2023

Modular Data Logger DL Series

PROGRAMMER GUIDE

LAN Interface:

DL series data Loggers are equipped with Ethernet port and it can connect to the PC via Ethernet cable. The data will send with UDP Protocol in unicast mode. It is possible to set the IP address and port number on the device.

Data Frame (Packet) :

All the communication with the software will done using a standard data frame:

BG: beginning of the frame is always equal to 2

Len: total frame length

CMD: Command

D: Data

LRC: LRC value for error handling (refer to LCR formula)

*Each box is represented of one byte.

By sending a frame device at the first step calculate the LRC and check the equality with received LRC to be sure that the data is not corrupted. then recognize the required action by frame command value and send specified answer. there is two type of frame, the frame for an action for example start test, in this case device send back the received frame as an answer so software can check the received frame which should be as same as sent one. The other type of frame is for reading or writing data and settings value so for reading frame, device send back specific answer according to the received command. In all condition it is better to calculate the LRC and checked with received LRC to be sure that data is not corrupted.

LRC Formula:

It is a standard method for determine the data corruption. For calculate LRC, you have to XOR bytes of the frame and the final result will be LRC. Note that BG and LRC should not use in LRC calculation.

LRC=Len(XOR)CMD(XOR)D0(XOR)Dn

For example, in test frame LRC is equal to 52:

LRC=4(XOR)48=52

Frame Description:

Function	Command	Length	Description
Set Channel To Zero	0x20	5	
Get Channel Value	0x22	5	Data will represent the channel structure
Start Log	0x23	4	
Stop Log	0x24	4	
Read Log Parameter	0x30	4	Data will represent the Log structure
Write Log Parameter	0x31		Data will represent the Log structure
Read Sensor	0x32	4	Data will represent the Sensor structure
Write Sensor	0x33	37	Data will represent the Sensor structure
Read Net Parameter	0x34	4	Data will represent the Network structure
Write Net Parameter	0x35	43	Data will represent the Network structure

- **Set To Zero:** all the sample data will include in this frame; command value will indicate read or write action:

  Send Command: 02 04 0x20 Data LRC

  Receive Write Answer: as same as sent frame.

  Data:

  D0 = Channel Index (Start From 0)

- **Strat Log**

  Send Read Command: 02 04 0x23 LRC

  Receive Read Answer: 02 04 0x23 LRC

- **Stop Log:**

  Send Read Command: 02 04 0x24 LRC

  Receive Read Answer: 02 04 0x24 LRC

- **Read Channel data:** Analog channel readings and device status include in this frame.

  Send Read Command: 02 05 0x22 LRC

D0= Index (Start from 0; each index represents 16 channels, for example index 0 represent the value of the first 16 channels (0 to 15) and etc.)

Receive Read Answer: 02 16 0x22 Data LRC

D0: Index;

D1: Status;

D2..D65: channels value in 32bit float format;

- **Log Parameter:**

  Send Read Command: 02 04 0x30 LRC

  Receive Read Answer: 02 104 0x30 Data LRC

  Send Write Command: 02 104 0x31 Data LRC

  Data is a byte array and contain the value of the Log Para structure:
  D0…D11: file name;
  D12..D75: Channels which are selected to log the value; 255: Selected , 0: Not Selected
  D76..D79: Log interval (32 unsigned Integer)
  D80..D83: Max Logged Sample (Log stop after log counter reach this value. Set to Zero for continues logging) (32 unsigned Integer)
  D84...D87: Log Counter (32 unsigned Integer)
  D88..D91: Log status; 255:Run , 0: stop (32 unsigned Integer)
  D92..D95: Log Mode; 255:Fast , 0:Normal (32 unsigned Integer)

  Data Structure:

```
public class Log
{
   byte [] File=new byte[12];
   byte [] Ch=new byte[64];
   uint Interval;
   uint Max_Sample;
   uint Counter;
   uint Run;
   uint Mode;
   uint Lost;
   uint Flag;
}
```

- **Sensor:**

Send Read Command: 02 04 0x32 LRC

Receive Read Answer: 02 33 0x32 Data LRC

Send Write Command: 02 33 0x33 Data LRC

Data is a byte array and contain the value of the Sensor structure:
D0…D9: Sensor Name (Byte Array)
D10=0
D11..D16: Sensor Unit (Byte Array)
D17: Decimal Point
D18: Log Active. 255: active; 0: not active
D19: High Alarm Flag. 255: active; 0: not active
D20: Low Alarm Flag. 255: active; 0: not active
D21..D24: High Alarm Value. (Float)
D25..D28: Low Alarm Value. (Float)
D29: Filter
D30: Gain
D31: Reserved
D32: Reserved

Data Structure:

```
public class sensor
{
   byte[] Name = new byte[10];
   byte [] SUnit=new byte[6];
   byte DecPoint = 2;
   byte Log = 0;
   byte  H_Alarm = 0;
   byte  L_Alarm = 0;
   float H_limit = 0;
   float L_limit = 0;
   byte Filter = 1;
   byte Gain = 0;
   byte reserved_1;
   byte reserved_2;
}
```

- **Network Parameter:**

Send Read Command: 02 04 0x34 LRC

Receive Read Answer: 02 43 0x34 Data LRC

Send Write Command: 02 43 0x35 Data LRC

D0..D15: Device Name (Byte Array)
D16..D19: IP (Byte Array)
D20..D23: Subnet Mask (Byte Array)
D24..D27: Gateway (Byte Array)
D28..D31: Client IP (Byte Array)
D32..D35: Port (Unsigned Integer)
D36..D39: Unicast (Unsigned Integer)

Data Structure:

```
public class NetWork

{
   byte [] Name=new byte[16];
   byte[] IP;
   byte [] Subnet=new byte[4];
   byte [] Gateway=new byte[4];
   byte [] C_ip=new byte[4];
   UInt Port;
   UInt Unicast;
}
```