Challenges in Emergence of Real Time Embedded System as a Tool for the Intelligent System

Sunita Tyagi
Research Scholar
Singhania University

Rakesh Jagdish Ramteke
Department of Information Technology,
North Maharashtra University, Umavinagar, Jalgaon, INDIA

Dawar, Nitin
Computer Centre,
Delhi University, Delhi

Abstract- Thinking an Intelligent System (IS) without exploring it in the context of Industrial Communication Bus, (ICB), Real Time Embedded System (RTEs) and Smart Field Devices (SFD) will be difficult in the span of economical and efficient innovation for a large class of industrial applications [1]. Initially, computers were used for the Industrial purposes in which its role was equally important to carry control functions. In the recent industrial requirement, PCs applications has been explored from the filed-level control to monitoring system, and from construction of data acquisition to a intelligent platform which is capable in decision making for efficient, high and quality productivity. In view of the inevitable disadvantages of size and power consumption of industrial PC, alternative systems were explored which resulted in the development of the embedded system. With the advancement in the low power consumption semiconductor technology, embedded systems were shaped to meet the changing requirement of the industries and resulted in the development of the RTEs. A step ahead and to cater the need of the future, RTEs are explored to work in a communication network over different ICBs in building Intelligent System (IS)

This paper examine the challenges associated with RTEs and how technology is working in developing intelligent system of the future

Keywords – RTEs, Industrial Communication Bus, Intelligent System.

I. INTRODUCTION

1.1 The RTEs (Real Time Embedded System)

RTEs, emerged as an alternative, certainly advantageous, is a kind of special computer which takes the application in the center and computer technology as a base, while the hardware and the software can be clipped on the bases of the requirements. It is essentially a special-purpose computational system built into another machine with the sole purpose of controlling the "Real-time", which usually means that the 'time' which is prescribed by external sources". The timings are generated by the requirements of the overall system.
The embedded system can adapt the versatile strict demands of application system well to the function, reliability, cost, size and power consumption etc. Given these merits, embedded electronics technology is contributing a significant rise in the number of automatic devices for industrial data acquisition, transmission, monitoring, diagnostics, control and supervision. High-performance of the embedded system is due to the ability to accommodate microprocessor as a core which can interact with new-generation interfaces therefore allowing easily to be connected with the different FDs (Field Device),SFDs (smart Field Device) and directly to the Industrial Communication Bus (ICB). At the each end of these devices including RTEs, FDs and SFDs are configurable and are capable of making multi way communication with other devices.

Primarily, RTEs (Real Time Communication systems) systems were considered to overcome the disadvantages of Industrial PC later on the adaptive technological advancement bought RTE architecture to fully connect and control FDs for controlling the process and to build intelligent manufacturing automation.

1.2 The Challenges and the Complexity

The recent advances in the software and high speed microprocessor technology further resulted to look RTEs in the Global prospective. Thereby a need arose in developing a worldwide, unified specification for RTEs in terms of interfacing and connectivity with the different networks. The controllers used in the RTEs provide typically on-chip signal conversion, data and signal processing, and communication functions [2]
In a strict rule to the IS there should be effective control at all the level of controlling and processing in the system.This is only possible if this rule is successfully incorporated for developing timely and predictable communication services in the RTEs for factories and industrial process & controls. For example, an automated factory is usually composed of several workcells, each of which contains devices such as robots, sensors, and transport mechanisms. All devices in a workcell need to be connected via a local area network mainly known as communication bus or ICBs such as FieldBus, CANBus, Profibus etc. This further allows multiple workcells to be connected with communication bus to bridges RTE,FDs and SFDs together on it. A number of cooperating tasks such as to collectively monitor & control manufacturing equipment and processes by communicating with one another via the underlying network is possible. This emergence of a wide-spread trend for networking of field devices around specialized networks frequently referred to as field area networks (FAN).

Given the technological challenges like how to make the introduction of embedded controllers cost effective, embedded controllers have become one of the most important drivers of innovation for a large class of industrial applications to develop IS.

In the rapidly changing scenario, system engineers are confronted with this a challenging task:, One of the most popular ways for achieving this goal is by an aggressive sharing of hardware resources amongst different control functions. The paradigm one application-one system is dead and buried and each computing unit is required to sustain several applications (either independent or interacting).

Broadly challenges to be confronted irrespective of functioning RTES in the IS is given below

1. Interfacing RTE with the different ICBs.
2. Interfacing RTEs in controlling different heterogeneous FDs and SFDs
3. Assuring effective communication among RTEs for commanding different systematic operation
4. Economical adaptability of heterogeneous FDs,SFDs including actuator, sensors, pneumatic devices etc .

1.3 The Structure of Embedded System

Unlike general-purpose computers, Daogang Peng argues that a generic architecture cannot be rigidly defined for Real Time Embedded System[3], It is the "Intelligence which can be privately defined or supplied over ICB based on processing, production, synchronization and events based requirement".
There are many types of architecture & platform was designed and developed to support multiple communication means such as serial port, Ethernet, ICB and etc. The platform is mainly to solve the problems in power system equipment, such as data acquisition, monitoring devices, building network communications and transmitting data for control function to upper DAS, DCS and monitoring systems over Ethernet or ICB. For the sake of our understanding we are discussing some of the common form of systems at the block diagram level. Any system can hierarchically divided into subsystems. Each sub-system may be further segregated into smaller systems. Each of these smaller systems, in turn, consists of discrete parts. Together, this is referred to as the hardware configuration.
Some of these discrete parts may be programmable and therefore must have memory to store the programs. In an RTEs the on-chip or on-board non-volatile memory is used to store these programs. These programs constitute both a Real Time Operating System (RTOS) as well as the application programs. Architecture of a typical embedded-system is shown in Fig. 3, The hardware unit consists of the above units along with a digital as well as an analog subsystem. The software in the form of a RTOS resides in the memory.

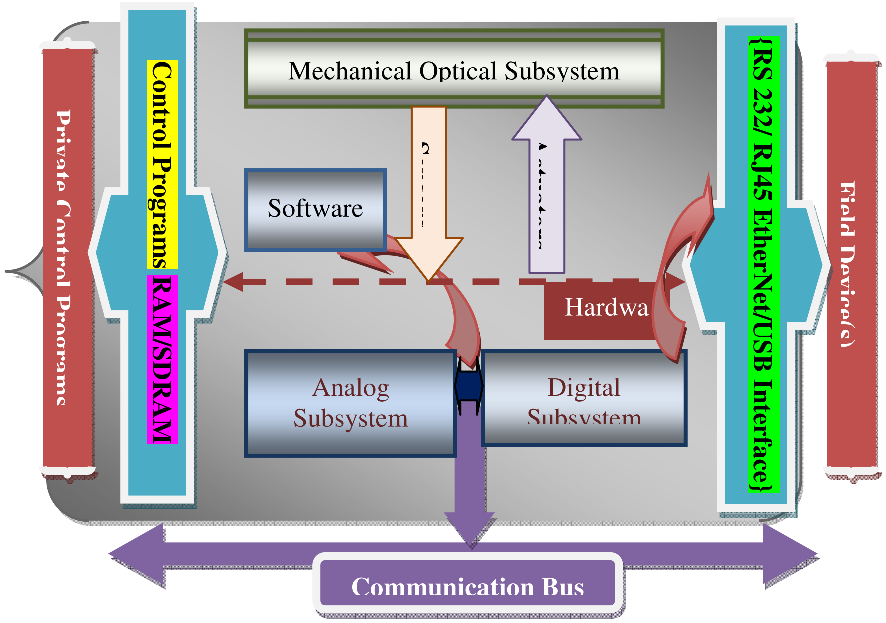

Fig 3 : Embedded system architecture for Intelligent System

A typical embedded system consists, by and large, interfacing and communication unit which is discussed below.

1.4 System Design for Interfacing Input/ Output Devices

Input/Output interfaces are necessary to make the RTES interact with the external world. Generally, FDs which are sensors, pneumatic shutters, actuators, cutters, feeders , crashers etc are non-intelligent devices but these are designed to connect with the controllers either on digital or analog interface while SFDs can be programmed to perform event critical functions only. The Physical Layer (P.L) of the RTEs send digital or analog signal to the FD or SFD which direct these to function or act [2]. These Systems are centred around processor or programmable controller which is supported by powerful communication capabilities. It also possessed RS232/485, expended to ICB interface to gather data which need to be monitored to PowerPC, and transmitting the data there after.
The embedded system development environment based on controller includes two aspects: host development system and target board development system. Host development system is used for compiling, linking, remote debugging, and other associated development while Target board development system always is used for application development and debugging. Target board is not a bare board which has some hardware driver like burnt on EPROM. Therefore it got a BootLoader in it. Fig 3.1b shows the common architecture to connect FDs and SFDs with the RTEs.

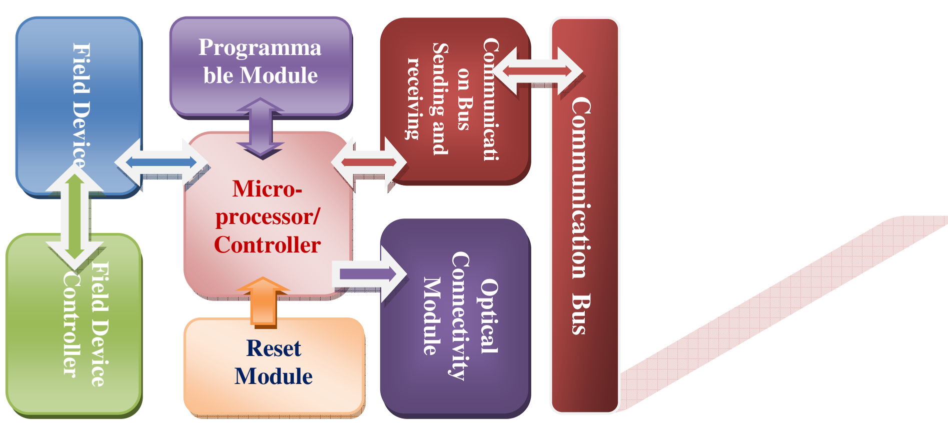

Fig 3.1 : Embedded system Design for I/O Interfaces

1.5 Designing Communication of RTE with Field Device

FD or SFD sets communication with the RTEs to build IS. The communication takes place in both the direction i.e. from FD to RTEs or vise versa. The communication which is referred as transferring the data, is divided in the two segments, a) control and b) limiting segment RTEs Microprocessor monitors communication port i.e. USB/Serial and the Network port[4]. In case device sends Interrupt on communication port, RTEs respond the device and make communication buffer available for receiving the data. On the bases of the data received, RTEs analyse, kind of FDs it is communicating with and what are request. Based on request and device it either allocates the resources {processing, storing etc} or provide instruction in-case device is smart. Also, a Real Time Operating System (RTOS) sitting on the non-volatile memory of the RTES, the application software which is essentially cyclic or interrupt driven programs, execute read-compute-write cycles. Fig 3.2 shows RTEs intelligence in building communication with the FDs. And SFDs. Multiple such programs execute under a priority driven scheduling policy managed by the RTOS, thus issuing control or limiting instruction or data to the FDs. This makes processes intelligent in which FDs or SFDs are capable in making its own decision without depending much on “Centre Control System”.

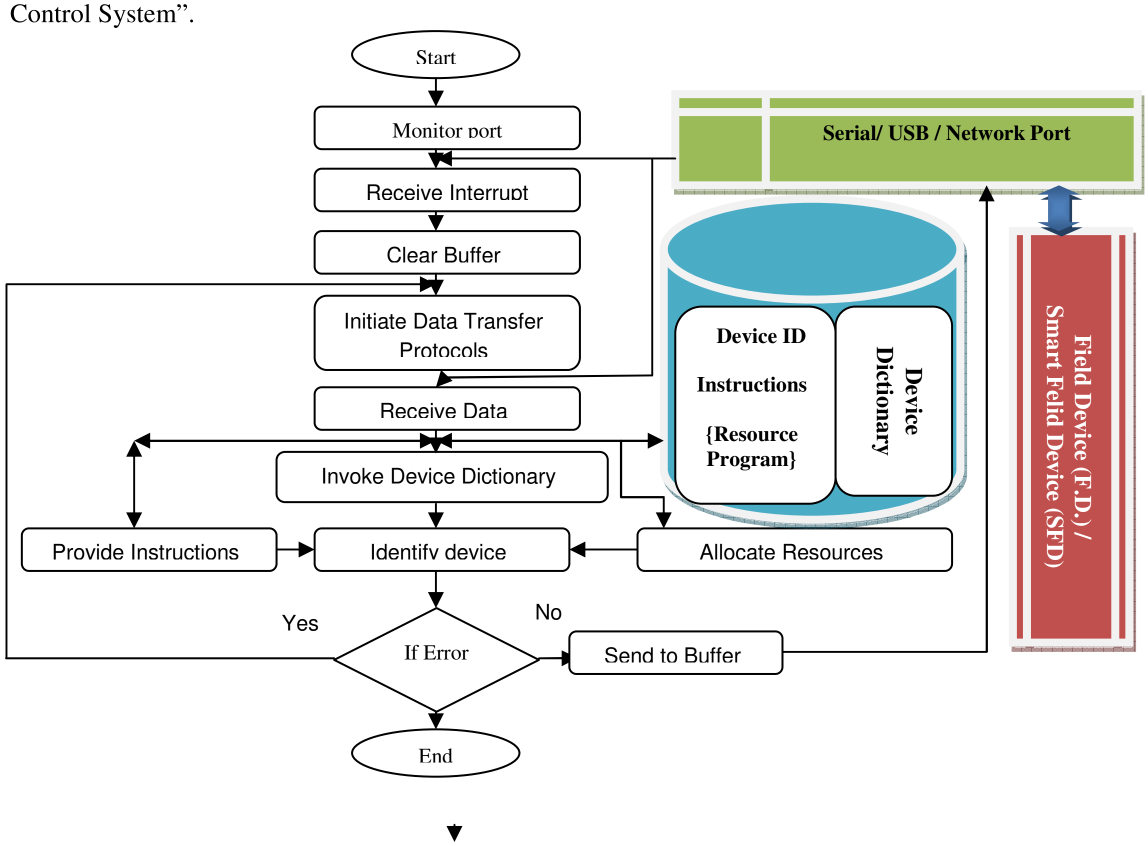

Fig 3.2 : Communication of RTEs with Field Device and Smart Field Device

II. EMERGENCE OF COMMUNICATION BUS

In industrial automation systems, the field signals from FD have been traditionally transmitted to the control room using point-to-point communication methods that employ analog technologies such as the 4-20 mA current loop or, more recently, digital ones such as the RS-422 or RS-485. The main disadvantages are

- Increased cost: highly increased cost of cabling due to the need for a separate pair of wires for each device connected to the mainframe. It also makes system complex as separate connecting pair of wires for each device is required.

- Poor monitoring and control: 4-20 mA analog current loop, signals can be transmitted only in one direction. With the need for more complex monitoring and control of a process plant, monitoring is required both ways so it is very difficult to monitor the processes and its control.

2.1 The Industrial Communication Bus objectives

Higher level of technology and automation, a large degree of autonomy, improved flexibility, user friendliness and high level of machine and personnel safety are basic expectations. Machines and production systems are engineering artifacts containing proper housing with actuators and sensors that provide desired functionality. Many of these actuators, sensors, transmitters and other field device express inherently event driven behavior [5] like constant speed drive, pneumatic valves, limit switches, etc. Proper machine functionality is achieved by appropriate sequence and synchronization operations of these machine production systems.

What is more, a set of conditions for the intelligent execution of a particular operation has to be met. Next to the control functions for basic machine technology, the control functions for machine automatic cycle, manual with setup operation and error recovery is required. Besides these adequate, the industrial transmitters/actuator and other field device must fulfill some special construction requirements pertinent to harsh operating conditions in industrial environments [6].

In industrial plants these systems like transmitters are often installed at measurement sites where the operating temperature belongs to a very wide range (-30°C to +95°C, i.e. the industrial temperature range). Mechanical stress (vibrations) and prolonged exposure to a harsh environment, including chemically aggressive and flammable substances, are quite possible. The distances between the measurement sites and the control rooms can be long (up to the order of km). For this reason, electrical current is chosen instead of voltage for signaling, because current signals are less susceptible to degradation when transmitted over long distances.

A number of industrial interfaces exist on the market, with different capabilities, but incompatible with each other. In the literature they are often called protocols (for example the HART protocol). The most widely used are HART, ProfiBus and Foundation Fieldbus [2][4]. An industrial transmitter with both digital signal processing and digital two-way communication using one of the industrial protocols is called an intelligent transmitter. Such transmitters belong to the fourth technology generation [1].

Fieldbus technology was designed for geographically distributed harsh environment of process control applications. Also, it was conceived that there would be frequent changes in the installations. To meet these requirements the protocol includes the following aspects which are not necessarily found in other Protocols:

- Control algorithms may be in field-mounted Devices, central controlled or a combination of both.
- The End User does not have to be concerned with Device numerical address allocation. The Protocol handles this task, so 'plug and play' services are available for commissioning, modification and replacement.
- Devices do not have to be 'configured' before they are attached to the network.
- Device Definition and Function Blocks create a standard vendor-independent device interface for each device type which, in turn, facilitate installation, commissioning and upgradation of multi-vendor applications.
- The Physical Layer of the Protocol was designed from the outset to cope with installed cables and flammable atmospheres (hazardous areas).
- Both precise cyclic updates as well as acyclic and sporadic communications are catered for within the Protocol.
- Each variable transmitted on the Fieldbus carries with it tags indicating the current health of the source. Using this information, recipient Devices can take appropriate action immediately (for example switch to Manual, Off-line, etc.).

2.2 Network Communication Architecture of the ICBs:

- Installation and maintenance of these point to point communication media and their signal integrity become more and more difficult. As an alternative the network communication architecture presented as an attractive option. Firstly the cabling requirements are marginally increased as more and more devices are added to the network. Secondly, a vast array of high speed networking technologies is available at attractive costs from the computer market. Thirdly, with the addition of intelligent devices, such a system enables advanced monitoring supervision and control, leading to improvements in productivity, quality and reliability of industrial operations.
- The Industrial Communication Bus (ICB) is a standard for Local Area Network (LAN) of industrial automation field devices that enables them to intercommunicate. Typical Fieldbus devices are sensors, actuators, controllers of various types, such as PLCs, and DCS, RTEs and other computer systems such as human-machine interfaces, process management servers etc. It includes standards for the network protocol as well as standards for the devices on the network. As shown in Fig. 4.2, Fieldbus allows many input and output variables to be transmitted on the same medium such as, a pair of metallic wires, optical fibre or even radio, using standard digital communication technologies such as baseband time-division multiplexing or frequency division multiplexing.

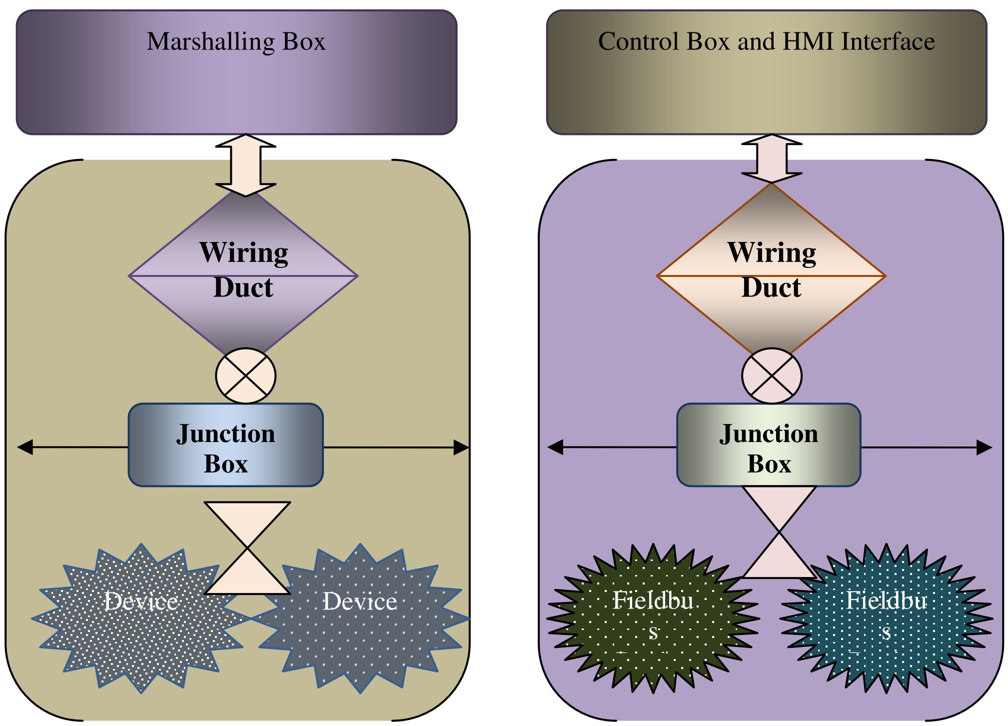

Figure 4.2., Wiring System for Point to Point on ICBs

Thus sensors transmit the measured signal values as well as other diagnostic information; the controllers compute the control signals based on these and transmit them to actuators. Further, advanced features such as process monitoring can be carried out leading to increased fault tolerance. Online process auto-tuning can be performed leading to optimized performance of control loops.

2.3 Standardization of ICBs in building adaptive networking and Electronic Device Description

The Open Systems Interconnect (OSI) model published by the International Standards Organization is a well known definition of network communications based on seven generic layers. It defines seven generic 'Layers' required by a communication standard capable of supporting vast networks. The first two layers, namely the Physical and the Data Link layers incorporate the technologies to realize a reliable, relatively error free and high speed communication channel among the communicating devices. It provides support for all standard and medium dependent functions for physical communication. DLL actually manages the basic communication protocol as well as error control set up by higher layers.

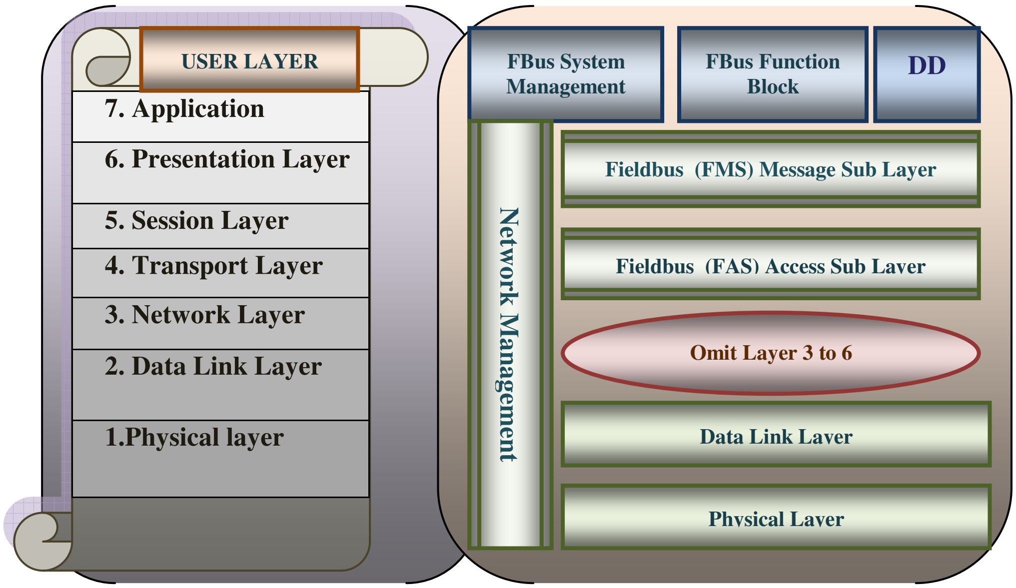

Figure 4.3, Industrial Communication Bus Network Architecture

In ICBs, since the communication takes place over a fixed network routing and transport layers are made redundant. Moreover, in an industrial control environment, the network software entities or processes are also generally invariant. Under such a situation, requirements of the session and the presentation layers are also minimized. Therefore, the third, fourth, fifth and sixth layers of the ISO protocols have been omitted in the ICB protocol. In fact the requirements of the omitted layers, although limited, have been included within the ICB Application Layer (7), which is sub-divided into two sub-layers, namely the ICB Message Sub-layer that builds up a message data structure for communication as per requirements of user layer and includes the roles of the session and presentation layers of the ISO-OSI model , and the Field Access Sub-layer that manages the functionality of the networking and transport layers to the extent needed and provides a virtual communication channel. Thus, the Foundation of ICB utilizes only three ISO model Layers (1, 2 and 7), plus an additional Layer referred to as the User Layer (8).

In the ICB standard, the User Layer (8) is also included in the specification. In this it differs from other communication standards. A typical function of the User Layer is to define control tasks for a process plant. This layer is embedded with the different standrised device driver which are capable of making controls and communication with the RTEs as in Fig. 3in accordance with Electronic Device Descriptions (EDDs) which is product specific for the FDs and SFDs. This help in solving typical tasks control and operation such as operation, The EDDs help in parameterizing and diagnosis of communication and other control function. In this fashion, one RTEs refer EDDs of FDs on the other RTEs and make effective communion using ICB.

Electronic Device Description Language (EDDL) allows vendors to describe their products in a way that may be interpreted by any compliant host systems thereby enabling compatibility and interoperability of devices. Also, the language allows vendors to include their specific product features while remaining compatible. Furthermore, the use of EDDL allows the development of new devices while still maintaining compatibility. EDDs primarily contain the description of all device parameters and their attributes (e.g. lower/upper value range, default value, write rights) and device functions, e.g. for the plausibility check, scaling, mode changes or tank characteristics. EDDs also include a grouping of device parameters and functions for visualization and a description of transferable data records.

Defining the User Layer functionality in terms of the open and published standards of Function Blocks enables interoperability of devices from different vendors. This is also allowing devices (RTES,FDS&SFDs) to interact by referring EDDs from application repository to implement the standard abstract block function interface which would interoperate, irrespective of their internal implementations.

III. CONCLUSION

Digital Communication is now became the requirement for a rapid development of digital data acquisition and control systems. By the end of the facts which are discussed above, catering the challenges of making two-way communication for the RTEs with the FDs and SFDs on ICBs has been explored and yet there are many challenges of the future. However, EDDs became a solution of the present which solve most of the issue pertaining to interfacing and communication complexity for the RTEs on ICBs.

REFERENCES

[1] C. L. Liu and J. W. Layland. Scheduling algorithms for multiprogramming in a hard-real-time environment. Journal of the Association forComputing Machinery, 20(1), 1973.

[2] Franc¸ois Jammes and Harm Smit. Service-oriented paradigms in industrial automation. Parallel and Distributed Computing and Networks, pages 716–723, 2005.

[3] Z.H Qian, H. Zhang and D.G. Peng, “Controlling methods of general purpose input/output for embedded PowerPC real-time system,” Measuring & Control Technology, 2007.26(12):58-60.

[4] G.S. Yang, H. Zhang, Z.G Niu, etc, “Reasearch and realization of embedded linux system based on MPC8248,” Measuring & Control Technology, 2008,27(1): 51-54.

[5] David S. Linthicum. Next Generation Application Integration: From Simple Information to Web Services. Addison-Wesley Professional, 2003.

[6] Holbrook, H. W., Singhal, S. K., and Cheriton, D. R. Log-Based Receiver-Reliable Multicast For Distributed Interactive Simulation. In Proc. Conference on Applications, Tech-nologies, Architectures, and Protocols for Computer Communication, ACM Special Inter- est Group on Data Communication (ACM/SIGCOMM'95) (1995), ACM Press, New York, pp. 328{341