Bartington Instruments

DG-1

Digitiser

Data Protocol

Customer Release

A version of Bartington Instruments generic data protocol applied to the DG-1

This data protocol document targets application developers or anyone who wishes to interface to the DG-1. It provides complete information on how to interface and control the DG-1 and among other things contains examples that illustrate the control of DG-1.

Bartington Instruments

Table of Contents

Change History

ISSUE	DATE	AUTHOR	DESCRIPTION
1	12/03/2023	Stephen Okuribido	First release
2	08/04/2023	Stephen Okuribido	Input Range setting in the Reply_Digitizer_Details has been modified to reflect a change in firmware. Range Setting in the Request_Set_Range has been modified to reflect a change in firmware.
3	24/06/2023	Stephen Okuribido	The low pass filter section in the appendix has been updated to reflect changes in the design of the FIR filter. Including the calculation of coefficient values. A third note is added to the REQ_ SET_COEF section. All references to sample rate of 32µs changed to 31.25µs to reflect change in DG1 functionality. In page 12, in the description section of START_DELAY and TIME DELAY. Text stating that 32us need to be added has been removed as it's no longer necessary to do so.
4	22/09/2023	Stephen Okuribido	Changed all references to 60Hz to 100Hz. Changed all references to 30Hz to 10Hz. In the appendix, updated instruction on how to send Autoscan command when using 10Hz and 100Hz filter settings.
5	7/12/2023	Stephen Okuribido	Status flag added for LPF functional state.
6	03/02/25	Stephen Okuribido	On page 14: Added LPF_PRESET field to the REP_DIGITIZER_DETAILS. On page 15: Removed the REQ_GET_COEFF functionality On page 16: Removed the REP_GET_COEFF functionality. On page 17: Replaced REQ_SET_COEFF with REQ_SET_LPF. On page 18: Removed the REQ_STORE_COEFF command. On page 12: Removed the DATA_TYPE parameter from the REQ_SCAN_AUTO_RESULT command.

			References to 100Hz and 10Hz has been removed from the LPF functionality. Removed the section “PROGRAMMING THE FIR LPF” in the appendix. On page 10: Removed REQ_GET_COEFF, REP_GET_COEFF and REQ_STORE_COEFF from the list of messages and added REQ_SET_LPF to the list. Removed 10Hz and 100Hz filter examples from Appendix

Introduction

The Data Protocol details the messages sent and received via the USB interface by the DG-1 digitiser.

AUDIENCE

This document is aimed at software developers and is intended to help them communicate with the DG-1 digitiser.

Frame Structure

The data structure is encapsulated in an HDLC frame comprising of:

- Two start of frame bytes, and two end of frame bytes
- The start frame consists of a DLE and an STX character
- The end of frame consists of a DLE and ETX character
- DLE character = 10h(16)
- STX character = 02h(2)
- ETX character = 03h(3)

Byte	Value
DLE	10h(16)
STX	02h(2)
	Data structure
DLE	10h(16)
ETX	03h(3)

Data Structure

Table 1: Shows the make-up of a data structure.

Bytes	Value	Function
1	UNUSED	
1	Message Type	See Table 2
<250	Parameters	See Table 2 for numbers of parameters for each message
2	Checksum	16-bit checksum

Response of DG-1 to the Message Type byte is detailed in subsequent sections.

Data is structured for minimum wasted time when sampling and transferring data. DG-1 only requires 5 bytes to begin a data transfer – an Address byte, Message Type byte, Parameter byte and two checksum bytes. However, Address byte in the DG-1 is treated as dummy data and is discarded.

DLE Within the Data Structure

When creating an HDLC packet, if a DLE occurs within a data structure, a second DLE should be added immediately after.

For example, a message that looks like:

	10h(16) 02h(2)	Start of frame
DLE within data structure →	0Fh(15) 14h(16) 10h(16) 05h(16) 37h(16) 00h(16)	Data Structure
	10h(16) 03h(16)	End of frame

The value 10h(16) occurs in the data structure. To differentiate from a DLE the value is repeated as follows:

	10h(16) 02h(2)	Start of frame
DLE within data structure → DLE repeated →	0Fh(15) 14h(16) 10h(16) 10h(16) 05h(16) 37h(16) 00h(16)	Data Structure
	10h(16) 03h(16)	End of frame

Therefore, if a packet is received with two consecutive DLE values, the second DLE character must be removed before being used as a valid message. In addition, a check must be made if the DLE character is received to ensure that a second DLE character is not received before interpreting as a genuine DLE.

If extra DLEs need to be inserted into the transmitted message, then the packet transmission time will be affected as extra data bytes will be sent.

Checksum

Checksum calculation is performed on all the bytes in the data structure before adding DLE values (see previous section).

Note: On reception of a HDLC frame any repeated DLEs should be removed first before performing checksum calculations.

The checksum is calculated by adding all the bytes in a data structure to FFFFh (65535) following removal of extra DLEs.

The string below is an example HDLC frame with the data to be used in the checksum in italics and the checksum itself in bold.

10,02,*0F,0A,05,02,04*,**23**,**00**,10,03

HDLC packet (hexadecimal values)

In the byte sequence above the first two byte represent the start of frame, the next byte the address, the next four are data, the next two are the checksum (in bold) and the last two are end of frame. The checksum is performed by adding the bytes shown in italic to 65535 which will result in the value 10023h (65571).

Only the lower two bytes are used so we end up with 0023h (35). When streaming out a packet the lower bytes are transmitted first so 23h will appear before 00h.

For example:

```
char DataBytes[5] = {0x0F,0x0A,0x05,0x02,0x04};
                                          // Data to be used in checksum calculation
char CheckSumBytesResult[2];              // Checksum result to be added to HDLC packet
int checksum = 65535;                     // 0xFFFF

// Now perform the addition
for ( int idx = 0; idx < 5; idx++ )
{
       checksum += DataBytes[idx];
}

// Now lets extract the two lower bytes to get the checksum.
checksum &= 0xFFFF;

//Extract the two checksum bytes
CheckSumBytesResult[0] = (checksum & 0xFF);
CheckSumBytesResult[1] = ((checksum>>8) & 0xFF);

// CheckSumBytesResult[0] is sent first then ChecksumBytesResult[1] within the HDLC packet.
```

Power Up Sleep Mode

Following power up or power cycle the DG-1 goes in to sleep mode to minimize current consumption to allow the USB port to perform the initial enumeration process as a device of less than 100mA.

To wake the device up dummy data of at least one byte must be send to it. After the device is woken up it will be ready to accept protocol commands.

List of Messages

Table 2: Standard messages supported by the DG-1 digitiser.

Name	Description
BRD_CANCEL_OP	DG-1 cancels any current operation and goes into idle mode.
REQ_SCAN_AUTO_RESULT	Scan and auto-reply for several times or until receive REQ_CANCEL_OP.
REP_SCAN_AUTO_RESULT	Reply to REQ_SCAN_AUTO_RESULT.
REQ_GET_DIGITIZER_DETAILS	Requests configuration details of DG-1 digitiser.
REQ_SET_RANGE	Sets the input range for the analogue inputs.
REP_DIGITIZER_DETAILS	Reply to REQ_GET_DIGITIZER_DETAILS with configuration information.
REQ _SET_COEF	Set coefficient data
REQ _GET_COEF	Request coefficient data
REQ_ SWITCH_AUX_PWR	Turns power to the ADCs and analogue circuitry on/off.
REQ_LPF_MODE	Sets the low pass filter (LPF) to active or bypass mode.
REQ_SET_LPF	Sets the pre-set setting of the LPF.
REP_GENERIC_REPLY	This reply returns the last message received and the result of processing that message in the form a pass/fail byte. Some request messages have their own respective replies and do not use this message. But for those messages that do not, this generic reply is used.

Details of Messages

This section only details the data structure. Messages must be encapsulated in an HDLC frame as described above. All data in messages are little endian.

Note: Following power cycle, at least one byte must be sent to the DG-1 before protocol commands can be sent.

[0x00] Cancel Operation (BRD_CANCEL_OP)

This message is used to cancel any current operation in the DG-1. After receiving this message, the DG-1 will go into its idle mode.

Name	Size (Bytes)	Format	Description	Position of Byte in data structure
UNUSED	1			1
MESSAGE_TYPE	1	Uint8	0x00	2
CHECKSUM	2	Uint16	16-bit checksum	3 - 4

[0x04] Repeated Scan Request (REQ_SCAN_AUTO_RESULT)

DG-1 is placed in an autonomous mode, carrying out measurements at a rate determined by the parameters supplied.

This message requests 'N' number of measurements (where 'N' is the repeat count) with a specified time gap between each measurement. The DG-1 will reply at the end of each measurement, before starting the next, repeating it for the requested number. To interrupt this process, send a **REQ_CANCEL_OP** message to the DG-1. If the repeat count is set to 0 then the DG-1 will indefinitely repeat the measurements. The DG-1 will only stop when a **REQ_CANCEL_OP** message is received.

It is recommended that after issuing this message that the only other message sent to the DG-1 should be the **REQ_CANCEL_OP,** as results could be unpredictable due to potential message clashes.

A *[0x10] Request Switch Auxiliary Power On/Off* command must be issued first with SWITCH_STATUS set to 1 prior to issuing this command.

Notes:
Auxiliary power must be switched on, or the device will not return any digitiser data. See *[0x10] Request Switch Auxiliary Power On/Off*.

SAMPLE_SUM_COUNT SETTING:
If SAMPLE_SUM_COUNT has been set to zero, then the DG-1 will not respond.

This setting causes the DG-1 to take several ADC samples equal to the SAMPLE_SUM_COUNT setting. Sample rate is approximately 31.25µs.

If the SAMPLE_SUM_COUNT multiplied by the sample rate equates to an acquisition time greater than the TIME_DELAY, then the TIME_DELAY setting will be ignored.

The packet transmission time is dependent on the data rate, and also will change if extra DLE's need to be inserted into the transmitted message (see section 'DLE Within the Data Structure'). The packet transmission time will influence the measurement.

If SAMPLE_SUM_COUNT is set too low ADC samples will be skipped. Conversely higher SAMPLE_SUM_COUNT settings will slow down the rate that messages are sent.

TIME DELAY SETTINGS:
Where TIME_DELAY is used to make timed interval measurements, users must ensure that the TIME_DELAY setting is greater than the packet transmission time, as well as the measurement time defined by the SAMPLE_SUM_COUNT.

Please make note of this where measurement interval timing is critical to your design.

START DELAY SETTINGS:

Where START_DELAY is used to delay the first measurements, users should be aware that the USB interface may have their own delay. This may be dependent on the PC used.

Name	Size (Bytes)	Format	Description	Position of Byte in data structure
UNUSED	1			1
MESSAGE_TYPE	1	Uint8	0x04	2
UNUSED	1			3
START_DELAY	3	Uint24	Delay period before the first measurement starts in milliseconds, range is 0 – 16777215 (0xFFFFFF) ms.	4 – 6
TIME_DELAY	3	Uint24	Delay period between measurements in milliseconds, range is 0 – 16777215 (0xFFFFFF) ms. In addition, if the SAMPLE_SUM_COUNT setting results in a larger delay, it will override this setting.	7 - 9
SAMPLE_SUM_COUNT	3	Uint24	Sets the number of samples that are summed together in the DG-1, resulting in one measurement. This is only valid for magnetic measurements. Limited currently to 0x0000FE (1 to 254)	10 - 12
REPEAT_COUNT	3	Uint24	Number of measurements requested, 1 - 0xFFFFFF. If zero, measurements continue indefinitely until a REQ_CANCEL_OP message is received.	13 - 15
UNUSED	3	Uint24		16 – 18
CHECKSUM	2	Uint16	16-bit checksum	19 – 20

[0x21] Reply scan result (REP_SCAN_AUTO_RESULT)

This reply returns measurements in response to REQ_SCAN_AUTO_RESULT.

Name	Size (Bytes)	Format	Description	Position of Byte in data structure
UNUSED	1			1
MESSAGE_TYPE	1	Uint8	0x21	2
X _VALUE	4	Int32	Magnetometer X axis value	3-6
Y _VALUE	4	Int32	Magnetometer Y axis value	7-10
Z _VALUE	4	Int32	Magnetometer Z axis value	11-14
PACKET_NUMBER	2	Uint16	A sequential count generated at the time the message was created. Counts from 0 to 65535 and then wraps.	15-16
CHECKSUM	2	Uint16	16-bit checksum	17-18

The DG-1 value (X_VALUE; Y_VALUE; and Z_VALUE) is converted to magnetic units as follows:

$$MagneticUnits(nT) = \frac{DG1_{Value} \times MAG_{FSD}}{SAMPLE \sum COUNT \times 2^{23}}$$

Where:
MAG_FSD is returned in the **REP_DIGITIZER_DETAILS** message.
SAMPLE_SUM_COUNT is sent to the DG-1 in the **REQ_SCAN_AUTO_RESULT** message.

[0x05] Digitizer Details Request (REQ_GET_DIGITIZER_DETAILS)

Message sent to the DG-1 to check that it is functional and to retrieve some details of the DG-1. See **REP_DIGITIZER_DETAILS**.

Name	Size (Bytes)	Format	Description	Position of Byte in data structure
UNUSED	1			1
MESSAGE_TYPE	1	Uint8	0x05	2
CHECKSUM	2	Uint16	16-bit checksum	3 - 4

[0x08] Set LPF Mode (REQ_LPF_MODE)

Determines whether sampled data should pass through the LPF to be filtered or bypassed and routed straight to the PC. Following each power cycle, the LPF mode is set to bypass. The digitiser replies to this message with a REP_SENSOR_DETAILS.

Name	Size (Bytes)	Format	Description	Position of Byte in data structure
UNUSED	1	Uint8		1
MESSAGE_TYPE	1	Uint8	0x08	2
MESSAGE_VALUE	1	Uint8	0 – Bypass and disable LPF. (Default) 1 – Enable LFP.	3
CHECKSUM	2	Uint16	16-bit CRC	4 - 5

[0x24] Reply Digitizer Details (REP_DIGITIZER_DETAILS)

The full set-up information in response to **REQ_GET_DIGITIZER_DETAILS**.

Name	Size (Bytes)	Format	Description	Position of Byte in data structure
UNUSED	1			1
MESSAGE_TYPE	1	Uint8	0x24	2
DEVICE_TYPE	2	Uint16	0x0007 = DG-1 Digitiser	3 - 4
UNUSED	2	Uint16		5 - 6
FIRMWARE_FPGA_VER	2	Uint16	Starts from 0x0001	7 - 8
PROTOCOL_VER	1	Uint8	Starts from 0x01	9
SERIAL_NO	2	Uint16	Unique 16-bit serial number	10 - 11
MAG_FSD	4	Uint32	Magnetometer full scale in nT e.g. 1000000 (1mT)	12 - 15
INPUT_RANGE	1	Uint8	0 = ±8V 1 = ±3V 2 = ±10V	16
STATUS	1	Uint8	Bit0 = 0 – Aux power off = 1 – Aux power on Bit1 = 0 – LPF disabled = 1 – LPF enabled	17
LPF_PRESET	1	Uint8	1 – 5KHz. 2 – 3KHz. 3 – 1KHz. Others = Invalid	18
UNUSED	11			19 – 29
CHECKSUM	2	Uint16	16-bit checksum	30 - 31

[0x42] Request Set Range (REQ_SET_RANGE)

This message sets the voltage and magnetic measurement range for the digitizer in non-volatile memory. The DG-1 replies to this message with a REP_DIGITIZER_DETAILS.

Name	Size (Bytes)	Format	Description	Position of Byte in data structure
UNUSED	1			1
MESSAGE_TYPE	1	Uint8	0x42	2
MAG_FSD	4	Uint32	Magnetometer full scale in nT e.g. 1000000 (1mT)	3 – 6
RANGE_SETTING	1	Uint8	0 = ±8V 1 = ±3V 2 = ±10V Others = UNUSED	7
UNUSED	3			8 – 10
CHECKSUM	2	Uint16	16-bit checksum	11 – 12

RANGE_SETTING and MAG_FSD should be set to match the magnetometer connected to the DG-1.

[0x15] Set Low Pass Filter (REQ_ SET_LPF).

This message sets the low pass filter characteristics to a preset value set by LPF_PRESET.

The digitiser replies to this message with a REP_DIGITIZER_DETAILS with the current LPF_PRESET setting.

Name	Size (Bytes)	Format	Description	Position of Byte in data structure
UNUSED	1			1
MESSAGE_TYPE	1	Uint8	0x15	2
UNUSED	1			3
LPF_PRESET	1	Uint8	1 – 5KHz. 2 – 3KHz. 3 – 1KHz. Others = UNUSED	4
UNUSED	1			5
UNUSED	4			6 - 9
CHECKSUM	2	Uint16	16-bit CRC	10 - 11

[0x10] Request Switch Auxiliary Power On/Off (REQ_ SWITCH_AUX_PWR)

Master sends this command to switch the power to the ADCs and the analogue circuitry on or off.

The power to the ADCs and analogue circuitry is off following power cycle.

The digitiser replies to this message with a REP_DIGITIZER_DETAILS.

Name	Size (Bytes)	Format	Description	Position of Byte in data structure
UNUSED	1			1
MESSAGE_TYPE	1	Uint8	0x10	2
SWITCH_STATUS	1	Uint8	0 = Off 1 = On	3
CHECKSUM	2	Uint16	16-bit checksum	4 – 5

[0x2A] Generic reply to messages (REP_GENERIC_REPLY)

Generic response used as a reply to a certain number of request messages. This reply returns the last message received and the result of processing that message in the form a pass/fail byte.

Name	Size (Bytes)	Format	Description	Position of Byte in data structure
UNUSED	1			1
MESSAGE_TYPE	1	Uint8	0x2A	2
LAST_MESSAGE	1	Uint8	Last received message	3
RESULT	1	Uint8	Process result: Success = 0xAA Fail = 0x55	4
CHECKSUM	2	Uint16	16-bit checksum	5 – 6

APPENDIX

PRESET SELECTION FOR THE FIR FILTER

The FIR filter consist of 3 corner frequency preset settings.
1kHz
3kHz
5kHz

These presets are selectable by writing a REQ_SET_COEFF command to the filter with the appropriate parameters. For each instruction below steps 1 and 2 only need to be carried out once then Autoscan commands can be issued repeatedly.

SETTING AND USING PRESET 1 (5kHz)

Step1 :
Send REQ_ SET_COEF command with COEF_PARAMETER = 2 and COEF_INDEX = 1

Step2 :
Send REQ_LPF_MODE command with MESSAGE_VALUE = 1

Then with each REQ_SCAN_AUTO_RESULT command ensure that TIME_DELAY = 0 and SAMPLE_SUM_COUNT = 1.

SETTING AND USING PRESET 2 (3kHz)

Step1 :
Send REQ_ SET_COEF command with COEF_PARAMETER = 2 and COEF_INDEX = 2

Step2 :
Send REQ_LPF_MODE command with MESSAGE_VALUE = 1

Then with each REQ_SCAN_AUTO_RESULT command ensure that TIME_DELAY = 0 and SAMPLE_SUM_COUNT = 1.

SETTING AND USING PRESET 3 (1kHz)

Step1 :
Send REQ_ SET_COEF command with COEF_PARAMETER = 2 and COEF_INDEX = 3

Step2 :
Send REQ_LPF_MODE command with MESSAGE_VALUE = 1

Then with each REQ_SCAN_AUTO_RESULT command ensure that TIME_DELAY = 0 and SAMPLE_SUM_COUNT = 1.