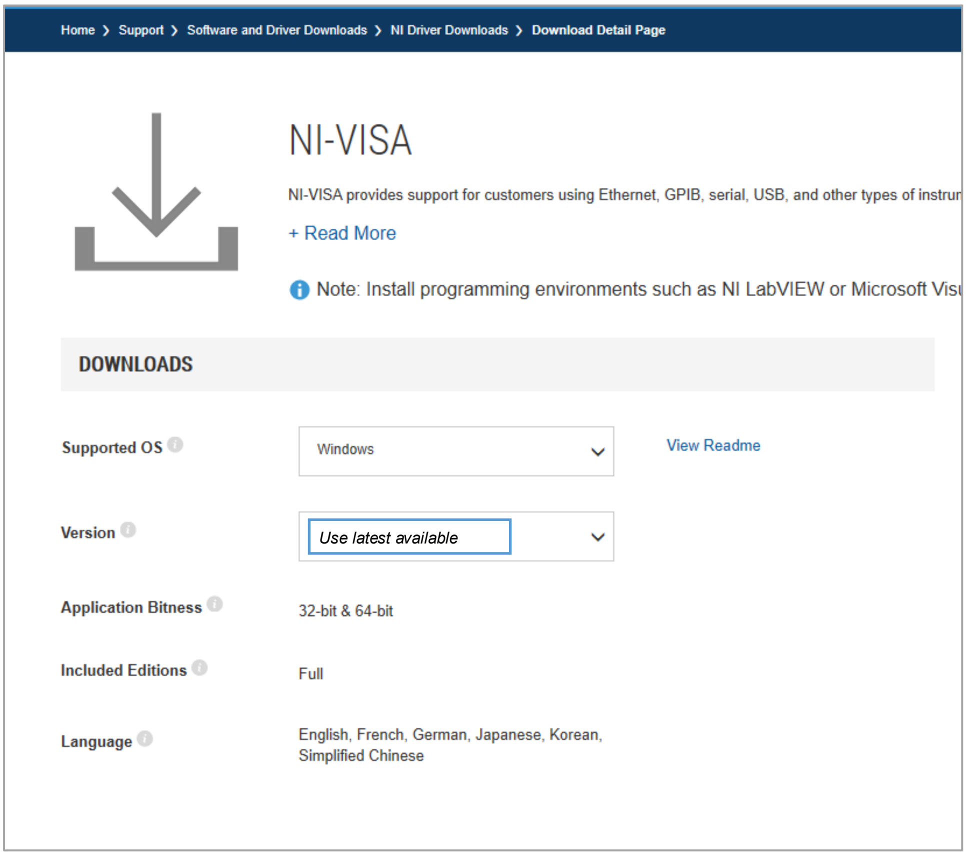


Figure 6: Download settings for NI VISA Runtime Engine

Additionally, a further USB driver install may be required for some older versions of Windows (not required for Windows 10).

Executable

An executable software package can be downloaded for your rake system, which facilitates direct communication between your PC and your rake using the rake's micro-USB connector. After launching the software, you should see the window reproduced in figure 8.

Starting procedure

1) Connect the computer to the rake's micro-USB socket.
2) The rake will perform a power-on self-test, lasting a couple of seconds. If all tests pass, a green LED beside the USB socket will illuminate.
3) Load the program [24HP USER INTERFACE]. It will start in the [ACTION] tab.
4) Using the [COM PORT] drop down menu, select the rake COM port.
5) Enter the desired setup options (see below)
6) Press the white arrow near the top left corner of the window to run the application.
7) Ensure that there are no errors in the [ERROR] dialogue box. If an error has occurred at this stage, it is most likely an invalid COM port selection. If using an older Windows version this may alternatively be a compatibility error requiring the additional installation described above.

Setup options

Within the [ACTION] tab, the user can select some options for data collection and processing.

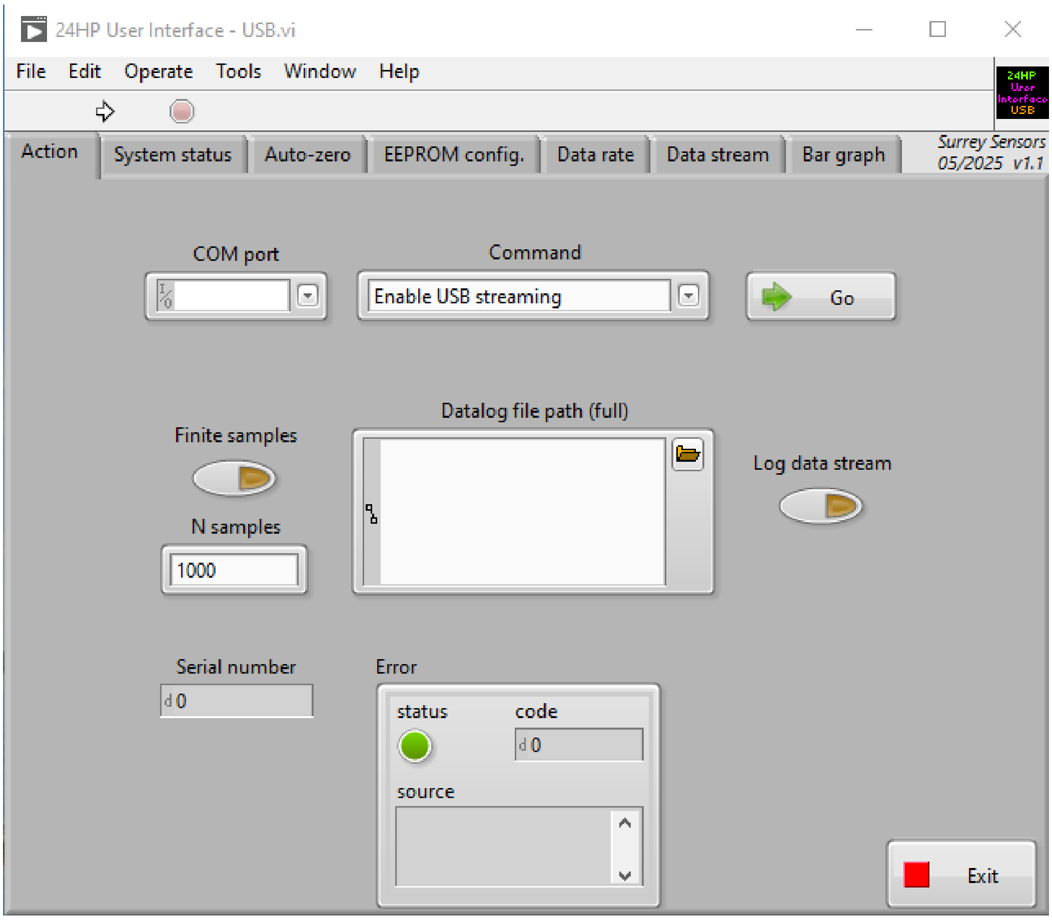


Figure 7: Rake system companion software screen-shot

- **Finite samples:** The rake can be instructed to collect a specified number *N* samples only. If this option is disabled, the rake will collect samples until manually stopped using the [DISABLE USB STREAMING] command.

- **Log data stream:** A tab-delimited ASCII text output of a time history of all the sensor values can be saved to disk by enabling the [LOG DATA STREAM] switch. If the [DATALOG FILE PATH] box is left empty, the user will be prompted where to save the file. It is helpful to put a known extension on the file such as ".txt". Data is written to disk every second and stopped upon disabling USB streaming. If streaming is resumed and the filename is not changed, the user will be prompted to either replace the file or cancel. If cancel is chosen, a prompt will appear to allow a new filename to be chosen. If this is also cancelled, data streaming will begin but no log file will be created and the [LOG DATA STREAM] switch will be disabled.

Available commands

The [COMMAND] drop-down menu in the [ACTION] tab contains the following commands. A command is run by pressing the [GO] button to the right of the [COMMAND] menu. The application will automatically switch to the relevant tab. Once finished, return to the [ACTION] tab to select another command and run as desired.

- **Enable USB streaming:** this command activates the USB data stream. The [DATA STREAM] tab is activated and a running plot of all 24 pressure sensor values is displayed (whether or not they are used), along with values for all the other on-board sensors. This will continue until the user presses the [STOP] button or selects another tab, or exits the program by closing the window or until the desired number of samples has been acquired (if [FINITE SAMPLES] enabled). Note that the data stream from the UART serial output is not affected by enabling USB streaming.

- **Enable USB streaming (HW-trig.):** this command arms the USB data stream function, such that the rake will begin streaming data when a hardware trigger command is received. Hardware triggering allows the rake's UART port to be used for synchronization, with specific EEPROM settings. See appendix for details.

- **Get status info:** running this command will activate the [SYSTEM STATUS] tab. The system will be queried for the status of the last self-test. Green indicators indicate the test was passed, while red indicates a fault. Descriptive text beside each indicator is provided.

- **Run self-test:** the system will be made to perform the power-up self-test again, and then query for the result. This allows the user to easily check any changes made without having to power cycle the device.

- **Perform temporary auto-zero:** this command actives the [AUTO ZERO] tab and begins collecting samples from each pressure sensor to estimate the zero pressure reading. An offset is then temporarily stored in volatile memory that will remain until the system is powered down. During this procedure, ensure that the sensors are subjected to zero differential pressure.

- **Perform permanent auto-zero**: this function should not be used under normal operation. This command is similar to temporary auto-zero, but the zero values are written permanently to the EEPROM. ***WARNING*** *– this will overwrite the existing offsets that were used during calibration. Carrying out this procedure may invalidate your calibration.*

- **Set data rate***:* this command allows the user to set the system sampling rate using a drop-down menu.

- **Get data rate***:* this command allows the user to retrieve the current setting of the system sampling rate.

- **Get serial number:** each rake is encoded with its own unique serial number, which is displayed in the window below the [COM PORT] drop-down menu. This function will retrieve the rake's serial number and display it in the [ACTION] tab. Note that the serial number is also retrieved as part of the normal startup operations. The serial numbers are decimal values; the rake having serial number "1234" will display as [d1234].

- **Disable USB streaming:** this command stops data transmission over USB.

- **Get EEPROM values:** this function displays the [EEPROM CONFIG] tab and read off all the EEPROM values from the system.

- **Set EEPROM values:** The EEPROM is the rake's on-board memory, and is factory-set for normal operation. To make changes to the EEPROM values, first run [GET EEPROM VALUES] to populate the table with the current values. These values should be recorded separately to ensure that the rake can be returned to factory settings. To make changes, alter the values as desired; return to the [ACTION] tab and run the [SET EEPROM

VALUES] command. Finally, return to the [ACTION] tab and run the [GET EEPROM VALUES] command again to check the values are as desired and, importantly, that the EEPROM checksum indicator is green. A red indicator means the data has become corrupt in transit and should be re-transmitted. Note that rakes typically ship with UART data streaming disabled; to enable this, the [UART DATA OUTPUT ON POWER UP] setting in the EEPROM needs to be changed to [1] (for 'ON'). Note that older versions of the software may use 'TTL' rather than 'UART' in some labels.

- **System soft reset:** When enabled, this clears the rake's volatile memory and restarts the firmware.
- **Exit:** This function terminates the executable.

Notes on calibration and data reduction

For rakes with multi-hole heads, the velocity components are obtained using the generalized (sectorless) technique of Shaw-Ward *et al.* (2015). This process can be implemented separately using offline post-processing utilities.

6 TECHNICAL SUPPORT

Full technical support is available for this product and its associated software.

If you experience any difficulty in installation or use, or if you need additional support in the operation of the system, please contact your Surrey Sensors Ltd. account manager or technical representative.

Appendix 1: Additional communications details

Status bytes map

Byte num.	Bit	Description	Info		Byte default value
0	0	Bank 1 all sensor values in range	1 yes, 0 no	0	
	1	Bank 2 all sensor values in range	1 yes, 0 no	0	
	2	Bank 3 all sensor values in range	1 yes, 0 no	0	
	3	-	-	0	
	4	-	-	0	
	5	-	-	0	
	6	-	-	0	
	7	-	-	0	0
1	0	Bank 1 all sensor status good	1 yes, 0 no	0	
	1	Bank 2 all sensor status good	1 yes, 0 no	0	
	2	Bank 3 all sensor status good	1 yes, 0 no	0	
	3	-	-	0	
	4	-	-	0	
	5	-	-	0	
	6	-	-	0	
	7	-	-	0	0
2	0	On-board temperature sensor okay	1 yes, 0 no	0	
	1	External temperature sensor okay	1 yes, 0 no	0	
	2	MCU EEPROM checksum okay	1 yes, 0 no	0	
	3	IMU ident pass	1 yes, 0 no	0	
	4	IMU acc self test pass	1 yes, 0 no	0	
	5	IMU gyr self test pass	1 yes, 0 no	0	
	6	Env sensors ident pass	1 yes, 0 no	0	
	7	-	-	0	0

USB Command List

After @

Cmd. Byte	Description	Return	Additional information
s	Retrieve status byte	3x uint8	Bank_P_val_ok, Bank_P_status_ok, EEPROM_CRC16_pass
S	Perform self-test and retrieve status bytes	3x uint8	Bank_P_val_ok, Bank_P_status_ok, EEPROM_CRC16_pass
z	Perform temporary auto-zero	24x float32	P_offset(0 to 23)[96]
Z	Perform permanent auto-zero (write values to EEPROM)	24x float32	P_offset(0 to 23)[96]
e	Read all EEPROM values	141x uint8	See EEPROM map for details
E	Write all EEPROM values (including new checksum)	-	
D	Enable USB streaming	163x uint8	See serial data table for details. Returns at [Datarate] until stream is disabled.
d	Disable USB streaming	-	
H	Enable hardware trigger	-	
h	Disable hardware trigger	-	
F	Set data period	-	Send 1x uint32, Data_Period[4], containing the data period in microseconds
f	Get data period	1x uint32	Data period (μs)
N	Get serial number	1x uint32	Serial_number[4]
G	Get current data packet	163x uint8	See serial data table for details
J	Set power-up default data period	-	Send 1x uint32, Data_Period[4], containing the data period in microseconds
B	Set power-up default TTL baud rate	-	Send 1x uint32, TTL_Baud[4]
b	Get TTL baud rate	1x uint32	TTL_baud[4]
Q	Set power-up default TTL streaming enabled	-	
q	Get power-up default TTL streaming enabled	1x uint8	1 = yes, 0 = no
R	System soft reset	-	

IMU Modes

Accelerometer range		
Mode (byte)	Eng. Value	Unit
0	±2	g
1	±4	g
2	±8	g
3	±16	g

Gyroscope range		
Mode (byte)	Eng. Value	Unit
0	±125	°/s
1	±250	°/s
2	±500	°/s
3	±1000	°/s
4	±2000	°/s

IMU refresh rate		
Mode (byte)	Eng. Value	Unit
0	6.25	Hz
1	12.5	Hz
2	25	Hz
3	50	Hz
4	100	Hz
5	200	Hz
6	400	Hz
7	800	Hz
8	1600	Hz

EEPROM Table

byte index	Description	Type	Unit
0	P0 offset	float32	Pa
1			
2			
3			
4	P1 offset	float32	Pa
5			
6			
7			
8	P2 offset	float32	Pa
9			
10			
11			
12	P3 offset	float32	Pa
13			
14			
15			
16	P4 offset	float32	Pa
17			
18			
19			
20	P5 offset	float32	Pa
21			
22			
23			
24	P6 offset	float32	Pa
25			
26			
27			

28	P7 offset	float33	Pa
29			
30			
31			
32	P8 offset	float32	Pa
33			
34			
35			
36	P9 offset	float32	Pa
37			
38			
39			
40	P10 offset	float32	Pa
41			
42			
43			
44	P11 offset	float32	Pa
45			
46			
47			
48	P12 offset	float32	Pa
49			
50			
51			
52	P13 offset	float32	Pa
53			
54			
55			
56	P14 offset	float32	Pa
57			
58			
59			
60	P15 offset	float32	Pa
61			
62			
63			
64	P16 offset	float32	Pa
65			
66			
67			
68	P17 offset	float32	Pa
69			
70			
71			
72	P18 offset	float32	Pa
73			
74			
75			
76	P19 offset	float32	Pa
77			
78			

79			
80	P20 offset	float32	Pa
81			
82			
83			
84	P21 offset	float32	Pa
85			
86			
87			
88	P22 offset	float32	Pa
89			
90			
91			
92	P23 offset	float32	Pa
93			
94			
95			
96	External thermistor temperature offset	float32	°C
97			
98			
99			
100	P_atmos offset	float32	Pa
101			
102			
103			
104	Serial number	uint32	-
105			
106			
107			
108	Data Period	uint32	µs
109			
110			
111			
112	UART baud rate	uint32	bps
113			
114			
115			
116	UART streaming enabled on power-up	unit8	1 yes, 0 no
117	Number of active sensor banks	unit8	3 (fixed)
118	Power saving mode	unit8	1 yes, 0 no
119	Sync sensor & data clocks	unit8	1 yes, 0 no
120	Accelerometer xyz scale factor	float32	-
121			
122			
123			
124	Gyroscope x component offset	float32	deg / s
125			
126			
127			

128	Gyroscope y component offset	float32	deg / s
129			
130			
131			
132	Gyroscope z component offset	float32	deg / s
133			
134			
135			
136	Accelerometer range mode	unit8	-
137	Gyroscope range mode	unit8	-
138	IMU update rate mode	unit8	-
139	CRC16-CCITT	uint16	-
140			

Summary of CRC-16-CCITT Implementation in C++

Global Variables and Constants

```
uint16_t CRC16_LUT[256];
const uint16_t poly = 0x1021;
const uint16_t crc_init = 0xFFFF;
```

CRC-16 Lookup Table (LUT) Generation

The following function is called once at the start. The 1D array of length 256 “CRC16_LUT” is then stored in memory for all time and used whenever a CRC is computed.

```
void Generate_CRC16_LUT()
{
	for (uint16_t i = 0; i < 256; i++)
	{
		uint16_t Byte = i << 8;

		for (uint8_t Bit = 0; Bit < 8; Bit++)
		{
			if ((Byte & 0x8000) != 0)
			{
				Byte <<= 1;
				Byte ^= poly;
			}
			else
			{
				Byte <<= 1;
			}
		}

		CRC16_LUT[i] = Byte;
	}
}
```

Alternatively, the LUT can be hard-coded as a constant:

```c
// CRC-16 lookup table for CCITT polynomial 0x1021

static const uint16_t CRC16_LUT[256] =
{
  0x0000, 0x1021, 0x2042, 0x3063, 0x4084, 0x50A5, 0x60C6, 0x70E7,
  0x8108, 0x9129, 0xA14A, 0xB16B, 0xC18C, 0xD1AD, 0xE1CE, 0xF1EF,
  0x1231, 0x0210, 0x3273, 0x2252, 0x52B5, 0x4294, 0x72F7, 0x62D6,
  0x9339, 0x8318, 0xB37B, 0xA35A, 0xD3BD, 0xC39C, 0xF3FF, 0xE3DE,
  0x2462, 0x3443, 0x0420, 0x1401, 0x64E6, 0x74C7, 0x44A4, 0x5485,
  0xA56A, 0xB54B, 0x8528, 0x9509, 0xE5EE, 0xF5CF, 0xC5AC, 0xD58D,
  0x3653, 0x2672, 0x1611, 0x0630, 0x76D7, 0x66F6, 0x5695, 0x46B4,
  0xB75B, 0xA77A, 0x9719, 0x8738, 0xF7DF, 0xE7FE, 0xD79D, 0xC7BC,
  0x48C4, 0x58E5, 0x6886, 0x78A7, 0x0840, 0x1861, 0x2802, 0x3823,
  0xC9CC, 0xD9ED, 0xE98E, 0xF9AF, 0x8948, 0x9969, 0xA90A, 0xB92B,
  0x5AF5, 0x4AD4, 0x7AB7, 0x6A96, 0x1A71, 0x0A50, 0x3A33, 0x2A12,
  0xDBFD, 0xCBDC, 0xFBBF, 0xEB9E, 0x9B79, 0x8B58, 0xBB3B, 0xAB1A,
  0x6CA6, 0x7C87, 0x4CE4, 0x5CC5, 0x2C22, 0x3C03, 0x0C60, 0x1C41,
  0xEDAE, 0xFD8F, 0xCDEC, 0xDDCD, 0xAD2A, 0xBD0B, 0x8D68, 0x9D49,
  0x7E97, 0x6EB6, 0x5ED5, 0x4EF4, 0x3E13, 0x2E32, 0x1E51, 0x0E70,
  0xFF9F, 0xEFBE, 0xDFDD, 0xCFFC, 0xBF1B, 0xAF3A, 0x9F59, 0x8F78,
  0x9188, 0x81A9, 0xB1CA, 0xA1EB, 0xD10C, 0xC12D, 0xF14E, 0xE16F,
  0x1080, 0x00A1, 0x30C2, 0x20E3, 0x5004, 0x4025, 0x7046, 0x6067,
  0x83B9, 0x9398, 0xA3FB, 0xB3DA, 0xC33D, 0xD31C, 0xE37F, 0xF35E,
  0x02B1, 0x1290, 0x22F3, 0x32D2, 0x4235, 0x5214, 0x6277, 0x7256,
  0xB5EA, 0xA5CB, 0x95A8, 0x8589, 0xF56E, 0xE54F, 0xD52C, 0xC50D,
  0x34E2, 0x24C3, 0x14A0, 0x0481, 0x7466, 0x6447, 0x5424, 0x4405,
  0xA7DB, 0xB7FA, 0x8799, 0x97B8, 0xE75F, 0xF77E, 0xC71D, 0xD73C,
  0x26D3, 0x36F2, 0x0691, 0x16B0, 0x6657, 0x7676, 0x4615, 0x5634,
  0xD94C, 0xC96D, 0xF90E, 0xE92F, 0x99C8, 0x89E9, 0xB98A, 0xA9AB,
  0x5844, 0x4865, 0x7806, 0x6827, 0x18C0, 0x08E1, 0x3882, 0x28A3,
  0xCB7D, 0xDB5C, 0xEB3F, 0xFB1E, 0x8BF9, 0x9BD8, 0xABBB, 0xBB9A,
  0x4A75, 0x5A54, 0x6A37, 0x7A16, 0x0AF1, 0x1AD0, 0x2AB3, 0x3A92,
  0xFD2E, 0xED0F, 0xDD6C, 0xCD4D, 0xBDAA, 0xAD8B, 0x9DE8, 0x8DC9,
  0x7C26, 0x6C07, 0x5C64, 0x4C45, 0x3CA2, 0x2C83, 0x1CE0, 0x0CC1,
  0xEF1F, 0xFF3E, 0xCF5D, 0xDF7C, 0xAF9B, 0xBFBA, 0x8FD9, 0x9FF8,
  0x6E17, 0x7E36, 0x4E55, 0x5E74, 0x2E93, 0x3EB2, 0x0ED1, 0x1EF0
};
```

CRC-16 Computation

The following function is called whenever a CRC-16 is required from an array of data.

```c
uint16_t Calc_CRC16(uint8_t *Data, uint16_t DataLen, uint16_t crc)
{
	for (uint16_t i = 0; i < DataLen; i++)
	{
		uint8_t index = Data[i] ^ (crc >> 8);

		crc = CRC16_LUT[index] ^ (crc << 8);
	}

	return crc;
}
```

CRC-16 Function Call Example

The data for which the CRC is to be computed is first of all typecast into an array of unsigned char (uint8_t) "DataBytes". This can be done using the `memcpy` function. When generating a CRC value for an array of data the length value "Len" passed to the function is that of the number of bytes in the entire array. However, when checking a CRC value appended to an array of data, the length value passed to the function is two less than that of the entire array so as to exclude the appended CRC word. The CRC value passed to the function is that of the initialiser constant "crc_init", which, for the CCITT specification, is hexadecimal `0xFFFF`.

```
uint16_t CRC_computed = Calc_CRC16(&DataBytes, Len, crc_init);
```

Checksum Test

A checksum test is passed if the computed and transmitted checksum values are equal. With the CRC appended at the end of the transmitted data array the test is carried out as follows

```
uint16_t CRC_appended;

memcpy(&CRC_appended, &DataBytes[Len - 2], 2);

bool CRC_pass = (CRC_appended == CRC_computed);
```

Code implementation can be validated by cross-checking results with a reputable online CRC calculator such as https://crccalc.com/

CRC-16-CCITT Algorithm Parameters:

Polynomial divisor:	`0x1021`	$(x^{16} + x^{12} + x^{5} + 1)$
CRC initialiser:	`0xFFFF`	
Input reflection:	`False`	
Output reflection:	`False`	
Output XOR:	`0x0000`	

Appendix 2: Reference coordinate systems

The spherical coordinate system commonly used in physics (ISO 80000-2:2019 convention) is adopted here and used throughout. By convention, the pitch angle α increases with rotation toward the *z*-axis, and the yaw angle β is positive rotating anti-clockwise about the *z*-axis (see figure 9). Here, *u*, *v* and *w* are the velocity components along *x*, *y* and *z*, respectively.

In some early data reduction algorithms, the cone angle θ and roll angle φ were used. These are not relevant here.

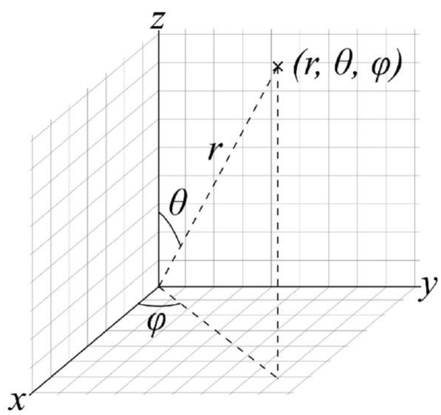


Figure 8: Coordinate conventions

In all cases, by convention, the pitch angle α and yaw angle β are defined as

$$\alpha = 90° - \theta$$
$$\beta = \varphi$$

Probe-based coordinate system:

This convention is intended for use on a moving platform, such as an aircraft or vehicle.

$$u = |U|\cos(\beta)\cos(\alpha)$$
$$v = |U|\sin(\beta)\cos(\alpha)$$
$$w = |U|\sin(\alpha)$$

Wind tunnel-based coordinate system:

This convention is intended for use in stationary-probe wind tunnel applications with a right-handed coordinate system with the probe pointing in the upstream *x* direction, such that the z-axis is vertical (environmental flow convention).

$$u = |U|\cos(\beta)\cos(\alpha)$$
$$v = -|U|\sin(\beta)\cos(\alpha)$$
$$w = |U|\sin(\alpha)$$

Rotated wind tunnel-based coordinate system:

This convention is intended for use in stationary-probe wind tunnel applications with a right-handed coordinate system with the probe pointing in the upstream *x* direction, such that the y-axis is vertical (aerospace flow convention).

$$u = |U| \cos(\beta) \cos(\alpha)$$
$$v = |U| \sin(\alpha)$$
$$w = |U| \sin(\beta) \cos(\alpha)$$

Appendix 4: Hardware triggering

The rake includes an internal clock for data acquisition and timing. Occasionally, it may be necessary to begin acquisition on receipt of an external trigger pulse. If so, the rake's UART Rx pin can be reconfigured as a trigger input, and the Tx port as a trigger output. Consequently the rake can only be operated via USB if the external trigger function is used.

If external triggering was specified as a requirement at the time of order, then the rake would have shipped with the required settings. If not, the settings can be modified by the user.

Instructions

Step 1: Launch the user interface application, select the appropriate COM port and run the application by clicking the white arrow in the menu bar.

Step 2: Under the [ACTION] tab, select [GET EEPROM VALUES] from the [COMMAND] drop-down menu and click [GO]. This will populate the [EEPROM CONFIG] tab.

Step 3: It is strongly recommended that a copy of the factory-set EEPROM values are saved in case of accidental corruption.

Step 4: Make the following changes to the EEPROM configuration (if required), following instructions in the manual:

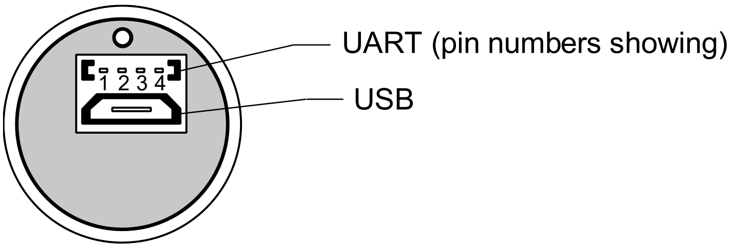
- Set [UART BAUD RATE] to [0] (zero) for trigger on rising edge, and [-1] (negative one) for trigger on falling edge.
- Set [UART STREAMING] to [0] (zero)

Note that the rake can only stream data via USB when the hardware trigger function is enabled.

Step 5: Connect the input trigger signal to the Rx pin (pin 3) on the rake's UART connector (see Figure 11) and the trigger system ground to the GND pin (pin 1).

IMPORTANT: Be sure to use the correct cable assembly for the UART connection, and that this is inserted in the correct orientation.

WARNING: The rake logic operates at 3.3V levels but is 5V tolerant (absolute maximum 5.5 Vdc). Any voltage values applied outside of this range to the Tx/Rx pins may damage the rake and will void the warranty.



UART
Pin 1 = GND
Pin 2 = Tx
Pin 3 = Rx
Pin 4 = V+

Figure 9: connector interface and pin assignment

Note that the pin numbering shown here is for the hardware end. Cable assemblies are not cross-linked: if a mating cable assembly is used, the pin order will be reversed on the cable end.

Step 6: Enter the desired parameters for acquisition in the user interface software, under the [ACTION] tab.

Step 7: From the [COMMAND] drop-down menu, select [ENABLE USB STREAMING (HW TRIG)] and click [GO].

The rake trigger input is now armed, and acquisition will begin on receipt of the trigger signal. Note that the latency in response is significantly smaller than the smallest sampling time, and can be considered as negligible.

When configured for hardware triggering, the rake's UART Tx line will automatically act as an output trigger. Whenever the rake is streaming data, the polarity of the UART Tx line will switch (following the same convention as the trigger polarity selection). The polarity will switch when the rake is triggered either by a hardware trigger or software command, and will return to its original state when acquisition stops. The use of the Tx line is optional (it can be left open or unconnected), but care must be taken to ensure that the line is used only to provide trigger pulses when in this mode.

The content of this user manual is for general information only and is subject to change without notice. It may contain inaccuracies or errors and Surrey Sensors Ltd. expressly exclude liability for any such inaccuracies or errors to the fullest extent permitted by law. Your use of any information is entirely at your own risk, for which Surrey Sensors Ltd. shall not be liable.