IJIREEICE

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

*International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering*

*Nitte Conference on Advances in Electrical Engineering NCAEE-2016*

*NMAM Institute of Technology, Nitte*

*Vol. 4, Special Issue 2, April 2016*

# PORTING OF DLMS CLIENT APPLICATION ON EMBEDDED PLATFORM FOR MODEM DEVELOPMENT

**Chaithra Bhat Padmar**

Student, 2nd year M.Tech (Electronics), Department of ECE, CEC, Benjanapadavu

**Abstract:** Current metering trend collects the data from the energy meter through MRI. This paper presents the idea for data collection from energy meters using a modem. To develop a modem, a cost effective embedded platform is essential. In this paper development of such a user friendly, economical embedded platform for RENESAS controller is carried out and discussed in detail.

**Keywords:** Modem, Client, DLMS (Device Language Message Specification), renesas.

## I. INTODUCTION

Traditional electric meter reading and billing for consumption of electricity is done by operators from houses to houses or buildings to building. It requires large number of labors as operators and large number of working hour to reach complete billing. Human operators may cause errors in reading or billing. In order to achieve efficient billing by reducing billing errors and operational costs AMR (automatic meter reading) plays an important role. AMR is an effective method of data collection that gives substantial saving to greater data accuracy through meter data reduction. With the advent of digital technology, Analog electro-mechanical meters are continuously replaced by digital meters. Digital energy meter offers greater convenience to adopt, implement and establish AMR. This paper presents the idea for data collection from energy meters using a MODEM, and also presents an embedded platform for modem development using a client stack. Section II includes the details of the DLSM/COSEM protocol structure followed by client stack and its implementation in section III. Section IV includes the result analysis which is succeeded by future scope in section V.

## II. DLMS/COSEM PROTOCOL STRUCTURE

The Device Language Message Specification / Companion Specification for Energy Metering (DLMS/COSEM) specify an interface model and communication protocols for data exchange with metering equipment [2]. COSEM addresses the challenges by looking at the meter as an integrated part of a communication system which requires all the ability to convey measurements of the delivered product (energy) from the diverse points where these measurements are made to the business processes which use them, over a variety of connecting media [3].

Communication follows the fundamental Open Systems Interconnection (OSI) model of telecommunication. Major part of COSEM is specified by the interface classes, objects defined in the protocol. The exchange of collected data with metering device using the COSEM model is based on server/client paradigm and generally meter will be the server. All the concepts and terminologies in the COSEM are related to the Open System Interconnection Reference Model described in ISO/IEC7498-1 (International Organization for Standardization/International Electro technical Commission). Communication in COSEM happens via data exchange (exchanging messages (service requests /responses)) of data between client and server AP's (application processes).

### *A.* CLIENT SOURCE CODE LIBRARY (CSCL)

The DLMS/COSEM Client Source Code Library provides a complete implementation of a 62056 DLMS/COSEM protocol stack to enable interested Original Equipment Manufacturers (OEMs) to create DLMS Client Applications to read/write DLMS-enabled meters and devices [1]. The OEM can build this sample implementations and start running the client on Windows/Linux machine and test against a server to trace the flow and logic of the protocol-stack. The key features of the client stack are like platform independent with default implementation for windows and Linux and supports HDLC and User Datagram Protocol, Transmission Control Protocol/Internet Protocol (UDP, TCP/IP) communication profiles.


DOI 10.17148/IJIREEICE/NCAEE.2016.15

IJIREEICE
ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526
International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering
Nitte Conference on Advances in Electrical Engineering NCAEE-2016
NMAM Institute of Technology, Nitte
Vol. 4, Special Issue 2, April 2016

The Source Code Library (SCL) functions provide a means to open and initialize a serial or IP port establish a HDLC connection (only for HDLC based communication profile), establish an application association, and then functions to read/write/execute attributes and methods in the connected DLMS server (meter)[1]. All the functions in the selected client are modified as per the requirement to build the platform. Cygwin is a Linux-like environment for Windows. It consists of a Dynamic Link Libraries (DLL) (cygwin1.dll), which acts as an emulation layer providing substantial POSIX (Portable Operating System Interface) system call functionality and a collection of tools, which provide a Linux look and feel [4]. The Application Program Interface (API) follows the Single Unix specification as much as possible, and then Linux practice.

A. IMPLEMENTATION

The task performed in Cygwin is building of Kalkitech client source code in windows platform using cygwin environment. The selected client is built using cygwin libraries and compiled successfully in cygwin environment. Data collection along with successful authentication is carried out in cygwin. After the successful compilation on cygwin, an alternate logic is used to store the collected data. The Data from the meter gets transferred either in normal mode or in block wise. Two different logics used to store the data into a text file and the stored data is compared with the actual data collected through Vinplus. The .exe file is generated as a result of successful compilation which will link the linker in the next stage initially with dynamic link libraries and in later stage it was made static library. All the Linux dependencies in the client code are replaced with controller available library functions. This was done by avoiding the shared library object and by the direct access of library functions. The collected data is compared with Vinplus format at each stage. Allocating static storage for client variables, analyze about dynamic memory used by client, removal of dynamic memory allocation was carried out. The dynamic data collection was made static by allocating the definite (1024) size to the defined buffers and modification of the code accordingly. The complete code is optimized to attain the memory requirements. The RAM, ROM and stack sizes are constantly checked at all stages of stack modification, comparison of actual and collected data was also carried out at each modification level. Continuous data collection was achieved by using XML (Extensible Markup Language) Parsing. Once it was working efficiently on this platform, later the code has to be moved to controller platform and compiled successfully in Cube Suite++ IDE. The Choose of controller was RL78/G13 R5F100GG for this modem.

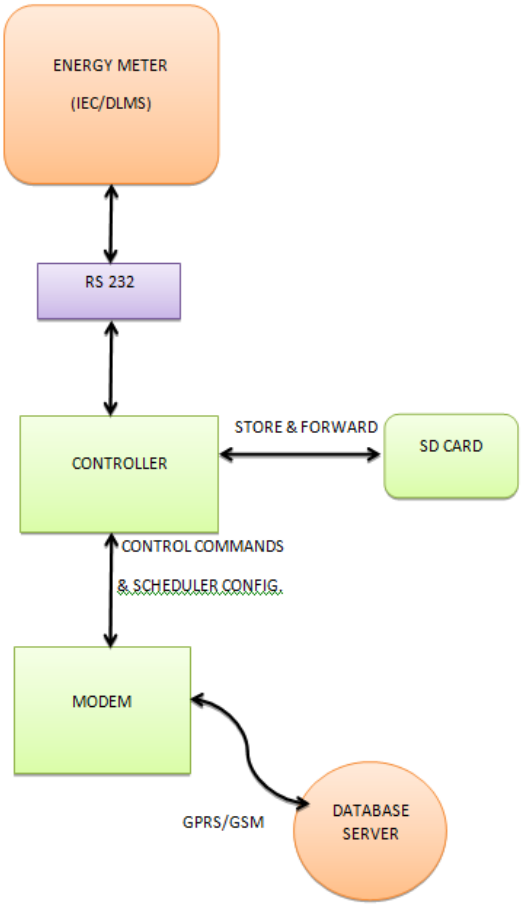


Fig 1. Block diagram of the GSM/GPRS modem


DOI 10.17148/IJIREEICE/NCAEE.2016.15

IJIREEICE
ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526
International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering
Nitte Conference on Advances in Electrical Engineering NCAEE-2016
NMAM Institute of Technology, Nitte
Vol. 4, Special Issue 2, April 2016

### *B.* RESULT

Building of Kalkitech client source code in windows platform using Cygwin. Kalki client source code is understood and it is compiled in Cygwin using make file. The Storing of "Get NORMAL REQUEST" DLMS data in vinplus format (Data which fits within single APDU size of client) and storing of "Get BLOCK REQUEST" DLMS data in vinplus format (Data which is greater than single APDU size of client, every received APDU packet is converted into vinplus required format) was carried step by step, block by block data collection and writing the data to the file about 1024 bytes.

### *C.* FUTURE SCOPE

The idea of GPRS/GSM modem which in different phases requires many sub-steps like client code stack, controller configuration followed by SD card design and communication implementation. The compilation of the client code stack in controller platform is achieved. Once the rest steps are implemented a complete modem can be seen as in Fig.1 which shows the block diagram of the GPRS/GSM modem which in different phases.

## REFERENCES

[1]. DLMS/COSEM Client Source Code Library User Manual, Version 3.1.4 FEBRUARY 2013, from Kalki Communication Technologies Limited.
[2]. A TECHNICAL REPORT on Companion Specification for Energy Metering, Green Book, 7th edition, on COSEM Identification System and Interface Classes .
[3]. A TECHNICAL REPORT on Companion Specification for Energy Metering, bluebook, 10th edition, on COSEM Identification System and Interface Classes.
[4]. A Introduction Guide on RL78 Family Open Source FAT File System M3S-TFAT-Tiny, Rev.1.02 Nov 08, 2013
[5]. Hardware User's Manual RL78/G13, 16-Bit Single-Chip Microcontrollers, Rev.3.20 Jul 2014.


DOI 10.17148/IJIREEICE/NCAEE.2016.15