Real Time Embedded Systems

Course code: CoSc3026/SEng4033

By- Dr. Mohammad Nasre Alam

Chapter 1

Introduction

Contents

1. Definition, Characteristics and Example of RTES

Definition

Embedded system

- An embedded system is a combination of computer hardware and software designed for a specific function.
- The systems can be programmable or have a fixed functionality.
- Industrial machines, consumer electronics, agricultural and processing industry devices, automobiles, medical equipment, cameras, digital watches, household appliances, airplanes, vending machines and toys, as well as mobile devices, are possible locations for an embedded system.

1. Definition, Characteristics and Example of RTES

Different types of Embedded Systems

Standalone Embedded Systems

- This type of embedded software, as the name suggests, can work by itself, without requiring a host like a computer or processor. It simply takes input data in its digital or analog form and delivers an output that might be displayed through a connected device. Examples- cameras, digital watches, and MP3 players.

Network, Or Networked, Embedded Systems

- This type of embedded system relies on wired or wireless networks for output generation. Often, these kinds of platforms are built on general-purpose processors and consist of various components like sensors, controllers, and the like devices.
- Most popular examples of network embedded systems are home or office security systems, point-of-sale tools, and ATMs.
- For instance, a security system includes sensors, cameras, alarms, and similar gadgets to monitor for intrusions and alert the relevant staff.

Mobile Embedded Systems

- This type of embedded systems are portable and easy to move around. Typically, they are used in different kinds of mobile devices but naturally have some constraints when it comes to memory size.
- As you can imagine, despite the memory and functionality limitations, mobile embedded systems are quite popular solely because they work on-the-go.

1. Definition, Characteristics and Example of RTES

Different types of Embedded Systems

Real-time Embedded Systems

- This kind of embedded software needs to deliver outputs promptly, within a particular time frame. Hence, it is often used in time-sensitive sectors like transportation, manufacturing, and even healthcare, as all of these spheres rely on delicate processes to perform their business.
- Some examples of real-time embedded systems are aircraft or autonomous vehicle controls, traffic monitoring tools, and the like.
- Real-time embedded systems can be further broken down into **“soft”** and **“hard”** categories.
  - ➢ Soft Real-time Embedded Systems: Temperature or humidity monitoring tools. A minor delay in the acquisition of real-time temperature data might not be considered too critical, and obtaining the information a bit later will still be valuable.
  - ➢ Hard Real-time Embedded Systems: Aircraft control platforms. In this case, even a small delay in data acquisition can lead to disastrous consequences as the pilot might make a decision based on outdated information.

1. Definition, Characteristics and Example of RT and ES

Real-time Embedded System

- Needs timely computation
- Deadlines, jitters, periodicity
- Temporal dependency

Characteristics of RTES

• Speed (bytes/sec) :	Should be high speed
• Power (watts) :	Low power dissipation
• Size and weight :	As far as possible small in size and low weight
• Accuracy :	Must be very accurate
• Adaptability :	High adaptability and accessibility.
• Reliability :	Must be reliable over a long period of time.

1. Definition, Characteristics and Example of RT and ES

Examples of embedded systems

Automobiles

- Some embedded systems in consumer vehicles include cruise control, backup sensors, suspension control, navigation systems and airbag systems.

Mobile phones

- GUI software and hardware, cameras, microphones, and USB (Universal Serial Bus) I/O (input/output) modules.

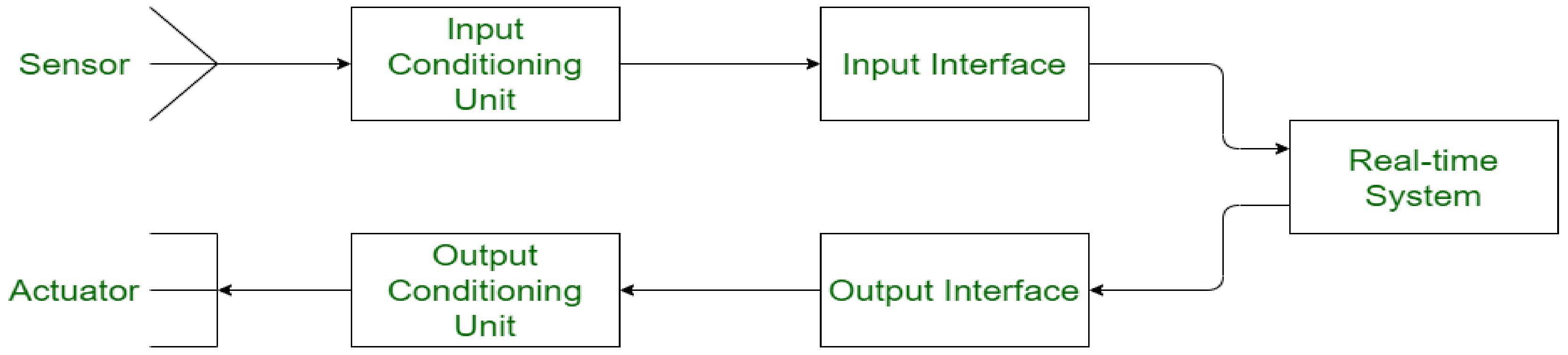
Industrial machines

- They can contain embedded systems, like sensors, and can be embedded systems themselves. Industrial machines often have embedded automation systems that perform specific monitoring and control functions.

Medical equipment

- These may contain embedded systems like sensors and control mechanisms.

2. Model of Real Time Embedded System



Sensor:

- Sensor is used for the conversion of some physical events or characteristics into the electrical signals.
- These are hardware devices that takes the input from environment and gives to the system by converting it.
- For example, a thermometer takes the temperature as physical characteristic and then converts it into electrical signals for the system.

Actuator:

- Actuator is the reverse device of sensor.
- It converts the electrical signals into the physical events or characteristics.
- It takes the input from the output interface of the system.
- The output from the actuator may be in any form of physical action. Some of the commonly used actuator are motors and heaters.

2. Model of Real Time Embedded System Cont..

Signal Conditioning Unit:

- When the sensor converts the physical actions into electrical signals, then computer can’t used them directly.
- Hence, after the conversion of physical actions into electrical signals, there is need of conditioning.
- Similarly while giving the output when electrical signals are sent to the actuator, then also conditioning is required.
- Input Conditioning Unit: It is used for conditioning the electrical signals coming from sensor.
- Output Conditioning Unit: It is used for conditioning the electrical signals coming from the system.

Interface Unit:

- Interface units are basically used for the conversion of digital to analog and vice-versa.
- Signals coming from the input conditioning unit are analog and the system does the operations on digital signals only, then the interface unit is used to change the analog signals to digital signals.
- Similarly, while transmitting the signals to output conditioning unit the interface of signals are changed i.e. from digital to analog. On this basis, Interface unit is also of two types:
- Input Interface: It is used for conversion of analog signals to digital.
- Output Interface: It is used for conversion of digital signals to analog.

3. Types of Real Time Embedded System tasks

Real-time tasks

- The tasks associated with the quantitative expression of time. This quantitative expression of time describes the behavior of the real-time tasks.
- All the real-time tasks need to be completed before the deadline. For example, Input-output interaction with devices, web browsing, etc.

Types of Tasks in Real-Time Systems

Periodic Task

- In periodic tasks, jobs are released at regular intervals. A periodic task repeats itself after a fixed time interval.

Dynamic Tasks

- It is a sequential program that is invoked by the occurrence of an event. An event may be generated by the processes external to the system or by processes internal to the system.

Critical Tasks

- Critical tasks are those whose timely executions are critical. If deadlines are missed, catastrophes occur.

Non-critical Tasks

- Non-critical tasks are real times tasks. As the name implies, they are not critical to the application. However, they can deal with time, varying data, and hence they are useless if not completed within a deadline.

4. Modeling Time Constraints in RTES

- Timing constraints decides the total correctness of the result in real-time systems.
- The correctness of results in real-time system does not depends only on logical correctness but also the result should be obtained within the time constraint.
- There might be several events happening in real time system and these events are scheduled by schedulers using timing constraints.

Classification of Timing Constraints

Delay Constraint –

- A delay constraint describes the minimum time interval between occurrence of two consecutive events in the real-time system.
- If an event occurs before the delay constraint, then it is called a delay violation.
- The time interval between occurrence of two events should be greater than or equal to delay constraint.

Deadline Constraint –

- A deadline constraint describes the maximum time interval between occurrence of two consecutive events in the real-time system.
- If an event occurs after the deadline constraint, then the result of event is considered incorrect.
- The time interval between occurrence of two events should be less than or equal to deadline constraint.

Duration Constraint –

- Duration constraint describes the duration of an event in real-time system.
- It describes the minimum and maximum time period of an event.

5. Computer Organization Concepts and Memory

Basic Terminology

• **Input:**	Whatever is put into a computer system.
• **Data:**	Refers to the symbols that represent facts, objects, or ideas.
• **Information:**	The results of the computer storing data as bits and bytes; the words, umbers, sounds, and graphics.
• **Output:**	Consists of the processing results produced by a computer.
• **Processing:**	Manipulation of the data in many ways.
• **Memory:**	Area of the computer that temporarily holds data waiting to be processed, stored, or output.
• **Storage:**	Area of the computer that holds data on a permanent basis when it is not immediately needed for processing.
• **Assembly language program (ALP):**	Programs are written using mnemonics.
• **Mnemonic:**	Instruction will be in the form of English like form.
• **Assembler:**	is a software which converts ALP to MLL (Machine Level Language)
• **HLL (High Level Language):**	Programs are written using English like statements
• **Compiler:**	Convert HLL to MLL, does this job by reading source program at once.
• **Interpreter:**	Converts HLL to MLL, does this job statement by statement
• **System software**	Program routines which aid the user in the execution of programs e.g.: Assemblers, Compilers.
• **Operating system:**	Collection of routines responsible for controlling and coordinating all the activities in a computer system

5. Computer Organization Concepts and Memory Cont.…

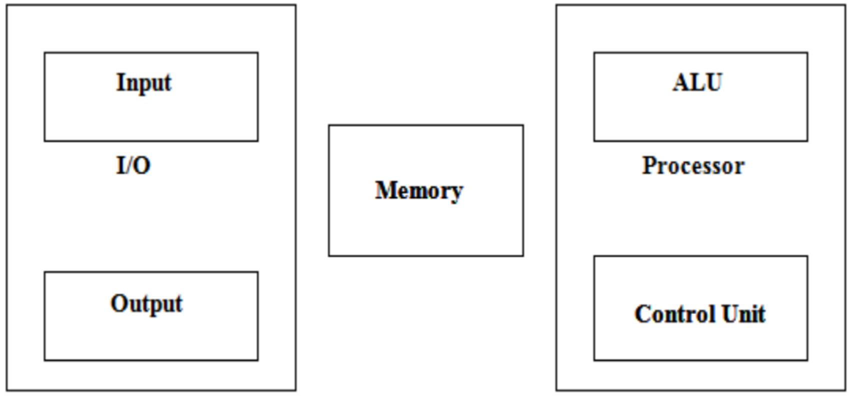
Computers has two kinds of components:

Hardware - Consisting of its physical devices (CPU, memory, bus, storage devices, ...)

Software - Consisting of the programs it has (Operating system, applications, utilities, ..)

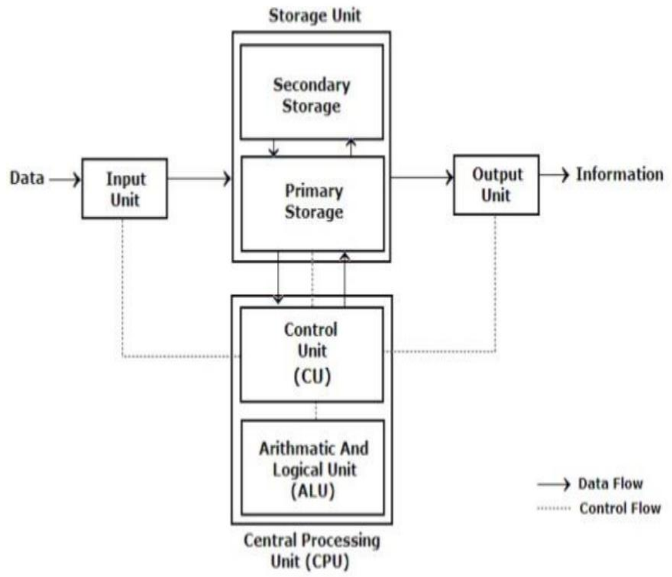
Functional Unit

- A computer consists of five functionally independent main parts:- Input, Memory, Arithmetic Logic Unit (ALU), Output And Control Unit.
- Input device accepts the coded information as source program i.e. high level language.
- This is either stored in the memory or immediately used by the processor to perform the desired operations.
- The program stored in the memory determines the processing steps.
- Basically the computer converts one source program to an object program. i.e. into machine language.
- Finally the results are sent to the outside world through output device. All of these actions are coordinated by the control unit.



Functional units of computer

Block diagram of computer



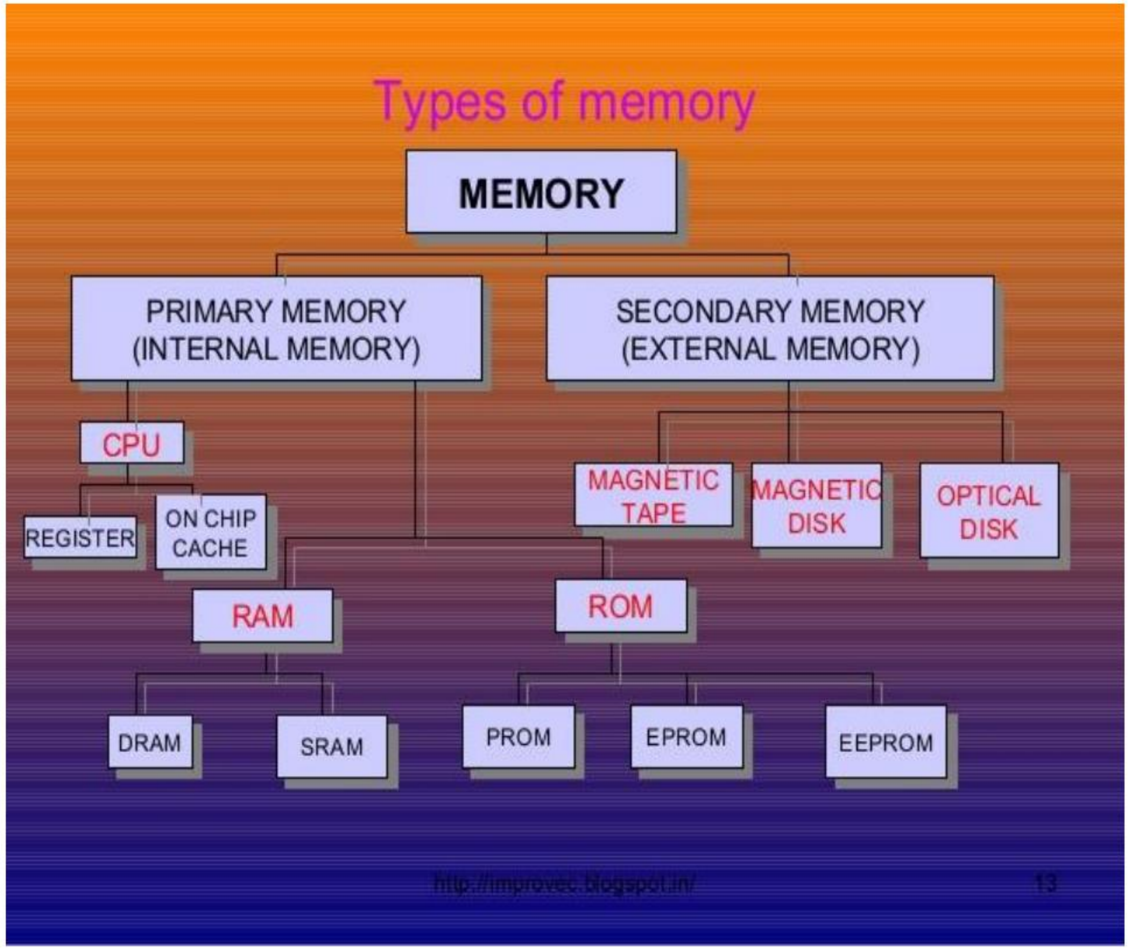
5. Computer Organization Concepts and Memory Cont.…

Input unit

- The source program/high level language program/coded information/simply data is fed to a computer through input devices keyboard is a most common type.
- Whenever a key is pressed, one corresponding word or number is translated into its equivalent binary code over a cable & fed either to memory or processor.
- Joysticks, trackballs, mouse, scanners etc. are other input devices.

Memory unit

- Its function into store programs and data. It is basically to two types-
  1. Primary memory
  2. Secondary memory



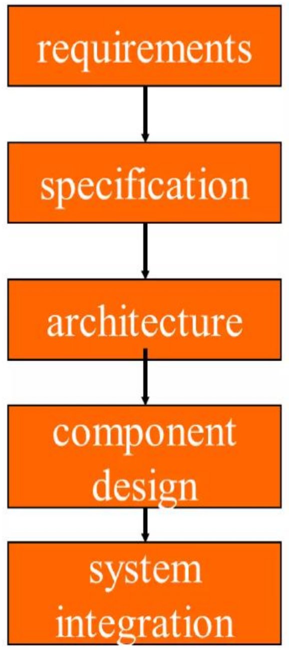
6. Embedded System Design Process

Design Methodologies

- It is a procedure for designing a system.
- It allows us to keep a scorecard on a design to ensure that we have done everything we need to do,
- It allows us to develop computer-aided design tools.
- A design methodology makes it much easier for members of a design team to communicate.

Major Goals of the Design

- Manufacturing Cost.
- Performance Speed.
- Power Consumption
- At each step in the design process
  - ➢ We must Analyze, Refine and Verify



➢ **Top-down design:**
- ➢ start from most abstract description.
- ➢ work to most detailed.

➢ **Bottom-up design:**
- ➢ work from small components to big system.

6. Embedded System Design Process Cont…

Requirements

- Before we design a system, we must know what we are designing.
- Plain language description of what the user wants and expects to get.
- May be developed in several way:
  - Talking directly to customers.
  - Talking to marketing representatives.
  - Providing prototypes to users for comment.
- **Functional Requirements:**
  - Output as a function of input.
- **Non-functional requirement:**
  - Performance- time required to compute output.
  - Cost
  - Size, weight, etc.
  - Power consumption.
  - Reliability etc.

Requirements Form

- **Name:**
- **Purpose:**
- **Inputs:**
- **Outputs:**
- **Functions:**
- **Performance:**
- **Manufacturing cost:**
- **Power:**
- **Physical size/weight:**

6. Embedded System Design Process Cont…

Specification

- It serves as the contract between the customer and the architects. As such, the specification must be carefully written so that it accurately reflects the customer's requirements and does so in a way that can be clearly followed during design.
- The specification should be understandable enough so that someone can verify that it meets system requirements and overall expectations of the customer.

Architecture Design

- Architecture is the plan for the overall structure of the system that will be used later to design the components that make up the architecture.
- Architecture design must give an idea about-
  - ➢ What major components need for satisfying the specification?
  - ➢ What hardware components need? Like CPUs, peripherals, etc.
  - ➢ What Software components need?
  - ➢ Must take into account functional and non-functional specifications.

6. Embedded System Design Process Cont…

Components Design

- The component design effort builds those components to satisfy architecture and specification.
- The components will in general include both hardware- FPGAs, boards and software module.
- Some components are ready-made, some can be modified from existing designs, others must be designed from scratch (that is new design).

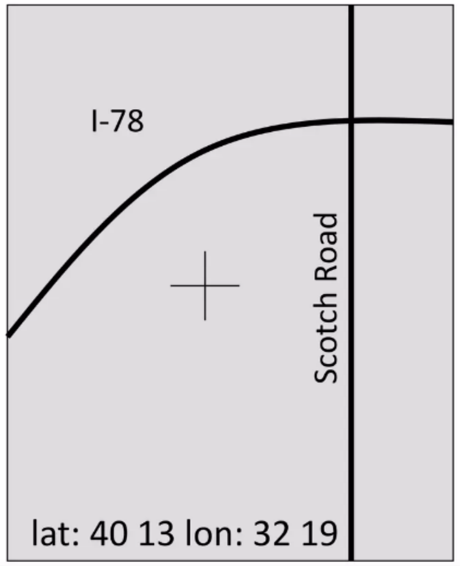
System Integration

- System integration is putting together the components.
  - Many bugs appear only at this stage.
- Bugs are typically found during system integration, and good planning can help us find the bugs quickly.
- System integration is difficult because it usually uncovers problems. It is often hard to observe in the system, to determine exactly what is wrong- the debugging facilities for embedded systems are usually much more limited.
- Inserting appropriate debugging facilities during design can help ease system integration problems would find on desktop system.

6. Embedded System Design Process Cont…

GPS moving map requirements

Name:	GPS moving map
Purpose:	consumer-grade moving map for driving
Inputs:	power button, two control buttons
Outputs:	back-lit LCD 400 X 600
Functions:	5-receiver GPS; three resolutions; displays current lat/lon
Performance:	updates screen within 0.25 sec of movement
Manufacturing cost:	$100 cost-of-goods-sold
Power:	100 mW

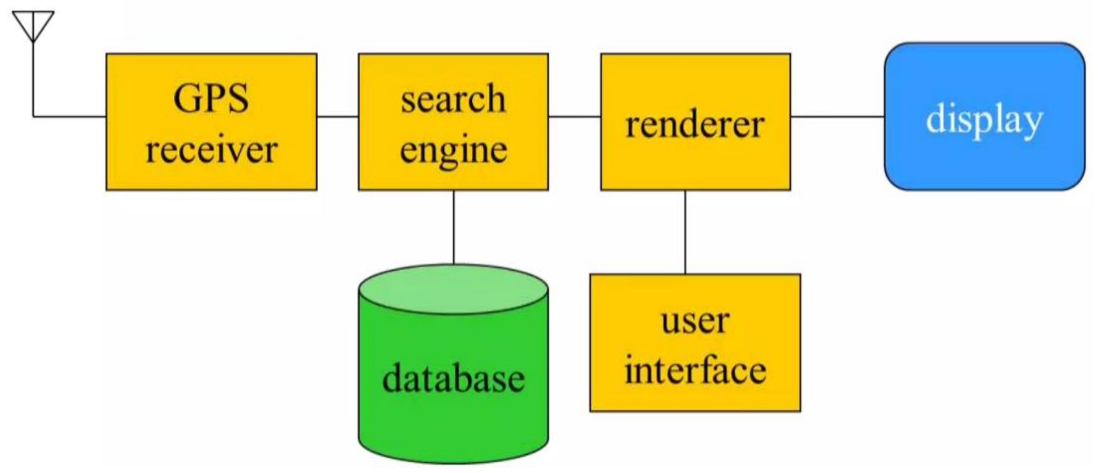


GPS Specification

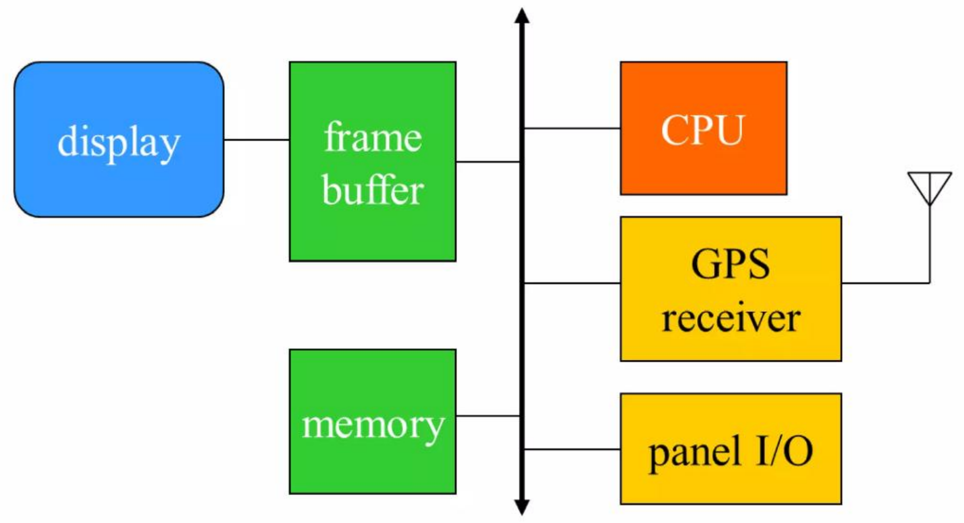
- Should include:
  - What is received from GPS.
  - map data.
  - user interface.
  - operations required to satisfy user requests.
  - background operations needed to keep the system running.

6. Embedded System Design Process Cont…

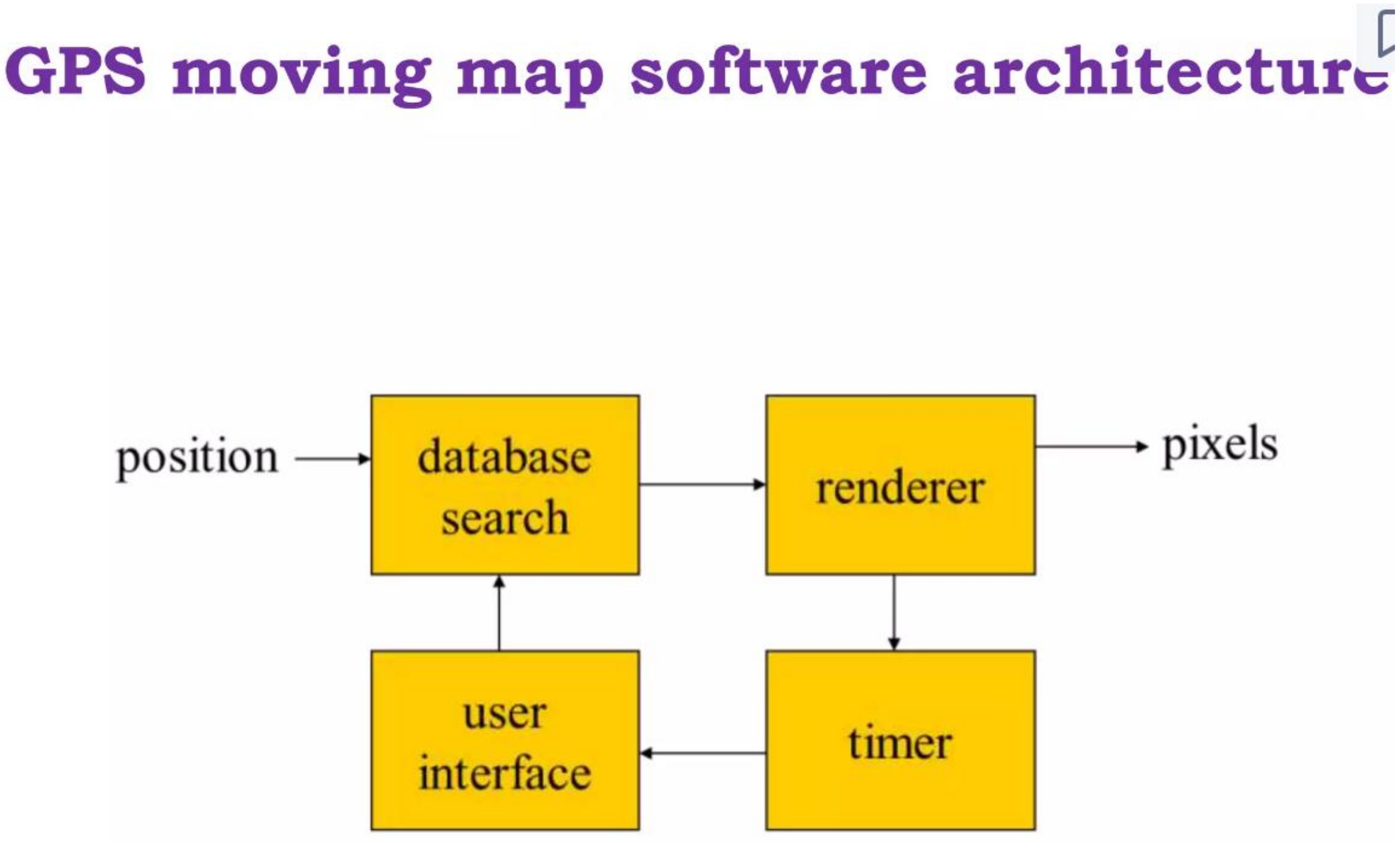
GPS moving map block diagram



GPS moving map hardware architecture



6. Embedded System Design Process Cont…



GPS moving map software architecture

Software's used for LAB

- Proteus Professional 8
- Arduino IDE

Project (Home Automation)

Group 1: Home temperature control
Group 2: Home Door control system
Group 3: Home light Control system
Group 4: Home fire alarm control system
Group 5: Home radar system

THANK YOU VERY MUCH

CHAPTER 1 COMPLETED