Log4 Serial Communications Protocol

Version 1.0

TEKTYTE

1 Contents

2 Description

This document outlines the general protocol followed by Log4 Devices for communication over Serial connection. The physical layer is RS232 UART with 5V logic levels, No hardware flow control, one stop bit, 8 data bits, and no parity bits. The default Baud rate is set to 115200. This value can be changed.
Throughout the document:

- Master is the central data collection unit.
- Slave is the (possibly multiple) data collection devices which may have one or more channels of data (current, voltage, temp, hall-effect)

3 Requirements

Following requirements needs to be a fulfilled for this document to be applicable:

- Able to change device specific parameters.
  - Sampling rate / baud rate
- Able to read device specific features and configurations
  - Device type identifier (will correspond with a master database entry of slave capabilities)
- Able to receive data from the device.

4 General structure of a packet

Data from Log4 devices are sent in the form of packets. Following provides rules on which a packet is structured:

- Start byte ':'
- Address (1 byte) (currently unused, set to 0x01)
- Command (1 byte)
- Data byte count (1 byte)
- Data (0 – 255 bytes)
- Stop byte '\n'

When multi-byte "Data" values are transmitted, they will be sent least-significant byte first.

5 Command Codes

Table 1 provides information for different commands available in the protocol.

Table 1 Command codes and description

Command Name	Code	Expected data size[1] (master, slave)	Long description
CMD_ERROR	0x00	(1-255,1-255)	Indicate an error has occurred
GET_ID[2]	0x01	(0,5-255)	Get device identification & Version
KEEP_ALIVE	0x02	(0,0)	Poll the slaved device to ensure its still there.
GET_CHANNELS[2]	0x03	(0,1-255)	Get the number and type of device channels
SET_BAUD_RATE[3]	0x04	(4,-)	Set the baud rate for the device
GET_BAUD_RATES[2]	0x05	(0,0-255)	Get compatible board rates of the device
SET_SAMPLING[3]	0x06	(14+,-)	Set Sampling Mode
GET_SAMPLING[2]	0x07	(0,19)	Get Current Sampling Mode
SET_DATE_TIME[3]	0x08	(7,-)	Set the current date/ time on the device
GET_DATE_TIME[2]	0x09	(0,7)	Get the current date/time on the device
SLAVE_DATA[4,5]	0x0B	(-,1-255)	Contains a sample of logged data

[1] If master or slave do not transmit this type of packet a '-' is put instead of a size range.
[2] Packet has differing data layouts whether master or slave is sending this type of packet.
[3] Only master can send this command
[4] Only slave can send this command
[5] Packet size and format differ between Log4.USB and Log4.PoE

6 Error Codes

If the received command in the packet is the error command, the first byte in the data section is the error type. Table 2 and Table 3 details which error was sent.
If the data byte count is greater than 0x01, then the rest of the data section is an ASCII string that accompanies the error and should be displayed to the user.

6.1 Generic Packet Error codes

Table 2 Generic packet error codes with description

Error name	code	Long description
ERR_INVALID_DATA	0x01	The previous packet contained invalid data for the command
ERR_INVALID_CMD	0x02	The previous command was not recognized as valid
ERR_TIME_OUT	0x03	The previous packet took too long to complete
ERR_INVALID_COUNT	0x04	The data count did not match the amount of data bytes received
ERR_BUSY	0x05	Unable to process request at the moment
ERR_PACKET_TOO_LARGE	0x07	The last received packet was too large to buffer
ERR_SLAVE_DEBUG_MSG	0x08	The slave encountered internal error

6.2 Packet Specific Error codes

Table 3 Packet specific error codes with description

Error name	code	Long description
ERR_INVALID_CHAN	0x21	The channel you are addressing doesn't exist on this device
ERR_INVALID_SAMPLE_RATE	0x22	The sample rate was unable to be selected

7 Master Commands

7.1 GET_ID

Data byte count: 0-255
Command: 0x01
Data structure:
Description:

Use to poll the Virtual COM Ports for the correct slave. The slave returns its serial number product model and firmware version number.

7.2 KEEP_ALIVE

Data byte count: 0
Command: 0x02
Data structure:
Description:

Use to poll slave device to ensure it is still connected. The slave will respond with the same packet command and no data.

7.3 GET_CHANNELS

Data byte count: 0
Command: 0x03
Data structure: No data
Description:

Used to poll the slave device for the number and type of channels it supports. See slave Command section for the response data format.

7.4 SET_BAUD_RATE

Data byte count: 4
Command: 0x04
Data structure:

Bytes 0-3: Unsigned 32bit Integer (*Baud_rate*)

Description:

Set the baud rate of the slaved device. Device should return CMD_ERROR if the baud rate is not available.

7.5 GET_BAUD_RATE

Data byte count: 0
Command: 0x05
Data structure:
Description:

Request an ASCII CSV list of the supported device baud rates.

7.6 SET_SAMPLING

Data byte count: 14+
Command: 0x06
Data structure:

Byte 0: (RESERVED, Set to 0x01)
Bytes 1-4: Unsigned 32-bit Integer (Sampling Period, milliseconds)
Byte 5: Alarm Type (0: none, 1: audio, 2: visual, 3: both)
Bytes 6-9: (RESERVED, Set to 0x00000001)
Bytes 10-13: Unsigned 32-bit Integer (Alarm enable mask)
Bit mask of enabled alarms (1: enabled, 0: disabled): Low alarms are alarms that are triggered when the channel goes under the specified value, High alarms are trigger when the channel goes over the specified value.
Channel number can be deduced by reading the contents of the GET_CHANNELS command and the order of the channels returned. For example the Log4.PoE will return `"-4uI,-4mV,-4nP,-4uI,-4mV,-4nP"` from the GET_CHANNELS command. This indicates that Ch0 is the first current channel and Ch3 is the second current channel.

	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
Byte1	Ch0 Low	Ch0 High	Ch1 Low	Ch1 High	Ch2 Low	Ch2 High	Ch3 Low	Ch3 High
Byte2	Ch4 Low	Ch4 High	Ch5 Low	Ch5 High	Ch6 Low	Ch6 High	Ch7 Low	Ch7 High
Byte3	Ch8 Low	Ch8 High	Ch9 Low	Ch9 High	Ch10Low	Ch10High	Ch11Low	Ch11High
Byte4	Ch12Low	Ch12High	Ch13Low	Ch13High	Ch14Low	Ch14High	Ch15Low	Ch15High

Bytes 14-17: 32-bit Integer (First Enabled Alarm Value)
Bytes 18-21: 32-bit Integer (Second Enabled Alarm Value)
Bytes 22-25: 32-bit Integer (Third Enabled Alarm Value)
……
Byte ((n-1)*4+14) to ((n-1)*4+17): Unsigned 32-bit Integer (n^{th} Enabled Alarm Value)
Only Alarm values for enabled alarms should be in the packet and must appear in the same order as described in the Bitmask table above.

Example:

- Ch0 High and Ch1 High alarms are to be enabled with the values (0x0123, 0x456789AB) respectively
- sample period of 7 ms.
- Both audible and visual alarms are enabled

Data Section Contents (Hexadecimal Values)

(reserved)	Sample period	Alarm type	(reserved)	Alarm Bitmask	First Alarm (ch0 High)	Second Alarm (ch1 High)
01	07 00 00 00	03	01 00 00 00	0A 00 00 00	23 01 00 00	AB 89 67 45

Description:

Set global sampling rate and individual channel alarms.

7.7 GET_SAMPLING

Data byte count: 0
Command: 0x07
Data structure:
Description:

Get the current sampling mode, the sampling rate and bitmask indicating active channel alarms and list of active channel values.

7.8 SET_DATE_TIME

Data byte count: 7
Command: 0x08
Data structure:

Byte 0-1: uint16 year (accepted range: 2012 - 2075)
Byte 2 : uint8 month (accepted range: 1-12)
Byte 3 : uint8 day (accepted range: 1-31)
Byte 4 : uint8 hour (24 hour format, accepted range: 0-23)
Byte 5 : uint8 min (accepted range: 0-59)
Byte 6 : uint8 sec (accepted range: 0-59)

Description:

Sets the current date and time on the micro controller. No packet is sent in response when the data format is correct. An error packet is sent if an incorrect packet was sent.

7.9 GET_DATE_TIME

Data byte count: 0
Command: 0x09
Data structure:
Description:

Requests the current date and time on the slave. The slave response packet data is in the same format as SET_DATE_TIME and has the GET_DATE_TIME command byte.

8 Slave Commands

8.1 GET_ID

Data byte count: 6-255

Data structure:

Byte 0-3: Unsigned 32-bit Integer (Device Serial Number)

Byte 4: Unsigned 8-bit Integer (Hardware Revision Number)

Byte 6-255: product model string and firmware version string separated with ":"

Description:

Response packet to Master GET_ID request packet

8.2 GET_DATE_TIME

Data byte count: 0

Command: 0x09

Data structure:

Description:

Gets the current date and time on the microcontroller.

8.3 SLAVE_DATA

Data byte count: USB 20 / PoE 32

Data structure:

Log4 USB

Byte 0-7 : uint64_t Timestamp in milliseconds since epoch (midnight 01/01/1970)

Byte 8-11 : int32_t Current in micro amps

Byte 11-15 : int32_t Bus Voltage in millivolts

Byte 16-19 : int32_t Power in microwatts

Log4 PoE

Byte 0-7 :uint64_t Timestamp in milliseconds since epoch (midnight 01/01/1970)

Byte 8-11 : int32_t Ch1 Current in microamps

Byte 11-15 : int32_t Ch1 Bus Voltage in millivolts

Byte 16-19 : int32_t Ch1 Power in microwatts

Byte 20-23 : int32_t Ch2 Current in microamps

Byte 24-27 : int32_t Ch2 Bus Voltage in millivolts

Byte 28-31 : int32_t Ch2 Power in microwatts

Description:

The data captured by the slave (e.g. voltage, current, temperature etc.). It is sent out for every sample taken on the device.

8.4 GET_CHANNELS

Data byte count: 1-255

Data structure:

Comma separated ASCII values describing each channel: Comma's separate channel descriptions which contain four character. Each character describes a different property of the channel and are listed as follows

1. whether it's signed (-) or unsigned(+),
2. data size in bytes(1-9),
3. unit scaling (shown in table below)
4. Type of value (shown in table below).

Scaling unit	Designator	Scaling Factor
pico	'p'	10^{-12}
nano	'n'	10^{-9}
micro	'u'	10^{-6}
milli	'm'	10^{-3}
none	'0'(zero)	1
kilo	'k'	10^{3}
mega	'M'	10^{6}
giga	'G'	10^{9}

Value Type	Value designator	Base measurement unit
Current	'I'	Amp
Voltage	'V'	Volt
Power	'P'	Watt

Example:

For a device that has 3 channels with the following characteristics

Channel 1	Channel 2	Channel 3
16-bit signed Current in microamps	16-bit unsigned Voltage in millivolts	32-bit signed Power in nanowatts

The data field would be "-2uI,+2mV,-4nP"