

Distributed, Modular, Network Enabled Architecture For Process Control Military Applications

Abhijit Kamble*, Ashish Dubey**, C L Waghmare*, Rajesh Chary*, Arun Barde*, Moiz Chasmai*, Guru Prasad*

*(Research & Development Establishment (Engineers), DRDO, Pune)

***(Research & Development Establishment (Engineers), DRDO, Pune)

ABSTRACT

In process control world, use of distributed modular embedded controller architecture drastically reduces the number and complexity of cabling; at the same time increases the system computing performance and response for real time application as compared to centralized control system. We propose a design based on ARM Cortex M4 hardware architecture and Cortex Microcontroller Software Interface Standard (CMSIS) based software development. The ARM Cortex-M series ensures a compatible target processor and provides common core peripherals whereas CMSIS abstraction layer reduces development time, helps design software reusability and provides seamless application software interface for controllers. Being a custom design, we can built features like Built-In Test Equipment (BITE), single point fault tolerance, redundancy, 2/3 logic, etc. which are more desirable for a military applications. This paper describes the design of a generic embedded hardware module that can be configured as local I/O controller or application controller or Man Machine Interface (MMI). This paper also proposes a philosophy for step by step hardware and software development.

Keywords—COTS; ARM; Cortex; Ethernet; CMSIS

I. INTRODUCTION

The first industrial plants automation projects, developed in the 1970s, used electrical logic hard-wired into cabinets. These cabinets commanded all electrical equipment located in the plant, and contained all logic needed to perform any operating sequence. Cabinets were modular and composed of local panels, each of which acquired several input signals, and the outputs were set according to hardwired logic. Usually, no communication was established between these panels [1]. Change in requirement needs redevelopment of system. One possible way to cope with this problem is the use of programmable logic controllers (PLCs) and field I/Os. These devices are easily reconfigurable and can be adapted to a variety of situations.

In MIL Grade systems, there are two schools of thoughts one is to use centralised control system based on COTS architecture and other is to use completely distributed architecture. Use of standard and off-the-shelf products (COTs) for centralised control based on CPCI and VME architecture lowers the development cost, but usually raises production cost. The COTs products based on distributed/remote I/Os are available but do not meet MIL specifications. Unavailability of such hardware is the genesis of distributed, modular, network enabled architecture.

Today's embedded market has lot more to offer; from PIC to PowerPC/x86 architectures.

However, ARM architecture has grown with the fastest pace in last 5 years and has become a de-facto Industry standard in the embedded development. ARM has further refined and customised their architecture and come up with specific processors for bare metal to Operating System based applications. This architecture also ensures a compatible target processor and provides common core peripherals.

A systematic software development life cycle needs to be followed for development of applications on such a complex hardware. Development of such software is acknowledged as a major cost factor by the embedded industry. The cost and efforts can be reduced by standardization of the software.

The ARM Cortex architecture provides us an opportunity to do this by using CMSIS. A consortium of all the ARM silicon vendors ensures optimized, accurate and timely availability of CMSIS libraries.

Major thrust for our research was to design and develop MIL grade network enabled embedded controller hardware with modular and reusable software. We can develop variety of control systems for numerous process control applications using these ingredients.

II. SYSTEM DESCRIPTION

1. System Block Diagram

We have designed ARM Cortex based, network enabled, and configurable embedded controller modules. A module can be configured as Application Controller or Intelligent I/O Controller or MMI Controller. A control system can be built using these modules communicating over Ethernet/ RS485/ CAN backbone network as shown in Fig. 1, which can be used for variety of process control applications. The master application software will run on Application Controller; the heart of the control system. The Intelligent I/O Controller will be used for acquiring sensors data and controlling the actuators. The MMI Controller will be used for reading the switches and displaying the status of various activities. The number of modules and their role depends on the system requirements.

2. Architecture of Network Enabled Embedded Controller

The concept of the distributed controller is based on networked memory sharing architecture. Each embedded controller module will have two Cortex processors, both interfaced to a dual port RAM in such a way that both can simultaneously read and write to any memory location of the RAM. One of the processor will run Network Interface application called as Network Processor and the other one called Application processor will run Control, MMI or I/O application, based on the system requirements. Fig. 2 shows the detailed internal architecture of this embedded controller hardware. The Application processor will provide all standard interfaces to the outside world through discrete, analog, PWM, Quadrature encoder and serial I/Os. It will also have a dedicated interface for LCD and Debug/Trace. The Network processor shall have Ethernet/ RS485/ CAN bus interface. It will have add-on power supply module which can be selected based on the module power requirement.

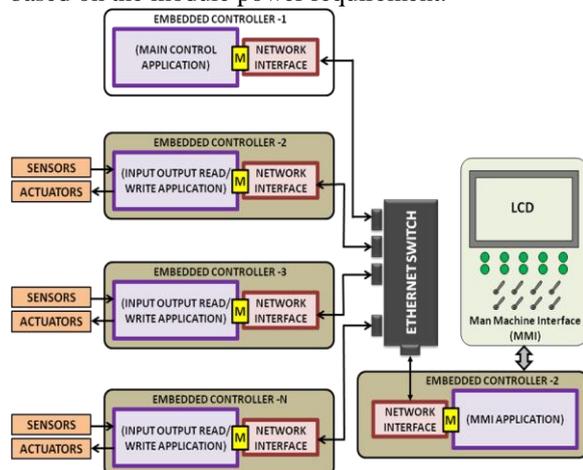


Fig.1. System Block Diagram

III. SELECTION OF PROCESSOR

The selection of right device on which to base our design was a challenging task; typically in the scenario where there is a need to have a right balance of price, performance and power consumption [2]. The x86(intel) and PowerPC processors are proven architecture which are currently being used. The ARM offers goodness of 32 bit processing and peripherals on a single chip (SoC) which makes the hardware design extremely simple. ARM eliminates the monopoly of a single silicon vendor as it is an IP core company with a huge silicon vendor base. Hence ARM was an obvious choice for this architecture.

We carried out comparative studies of different ARM flavors from different silicon vendors for parameters like processing power, latency, peripherals, support for floating point arithmetic, support for DSP functions and defined road map of at least 10 years.

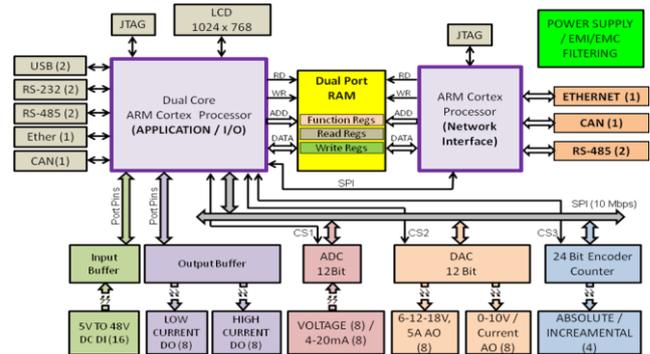


Fig.2. Internal Architecture of Embedded Controller

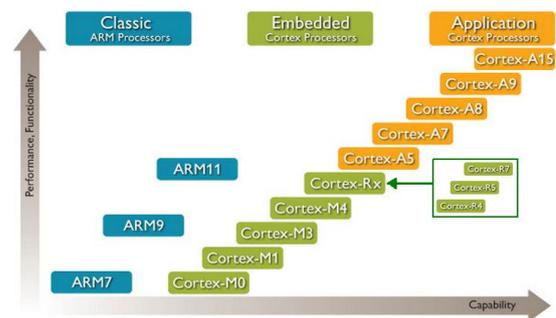


Fig. 3. ARM Processors Chart [3]

Fig. 3. shows the comparative chart for various ARM Processors for its capability, performance and functionality. Cortex-A series offer high performance processor for Operating System (OS), mainly used for smart phones and PDA applications. Cortex-R series offer exceptional performance processor for real time applications. The consistent interrupt handling structure and programmer's model of Cortex-M series provides a full upwards-compatible path for processors from Cortex-M0 to Cortex-M4.

Theoretically we should have chosen Cortex-R family for MIL applications. As Cortex-M4 has same performance as Cortex-R with wider silicon vendor support, we opted for Cortex-M4 over Cortex-R. The comparison of Cortex-M family is given in Fig. 4. LPC4350 Dual Core Cortex M4 & Cortex M0 (204MHz) [4] has been selected for Application/ IO/ MMI Controllers.

IV. SOFTWARE DEVELOPMENT APPROACH

Creation of software is acknowledged as a major cost factor by the embedded industry. The cost is significantly reduced by standardizing the software interfaces across all Cortex-M silicon vendor products, especially while creating new projects or migrating from existing software to a new device.

The Cortex-M4 architecture offered us an additional advantage in software development through its Cortex Microcontroller Software Interface Standard (CMSIS). It is a vendor-independent hardware abstraction layer for the Cortex-M processor series. All the major silicon vendors are part of a consortium which has standardized the software interfaces and are bound to release CMSIS libraries for the processors they are producing.

The CMSIS enables consistent and simple software interfaces to the processor and the peripherals, simplifying software re-use, reducing the learning curve for new microcontroller developers and reducing the time to market for new devices. CMSIS architecture is shown in Fig. 5.

The CMSIS consists of the following components:

1. Peripheral Register and Interrupt Definitions: A consistent interface for device registers and interrupts.

	Cortex-M0	Cortex-M3	Cortex-M4
Architecture Version	V6M	v7M	v7ME
Instruction set architecture	Thumb, Thumb-2 System Instructions	Thumb + Thumb-2	Thumb + Thumb-2, DSP, SIMD, FP
DMIPS/MHz	0.9	1.25	1.25
Bus interfaces	1	3	3
Integrated NVIC	Yes	Yes	Yes
Number interrupts	1-32 + NMI	1-240 + NMI	1-240 + NMI
Interrupt priorities	4	8-256	8-256
Breakpoints, Watchpoints	4/2/0, 2/1/0	8/4/0, 2/1/0	8/4/0, 2/1/0
Memory Protection Unit (MPU)	No	Yes (Option)	Yes (Option)
Integrated trace option (ETM)	No	Yes (Option)	Yes (Option)
Fault Robust Interface	No	Yes (Option)	No
Single Cycle Multiply	Yes (Option)	Yes	Yes
Hardware Divide	No	Yes	Yes
WIC Support	Yes	Yes	Yes
Bit banding support	No	Yes	Yes
Single cycle DSP/SIMD	No	No	Yes
Floating point hardware	No	No	Yes
Bus protocol	AHB Lite	AHB Lite, APB	AHB Lite, APB
CMSIS Support	Yes	Yes	Yes

Fig. 4. Cortex M Family Comparison [3]

2. Core Peripheral Functions: Access functions for specific processor features and core peripherals.
3. DSP Library: Optimized signal processing algorithms and for Cortex-M4 support of SIMD instructions.

4. System View Description (SVD): XML file that describes the device peripherals and interrupts. [3]

As shown in Fig. 6, a typical control application will consists of following software modules: (1) I/O Read/Write, (2) Communication, (3) Diagnostic/ Monitoring, (4) Typical Control Algorithm, (5) User Interfaces, (6) Timing/ Event/ Exception Handling, (7) Data Logging. These modules are identified based on the applications, i.e., Control, MMI, I/O and Network Interface. These software modules are either completely or partially reusable.

Keil Microcontroller Development Environment will be used as a platform for software development, as it provides tool support for Cortex-M series and widely used by the embedded industry all across the globe [5]. However, another widely used software development platform - Labview, which takes graphical code and generates procedural c code, can also be used in traditional programming environment [6].

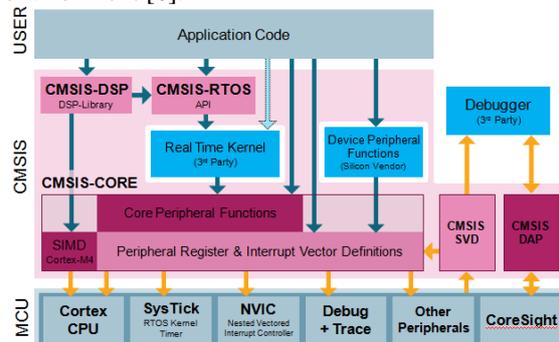


Fig. 5. CMSIS Architecture [7]

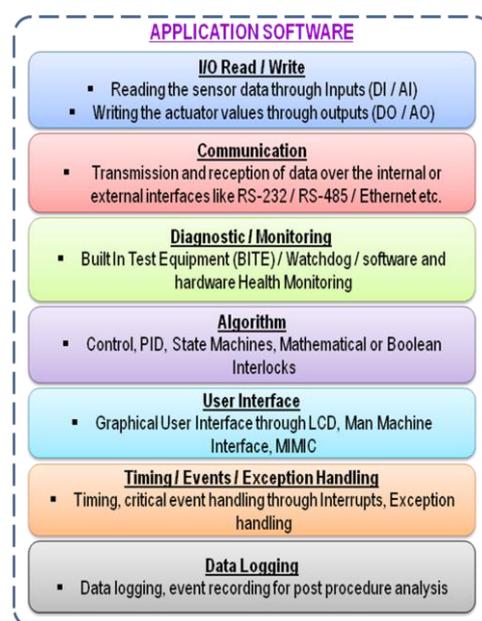


Fig. 6. Typical Software Module

V. SUMMARY

After studying the theoretical aspects of various architectures, we arrived at the conclusion that ARM Cortex-M4 Processor is best suited for our application. The concept of the distributed controller based on networked memory sharing architecture will certainly help in developing a more efficient control system in the field of military control applications. CMSIS software development standard shall be used to develop the software application for the system. It is quite obvious that in future this architecture will become pervasive because of its features like modularity, inter-changeability, space saving, low latency, connectivity with external networks and scalability which are unattainable with the traditional architecture. The design and architecture discussed in the paper will certainly help in developing process control applications in industrial and military environments.

VI. FUTURE SCOPE OF WORK

In future, the hardware will be realized using MIL Grade components and proper care will be taken in packaging in order to withstand harsh environmental conditions. Subsequently, the designed controller will be evaluated for different military process control applications. Also a future roadmap for system up-gradation will be developed. Key issues such as real time response, synchronization, reliable communication of different controllers, power consumption management,

cooperation, coordination, and security needs to be looked into in the future development.

VII. ACKNOWLEDGMENT

We are grateful to Dr. S. Guruprasad, Director R&DE (E), DRDO, Pune, India for allowing us to publish this work. We are also thankful to VV Parlikar and AN Ansari for their constant encouragement and scientific discussions.

REFERENCES

- [1] Juan Garcia, Francisco Rogelio Palomo, Antonia Luque, Jose M Quero, Daniel Carrion, Francisco Gamiz, " Reconfigurable Distributed Network Control System for Industrial Plant Automation" IEEE Transactions on Industrial Electronics, Vol.51 No.6, December 2004.
- [2] Frederic Gaillard, MCU VS MPU, pg. 1, atmel.com
- [3] www.emcu.it/CortexFamily/CortexFamily.html
- [4] www.nxp.com/documents/datasheet/lpc4350_30_20_10.pdf
- [5] www.keil.com/arm/mdk.asp
- [6] www.arm.com/products/processors/cortex-m/cortex-microcontroller-software-interface-standard.php
- [7] www.eeweb.com/company-blog/arm/the-choices-for-programming-arm-cortex-m-microcontrollers/