DI-1000UHS-1K Output Formats

Introduction

This Tech Note provides a summary of commands you can use to write your own software application using the DI-1000UHS-1K USB Sensor Interface.

Background

The DI-1000U is a 24-bit USB sensor interface with the ability to accept inputs between -2V to +2V and output either raw or calibrated values. It is generally used to amplify and digitize resistive load cells. The maximum data rate of the output of the base model of the DI-1000U is about 60-80 Hz. The DI-1000UHS provides a max output of 500 Hz. Those two devices have a command set that is more extensive than the ones we offer for the DI-1000UHS-1K.

The DI-1000UHS-1K offers data rates of at a minimum 1 KHz. This device has only a few ASCII commands and offers only simple linear fit using the mV/V calibration value. You can only enter the mvolt value and adjust it till the calibrated values match the force applied.

Connections

You can connect the device via the provided miniB to USB A Cable (no external power supply needed) to a Windows PC, Mac or Linux machine.

You should set the latency timer to 1 ms for the port being used. On your PC you can do this using the device manager utility. See the separate app note on how to set the latency timer. You can then connect to the device using Putty, HyperTerm or any other Terminal emulation utility. The connection settings are:

Baud rate: 230400

Data Bits: 8

Parity: None

Stop Bits: 1

Flow Control: None

Commands

If you are writing your own software to read sensor values from the DI-1000HS-1K interface unit, two commands are available to read the values. The commands are case insensitive. Both commands will stream the output continuously - simply send a carriage return (<CR>) to stop the output.
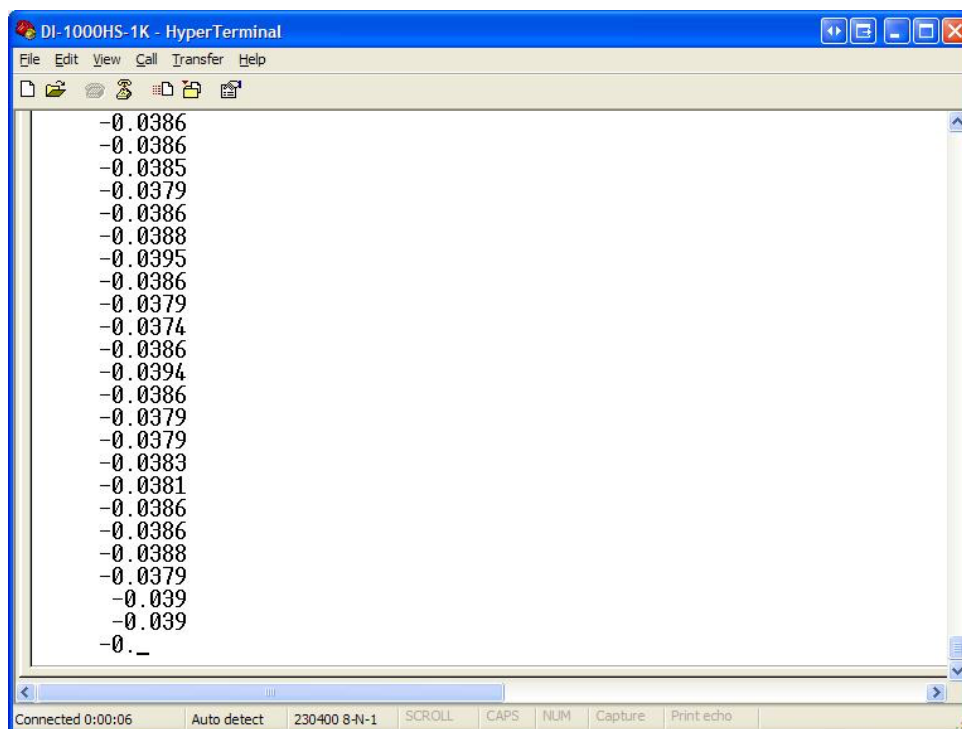
'WC' Command

About 575 readings/sec of data output.

This command outputs the load values in decimals according to the calibrated units (the calibration units are obtained using the UNITS command. The output format is:

```
%12.4f
```

A sample screen shot is shown below.

'H' Command

Above 1000 readings/sec of data output.

This command outputs the raw counts in hexadecimal format. These are the A2D values, not load values. To decode the hexadecimal values easily when writing your own programs,

- the negative hexadecimal values are multiplied by -1 and resulting positive values are simply output with a prefix of "-"

- positive values are prefixed with a space character and then output.

For each value, a print width of 7 is used

- One for the -ve sign if needed, otherwise a space character

- Six for the hexadecimal value

- In order to maintain as high an output rate as possible, a line feed is not printed after each value, only a carriage return.

For example, the raw counts of -193 will be displayed as:

```
–0000C1
```

In Microsoft Excel, you can use the formula **DEC2HEX** and **HEX2DEC** to convert between the 2 formats. Take care to exclude the –ve sign when applying these formulas and only then to multiply the converted values by -1.

In Microsoft.NET (C# or Visual Basic.NET) you can use the following statement:

> decimal_value = Convert.ToInt32(hex_value_string,16)

To convert the raw count values to load values:

1. Issue the SWC command (**S**how **W**eight per **C**ount) to display the Weight/Count in calibrated units. (Of course, you need to issue this command only once at the start of your application.)

2. Multiply the Weight/Count value by the counts (obtained by sending the H command above) to obtain the loading in calibrated units.

3. Perform your own unit conversion as necessary.

Loadstar
SENSORS
Sensing Made Easy

loadstarsensors.com

A sample screen shot is shown below.
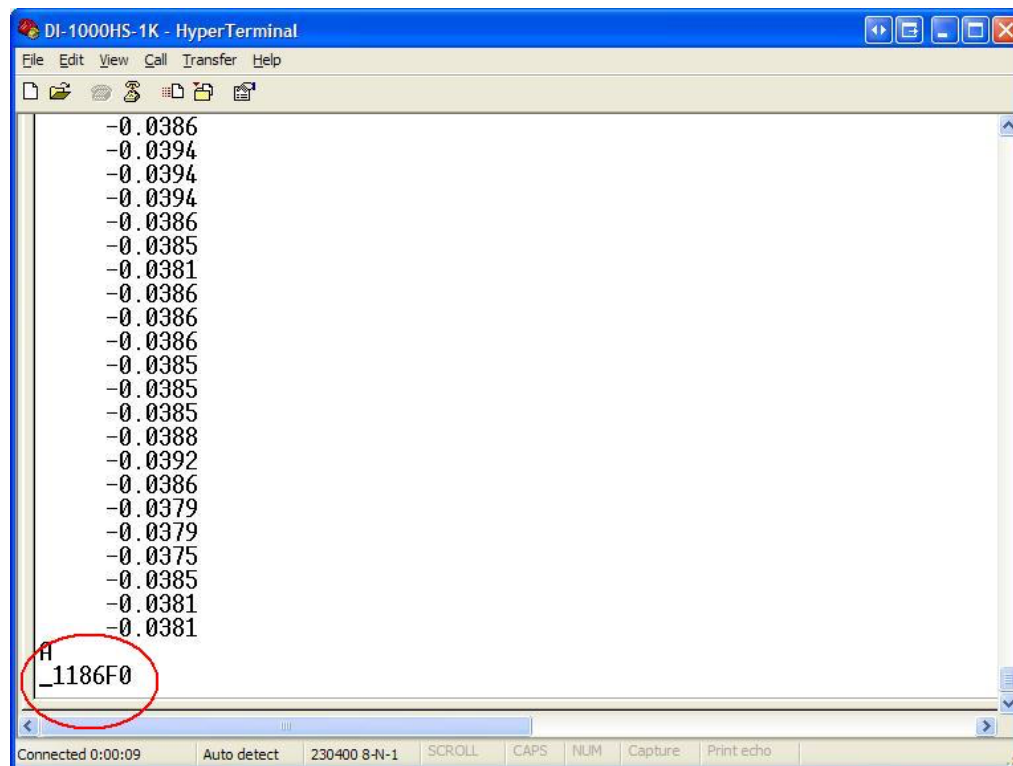
'TARE' or 'CT0' Command

Send the CT0 or TARE command to zero the sensor at any point. For e.g. if you have a scale and want to place an empty container on it and want to refer to that "state" as the zero load state, then issue this command to output zero as the measured load. All subsequent loads or forces will be relative to this zero-load state.

Note: Occasionally, the DI-1000HS-1K will output a -1 as the raw count. Just ignore this value. It is just an artifact of the microprocessor missing a beat while performing the data gathering operation as fast as possible.