

The DLMS/COSEM specification specifies an interface model and communication protocols for data exchange with metering equipment.

The interface model provides a view of the functionality of the meter as it is available at its interface(s). It uses generic building blocks to model this functionality.

The DLMS/COSEM specification follows a three-step approach.

* Step 1, Modelling: This covers the interface model of metering equipment and rules for data

Identification.

* Step 2, Messaging: This covers the services for mapping the interface model to protocol data units (APDU) and the encoding of this APDUs.
* Step 3, Transporting: This covers the transportation of the messages through the communication channel.

DLMS/COSEM provides built-in security mechanisms from the outset. Initially, it provided mechanisms for the identification and authentication of clients and servers, as well as specific access rights to COSEM object attributes and methods within application associations (AAs) established between a client and a server. Ciphered APDUs were also available to allow protecting the messages exchanged between clients and servers.

**Information exchange in DLMS/COSEM**

The objective of DLMS/COSEM is to specify a standard for a business domain oriented interface object model for metering devices and systems, as well as services to access the objects. Communication profiles to transport the messages through various communication media are also specified. The term "metering devices" is an abstraction; consequently “metering device” may be any type of device for which this abstraction is suitable.

The key characteristics of data exchange using DLMS/COSEM are the following:

* metering devices can be accessed by various parties: clients and third parties;
* mechanisms to control access to the resources of the metering device are provided; these mechanisms are made available by the DLMS/COSEM AL and the COSEM objects (“Association SN / LN” object, “Security setup” object);
* security and privacy is ensured by applying cryptographical protection to xDLMS messages and to COSEM data;
* low overhead and efficiency is ensured by various mechanisms including selective access, compact encoding and compression;
* at a metering site, there may be single or multiple metering devices. In the case of multiple metering devices at a metering site, a single access point can be made available;
* data exchange may take place either remotely or locally. Depending on the capabilities of the metering device, local and remote data exchange may be performed simultaneously without interfering with each other;
* various communication media can be used on local networks (LN), neighbourhood networks (NN) and wide area networks (WAN).

The key element to ensure that the above requirements are met is the Application Association (AA) – determining the contexts of the data exchange – provided by the DLMS/COSEM AL.

DLMS/COSEM uses the concepts of the Open Systems Interconnection (OSI) model to model information exchange between meters and data collection systems.

Data exchange between data collection systems and metering devices is based on the client/server model where data collection systems play the role of the client and metering devices play the role of the server. The client sends service requests to the server which sends service responses. In addition the server may initiate unsolicited service requests to inform the client about events or to send data on pre-configured conditions.

In general, the client and the server APs are located in separate devices. Therefore, message exchange takes place via a protocol stack

Client AP

)

Data collection system

(

Server AP

Metering device

)

(

.request

.response

SERVICE .request

SERVICE .response

Application layer

Intermediate layers

Physical layer

Communication media

Unsolicited

SERVICE .request

**The COSEM interface classes**

Data collection systems and metering equipment from different vendors, following these specifications, can exchange data in an interoperable way.

***Class*** *Methods* ***Object*** *Attribute Values*



*class identifier Attributes Instantiation*

g

**Total Positive Active Energy: Register**

logical\_name = [1 1 1 8 0 255]

value = 1483

…

**Total Positive Reactive Energy: Register**

logical\_name = [1 1 3 8 0 255]

value = 57

…

reset

logical\_name: octet-strin value: instance specific

...

**Register** class\_id=3

**An interface class and its instances**

**COSEM Object Identification System (OBIS)**

The OBject Identification System (OBIS) defines the identification codes for commonly used data items in metering equipment.

OBIS provides a unique identifier for all data within the metering equipment, including not only measurement values, but also abstract values used for configuration or obtaining information about the behaviour of the metering equipment. The ID codes defined in this document are used for the identification of:

* + logical names of the instances of the ICs, the objects.
  + data transmitted through communication lines.
  + data displayed on the metering equipment.

This applies to all types of metering equipment, such as fully integrated meters, modular meters, tariff attachments, data concentrators etc.

**OBIS code structure**

**Value groups and their use**

|  |  |
| --- | --- |
| **Value group** | **Use of the value group** |
| **A** | Identifies the media (energy type) to which the metering is related. Non-media related information is handled as abstract data. |
| **B** | Generally, identifies the measurement channel number, i.e. the number of the input of a metering equipment having several inputs for the measurement of energy of the same or different types (for  example in data concentrators, registration units). Data from different sources can thus be identified. It may also identify the communication channel, and in some cases it may identify other elements.  The definitions for this value group are independent from the value group A. |
| **C** | Identifies abstract or physical data items related to the information source concerned, for example  current, voltage, power, volume, temperature. The definitions depend on the value in the value group A.  Further processing, classification and storage methods are defined by value groups D, E and F.  For abstract data, value groups D to F provide further classification of data identified by value groups A to C. |
| **D** | Identifies types, or the result of the processing of physical quantities identified by values in value groups A and C, according to various specific algorithms. The algorithms can deliver energy and demand quantities as well as other physical quantities. |
| **E** | Identifies further processing or classification of quantities identified by values in value groups A to D. |
| **F** | Identifies historical values of data, identified by values in value groups A to E, according to different billing periods. Where this is not relevant, this value group can be used for further classification. |

Our project also include the use of Arduino Uno with the Atmega Microcontroller. This particular board and microcontroller is chosen because of fair advantages of using it over other boards. To begin with, the availability of the Uno Board is certainly very easy and cheap. The library support for Arduino is immense and the Arduino IDE being free to use gains a lot of favour from the Open Source Community.

The storage, easy usability, low maintenance, supported languages, efficiency and the support for various different modules are some of the many factors which made us choose arduino for our project.

**Integrated development environment**