
Defining the DI-145 Protocol

(Firmware revision 1.01 only)

DATAQ Instruments

[Part 1 — Legacy protocol with backward DI-194RS compatibility](#)

[Part 2 — Extended protocol without DI-194RS backward compatibility](#)

Part 1 — Legacy protocol with backward DI-194RS compatibility

Although DATAQ Instruments provides ready-to-run WinDaq software with its DI-145 Data Acquisition Starter Kit, programmers will want the flexibility to integrate the DI-145 in the context of their own application. To do so they want complete control over DI-145 hardware, which can be accomplished by using the device at the protocol level. This white paper describes how protocol-level programming of the DI-145 is implemented across the Windows and Linux and operating systems. Part one explores the legacy command set and response that maintains backward compatibility with the DI-194. It also defines how the virtual COM port software interface is implemented. Part two explores the DI-145's expanded protocol instruction set.

Virtual COM Port Operation

Installing the DI-145 driver package and connecting DI-145 hardware to the host computer's USB port results in a COM port being hooked by the operating system and dedicated to the DI-145 device. Multiple DI-145 devices may be connected to the same PC without additional driver installations, which results in a unique COM port number assigned to each by the operating system. Hooking a COM port in this manner facilitates ease of programming from any operating system and programming language by simply writing commands to and reading responses from the port, but before any meaningful communication with a connected DI-145 can begin the controlling program must determine the COM port number assigned to the device. The method used for this varies as a function of the host operating system.

Virtual COM Driver (Windows)

DATAQ Instruments provides a minimum installation for Windows that you can download and use at no charge, even for OEM applications. This is a scaled-down version of the standard installation that omits WinDaq software and other utilities that are extraneous in a pure programming environment. The download provides a Microsoft-signed INF file that ensures trouble-free operation with Windows XP (32-bit only), and both 32- and 64-bit Windows Vista and Windows 7. The installation depends upon driver *usbser.sys*, a Windows component located in path *%SystemDrive%\Windows\System32\Drivers*.

COM Port Number Discovery (Windows)

Using the DI-145's vendor and product IDs Windows' registry can be accessed programmatically to determine the COM port number that is assigned to one or more connected DI-145s. The

Vendor and Product ID combination for the DI-145 is: *Vid_0683&Pid_1450*. With this information and at least one connected DI-145, determining assigned COM port numbers is a two-step process:

1. The registry tree *HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\usb-ser\Enum* will contain one Device Instance ID for each DI-145 connected to the PC. The Device Instance ID assumes the following typical string value:
USB\VID_0683&PID_1450\5&18B6E64F&0&8. The first two sections of this string (*USB\VID_0683&PID_1450*) are constant for all DI-145s. The second section (*5&18B6E64F&0&8*) will vary depending upon where in the USB port hierarchy the DI-145 is physically connected. Since more than one DI-145 cannot be connected to the same USB port this value will be unique for each concurrently connected DI-145. The entire string value is required for the second step.
2. Registry tree *HKEY_LOCAL_MACHINE\CurrentControlSet\Enum\USB\VID_0683&PID_1450\5&18B6E64F&0&8* (continuing with the above example) shows a variable called *FriendlyName* set to string value *DATAQ DI-145 (COMXX)*, where *XX* is the COM port number assigned to the specified DI-145. This string may be parsed to extract the port number assigned to the DI-145. The process may be repeated using other Device Instance IDs determined from step (1) for other connected DI-145 instruments.

COM port number assignments may also be determined manually from Windows' Device Manager in its *Ports (COM & LPT)* section. However, the assigned value will change depending upon the physical USB port connected to the DI-145 and any other devices that may hook COM ports.

Virtual COM Driver (Linux)

Linux has two different generic drivers, which are appropriate for a USB to COM port converter. The first is an Abstract Control Model driver designed for modem devices, and is simply named *acm*. The other one is a generic USB to serial driver named *usbserial*.

COM Port Number Discovery (Linux)

If support for the *acm* driver has been compiled in the kernel, Linux will automatically load it and a new terminal device will be created under */dev/ttyACMx*, where *x* is the COM port number. The path */dev/serial/by-id/usb-0683_1450-** presents links to */dev/ttyACM** device files of currently plugged in DI-145 devices.

The second driver, *usbserial*, must be loaded manually by using the *modprobe* command with the vendor ID and product ID values used by the DI-145:

```
modprobe usbserial vendor=0x0683 product=0x1450
```

Once the driver is loaded, a new terminal entry appears and should be named */dev/ttyUSBx*, where *x* is the COM port number.

DI-145 Legacy Command Set

DI-145 Instruments support a dual command set, one for backward compatibility with its predecessor model DI-194RS. A second offers an enhanced instruction set that taps DI-145 features that were not available on earlier models. Examples include the ability to report firmware version, output delimited ASCII data, and more. The enhanced command set is explored in part two of this document. Both command sets are designed around standard ASCII characters to allow efficient experimentation and verification using terminal emulators, like HyperTerminal for Windows.

DI-145 Legacy Command Set					
Command*			Command*		
Hex	ASCII	Action	Hex	ASCII	Action
0x4E5A	NZ	Returns the DI-145's serial #	0x4339	C9	Enable channels 1 and 4
0x4331	C1	Enable channel 1	0x433A	C:	Enable channels 2 and 4
0x4332	C2	Enable channel 2	0x433B	C;	Enable channels 1, 2, and 4
0x4333	C3	Enable channels 1 and 2	0x433C	C<	Enable channels 3 and 4
0x4334	C4	Enable channel 3	0x433D	C=	Enable channels 1, 3, and 4
0x4335	C5	Enable channels 1 and 3	0x433E	C>	Enable channels 2, 3, and 4
0x4336	C6	Enable channels 2 and 3	0x433F	C?	Enable channels 1, 2, 3, and 4
0x4337	C7	Enable channels 1, 2, and 3	0x5331	S1	Start data acquisition
0x4338	C8	Enable channel 4	0x5330	S0	Stop data acquisition

* The DI-194 command set required that all commands be preceded with a null character (0x00), but the DI-145 does not. Null characters are ignored by the DI-145.

A Minor Backward Compatibility Issue

At this point the DI-145 takes a minor departure from complete compatibility with its DI-194 predecessor. The DI-194 did not have the intelligence on-board to perform the scaling and offset operation for each sample that's required to deliver to a fully calibrated reading to the host PC. DI-194 end users were instead shown how they could calibrate any of its four channels using the scaling and offset facilities built into WinDaq software. Of course, anyone interested in using the DI-194 with their own program was burdened with this task and one of the goals of the DI-145's design was to eliminate it, since the new design employs a much more powerful on-board processor that can easily apply the scaling constants transparently. However, this approach causes a minor incompatibility between the legacy DI-194 and resulting DI-145 protocols.

During the calibration process performed at DATAQ Instruments and before the DI-145 is shipped to a customer, known voltage values are applied to each of its channels and a connected PC calculates the required scale and offset factors so the instrument meets or exceeds its accuracy specification. Those factors are written to nonvolatile memory built into the DI-145 for the instrument's processor to apply going forward. An unavoidable problem is that this procedure consumes a small amount of the DI-145's dynamic ADC range. Since the DI-145 is specced as

having ten bits of measurement resolution, and if we were to report only those ten bits like we do for the DI-194, resulting values would show missing counts or codes. If you applied a very precise ramp to an analog channel and slowly changed the voltage between positive and negative full scale you'd see a response that for the most part changed by only one ADC count as the applied voltage passed through the ADC's bit resolution thresholds, but occasionally you'd also see the ramp jump, skipping one count and resulting in a discontinuity called a missing code. Moreover, the places where missing codes exist would be different from unit to unit and even from channel to channel within the same unit. Obviously, this situation is unacceptable.

Fortunately, the DI-145's ADC and protocol are designed for 12 bits, allowing us to use this improved resolution to provide an accurate calibration at nearly the 10-bit level without missing codes, but with the incongruity that we report a 12-bit number for an instrument specced for 10 bits of ADC resolution. The DI-145 is actually not a 12-bit device, since there will be between 993 and 1,024 active counts within the 4,096 possible 12-bit combinations. Usually the DI-145 will register a four count change for every 19.53 mV change of applied voltage, but sometimes it will register a 5 count change of 24.41 mV. This value is a composite of a four-count change of the 12-bit converter (or a one count change of the 10-bit converter) equal to 19.53 mV, plus a one count change of the 12-bit converter equal to 4.88 mV. These large changes are approximately evenly distributed, with at least seven 19.53 mV changes between each. The largest weighted average step size is 20.14 mV, yielding an effective average ADC resolution of 9.96 bits. A device with the minimum 993 active counts would have a total of (869) 19.53 mV steps and (124) 24.41 mV steps.

DI-145 Response Stream

It is noteworthy that the DI-145 always transmits data at a 240 Hz throughput rate and a mechanism to reduce sample rate is not provided. The host program may achieve a reduction in sample rate using selective sampling methods whereby every *n*th point is selected as the converted value. For example, assuming a single enabled channel, a sample rate of 120 Hz is achieved by selecting every other value from the reported data stream. Every third reading is effectively 80 Hz, every fourth yields 60 Hz, and so on. Averaging every *n* values on each channel is more difficult but recommended because it reduces the noise by a factor of the square root of *n*.

The DI-145 legacy data output format is a compressed binary stream of bytes in logical channel sequence. 10-bit ADC channel readings are output as 12-bit numbers as described in the previous section from the lowest to highest numbered enabled channel and the least significant bit of the first (lowest enabled channel) is always cleared. This bit is set in all other bytes to allow the host program to synchronize with the data stream.

The state of the DI-145's digital input bits DO (Remote event) and D1 (Remote stop/start) is also embedded in the binary stream of each transmitted channel. A logic low applied to either bit on DI-145 hardware results in a '0' value inserted for that bit in the data stream.

Stream sequence repeats until data acquisition is halted by the S0 command (0x5330).

Binary Data Stream Examples										
All channels enabled:										
Physical Channel	Word Count	Byte Count	B7	B6	B5	B4	B3	B2	B1	B0
1	1	1	AD4	AD3	AD2	AD1	AD0	D1	D0	0
		2	AD11	AD10	AD9	AD8	AD7	AD6	AD5	1
2	2	3	AD4	AD3	AD2	AD1	AD0	D1	D0	1
		4	AD11	AD10	AD9	AD8	AD7	AD6	AD5	1
3	3	5	AD4	AD3	AD2	AD1	AD0	D1	D0	1
		6	AD11	AD10	AD9	AD8	AD7	AD6	AD5	1
4	4	7	AD4	AD3	AD2	AD1	AD0	D1	D0	1
		8	AD11	AD10	AD9	AD8	AD7	AD6	AD5	1
Channels 2 and 4 enabled:										
2	1	1	AD4	AD3	AD2	AD1	AD0	D1	D0	0
		2	AD11	AD10	AD9	AD8	AD7	AD6	AD5	1
4	2	3	AD4	AD3	AD2	AD1	AD0	D1	D0	1
		4	AD11	AD10	AD9	AD8	AD7	AD6	AD5	1
Channel 2 enabled:										
2	1	1	AD4	AD3	AD2	AD1	AD0	D1	D0	0
		2	AD11	AD10	AD9	AD8	AD7	AD6	AD5	1

DI-145 ADC Coding

The DI-145 transmits a 12-bit binary number for every analog channel conversion. Meaningful information is extracted from these readings by inverting the most significant bit, and treating the result as a two's complement number. Note that counts will change roughly by four for each 19.5 mV step change in applied voltage to yield approximately 10 bits of resolution.

Ideal DI-145 ADC Coding											
AD9*	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	ADC Counts	Applied Voltage
0	1	1	1	1	1	1	1	1	1	2047	+9.995
0	1	1	1	1	1	1	0	1	1	2043	+9.975
...											
0	0	0	0	0	0	1	0	0	0	8	+0.039
0	0	0	0	0	0	0	1	0	0	4	+0.01953
0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	0	0	-4	-0.01953
1	1	1	1	1	1	1	0	0	0	-8	-0.03906
...											
1	0	0	0	0	0	0	1	0	0	-2044	-9.9805
1	0	0	0	0	0	0	0	0	0	-2048	-10.0000

* ADC counts versus applied voltage may vary depending upon calibration differences between individual DI-145 units.

**After inverting to yield a two's complement number.

Part 2 — Extended protocol without DI-194RS backward compatibility

This second section defines the extended command and response set supported by the DI-145, which is not supported by the legacy DI-194RS data acquisition starter kit. It defines a new command that directs the DI-145 to output data using an ASCII format:

DI-145 Extended Command Set		
Command		Action
Hex	ASCII	
0x4830	H0	Places the DI-145 in its extended mode. Data is output as delimited ASCII.
0x4831	H1	Places the DI-145 IN its DI-194 compatibility mode. Data is output as packed binary.
0x4130	A0	Returns "DATAQ"
0x4131	A1	Returns device name (4 bytes)
0x4132	A2	Returns firmware revision, 2 bytes (e.g. $65_{16} = 101_{10}$ for firmware revision 1.01)

The extended mode of operation allows the DI-145 to output sampled analog and digital channel data in an ASCII format, which may be easier to work with in some situations at the application programming level. All the legacy commands previously discussed operate exactly as defined. Each scan (row) of data in an ASCII format consists of a preceding "sc" fixed character pair, then at least one channel of analog data followed by one character indicating the states of the digital inputs. A space delimiter is used between fields, and a carriage return and line feed (x0D0A) terminates each line. Here are example outputs with various numbers of enabled analog channels:

Typical Output as a Function of Enabled Analog Channelst			
Number of enabled analog channels			
4	3	2	1
sc -006 -012 -008 -2048 3	sc -012 -008 -2048 3	sc -008 -2048 3	sc -2048 3
sc -011 -012 -012 -2048 3	sc -012 -012 -2048 3	sc -012 -2048 3	sc -2048 3
sc -001 -012 -012 -2048 3	sc -012 -012 -2048 3	sc -012 -2048 3	sc -2048 3
sc -012 -012 -012 -2048 0	sc -012 -012 -2048 0	sc -012 -2048 0	sc -2048 0
sc -012 -012 -012 -2048 3	sc -012 -012 -2048 3	sc -012 -2048 3	sc -2048 3
sc -012 -012 -012 -2048 3	sc -012 -012 -2048 3	sc -012 -2048 3	sc -2048 3
sc -012 -012 -008 -2048 2	sc -012 -008 -2048 2	sc-008 -2048 2	sc -2048 2
sc -012 -012 -012 -2048 3	sc -012 -012 -2048 3	sc -012 -2048 3	sc -2048 3
sc -016 -012 -012 -2048 1	sc -012 -012 -2048 1	sc -012 -2048 1	sc -2048 1
sc -012 -012 -008 -2048 1	sc -012 -008 -2048 1	sc -008 -2048 1	sc -2048 1
sc -012 -012 -012 -2048 1	sc -012 -012 -2048 1	sc -012 -2048 1	sc -2048 1
sc -012 -012 -008 -2048 3	sc -012 -008 -2048 3	sc -008 -2048 3	sc -2048 3
sc -012 -012 -012 -2048 3	sc -012 -012 -2048 3	sc -012 -2048 3	sc -2048 3

Analog channel values are output as decimal ADC counts as defined in the *DI-145 ADC Coding* section above. Like the binary data format, analog channel data appears in logical channel order, the lowest numbered enabled channel first, and the highest numbered enabled channel last. For example, if channels 2 and 4 are enabled channel 2 data (the lowest enabled channel number) appears first..

The last field in the ASCII mode is the state of the DI-145's two digital inputs:

DI-145 Digital Input Coding		
Applied State (0 = TTL low)		Displayed Value (Hex)
D1 (remote stop/start)	D0 (remote event)	
0	0	"0" (0x30)
0	1	"1" (0x31)
1	0	"2" (0x32)
1	1	"3" (0x33)

[Go to the DI-145 Product Page](#)