Reprinted from conference "Power System Protection and Automation", December 2010, New Delhi, India.

Challenges in Smart Meter Design

M.S.Jaganmohan, K. Manikandan
Easun Reyrolle Ltd

Abstract

This paper briefs on the smart meter functionality requirements and discusses the challenges it poses to the energy meter design and proposes methods to achieve an architecture which is scalable and modular. Also, the paper discusses the key care about for choosing a Real time operating system for smart meter application and silicon SOC architecture required for implementing the software.

1 Introduction

Similar to the way embedded systems used in mobile phones have evolved from a simple 8bit embedded microcontroller with a wireless modem and single line number display to a phone with internet, mail, gaming and graphics etc, the energy meter is also evolving from a simple single line LCD display electronic energy meter to a Smart meter. Smart meter will be an integral part of smart grid architecture. Smart meter will have the capability to communicate over many communication methods based on data rate, distance, type of data and protocols. It will have touch screen based graphic home display units, home energy management software for Home etc. Smart meters can communicate to a central database over a range of disparate yet commonly networked communications media. This will provide utilities/consumer with more data about energy usage and optimize the energy use. Smart meters can help quantify demand habits so that utilities and consumers can better predict their base and peak loads. Enable additional high value functions including flexible pre-paid metering solutions, demand side management (DSM), and rapid automatic outage detection and restoration. Smart meters can act as the gateway to intelligent devices and appliances in the home, and provide the foundation of new tariff programs.

As we are aware the metering systems are deployed with a long life time expectation exceeding 20 years. There is a need to look back, redefine the current Meter Hardware & Software architectures before the existing meters are modified to be used for Smart meter applications. This step will ensure that the Smart meters are scalable, reliable and the architecture is modular to accommodate new features and functionalities. This paper brings out the challenges in evolving the architecture of the present generation meters and proposes new architecture for smart meter.

2 Smart Meter Functionality

The smart meter designs will need to support different types of communication functionalities based on some or many of the following methods: UART, RS485, IrDA, LPRF, Zigbee, Bluetooth, GSM/GPRS, Wifi, & PLCC. The smart meter can be used to implement the features of Billing, Credit management, communication, remote connection / disconnection, revenue protection, power quality measurement, load and loss control, load forecast, common user console (Electricity, Gas, Water) with features of Power management in home. The new designs will support advanced tamper detection using accelerometer and as well support Geographical Information by use of built in GPS. Other than the above the Smart Meter will have features like Remote Display Unit with Trend display, Touch screen to set different load profile options and control of Home equipments to achieve optimal use of electric energy.

3 Challenges in Smart meter design

The design challenges in realising the smart meter are multi faceted. One of the first challenges is clear definition of features, as Smart meters for Home and Utility occupy different spectrum of feature sets with minimal overlap. Clarity in the definition of requirements by user community can optimise the cost of the Smart meter by use of optimal hardware and software components. Some of the features which can impact the cost and modularity of the design are listed below:

a. Communication Protocol: It is important that specific communication protocols to be implemented are specified /mandated by central body for Smart Meters to ensure that proprietary protocols do not find way into the Smart Meter.
b. Communication Security: Communication security in an AMI/ Smart Grid environment will be a prime requirement as remote control features are enabled and sensitive revenue data is transmitted. It is important that the specific sections of the standards (IEC 62351) to be incorporated in the Meter Firmware are specified which are compatible to the AMI and MDM systems.
c. Interoperability: The control commands and other data formats used in smart meters should be ensured that they are interoperable with the existing AMI infrastructure.
d. Meter Data: Different Consumers (Utilities, Industry and Home) require different measurement parameters. Standardisation of the parameter requirements across consumer segments is a must.
e. Communication ports: Since the Smart meters will be provisioned with Ethernet, optical Ethernet other than the Serial connector and USB. Specification of the ports will help optimise the case design and upgradeability.
f. Communication: There is a varied choice of communication modes available for the smart meters like wireless (Zigbee, WiFi, LowPowerRF) or power line. It will be desirable to define the choice of these modes as focussed study, analysis and implementation can be carried out.
g. Standardization of Smart meter size and foot print would also be an important consideration.
h. Power Quality Requirements: The specification on the type of power quality measurements required across different consumer segments will need to be defined for higher order harmonic distortions, Sag, Swell, Outages, dips, transients recording, duration of records. Clear specification on these requirements will optimise the cost of the smart meter as it will impact use of Digital Signal processors and high speed memories for implementing the Power quality measurement features.
i. Standardisation of user interface Options like Touch screen displays or conventional LCD display.
j. Time synchronisation: Specification on synchronisation method for RTC time based on GPS or IRIG-B or from network needs to be specified to design the internal interfaces and choice of devices for these features.
k. I/O expansion capability: Definition of type of I/O capability like electro-mechanical relay or solid state, analog inputs and

outputs and the number of Inputs and Outputs. This will determine if the I/O can be accommodated in the smart meter or a separate extension box needs to be provisioned with necessary interfaces to the meter.

l. Special features like language support, email SMTP client, ftp, remote firmware upgrades with security features are better to be specified at the initial phase of the design to make right choices of operating systems and protocol stacks. Inclusion of these features later will not be optimal as the real time operating system chosen may not support certain client features.

4 Present Generation Meter Architecture

The present generation meter designs are developed with limited interface for specific purpose with minimal user interface. The present meter software architectures will fall into a class of monolithic software architecture as represented in the Figure 1, where the memory allocations, the peripheral driver interfaces to the applications are written in a proprietary way. The memory allocations are managed manually during coding. The modules are tightly coupled with proprietary API Interfaces.

The inclusion of additional modules requires careful analysis of memory allocation and memory handling. The Monolithic software architecture, also does not allow easy integration of off the shelf Modules like protocols and functions. If the coding has been carried out in assembly language, the portability of the code from one platform to another becomes a nightmare. Thus the development time for providing additional features increases exponentially affecting the Time to Market.

The hardware used in current meter design is typically 8 bit microcontrollers, with a limitation in memory addressing capability and performance.

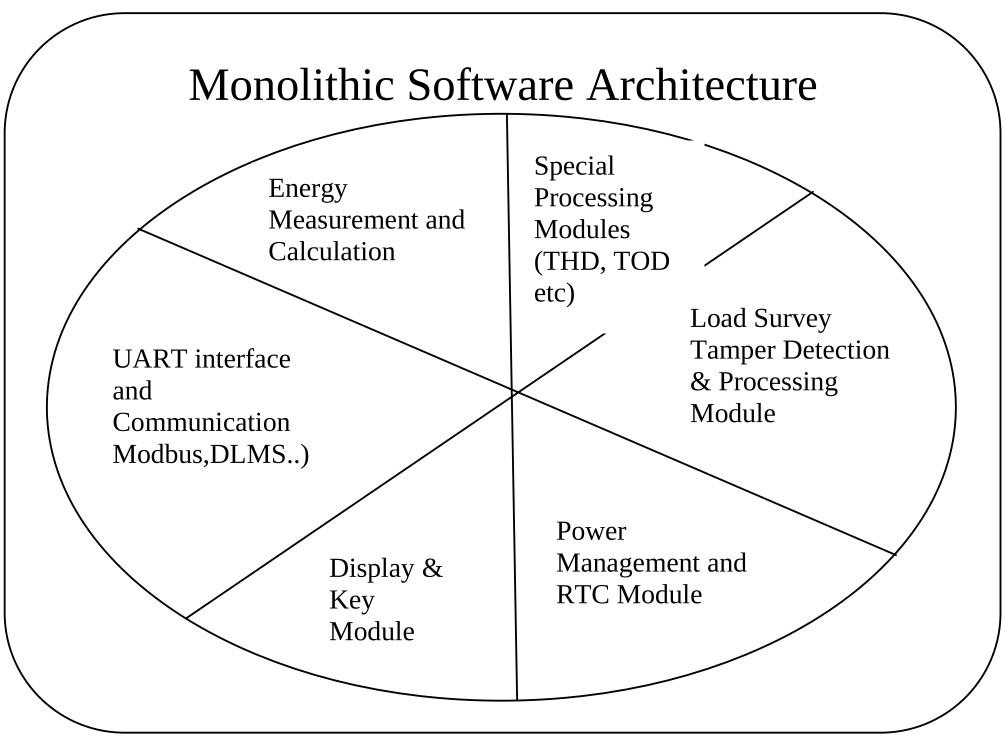

FIGURE 1

5 Smart meter Architecture

5.1 Hardware Architecture

It is desirable integrated Silicon on Chip (SOC) architecture is considered for Smart meter design. SOCs would integrate a 32Bit microprocessor with an optional digital signal processor for running advanced signal processing functions for power quality measurement features. It is efficient if the multi-core processor SOC chosen is implemented on a shared Peripheral and Memory Interface Bus. This method of interconnection of resources in a Smart meter SOC would make the system realization to be modular and scalable in architecture. A typical SOC which would meet the requirement of a Smart meter is represented in Figure 2. The Meter SOC should support core energy measurement and calculation functions as specialised co-processor with additional peripherals for Communication such that the respective communication stacks can be run. The hardware SOC architecture should support sufficient Flash and internal memory to run a Real Time Operating System which can support most of the commercially available or open source Protocol Stacks for Wi-Fi, Modem (2G/3G), TCP/IP, Zigbee and Bluetooth Peripherals.

5.2 Software Architecture

As the metering applications evolve into ever more complex and feature rich designs, the software development time also increases. The software development time could be optimized and made reliable if standard and proven components of software to manage the resource handling, scheduling is used. This can be achieved by using Commercial off the Shelf (COTS) Real Time Operating Systems (RTOS). The software architecture and implementation becomes modular by use of Real Time operating Systems. RTOS offers an application developer a number of aids that allow a complex design to be completed in a timely fashion, permit easy integration of existing components and allow for simpler code re-use in the future. The RTOS provides following tested components as part of the Core Kernel

- Memory Management
- Scheduling Policies
- Inter Process Communication and Resource Sharing
- Interrupt Handlers
- Task Priority Assignment
- Synchronize Concurrent process and threads

In modern systems, a RTOS consists not only of a real-time kernel, but also

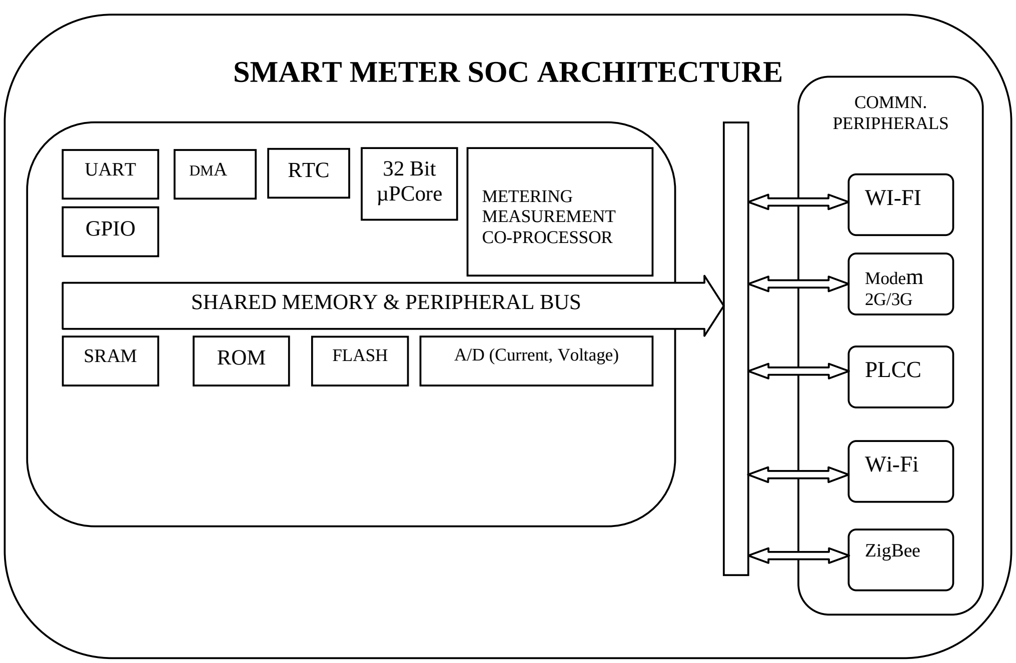

FIGURE 2

higher-level functions such as device management (USB, UART, Ethernet, LCD etc), file systems, protocol stacks (CAN, TCP/IP, HTTP etc) and graphical user interfaces (GUI).

A typical Smart Metering application can use tested protocol libraries which are re-used across applications like

- TCP/IP Ethernet Stack
- Zigbee® Protocol Stack
- Networking Protocol Stack
- USB Host and Client Stacks
- IrDA® Stack
- Graphics Library
- Touch Screen Sensing Library
- Flash Disk Drive (MDD)

A typical Real Time operating system with the above description is shown in Figure 3. The applications for Energy Meter like Energy Measurement, Communication, Load Survey, TOD, Tariff, Billing, Power Quality & THD Measurement, Remote connect/disconnect, Display, Touch screen, calibrations are developed on top of the RTOS kernel and uses the Device drivers, libraries available with the RTOS for the specific SOC. This enables faster and reliable development of the Smart Metering Solution.

5.3 Considerations for Choice of RTOS

There are many COTS RTOS. It is very important that the right RTOS is chosen for the smart meter application. The most important factors of real time systems are the worst case response time of a task and worst case response time of an interrupt [Sohal 2001]. However, it makes no sense to analyze real time operating systems metrics such as interrupt latencies and task switching time without considering different CPU usage scenarios [Timmerman et al. 1998], as it is easier for a system to be more predictable when it is not overloaded. Some of the considerations while Choosing RTOS for

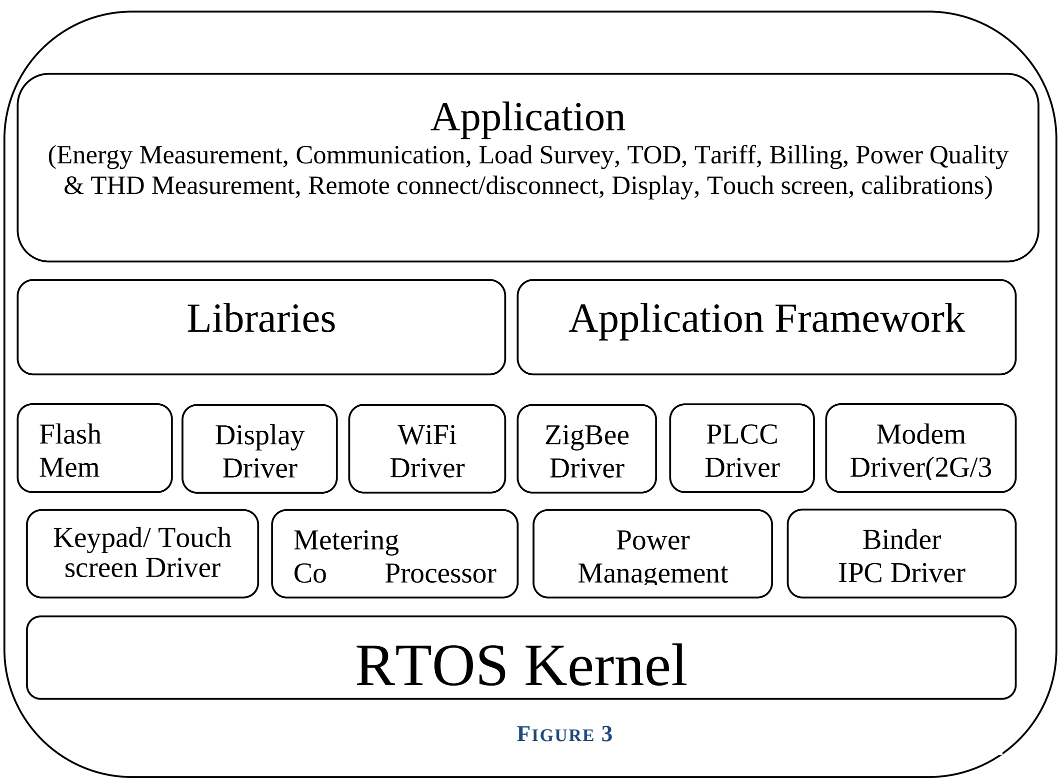

FIGURE 3

TABLE 1

Operating System	License	POSIX	Processor SOC	Foot Print**	Context switch Time **	Interrupt latency**
MQx	Proprietary	Yes	MPC8248 (Freescale)@200MHz	10KB	0.558 µsec	0.331 µsec
QNx	Proprietary	Yes	ARM AT91SAM9G45 @400MHz	12KB	4.523 µsec	
VxWorks	Proprietary	Yes	MPC8260 (Freescale)		11µsec	98µsec
RT Linux	GNU GPL	Yes	MPC8260 (Freescale)		13.4µsec	132µsec
Theradx	Proprietary	No	Freescale i.MX @ 40MHz	4KB	1.7us	

Smart Metering application would be

- POSIX Standard (IEEE Std 1003.1-1988, ISO/IEC 9945) Compliant
  - This feature would ensure that the applications written on a given RTOS can be migrated/re-used on any other POSIX Compliant RTOS
- Small Foot Print
  - Since most of the embedded systems are optimized for power and performance, limited memory will be available to hold a complete RTOS Kernel image during run time.
- Worst Case Interrupt Latency Time.
  - This is specifically applicable for operating systems used for real time application and specifies the time required to respond to an interrupt request..
- Worst case Response time of a Task
  - The time required to switch between different processes based on priority.
- Memory Management

- Availability of RTOS for a specific SOC architecture.
- List of Libraries supported by RTOS, like Database Manager, email client, ftp, web client etc.
- Availability of device drivers ported and tested by the Silicon Manufacturer (this can reduce substantially the development Time)
- Royalty cost/ free, one time Licensing cost/ free (O/S like Linux and FreeRTOS are free and no licensing cost is involved)
- Availability of source code of kernel (this can help debug deeper issues in real time systems and to optimize the memory/performance for a specific application)
- Is the RTOS field proven in many applications and certifications if any.
- Does the operating system meets real time criteria (interrupt latency, Context switch time etc.).
- Sleep modes to minimize power consumption

The important features of the above care about is summarized in Table 1

Note: ** Depends on the Hardware performance and the features included to create the Kernel

6 Conclusion

In summary, the paper discussed on the architecture of the present generation meters and the type of software and system architecture required for the smart meter. The paper also brought out the care about while choosing or specifying a RTOS for smart meter application. The implementation of a RTOS based system requires the engineers to be trained to understand and work in a RTOS environment for smart meter employing RTOS. Also, the architecture of the smart meter will also be determined / driven by the specifications required in such meters.

7 Acknowledgements

The author wishes to thank Mr.George Punnoose, President, Easun Reyrolle Ltd. I also thank the Easun Reyrolle Metering team for the innovative ideas shared with me which helped me author this paper.

References

1. Performance Analysis of VxWorks and RTLinux by Benjamin
2. Commercial real time operating systems by Swaminathan Sivasubramanian
3. http://rtos.com/products/threadx/
4. http://www.enea.com.cn/uploads/content/11260860226742.pdf