Real-Time Systems and Application- Introduction

Dr. J. K. Das, Dr. B. S. Patro

School of Electronics Engineering, KIIT DU

Table of Contents

RTES Intro

A real-time in this context of Real-Time System is quantitative notion of time measured using a physical clock.

For example:

take a chemical plant where temperature of reaction chamber exceeds 500°C, then the coolant should be occur within 100ms from time when temperature rises above 500° C. This occurs in real time is a quantitative notion of time.

RTES Intro...

Surge in Realtime application

Since many of the embedded real time systems are real time and there is a surge in embedded application, as the applications of real time systems has increased tremendously and is growing at a very fast rate. Look at all these different appliances and so on, starting with transportation, home appliances printer and many others.

Why surge in Embedded Application?

- Trends in reducing computers Processor, Memory
- Increasing flexibility due to internet
- Reducing power consumption
- Reducing in size
- increasing processing power and capability

Embedded System Applications

- Industry - *Chemical plant control, Automated vehicle assembly* plant and many more.
- Medical- *Robotic surgery, CT Scanning and many more.*
- Consumer Electronics - *Smart TV, Smart Lighting etc.*
- Peripheral Equipment-*Laser printers, digital camera etc.*
- Transportation- *MPFI Engine(multi point Fuel Injection) system, Automated car*

More on Embedded System Applications

- Aerospace- *cruise control*
- Telecommunication Application- *cellular application*
- Internet
- Multimedia application- *Video conferencing*
- Defense system applications- *missile guided system*


Patient
Video conference
Medical store
Emergency
Internet
Cardiolody
WBAN
Coordinator
Panel of Doctors
Telemedicine Hospital
Doctor Home

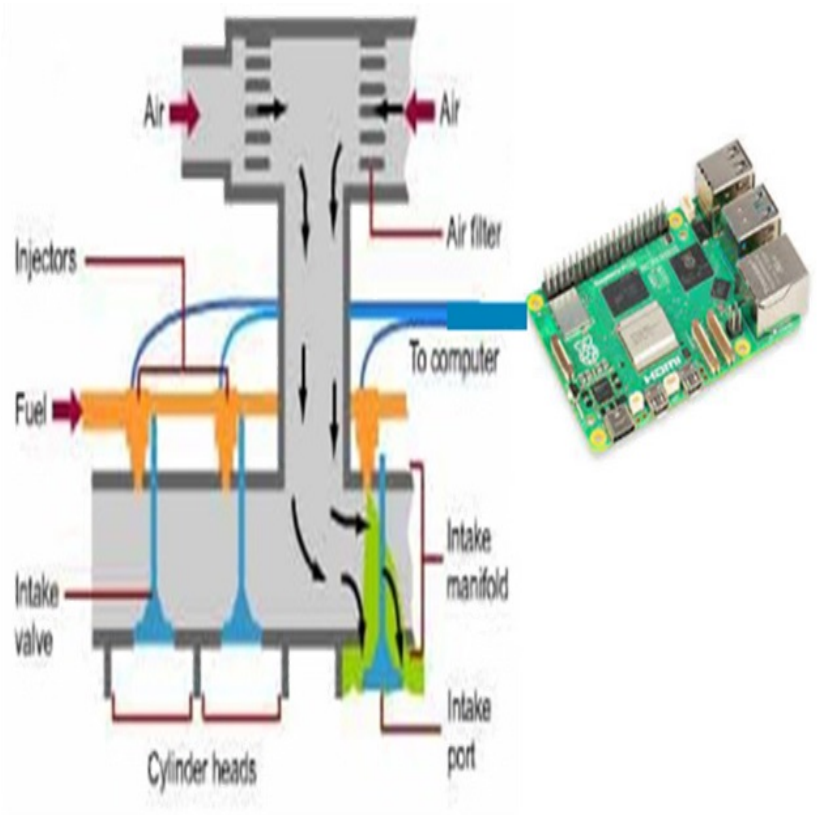
Typical Example – An Automotive Application

- About 100 processor per car used for Engine control, brake system, airbag system Wiper, door locks, entertainment
- Example: BMW 745i
  - 2,000,000 loc
  - Windows CE
  - Over 60 microprocessors
  - Multiple networks

MPFI- Multi-Point Fuel Injection

Controls the timing and amount of fuel injected:

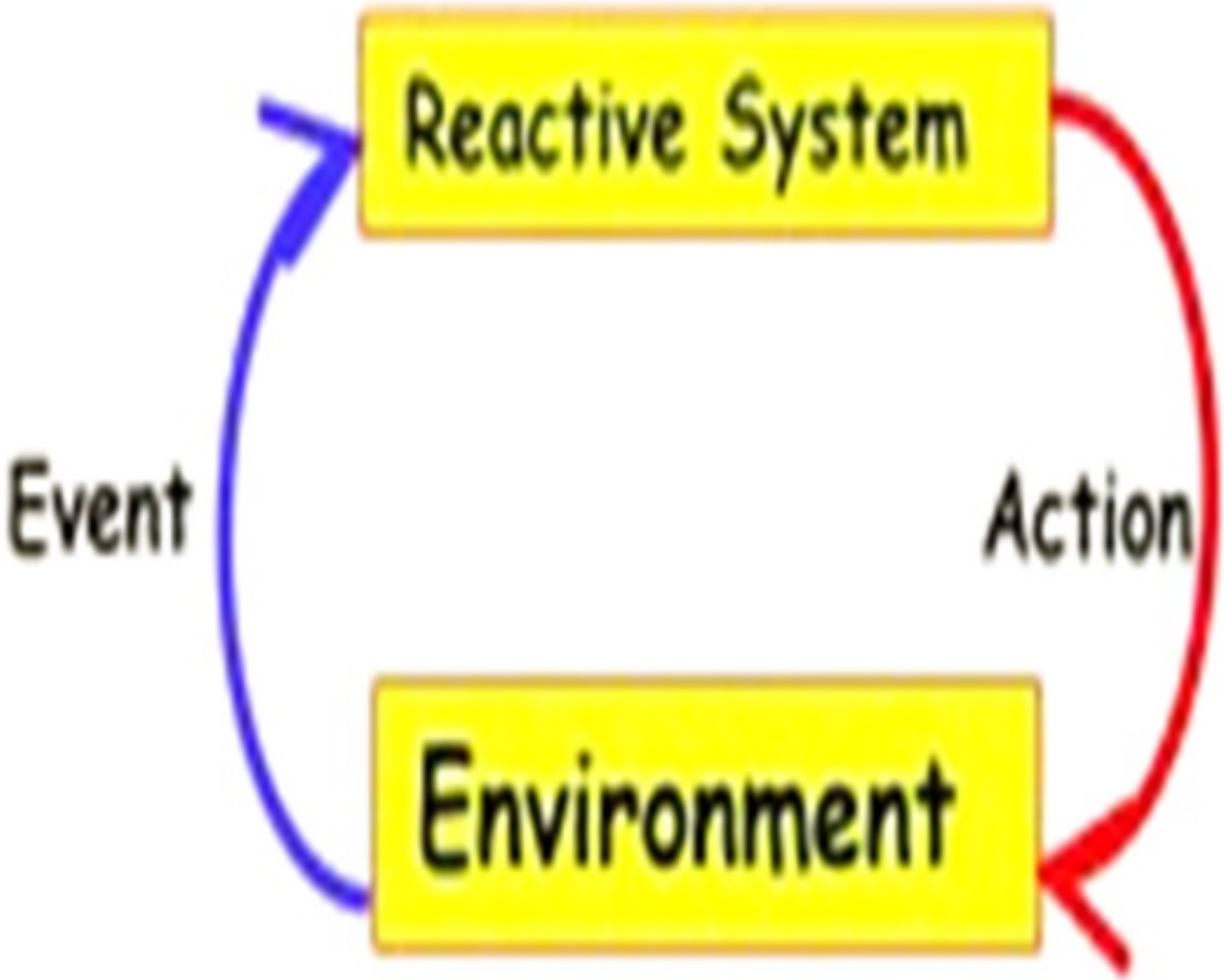
- Receives signal from various sensors
- Process the signals
- Send control signals to open the actuators.



What is Reactive System?

- Non-terminating interaction with environment
- responds to inputs
- event happens when there is a change in environment
- responses to all called actions or tasks.
- **Embedded System is generally reactive always gives responses to inputs.**

Reactive system



Events

- **Periodic:**
  - occurs according to a timer
  - almost all reactive systems are periodic in nature i.e. polling a temperature, pressure etc.
- **Aperiodic:** occurs randomly but not critical i.e reporting about an event or temperature etc.
- **Sporadic:** occurs randomly but critical in nature i.e. fire alarm

How to specify Embedded System?

- Model all actions and events
- large and complex system:
  - can have thousands of actions and events.
  - best modeled by taking subsystems

Basic Model of Embedded System

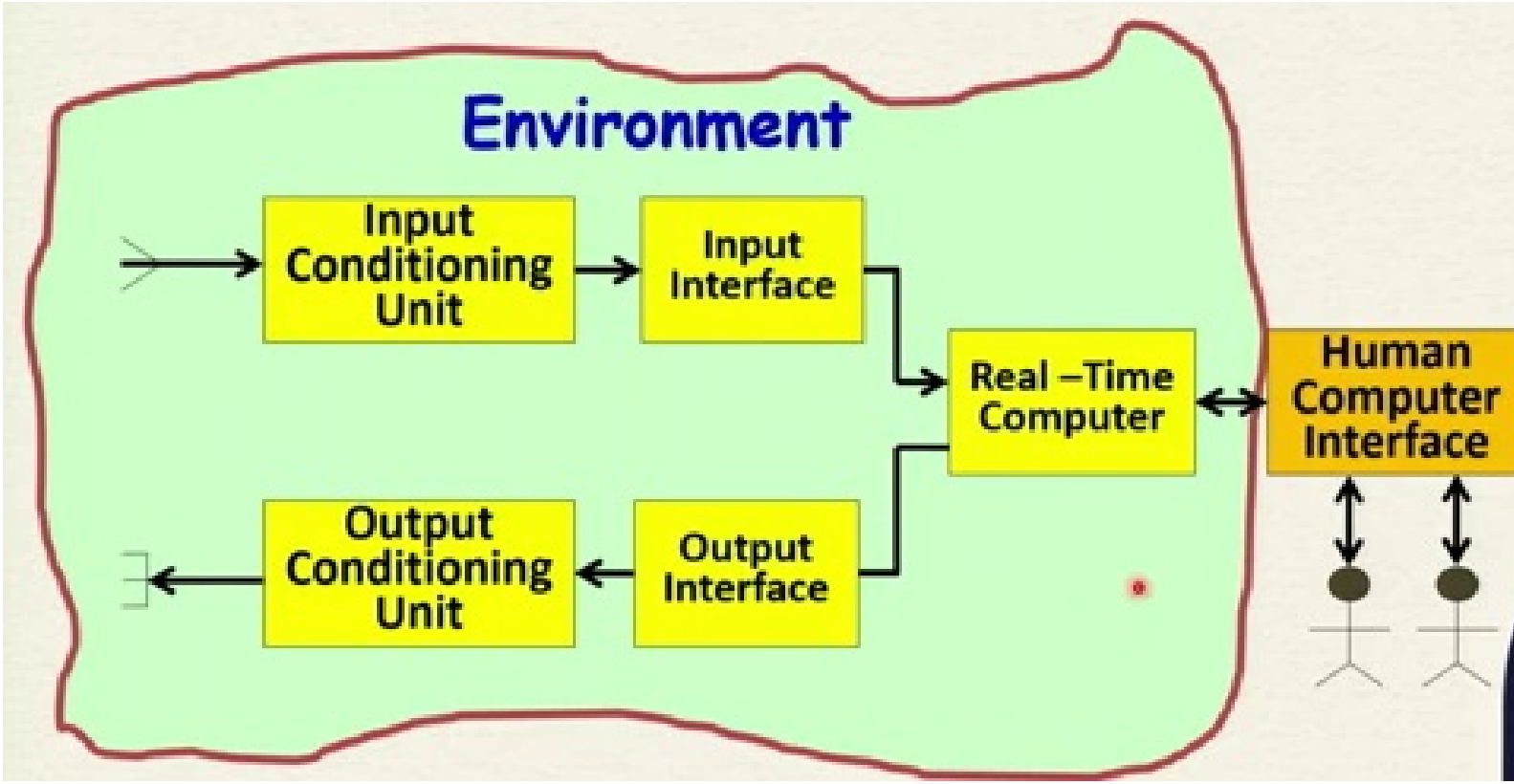


Figure: basic_model_ESD

Simplified Model of Embedded System

Many simple real-time systems are control Systems.

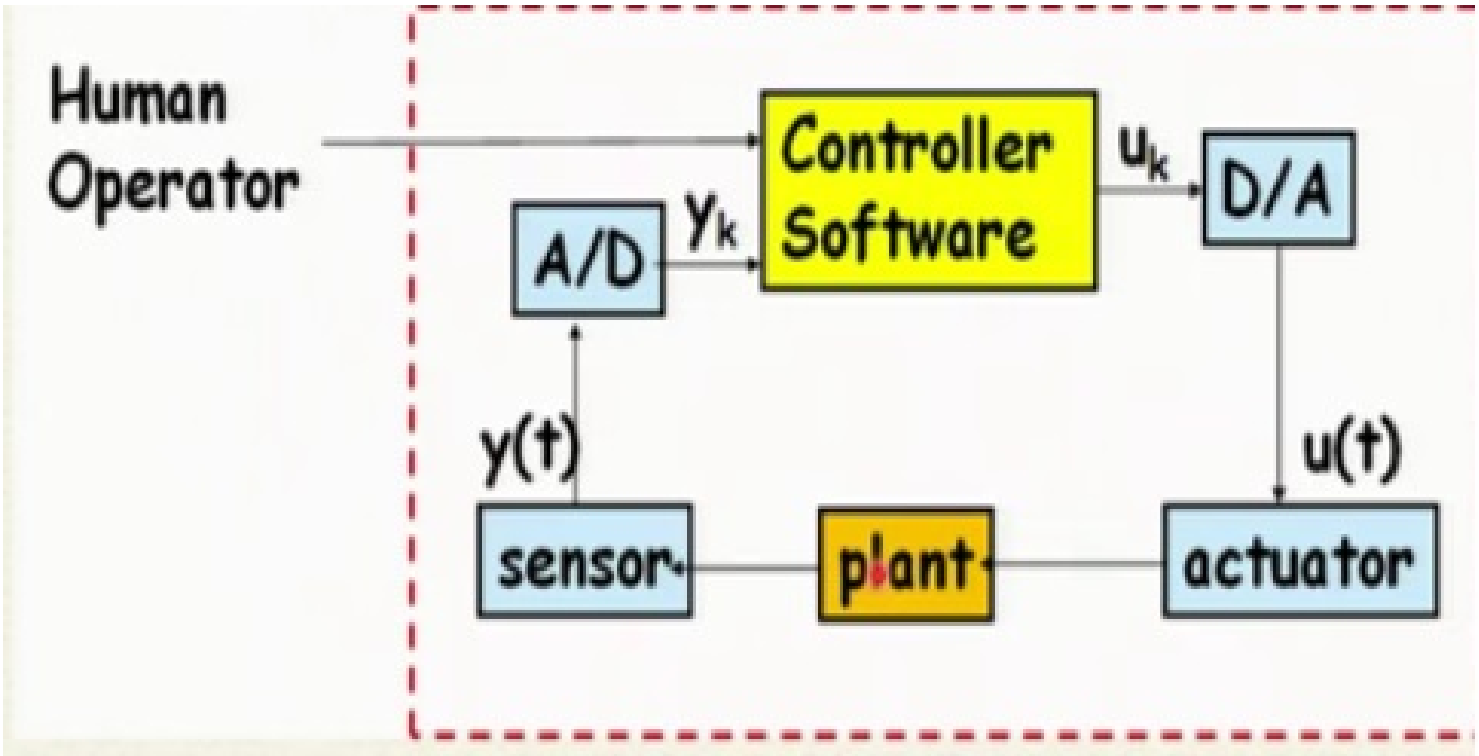
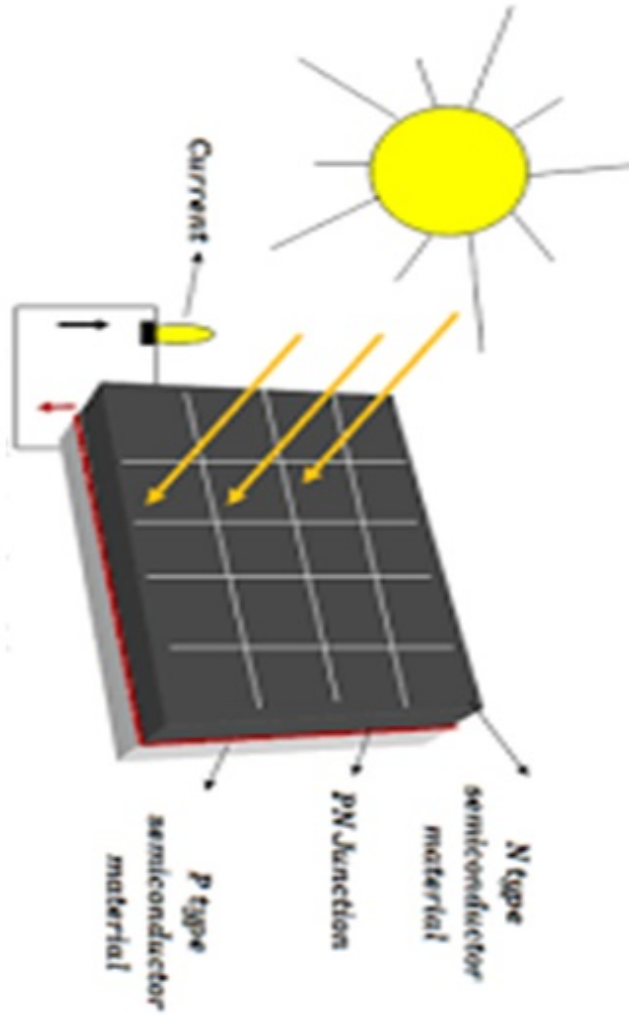


Figure: ESD_Simple_model

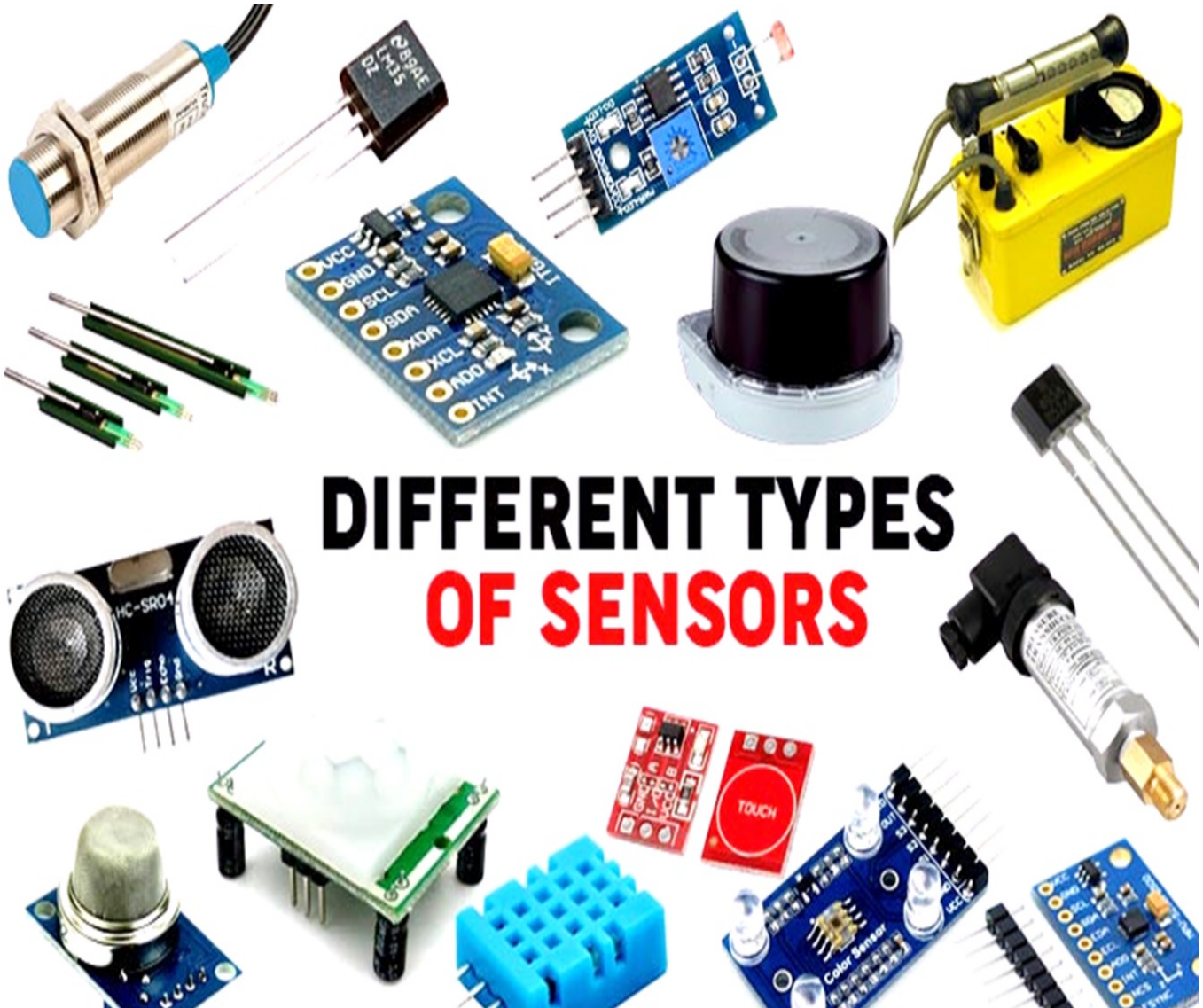
Sensors

A sensor converts some physical quantity/ characteristics of its environment to Electrical signals Examples:

- A photo-voltaic cell converts light energy into electrical energy
- A temperature sensors typically converts temp into equivalent electrical signal-thermocouple
- A pressure sensor.



Some Low cost Sensors



Actuators

An actuator converts electrical signal into some physical actions. The typical physical action may be Motion change of thermal, electrical, pneumatic, or physical characteristics of an object.

Example: Motors, Heaters, hydraulic and pneu-

Servos

A servo is a wired device that has a shaft and the shaft can be positioned at specific angular position.
For example this is done by a digitally coded signal.
Servos are mainly used in robots.

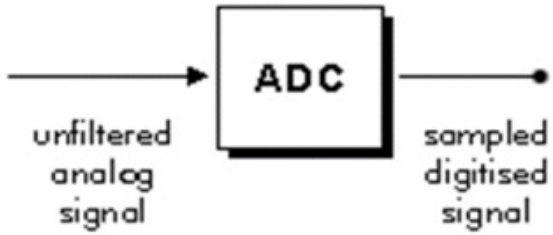
Signal conditioning

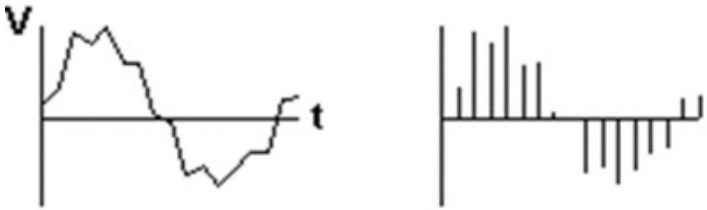
- Sensors generates signal:
  - A photovoltaic cell normally generates in millivolts range
  - Need to be conditioned before they can be processed by a computer.
- Important types of conditioning:
  - Voltage amplification
  - Frequency range shift and filtering
  - Signal modification.

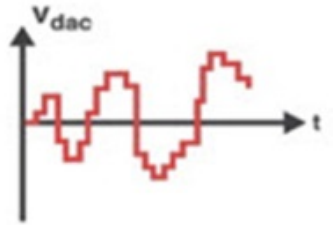
ADC& DAC

Analog-to-Digital Converter(ADC): Converts continuous signals to discrete digital signal numbers

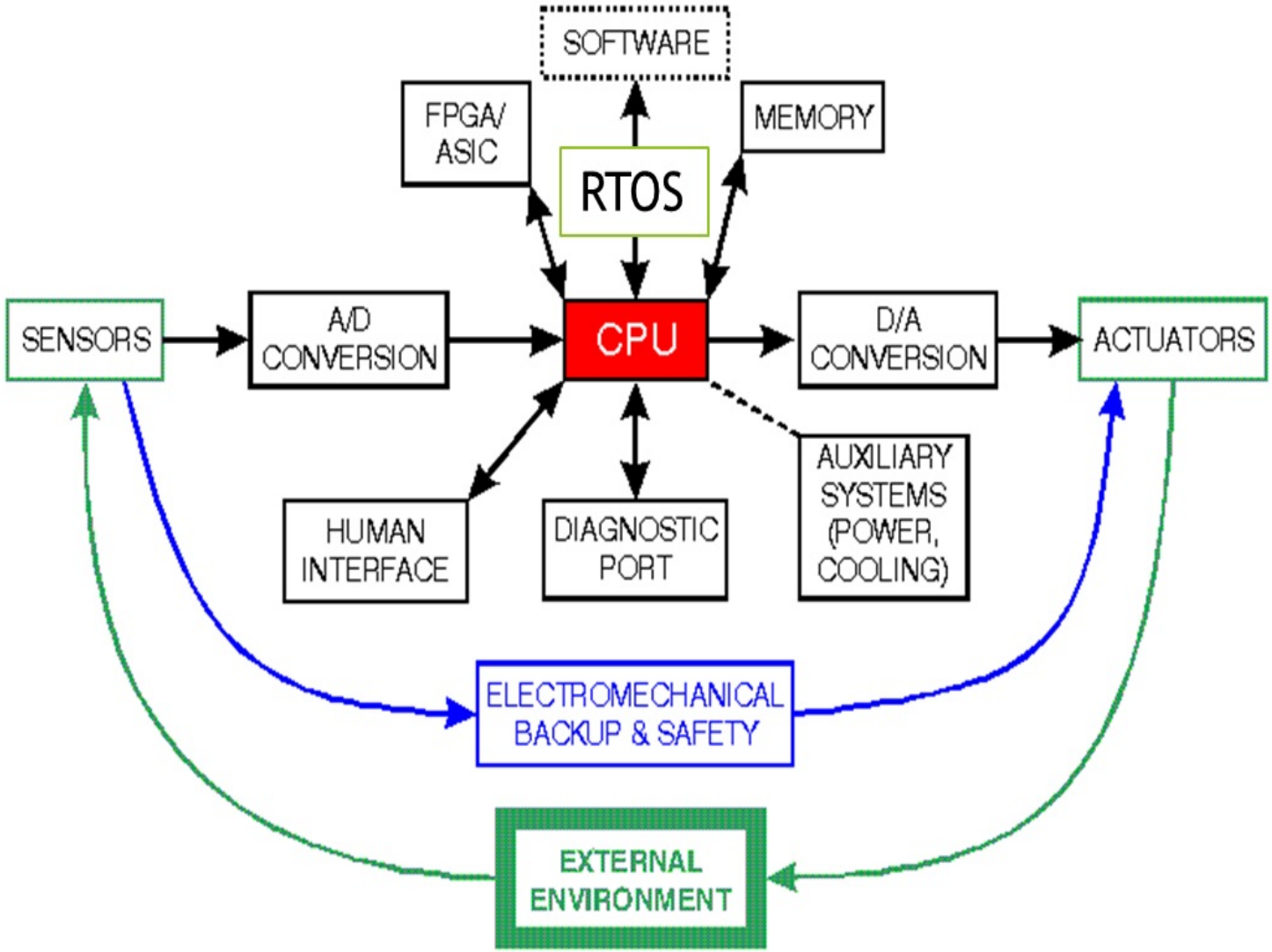
The reverse operation : Performed by a digital-to-analog converter(DAC)





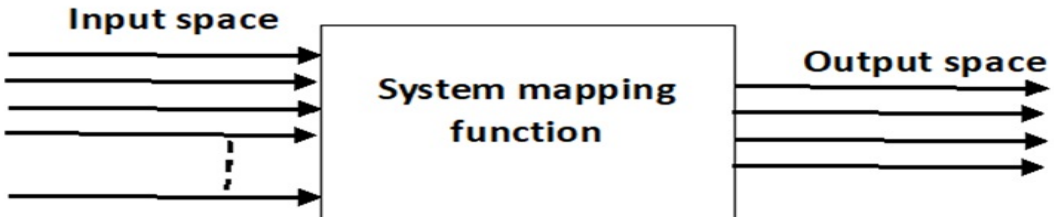


An Embedded System



Real-time System

A system is a mapping process is used to map n set of inputs into a set of output as shown in figure.



Response time

- The time between the presentation of set of inputs to a system and the realization of required behavior with available resources and the responses as a set of output called the response time.
- Response time may be slow or fast depend son the characterization of specific system.

Real-Time System *contd...*

- A real-time system is a computer system that must satisfy the bounded response time constraints or risk severe consequences including failure.
- a failed system is a system that can not satisfy one or more the requirements mentioned in system requirements specification.
- A real-time system is one whose logical correctness is based on both the correctness of outputs and their timeliness.
- Real time systems are often real-time or embedded systems. Real-time systems have continuous interaction with environment like fire control system etc.

Firm real-time system

- Firm real-time systems are those with hard deadlines where some arbitrary small number of missed deadlines can be tolerated.
- A firm real-time system is one which a few missed deadlines will not lead to total failure but missing more than few lead to complete or catastrophic system failure.

Example:
In a automated teller machine, missing too deadlines leads to a significant customer dissatisfaction and potential even loose of business to threaten existence of bank.

Soft Real-Time System

In soft Real-Time Systems,deadlines are soft deadlines.
for example on-line transaction. There is no rigorous about it that it meets its time constraints and less rigorous validations are required also meeting a deadline or aborting execution will not make loss to system.

Hard Real-Time System

Before implementation of Hard Real-Time systems, it must be validated invariable places where restriction on the design implementation of hard real-time applications are desired to decide the architecture and software requirements.

Real-Time System$_{contd...}$

- Thus system may either firm or soft or had real-time system is determined by system requirement that sets the realistic dead-lines expectations. In any situations, the principal goal of real-time system to trans-form into firm ones and firm ones to soft ones.
- Real-time Punctuality: It means that ev-ery response time has an average value tR with upper bound $t_R + \epsilon_V$ and lower bound $t_R - \epsilon_L$ respectively with $\epsilon_V, \epsilon_L \rightarrow 0^+$(finite time).

Response time Example

Take an elevator door which automatically operated and have sensors used for possible passengers between the closing door blades. Thus the door blades can be quickly reopened before they touch the passengers and else a disconnection which threatens the safety of passengers. Thus the response time has 5 components:

- Sensor response time t_{smin} =5ms, t_{smax} =15ms, t_{smean} = 9ms
- Hardware response time t_{HWmin} =1µs , t_{HWmax} = 2µs , t_{HWmean} =1.2µs
- System software response time t_{SWmin} =1.6µs , t_{SWmax} = 4s , t_{SWmean} = 3.7
- Application software response time t_{ASmin} = 0.5µs , t_{ASmax} = 0.5µs , $t_{}$ 0.5µs
- Door response time t_{DOmin} = 300ms , t_{DOmax} = 500ms , t_{DOmean} =400ms
- Total response time t_{min}=305ms, t_{max}=515ms, t_{mean}=409ms

Release Time

- Any occurrence that causes the program counter to change non-sequentially is considered a change of flow of control to another event.
- Release time is the time at which an instance of a schedule task is ready to run and generally associated with an interrupt. any event is always caused by an interrupt as well any branch instruction. The event may be either asynchronous or synchronous.

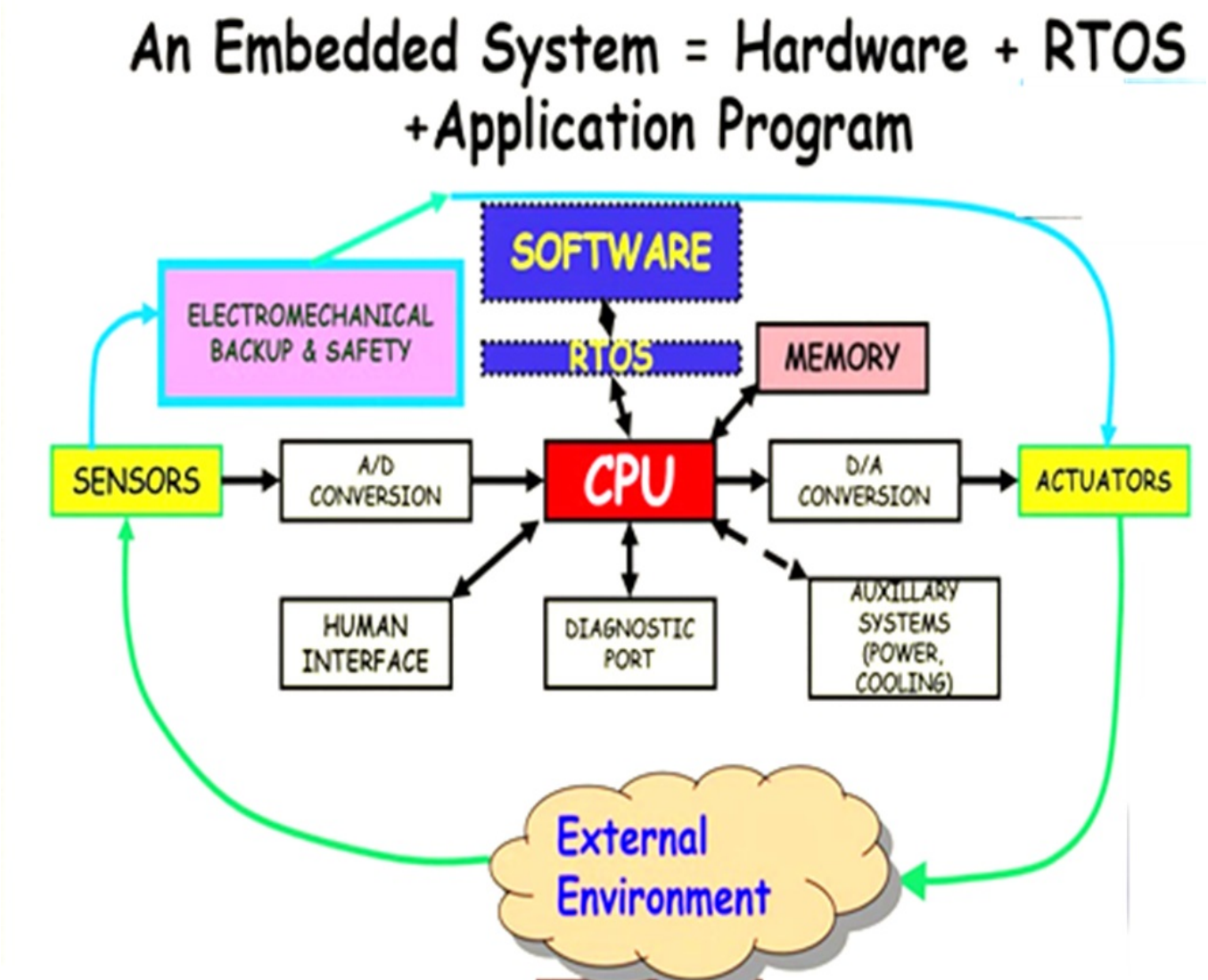
Release Time *contd...*

- Synchronous event occurs at regular interval of time which is predictable.
- Asynchronous events occurs unpredictable which occurs randomly caused by external sources.
- Event that does not happen periodically called aperiodic event and event which occurs very infrequently are called sporadic events.

Deterministic Event

A system is deterministic if for each possible state and each set of inputs, a unique set of outputs and next states of the syst1em are predictable or can be determined.
When a system is deterministic, next state and outputs also known for each set of inputs.

Characteristics of Real-Time Systems



Important Characteristics Real-Time Systems

An embedded system responds to events in time constrained manner.

Example: An automobile airbag system When the airbag's motion sensors detect a collision, the system needs to respond by deploying the airbag within 10ms or less or system fails!

Important Characteristics Real-Time Systems$_{contd...}$

- Time constraints:
  - Some tasks are real-time, not necessarily all tasks
  - Each real-time task is associated with some time constraints e.i. a deadline.
- New correctness criterion:
  - Result should be logically correct
  - And within the stipulated time
- Safety and task criticality
  - A critical task is one whose failure causes system failure
  - a safe system does not cause damage.
  - A safety critical real-time system is one where any failure causes severe damage.

Important Characteristics Real-Time Systems$_{contd...}$

Concurrency

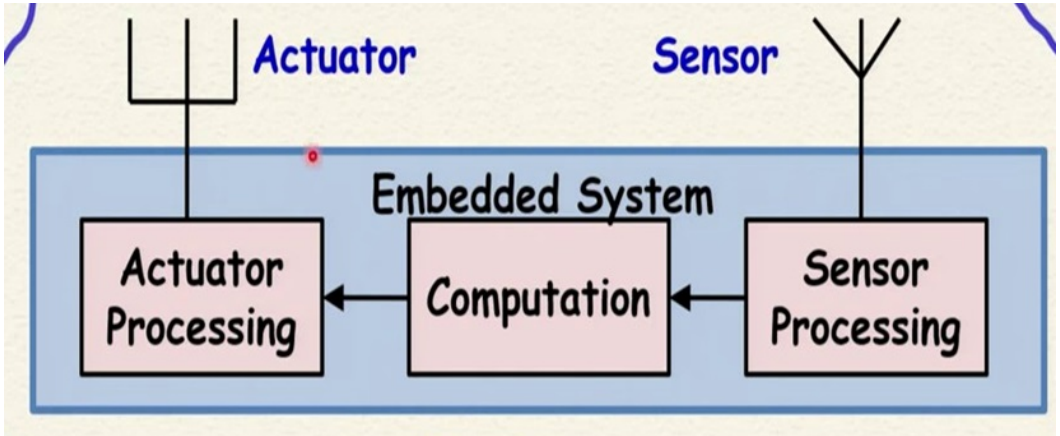
- A real-time system needs to respond to several independent events
- Typically for each independent event, separate tasks are created.
- For example, processing coolant level, temperature and pressure, process user requests etc.

Important Characteristics Real-Time Systems$_{contd...}$

- Distributed and feedback structure
- Embedded - Overwhelming majority are embedded system
- Custom hardware: An embedded system is often implemented on custom H/W that is specially designed and developed for the purpose.

Important Characteristics Real-Time Systems$_{contd...}$

☞ Feedback structure of real-time systems



- Should be Reactive – interaction between system and physical environment
- Stable – under overload condition critical task should meet their deadlines
- Exception handling

Safety & Reliability

- ☞ A safe system
- ☞ Does not cause damage even when it fails
- ☞ A reliable system- Operates for long time without any failure
- ☞ Independent concepts in traditional system so systems are generally unreliable and not safe to implement.

How to achieve High Reliability?

I order to achieve high reliability, followings should be followed:

☞ Error Avoidance- every possible error in the system should be minimized

☞ Error Detection and Removal- In spite of best possible way of error avoidance, some times error do occurs and in these cases it should be detected properly and removed.

☞ Fault-Tolerant -No matter how effectively error are avoided,system should tolerate the faults that may appear and compute correct results.

Real-Time Task

- Periodic: the periodic tasks, these repeat after certain fixed interval of time, typically the hard real time tasks are periodic in nature
- Sporadic: they occur at random instance, and have hard deadlines.
- Aperiodic:Reccuring irregular or random manner and usually it is soft realtime system. May occur zero delay time.

Real-Time Tasks

- Hard-Real-Time Task:it one which is constrained to produce its response within certain predefined time bounds. If the time constraint did not meet, system fails. Best example is Robot.
- Firm Real-Time Task: Every Firm real time task associated with a deadline but unlike hard real time task, it does not fail when time constraints are not met. The late results are simply discarded.
- Soft Real-Time Task: It has also time constraints, but have no absolute value of time constraints. instead the time constraints are expressed in terms of average value.

Events in Real-Time System

- stimulus event
  - generated by environment
  - typically asynchronous in nature bu can have periodic nature also
- response events
  - produced by the system in response to a stimulus
  - for example of switching a heater On or OFF depending on temperature sensed.

Classification of Time Constraints

The different time constraints are classified into 2 basic types which are

- Performance time constraints- imposed on response of the system
- Behavioral Time constraints - imposed on stimulus events

Types of Time Constraints

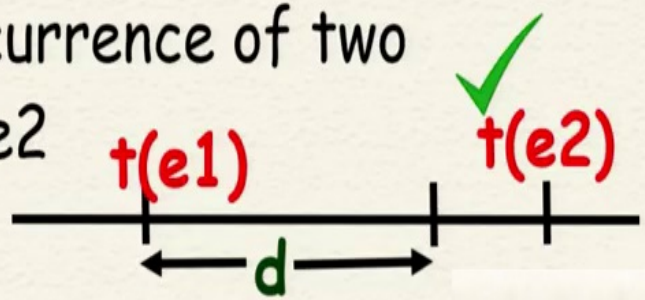
Both performance and behavioral time constraints are of three types:

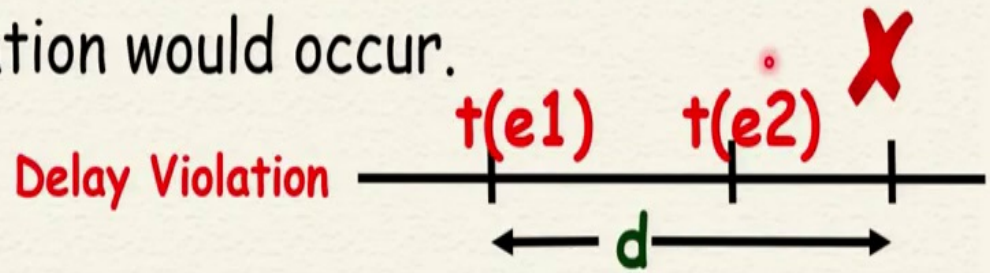
- Delay Constraints
- Deadline constraints item Duration Constraints

Delay Constraints

Delay Constraint

- Expresses minimum time delay d:
  - Needed between the occurrence of two arbitrary events e1 and e2
  - t(e2) - t(e1) ≥ d
  - if e2 occurs earlier than d then a delay violation would occur.

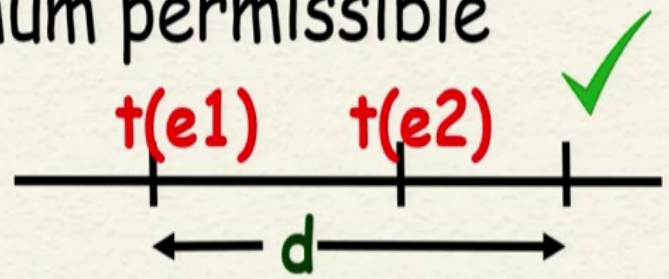




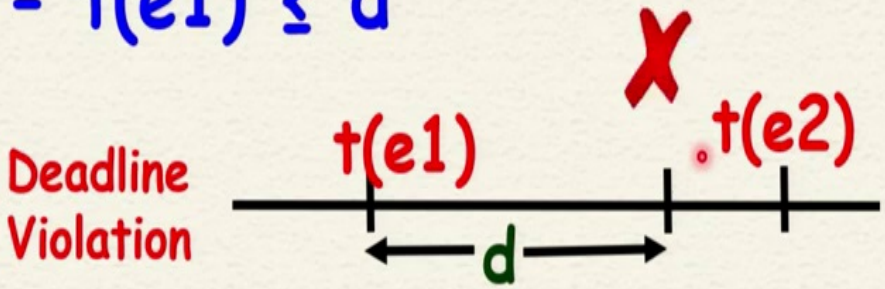
Deadline Constraints

Deadline Constraint

- Expresses the maximum permissible separation:



  - Between any two arbitrary events.
  - $t(e2) - t(e1) \leq d$



Duration Constraints

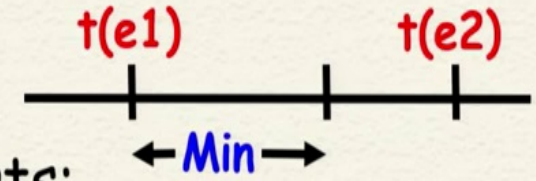
Duration Constraint

- A duration constraint on an event:
  - Specifies the time period over which the event acts.
- A duration constraint can be:
  - minimum type
  - maximum type.

Duration Constraints$_{contd...}$

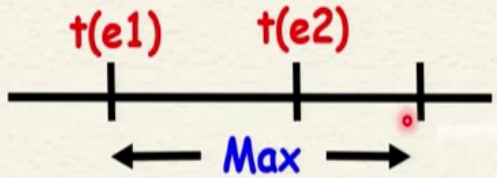
Duration Constraints

- **Minimum:**



  - Once a duration event starts:
    - It must not end before a certain minimum time.

- **Maximum:**



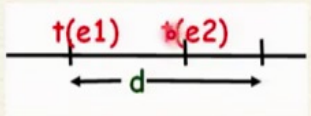
  - Once a duration event starts:
    - It must end before a certain maximum time.

Example of different Time constraints-SS Deadline

SS Deadline Example

- Deadline is defined between two stimuli.
  - A behavioral constraint.
  - Imposed on stimulus.

- Once a user completes dialling a digit,
  - He must dial the next digit within the next 5 seconds.



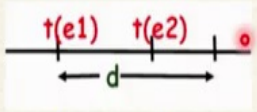
  - Otherwise an idle tone is produced.

Example of different Time constraints-RS Deadline

RS Deadline Example

- Deadline is defined on the stimulus from the respective response event.
  - A behaviorial constraint.
  - Imposed on stimulus.

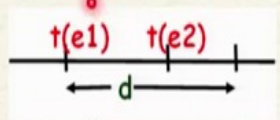
- Once the dial tone appears:
  - The first digit must be dialled within 30 seconds
  - Otherwise the system enters an idle state and an idle tone is produced



Example of different Time constraints-RR Deadline

RR Deadline Example

- Deadline is defined on the response time from another response.
- A performance constraint.
  - Imposed on response.
- **Example:**
  - Once ring tone is given to the callee,
  - Ring back tone must be given to the caller within two seconds,
  - Otherwise the call is terminated.

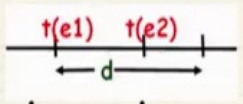


Example of different Time constraints-SR Deadline

SR Deadline Example

- Deadline is defined on the response from the respective stimulus.
  - A performance constraint.
  - Imposed on response.

- **Example:**
  - Once the receiver of the hand set is lifted:
  - The dial tone must be produced by the system within 2 seconds,



  - Otherwise a beeping sound is produced until the handset is replaced.