SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND TECHNOLOGY

COURSE NAME: 19CS603-Mobile Application Development

III YEAR /VI SEMESTER

UNIT – II Basic Design

Introduction – Basics of Embedded systems design

As computer technology advances, so its technology becomes embedded in more and more electronic products. The capabilities provided by embedded systems enable electronic equipment to have far greater capabilities than would be possible if only hardware techniques were used.

As a result, embedded systems are found in all manner of electronic equipment and gadgets. From small amounts of processing in items like electronic timers, to far more complicated embedded systems like gaming consoles and even major factory and other industrial systems.

The technique gains its name from the fact that the software is embedded into it for a particular application. The embedded system is not like a PC or other computer that can run a variety of programmes and fulfil a whole host of tasks, but instead the item using embedded software is focussed on one task or application. To meet this need, the item using an embedded techniques has its software preloaded, although updates may be undertaken from time to time.

Embedded systems basics

It may be asked what an embedded system is. With many processor-based systems and computers it is useful to define what an embedded system is. A convenient definition for an embedded system is: An embedded system is any computer system contained within a product that is not described as a computer.

Using this embedded system definition, it is possible to understand the various basic characteristics one. Typically, they are:

- Embedded systems are designed for a specific task. Although they use computer techniques, they cannot be used as a general-purpose computer using a variety of different programmes for different task. In this way their function can be focussed onto what they need to do, and they can accordingly be made cheaper and more efficiently.
- The software for embedded systems is normally referred to as firmware. Rather than being stored on a disc, where many programmes can be stored, the single programme for an embedded system is normally stored on chip and it is referred to as firmware.

Embedded systems contain two main elements:

- ***Embedded system hardware:*** As with any electronic system, an embedded system requires a hardware platform on which to run. The hardware will be based around a microprocessor or microcontroller. The embedded system hardware will also contain other elements including memory, input output (I/O) interfaces as well as the user interface, and the display.
- ***Embedded system software:*** The embedded system software is written to perform a particular function. It is typically written in a high-level format and then compiled down to provide code that can be lodged within a non-volatile memory within the hardware.

Embedded systems hardware

When using an embedded system there is a choice between the use of a microcontroller or amicroprocessor.

- ***Microcontroller based systems:*** A microcontroller is essentially a CPU, central processor unit, or processor with integrated memory or peripheral devices. As fewer external components are needed, embedded system using microcontrollers tend to be more widely used
- ***Microprocessor based systems:*** Microprocessors contain a CPU but use external chips for memory and peripheral interfaces. As they require more devices on the board, but they allow more expansion and selection of exact peripherals, etc, this approach tends to be used for the larger embedded systems.

Embedded systems software

One of the key elements of any embedded system is the software that is used to run the microcontroller.

There is a variety of ways that this can be written:

- ***Machine code:*** Machine code is the most basic code that is used for the processor unit. The code is normally in hex code and provides the basic instructions for each operation of the processor. This form of code is rarely used for embedded systems these days.
- ***Programming language:*** Writing machine code is very laborious and time consuming. It is difficult to understand and debug. To overcome this, high level programming languages are often used. Languages including C, C++, etc are commonly used.

The code for the embedded system will typically be stored on a form of non-volatile memory held on the processor board. The code is called firmware - the idea is that it is not updated in the same way that software is, being held in the embedded system and it cannot be changed by the user. Often it is possible to update the software, but this can mean changing the memory card on which the firmware is held, or by updating it in another way.

Often additional tools may be used to help with the development of the firmware. Often programmes can become complicated, and it is necessary to ensure the firm ware for the embedded system operates correctly.

Embedded systems design tools

Many embedded systems are complicated and require large levels of software for them to operate. Developing this software can be timing consuming, and it must be very accurate for the embedded system to operate correctly. Coding in embedded systems is one of the main areas where faults occur. To help simplify the process, software development tools are normally used. These help the software developer to programme more quickly, and more accurately.

Microcontroller MCU in embedded systems:

when developing embedded system hardware there is a choice of using a microprocessor or a microcontroller - when using a microcontroller what are the best approaches. When developing an embedded system, one of the options is to base the computational hardware around a microcontroller, MCU rather than a microprocessor, MPU.

Both approaches have their attractions, but generally they will be found in different applications. Typically, the microcontroller, MCU, is found in applications where size, low power and low cost are key requirements. The MCU, microcontroller is different to a microprocessor in that it contains more elements of the overall processing engine within the one chip.

Bringing most of the processing engine components onto a single chip reduces size and cost. This enables it to become economical viable to digitally control even more devices and processes. Also, it is found that mixed signal microcontrollers are being increasingly used, integrating analogue components needed to control non-digital electronic systems.

Microcontroller basics

Microcontrollers comprise the main elements of a small computer system on a single chip. They contain the memory, and IO as well as the CPU one the same chip. This considerably reduces the size, making them ideal for small embedded systems, but means that there are compromises in terms of performance and flexibility.

As microcontrollers are often intended for low power and low processing applications, some microcontrollers may only use 4-bit words and they may also operate with very low clock rates - some 10 kHz and less to conserve power. This means that some MCUs may only consume a milliwatt or so and they may also have sleep consumption levels of a few nanowatts. At the other end of the scale some MCUs may need much higher levels of performance and may have very much higher clock speeds and power consumption.