17.1 DLMS/COSEM

Introduction

IEC 62056 is a set of standards for electricity metering data exchange by International Electrotechnical Commission.

The IEC 62056 standards are the international standard versions of the DLMS/COSEM specification.

DLMS or **Device Language Message Specification** (originally Distribution Line Message Specification),[1] is the suite of standards developed and maintained by the DLMS User Association (DLMS UA) and has been adopted by the IEC TC13 WG14 into the IEC 62056 series of standards. The DLMS User Association maintains a D Type liaison with IEC TC13 WG14 responsible for international standards for meter data exchange and establishing the IEC 62056 series. In this role, the DLMS UA provides maintenance, registration and compliance certification services for IEC 62056 DLMS/COSEM.

COSEM or **Companion Specification for Energy Metering**, includes a set of specifications that defines the transport and application layers of the DLMS protocol. The DLMS User Association defines the protocols into a set of four specification documents namely Green Book, Yellow Book, Blue Book and White Book. The Blue Book describes the COSEM meter object model and the OBIS object identification system, the Green Book describes the architecture and protocols, the Yellow Book treats all the questions concerning conformance testing, the White Book contains the glossary of terms. If a product passes the conformance test specified in the Yellow Book, then a certification of DLMS/COSEM compliance is issued by the DLMS UA.

The IEC TC13 WG14 groups the DLMS specifications under the common heading: "Electricity metering data exchange - The DLMS/COSEM suite". DLMS/COSEM protocol is not specific to electricity metering, it is also used for gas, water and heat metering.

Source: https://en.wikipedia.org/wiki/IEC_62056

DLMS Master

Overview

DLMS (Device Language Message Specification) is a suite of standards developed and maintained by the DLMS User Association. COSEM (Companion Specification for Energy Metering) includes a set of specifications that define the transport and application layers of the DLMS protocol.

In DLMS/COSEM all the data in electronic utility meters and devices are represented by means of mapping them to appropriate classes and related attribute values.

Objects are identified with the help of OBIS (Object Identification System) codes (as per IEC 62056-61).

The DLMS driver allows only for readout and displaying only numeric values of DLMS object data fields. Connection via TCP (HDLC or WRAPPER) or serial (RS232/RS485) port are supported.

The setup of the DLMS driver consists of communication and tag configuration. Protocol specific parameters (except for DLMS/IEC handshake mode) apply for both serial and IP connections.

Configuration

Devices section

serial_number, **physical_address** and **logical_address** define the meter addressing parameters. Either **serial_number** (meter serial number) or a combination of **physical_address** and **logical_address** is used. If a serial number is provided, physical and logical server addresses are ignored.

> ⚠ Before configuring the Device section it is best to first check the connection parameters with a 3rd party DLMS utility.

client_address is defined in hex and usually depends on the authentication used. Most meters support hex 11 for no authentication.

type defines the object referencing. SN should be used for short name referencing and LN for logical name referencing.

mode defines the communications mode. If IEC is used along with comms settings for serial readout, the connection is initiated as per IEC 62056-21, at the default initial baud rate (300 7E1). DLMS-HDLC shall be used for HDLC

connections via IP. DLMS-WRAPPER is also supported for IP connections. The default setting is DLMS-HDLC.

timeout_ms defines the reply timeout for telegrams both via serial and TCP.

auth and **password** define the authentication mode and password. This can be set to None, or other authentication variant (see table below), depending on the mode configured and supported by the particular meter.
ip and **port** define the IP address and TCP port for DLMS communication via IP.

> ⚠ Connection parameters are device specific and can differ between makes, models and utility companies. For initial connection settings please refer to the configuration of the particular meter.

> ℹ When ip and port are configured, any serial port settings are ignored and connection is initiated only via IP.

Device configuration parameters for DLMS meters acquisition:

Parameter	Type	Description	Required		Default Value (when not specified)	Range	
			TCP	RTU		Min	Max
name	string	User-friendly name for a device	Yes	Yes			
description	string	Description of a device	No	No			
device_alias	string	Alphanumeric string to identify a device	Yes	Yes			
enable	boolean	Enabling/disabling of a device	No	No	1	0	1
protocol	string	Protocol to be used	Yes	Yes		DLMS	
serial_number	integer	Meter serial number	No	No	0		
physical_address	integer	Meter physical server address	No	No	1600		
logical_address	integer	Meter logical server address	No	No	0		
address_size	integer	Meter address size in bytes	No	No	1	1	4
client_address	integer	Client address	Yes	Yes			
type	string	Meter object referencing: SN - short referencing, LN - logical referencing	No	No	SN	SN, LN	
mode	string	Initial handshake mode.	Yes	Yes	DLMS--HDLC	IEC, DLMS, DLMS-HDLC or DLMS-WRAPPER	
timeout_ms	integer	Timeout in milliseconds	No	No	2500		
auth	string	Authentication.	No	No	None	None, Low, High, HighMd5, HighSha1, HighSha256, HighGmac, HighEcdsa	
password	string	Password for authentication	No (when auth is None)	No (when auth is None)			

Parameter	Type	Description	Required TCP	Required RTU	Default Value (when not specified)	Range Min	Range Max
ip	string	IP address	Yes	-			
port	integer	TCP port	Yes	-			
device		Communication port	-	Yes		PORT1	PORT2
baudrate	integer	Communication speed (bauds/s)	-	No	9600	300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200	
databits	integer	Data bit count for communication	-	No	8	6	9
stopbits	integer	Stop bit count for communication	-	No	1	1	2
parity	string	Communication parity option	-	No	none	none, even, odd	
flowcontrol	string	Communication device flow control option.	-	No	none	none	
retry_counter	integer	Number of requests, before link is considered lost (device status signals are changed) and reconnect attempt will be issued	No	No	3		
scan_rate_ms	integer	If provided and positive all reads and writes will be executed within the timeframe in milliseconds	No	No	10000		
reconnect_time	integer	Defines how often (in milliseconds) the client will try to reestablish communication with the meter after an unsuccessful attempt.	No	No	1000		

Signals section

DLMS configuration parameters creating signals:

Parameter	Type	Description	Required		Default Value (when not specified)	Range	
			TCP	RTU		Min	Max
signal_name	string	User-friendly signal name	Yes	Yes			
device_alias	string	Device alias from a Devices tab	Yes	Yes			
signal_alias	string	Unique alphanumeric name of the signal to be used	Yes	Yes			
enable	boolean	Enabling/disabling of an individual signal	No	No	1	0	1
log	boolean	Enable logging in event log	No	No	0	0	1
short_name	integer	Address of value to read (Short name).	No	No	0		

obis_job	string	OBIS codes can be accompanied by an attribute index, eg.: 1.0.1.8.0.255:2. Objects of register and extended register types do not require indexes and the scalers are applied to values automatically (though they can still be used if attributes other than the value need to be read out).	Yes	Yes

Debugging the DLMS service

If the configuration for DLMS devices is set up, the handler for the protocol will start automatically. If the configuration is missing or contains errors, the protocol will not start. It is done intentionally to decrease unnecessary memory usage.

If DLMS does not work properly (e.g. no communication between devices, data is corrupted, etc.), a user can launch a debug session from the command-line interface and find out why the link is not functioning properly. To launch a debugging session, a user should stop dlms process and run dlms command with respective flags as in the table shown below.

Procedure for DLMS protocol service debugging:

- **Step 1**: Service must be stopped by entering the following command into the WCC Lite:
  /etc/init.d/dlms stop
- **Step 2**: After service is stopped it must be started with the preferred configuration file (JSON files found in /etc/dlms folder) and a debug level 7: **dlms -c /etc/dlms/dlms.json -d7 --dlms**
  Additional output forming options described in the table below.

- **Step 3**: Once the problem is diagnosed normal operations can be resumed with the following command: **/etc/init.d/dlms start**

DLMS command line debugging options

Option	Description
-h [--help]	Display help information
-V [--version]	Show version
-p [--port]	Show output for one port only
-d <debug level>	Set debugging level
-c [--config]	Config path