Embedded Systems in A new way

Anandaraj Shunmugam, Ramkumar Ramasamy
Lecturer II& Department of Computer Science & DMI- St. John the Baptist University
Mangochi, the Republic of Malawi

ABSTRACT: *Embedded System is combination of hardware and software that forms the component of larger system. Hardware is normally unique to given application. Computer chips are embedded into control electronics to manage the product's functionality. The embedded system can be categories into four categories viz.; stand alone system, real-time system, network appliance system and mobile devices. The new development tools available today make the task easy. Also the production cost is decreasing with increase in complexity. All these developments are leading to an era of invisible computing, or hidden computing where in computer does a job without a Ubiquitous and physical presence. Thus embedded devices are becoming smaller, smarter and more integrated. So needless to say, embedded software development is a very lucrative business these days.*

Keywords — *mobile devices, embedded system, hardware etc.*

I. INTRODUCTION

With advent of industrial automation, embedded systems have been around since the 1960s. An embedded system is a computing system hidden inside a product other than a computer. Hardware is designed Specific to the application. (Not a general purpose). Software is developed as per the requirement and specific to the hardware. All embedded systems use either a microprocessor or microcontroller. Some of the systems perform only some simple functions. Take in the case of an embedded system that controls printer.[1] This system has to execute more complex functions like checking for paper availability, printer ink, open door, paper jam, communication with host computer, data integrity etc are few of them. The software for the embedded system is called firmware. The firmware will be written in assembly language for time or resource critical operations or using higher-level languages like C or embedded C. The software will be simulated using microcode simulators for the target processor. Since they are supposed to perform only specific tasks, these programs are stored in Read Only Memories (ROM's).Embedded systems are also known as real time systems since they respond to an input or event and produce the result within a guaranteed time period.

A. CATEGORIES OF EMBEDDED SYSTEM

This categorization is based on whether the system has to work as an independent system or it has to be networked.

1) Stand-alone embedded systems:

As the name suggests, in a stand-alone mode work is taking input and producing output, which may be anything. Also the deadlines to carry out a specific task may not be strict; a few milliseconds variation not matter much. For example, an air conditioning unit can be set to turn on when the temperature is out of specified limit.

2) Real-time embedded systems:

Some embedded systems are required to carry out specific tasks in a specified time period. Such systems are called as *Real time system*. Systems in which real-time constraints are to strictly meet are called *hard real-time embedded system.* Systems in which real-time constraints are not so critical are called as *soft real-time embedded system*. For exam, opening a valve say 30 milliseconds when humidity crosses the limit

B. Embedded systems for Networked appliances:

Some embedded systems are connected to a network --typically, one based on a TCP/IP protocol suit, such as the Internet or companies Intranet. Even the Web server running HTTP can be embedded into the system. [2] A typical example is to monitor some parameter in the system and sending information on internet for on-line monitoring. A camera for this can be incorporated here

C. Embedded system for Mobile devices

With advent of wireless networks that support high speed, mobile devices are capable of supporting high data rates services in addition to the voice services. Accessing Internet services such as e-mail, the World Wide Web, and so on can be done while the person is on move. [3] So for downloading this all information, a power full processor is needed with high speed which can be an embedded processor.

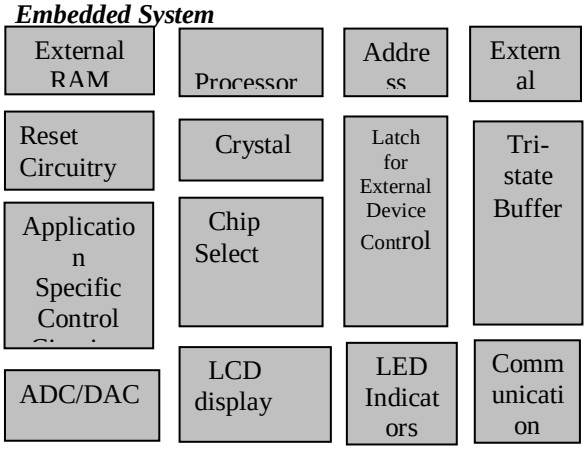
D. REQUIREMENT ISSUES

1) Throughput:
System may need to handle a lot of data in a short period of time.
2) Response: System may need to react to events quickly. Mostly for the real-time embedded system time constrains are strictly met.
3) Testability: Setting of equipment to test the embedded software can be difficult.
4) Debug ability: Without a screen or a keyboard, finding out what the software is doing wrong (other than not working) is a troublesome problem. A host system has the development tools are used for the development and initial testing [4].
5) Reliability: Embedded systems must be able to handle any situation without human intervention. Also system has to work without rebooting or resetting and so the hardware and software must be reliable.
6) Memory space: Memory is limited on embedded systems, you must make the software and the data fit into whatever memory exists as they don't have secondary memory mostly. The capacity depends upon the cost and its application.
7) Program installation: you will need special tools to get your software into embedded systems (programming kit or HDL).
8) Power consumption: Portable systems must run on battery power, and the software in these systems must conserve power. So the hardware design for the processor is simplified and is reduced.
9) Processor hogs: Computing that requires large amount of CPU time can complicate the response problem.
10) Cost: Reducing the cost of the hardware is a concern in many embedded system projects; software often operates on hardware that is barely adequate for the job. [5] As the embedded system is mostly for the dedicated application so for the mass production cost reduces. These requirement issues are met in today's era of Electronics.

E. HARDWARE ARCHITECTURE

The figure shows the common architecture of embedded

Fig 1: Hardware Architecture of Typical Embedded System



The components are explained bellow:

1) Processor:
The processor used in the embedded system can of three types:

- ***Micro-controller***
- ***Microprocessor***
- ***Digital signal processor***

Each of these processors is specified by clock speed and data word-length. The higher the clock speed and bigger the word-length the better the processor. The choice of the processor depends upon the application in use.

2) Memory:
The memory used in an embedded processor can be either internal or external. The internal memory is very limited. For the small application if this memory is sufficient then no need to use external memory. Internal memory is faster than the external, also the cost of memory increases with capacity and speed. Reprogrammable memories are now available.

3) Latches and buffers:
Processor-based system need to drive external devices such as LED displays, relays, etc. the processor does not interact directly with these devices. Flip-flop logic chips are used to drive external devices. These chips hold the processor output data to be sent to the external devices.

4) Application-specific control circuitry:
This is used for the control of the sensors and the relays etc. used to interface embedded system to other system under test.

5) Display units:
Some embedded system-for instance, those in an unnamed spacecraft or the process control system we do not need displays. But in system where the user interaction is important, LCD or other displays are used.

6) Keypads:
Every embedded systems offer different capabilities for providing user input. For example, for PDA's key pads are used for entering data and sensors in industries (machine).

7) Communication devices:
In order for the embedded system to interact with the external devices, they need communication interface. [6] Mostly it is serial. But in case of mobile embedded system it may be serial or parallel.

F. EMBEDDED SYSTEM DEVELOPMENT TOOLS

1) Hardware development tools:

- PCB making.

- Emulators.
- Logic analysers.

2) Software development tools:

- Embedded software development is typically done on a host machine, different from a target machine on which the software will eventually be shipped to customers.
- A tool for developing embedded software typically contains a cross-compiler, a cross-assembler, a linker/locator, and a method for loading the software into the target machine.
- The cross-compiler understands the same C languages a native compiler (with a few exemptions), but its output uses the instruction set of the target microprocessor.
- A cross-assembler understands an assembly language that is specific to your target microprocessor and outputs instructions for that microprocessor.
- A linker/locator combines separately complied and assembled modules into an executable image. In addition, it places code, data, start-up code, constant strings, and so on at suitable addresses in ROM and RAM.
- Linker/locators produce output in a variety of formats; it is up to you to ensure that your linker/locator's output is compatible with the tools you use for loading software into your target.
- You must find a way to load your software into the target system for testing. The most common ways include PROM programmers, ROM emulators, in-circuit emulators, flash memory, and monitors.

G. INTERACTIVE VOICE RESPONSE (IVR) SYSTEM: AN APPLICATION

Consider application of embedded system in IVRS.

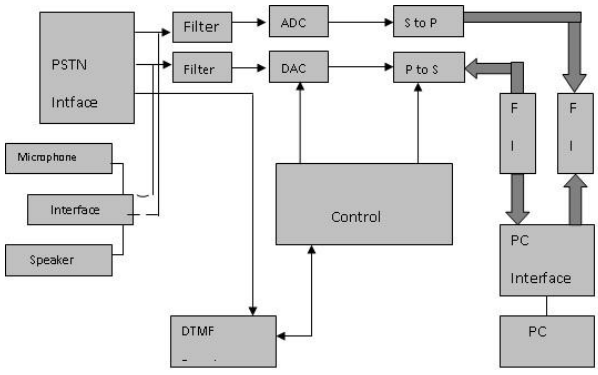


Fig 2: Interactive Voice Response System

Because of extensive automation tremendous lot of information is available on computer. Being able to access this information not just through computers but through telephones as will be a great boon. Interactive Voice Response Systems facilitates this. For instance, in many countries, you can use any telephone to retrieve your bank account balance from the IVR system installed in your bank. An IVR system is an embedded system connected to the computer holding the bank data base. Using IVR, the bank can eliminate the expense of additional employees' doing the mundane work of checking the database and informing the account holders. [7]The PSTN interface receives the telephone calls and answers them. Filters limit the audio signal to the desired frequency band (up to 4 kHz). The ADC (Analog-to-Digital Converter) converts the input signal to digital format to send it for further processing. The DAC (Digital- to-Analog Converter) takes the speech files stores in the IVR system and convert them into analog signal for transmission over the telephone line. The ADC outputs the digitized voice data in serial format, which is converted into parallel format using the S-to-P (Serial-to-parallel) converter. Similarly, the data in parallel format from IVR system is converted to serial format using P-to-S (Parallel-to-Serial) converter. FIFO is buffers that temporally hold the speech data. The digits entered by the subscriber (such as the account number) are in the form of DTMF (Duel Tone Multi Frequency) signals. An IC such as MT 8880 is used to decode the digits from the DTMF signals. Using this technology coupled with speech recognition and speech synthesis, you can now develop applications to browse the Web through voice commands.

H. ADVANTAGES AND PITFALLS

1) Advantages:

The overall cost of system is reduced. Even if hardware is not re-usable the *time-to-market* advantage is clear and important Consider the rapid evolution of domestic electronics, VCR's, televisions and microwave cookers need control panels/timers. These can be designed and taken to production quicker using the highly-integrated functionality of microcontrollers to form the heart of the system

1.1Other systems (machine tools, telephone switchgear...) can have software

1.2upgrades but utilise existing embedded hardware

1.3Any systems which would have required expensive hardware now upgrades in the past need only software changes this can sometimes be done remotely, using communication links [3].

1.4Mechanical systems can be more effectively controlled by microprocessor

1.5sensor derived data can lead to more effective control, thus reducing mechanical wear diagnostics are available

1.5Development time is reduced because of use of software on host system.

1.6More complex system can be developed in a simple way.

H. Pitfalls:

The pitfalls or problems with development in embedded system is that hardware fabricated no chip can never be changed. [8] Also embedded software programmers need to be aware of diversity of application areas and specific requirement for each.

I.APPLICATIONS OF EMBEDDED SYSTEMS

1) Consumer electronics:

Such as toys, air conditioners, refrigerators, microwave oven, CD player, mp3 players, TV sets, web enable TV sets are embedded system with varying processing power and memory requirements.

2) Control systems & industrial automation:

It includes fuel injection control, traction control, and climate control.

3) Biomedical systems:

Hospitals are full of embedded systems, including X- ray control unit, ECG and EEE units, equipment use for diagnostic testing such as Endoscopies.

4) Handheld computers:

Now with, low-cost 32-bit processors, the computing power on the handheld have increased multifold.

5) Data communication:

The modems that connect two computers-yours with one down by your ISP-is an embedded system.

6) Networked information appliances:

With network, a web camera with embedded web server can be to the Internet to access web pages from desktop browser.

7) Telecommunications:

Telecommunication infrastructure include networking component such as telephones switches, loop carriers, ISDN network terminations, ATM switches etc. [9]

8) Wireless communications:

These include mobile phones, PDA, Bluetooth devices such as fax machines, modems cellular phones, speakers etc.

I. CONCLUSION

The world of embedded system is dreamer's paradise with unlimited possibilities. Embedded system is rapidly becoming a catalyst for change in data communication, telecommunications, industrial control etc. New innovative application in these areas will roll out in near future. The global market size for embedded software development alone was $7 billion in 2001, which is expected to reach $31 billion by 2005. As per Nasscom-McKinsey survey, Indian R& D in embedded system was worth $1.1 billion in 2001, which would grow to $8 billion by 2008.

J. REFERENCES:

[1] Dr. K. V.K.K. Prasad, Vikas Gupta, Avinash Dass, Ankur Verma: "Programming For Embedded System" (Cracking The Code), Wiley- dreamtech India Pvt. (2003) Pg. 1-74.

[2] David Simon: "Primer Embedded system" (2003), Pg. 1-25

[3] B Ramakrishna Rau, Michael S. Schlansker: "Embedded Computer Architecture and Automation", computer-April-2001, Pg. 71-86

[4] Nilesh Kakde: "Possibilities Unlimited With Embedded System", Electronics For You, November-2003, 94-101

[5] Vahid and Givargis: "Embedded System Design" (2002), Wiley Pvt. Pg. 45-56

[6]"EmbeddedSystemandRTOS",www.cisco.com/embeddedsystem/tutorial.html

[7] "IntroductionToEmbeddedSystem", www.embedded.com

[8] "Chipdirectory", www.embeddelinks.com/chipdir

[9] "SystemOnChip", www.wipro.com/connect/webcast.htm