

Instromet Weather Systems Ltd

Datalogger : 480 1000 148

Protocol List

Version: 1.0

Date: 25/APR/2012

Description:

Detail of commands used to communicate between Datalogger & host PC.

Revision History:

VersionDate Changes

1.0 25/APR/2012 Initial release.

Hardware specification:

- RS232C (+/- 12V levels)
- 3-wire (Tx, Rx, GND)
- 115200bps 8N1

Protocol:

The Datalogger will always act as slave to the host PC as master. Sequence of operation for all commands is as follows:

Host PC action Datalogger action

Send command header	
Send data if specified by command	
Send command terminator	
[Perform requested operation]	
Return command acknowledgement	
Return data if specified by command	
Return checksum	
Return command terminator	

Error checking is included in data from the datalogger, but error correction is not.

In the event of the Datalogger receiving corrupt data from the PC it will not respond, or will respond with an error that does not conform to the expected response. Therefore, the PC must time-out & resend the data (and/or warn user of communication failure).

In the event of the PC receiving corrupt data from the Datalogger it must be discarded. The PC should then re-request the same data (and/or warn user of communication failure).

No non-repeatable commands will be included in the protocol. I.e. commands to download a data packet, and to delete the packet must be implemented separately so that download may be confirmed before the data is deleted.

Protocol Detail:

All commands are sent & received in ASCII format for convenience & compatibility.

Command format as sent from PC :

Example command:

RDLG,58

- Command mnemonic (RDLG). This describes the command (ReaD LoGs in this case) and is upper-case alpha (A-Z) only. Length may be from 1-31 characters.
- Comma delimiter(s) (,). Used to separate command and arguments.
- Optional data, as specified by command (58), and comma delimited (,). Numbers are presented in decimal format, and leading zeros should be removed (but may be accepted/ignored if present). In addition to 0-9, the decimal point and minus sign (-) are also permitted if within the context of the argument. String arguments must be presented within inverted commas (" "), and may contain any ASCII characters except the inverted comma, or .
- Terminator (.). The ASCII characters for Carriage Return (\$0D), and Line Feed (\$0A) appended to all commands. Software should treat as the terminator and ignore , although both must be transmitted for compatibility.

Reply format as sent from Datalogger:

Example reply:

rdlg,1,2,3,4,5,6,7,8,9,123

- Command mnemonic (rdlg). This will match the command sent, although will be in lower-case to distinguish it as the reply.
- Comma delimiter(s) (,). See above.
- Optional data, as specified by command (1,2,3,4,5,6,7,8,9). See above, although it should be noted that outgoing data & incoming data do not generally consist of the same number or type of fields.
- Checksum (123). This is the 8-bit decimal complement of the sum of all previous ASCII characters in the command string.
- Terminator (.). See above.

Some commands may invoke multiple replies, an example of this would be the RDLG,n command (Read Logs), which would require n replies. Each reply would be in the format detailed above, and should be confirmed by the command mnemonic, and identified by the arguments.

Timing:

Inter-byte interval:

PC to Datalogger Datalogger to PC

min. max. min. max.

0 Sec. 30 Sec. 0.0 mSec 1.0 mSec

(typical)

Response interval:

Equal to time taken to perform command, plus comms overhead.

min. max.

0.0 mSec 500.0 mSec

Memory organisation:

Settings & pointers are stored in EEPROM on the processor (1024 bytes, arranged as 170 32-bit error-corrected longwords). See Settings for details.

Logs are stored on a separate flash chip (2MB, arranged as 32768 60-byte error corrected entries).

Settings:

Internal EEPROM registers on the processor. 170 locations available, each is 32-bit.

0 = Spurious writes – do not use.

5 = Thermistor 1 calibration – signed value to add to temperature reading (millicelsius).

6 = Thermistor 2 calibration – signed value to add to temperature reading (millicelsius).

11 = Log interval (seconds) 1-65535.

12 = Log ID – Unique (wraps to zero after $2^{32}-1$), incrementing number written to every log.

13 = Internal read pointer.

14 = Internal write pointer.

Note – in normal operation, setting #11 (log interval) is the only one likely to be changed.

Command List:

Note: Checksum & terminators omitted for clarity.

RDLG - Read log(s)

Example: RDLG,n

Arguments: n = Number of logs to read. (8-bit)

Action: Read logs from the datalogger. Once the requested number, or all remaining logs have been sent, the command will finish. Note that the internal read pointer is NOT progressed by this command – see PRLG.

Response: rdlg,id,type,int,time,date,t1mn,t1mx,t1av,t2mn,t2mx,t2av,rhmn,rhmx,
rhav,psmn,psmx,psav,wsmn,wsmx,wsav,wdir,sun,rain

(Multiple responses permitted)

- 1 Arguments: id = Unique, incrementing ID number. (32-bit)
- 2 type = Log type - defines further fields. (8-bit)
- 3 int = Period that this log relates to (16-bit)
- 4 time = Time stamp in HH:MM:SS format.
- 5 date = Date stamp in DD:MM:YYYY format.
- 6 t1mn = Temperature #1 minimum value during log period (degC to 1 decimal place, eg. 15.6).
- 7 t1mx = Temperature #1 maximum value during log period.
- 8 t1av = Temperature #1 average value during log period.
- 9, 10, 11 t2.. = Temperature #2 min/max/average.
- 12, 13, 14 rh.. = Relative Humidity min/max/ave (% to 1 dp. eg. 12.3)
- 15, 16, 17 ps.. = Air pressure min/max/ave (mbar to 0 dp. eg. 1023)
- 18, 19, 20 ws.. = Wind speed min/max/ave (m/s to 1 dp. eg. 23.4)
- 21 wdir = Instantaneous wind direction (degrees to 0 dp. eg. 90)
- 22 sun = Rolling counter for sunshine (hours to 2 dp. eg. 12.34)*
- 23 rain = Rolling counter for rainfall (mm to 2 dp. eg. 56.78)*

* Rolling counters can be used to indicate the quantity by subtracting the figure from a previous log. If there is a wrap (result is <0), then add $(2^{16})/100$ (655.36).

Note that if data is not available in any field (sensor not present at the time the log was created), the field will be left blank but the delimiting comma will still be present.

PRLG – Progress logs(s)

Example: PRLG,n

Arguments: n = Number of logs to progress. (32-bit)

Action: Progress internal read pointer. This command must be used to advance the pointer past logs that have been correctly read. In effect, it 'discards' logs once they have been verified as ok at the PC. Typically used after a RDLG command.

Response: prlg,n

Arguments: n = Number of logs progressed. Will equal the requested amount unless there are fewer logs available on the datalogger. (32-bit)

RGLG – Regress log(s)

Example: RGLG,n

Arguments: n = Number of logs to regress. (32-bit)

Action: Regress internal read pointer. This command may be used to reverse the pointer over logs that have been previously progressed. In effect, it 'retrieves' logs once they have been discarded. Typically used if data has been lost on the PC.

Response: rglg,n

Arguments: n = Number of logs regressed. Will equal the requested amount unless there are fewer logs available on the datalogger. (32-bit)

LGCT – Log count

Example: LGCT

Arguments: none

Action: Request the number of logs available on the datalogger.

Response: lgct,n

Arguments: n = Number of logs available on the datalogger. (32-bit)

RDST – Read setting (EEPROM register)

Example: RDST,adr

Arguments: adr = Register address. (8-bit)

Action: Read setting from EEPROM memory. See Settings for details.

Response: rdst,adr,dat

Arguments: adr = Register address confirmation. (8-bit)

dat = Register data, or default if fail. (32-bit)

WRST – Write setting (EEPROM register)

Example: WRST,adr,dat

Arguments: adr = Register address. (8-bit)

dat = Register data. (32-bit)

Action: Read setting from EEPROM memory. See Settings for details.

Response: wrst,adr,dat

Arguments: adr = Register address confirmation. (8-bit)

dat = Register data confirmation, or default if fail. (32-bit)

RDTM – Read time (internal real-time clock)

Example: RDTM

Arguments: none

Action: Read current time & date.

Response: rdtm,date,time

Arguments: date = Date in YYYYMMDD format.

time = Time in HHMMSS format.

WRTM – Write time (internal real-time clock)

Example: WRTM,date,time

Arguments: date = Date in YYYYMMDD format.

time = Time in HHMMSS format.

Action: Read current time & date.

Response: rdtm,date,time

Arguments: date = Date confirmation in YYYYMMDD format.

time = Time confirmation in HHMMSS format.

RDLV – Read live data

Example: RDLV,n

Arguments: none

Action: Read live data from the datalogger. Note this is the current data, not historical information from the logs.

Response: rdlv,t1,t2,rh,ps,wspd,wdir,sun,rain

- 1 Arguments: t1 = Temperature #1 (degC to 1 decimal place, eg. 15.6).
- 2 t2 = Temperature #2 (degC to 1 decimal place, eg. -2.3).
- 3 rh = Relative Humidity (% to 1 dp. eg. 12.3)
- 4 ps = Air pressure (mbar to 0 dp. eg. 1023)
- 5 wspd = Wind speed (m/s to 1 dp. eg. 23.4)
- 6 wdir = Wind direction (degrees to 0 dp. eg. 90)
- 7 sun = Rolling counter for sunshine (hours to 2 dp. eg. 12.34)*
- 8 rain = Rolling counter for rainfall (mm to 2 dp. eg. 56.78)*

* Rolling counters can be used to indicate the quantity by subtracting the figure from a previous read of live data. If there is a wrap (result is <0), then add $(2^{16})/100$ (655.36). Note that if data is not available in any field (sensor not present), the field will be left blank but the delimiting comma will still be present.

END OF DOCUMENT.