Date
26/11/23

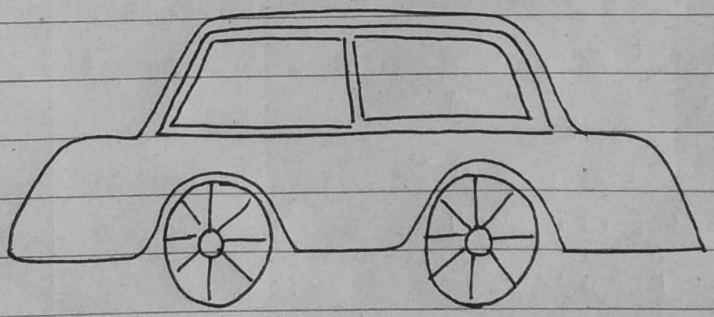
Page No.:
Date: / /

Question Bank 2 Unit 2.

(1) Automotive Domain Specific example of Embedded System.

A) = (A) High Speed Electronic Control Units (HECUs)
- These are used in critical control units requiring fast response.
- This system contains fuel injection systems, antilock brake system, engine control, steering control, transmission control unit & central control unit.

(B) Low Speed Electronic control Units (LECUs)
- These are used in applications where response time is not so critical.
- They built low cost microprocessor/microcontroller and digital signal processor.
- E.g. Audio controller, passengers and driver door locks, power windows, wipe control, mirror control head-tail lamp control.



(C) Embedded Systems in Automotive domain:

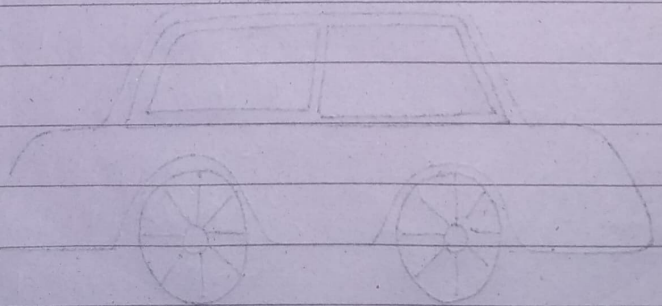
(1) Comfort & convenience.

- Voice Recognition
- DVD Players
- Internet Access
- Telematics
- Rear-Seat Entert.
- Multimedia
- Head-up
- Park/reverse
- Games consoles
- Display

Page No:
Date: / /

② Driver Assistance System.

- GPS
- Audio System
- Digital Radio
- Mobile phone
- Night vision
- Lane Warning
- Adaptive Cruise control & collision.
- Tire pressure Monitor.
- Reconfigurable instrument.



(2) Washing Machine - Application Specific domain example of Embedded System.

A) = (1) Water Inlet Control Valve.
- The water inlet activates water entry point.
- It opens & closes automatically on amount required.

(2) Water Pump.
- Circulate water in Machine
- Recirculating & then draining the water during spin are two directions of water pump.

(3) Tub.
- Inner & Outer Tub
- Clothes are washed, rinsed & dried with help of inner tub.
- External tub covers inner tub.

(4) Timer.
- Wash time for clothes manually.
- Depends on Number of clothes.

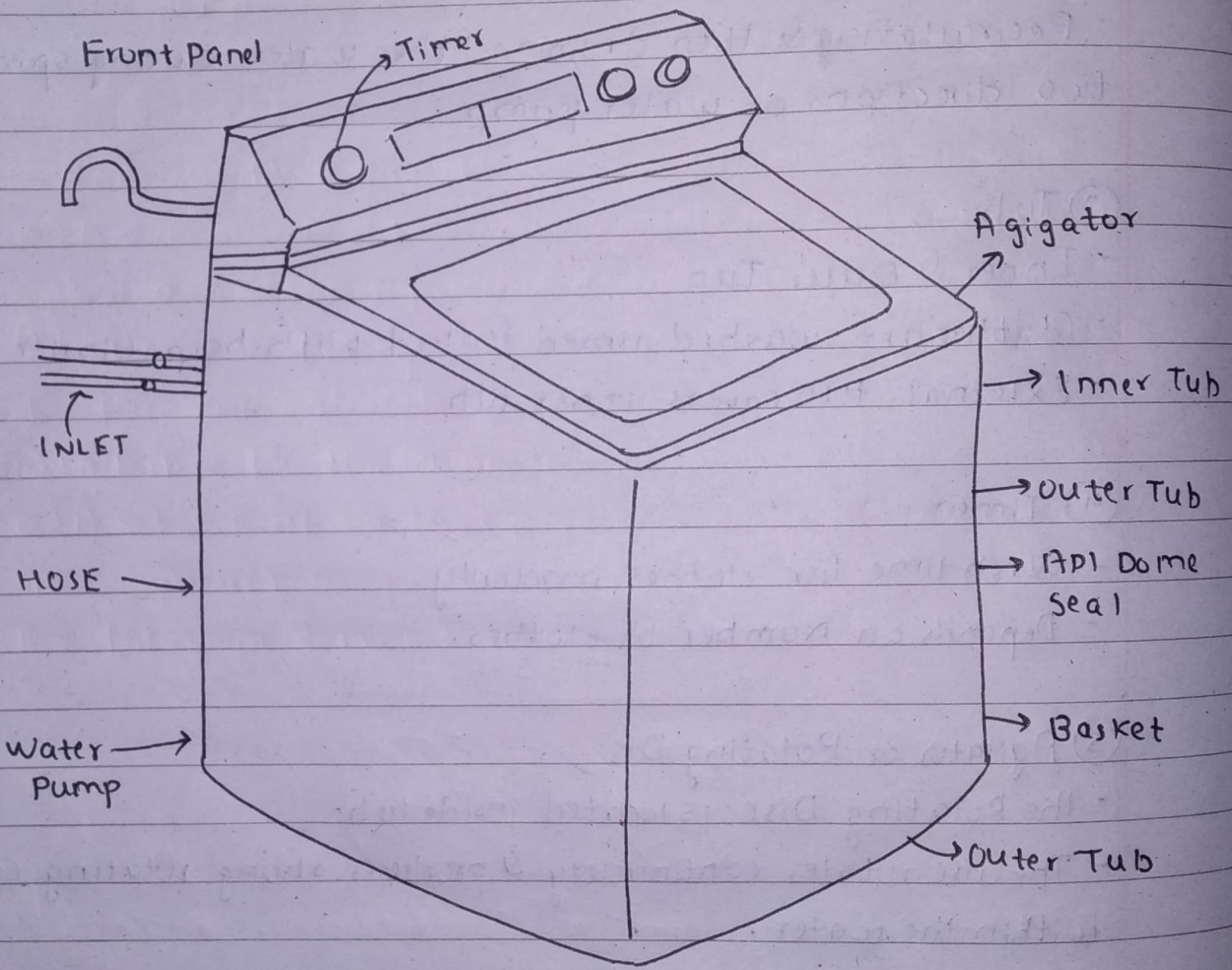
(5) Agitator or Rotating Disc.
- The Rotating Disc is located inside tub.
- The disc rotates continuously & produces strong rotating currents within the water.
- Detergent remove dirt / dust.

(6) Motor.
- Motor is attached to disc.
- Speed changes as per condition.

Page No:
Date: / /

⑦ PCB(Printed Circuit Board)
- Condition & amount of clothes are loaded in PCBs.
- AI technology containing fuzzy logic systems.
- PCB will calculate the total weight of clothes, and find out quantity of water.

⑧ Drain Pipe.
- Removes dirty water.



Page No.:
Date / /

(3) Explain different communication buses in Domain Specific Automotive Systems.

A):- - Automotive application use of serial buses for communication.
- The different types of serial interface buses which is used in Automotive Embedded Application are as follows:

(A) Controller Area Network (CAN):
- Medium speed and High Speed data transfer is supports to Controller Area Network (CAN).
- For error handling in data communication used CAN which is an event-driven protocol interface.
- It is generally employed in safety system like airbag control, power train system.

(B) Local Interconnect Network (LIN):
- It is single master multiple slave communication interface.
- It is a low speed, single wire communication interface with supports for data rates up to 20 kbps and is used for sensor/actuator interfacing.
- Application of LIN are mirror control, fan control, seat positioning controls, window controls.

(C) Media-Oriented System Transport (MOST):
- It is targeted for automotive audio/video interfacing & used primarily in European cars.
- A MOST bus is a multimedia fiber-optic point to point network used in a star, ring topology.

Page No.:
Date: / /

(4) Explain working of Control & Status Register.

A): - The interface between embedded processor and peripheral device are control & status register.
- These registers are a part of peripheral hardware.
- The registers vary from device to device.
- Peripheral devices are situated in processor's memory space or within I/O space.
- Some peripherals of each type contain embedded system which is called as Memory Mapped & I/O Mapped.

Control & Status Registers in Device Driver:

(A) A data structure that overlay the memory-mapped control and status registers of device.
(B) It involves listing out the required variables needed to keep track of state of hardware.
(C) Initialize the Hardware.
(D) A set of Routines that provide an API for Users of Device Driver.
(E) Interrupt service Routines.

Page No.:
Date. / /

(5) Explain working of Device Driver.

A)= • The concept of device driver is to hide the hardware completely.
• It is very difficult to attempts to hide the hardware.
• For sharing the concept of sectors flash memory devices are used.
• In this sector erase operation which can be performed.
• Erased bites or words can be rewritten.
• In embedded systems devices drivers are quite different from the workstation counter parts.
• In modern computers work station device drivers are most often concerned with satisfying the requirement of operating system.

- There are three benefits of good device driver:

I. Modularization which makes structure of overall software and which is easeier to understand.

II. Only one module is exists to interact directly with the peripheral's registers making communication easier.

III. software changes that result from hardware changes are localized to device driver.

Page No.:
Date: / /

(6) Components of a Device Driver.

A)= (A) A data structure that overlay the memory-mapped control and status registers of device.
- This step involves creating a C style structure that is actually a map of registers present in the device.
- These registers can be found out by referring to datasheet for device.
- A Table is created which maps the control register to irrelative offsets.

(B) A Set of Variables to Track Current state of Hardware and Device Driver.
- It invalves listing out required variables needed to keep track of state of hardware & Device Driver.

(C) Initialize Hardware.
Once variables to be used are known the next step in device driver programming is to initialize hardware. Next fuction can be written to control.

(D) A Set of Routines that Provide API for users of Device Driver.

This involves writing diff. func. that will implement various tasks listed to be performed.

(E) Interrupt service Routines.
Once functions are coded the thing remaining to be done is to identify interrupts.

Page No.:
Date: / /

(7) Explain watchDog Timer.

A)= (1) It is hardware kit.

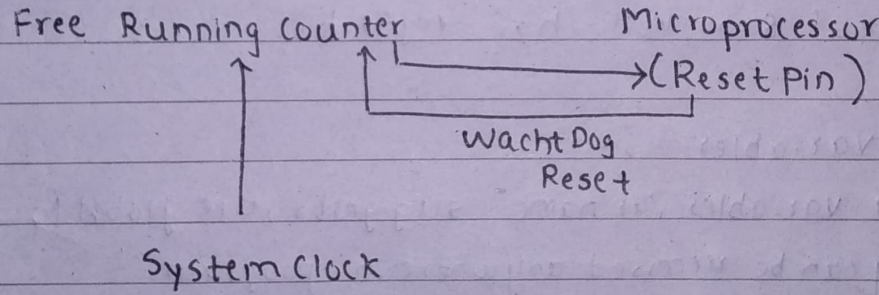
(2) This kit is used to protect the system from software hangs.

(3) watchDog Timer always counts down from some large number to zero.

(4) This process takes a few seconds to reset, in mean time, it is possible for embedded software to "kick" watch dog timer, to reset it's counter to original large number.

(5) The computer regularly resets watchdog timer to prevent it from timing and if the timer expire i.e. counter reaches zero, the watchDog Timer will assume system has entered software hang.

(6) Common way to recover from unexpected software hang.



• WatchDog Timer

Page No.:
Date: / /

(8) Explain structure of Embedded Program.

A)= 1. Comments:

- Comments are readable text are written to help us (reader) understand the code easily.
- There are two ways
  //:- Single line comment.
  /*....*/ :- Multi-line comment.

2. Preprocessor Directive.

- A preprocessor Directive in Embedded C is an indication to the compiler that it must look in to this file for symbols that are not defined in program.
- In C programming Language, preprocessor Directives are usually represented using #include or #define.
- In Embedded C, we usually use the preprocessor directive to indicative a header file specific to micro-controller.
- In case of 8051, Keil has reg51.h.

3. Global Variables:

- Global variables, as name suggests, are global to program i.e. they can be accessed anywhere in program.

4. Local Variables:

- Local variables in contrast to Global variables are confirmed to respective function.

5. Main Func:

Every C has one function where execution begins.

Page No.:
Date: / /

(9) Compiling, Linking, Debugging.

A) :- ① Compiling

- Compiler means to translate programs written in some human-readable language into an equivalent set of op-codes for a particular processor.
- Everything in this section applies to compilers & assemblers.
- Processor has it's own unique machine language.
- Compiler almost always runs on host computer.
- It simply doesn't make sense to execute compiler on Embedded system itself.
- A compiler such as that runs on one computer platform and produces code for another is cross-compiler.

② Linking.

- The entire object files resulting from compilation in step one must be combined.
- The output of linker is a new object file which contains all of code and data.
- When linker finishes executing, machine language code from input object files will be in text section.
- Linker is also on lookout for unresolved symbols.
- The unresolved reference to be replaced with actual variable.

③ Debugging.

- Run Control of Target - Start and stop program execution.
- Ability to change code & data on target - Fix errors, test alternatives.
- Real Time Monitoring of target execution - Non-intrusive.
- Timing & Functional Accuracy.

- Debugging is process of finding and resolving of defects or problem within program that prevent correct operation of computer system.

- Debugging ranges in complexity from fixing simple errors to performing lengthy and tiresome tasks of data collection, analysis & scheduling updates.