Sub: Embedded System (KOE-062)

Notes: Unit-5

Topic

1.1. Embedded System Application Development :
1.2. Design issues and techniques Case Study of Washing Machine
1.3. Automotive Application
1.4. Smart Card System Application

1.1. Embedded System Application Development :

Embedded systems have been a game-changer in various industries like automotive, aerospace, medical, and defence.

These systems are special-purpose computers that perform specific tasks and operate within a larger system or device.

They are designed to work with limited resources, power, and memory, making them an ideal choice for applications that require real-time processing, reliability, and efficiency.

Application development for embedded systems is a complex and critical process that requires specialized knowledge and skills.
Embedded systems are computer systems that are integrated into other devices or products, such as cars, appliances, medical devices, and industrial equipment.

To develop applications for embedded systems, developers need to consider various factors, such as the system architecture, hardware and software interfaces, programming languages, development tools, testing and debugging techniques, and deployment strategies.

Embedded Application Development Process

1. Requirement Analysis: Clearly define the application's requirements, including hardware specifications, performance targets, and functionality expectations.

2. System Design: Design the software architecture, considering components like drivers, communication protocols, and user interfaces.

3. Software Development: Write and optimize code according to the chosen programming language (often C/C++), keeping in mind resource constraints and real-time requirements.

4. Testing and Validation: Rigorously test the application under various conditions to identify and address bugs, performance issues, and compatibility problems.

5. Integration: Integrate the software with the hardware, ensuring that all components work seamlessly together.

6. Optimization: Fine-tune the code for better performance, memory usage, and power efficiency.

7. Deployment: Deploy the application onto the target embedded system, either through direct programming or over-the-air updates.

Examples of Embedded Application Development

There are some fundamental concepts related to embedded application development.

- Real-Time: some embedded applications must receive continuous inputs, processing them, and generate outputs of data, where such processing must be performed in a very short period of time. As an example, some applications demand object detection in live stream videos, where processing involves inference and bounding boxes with labels displayed on each frame. This process must be performed as fast as possible.

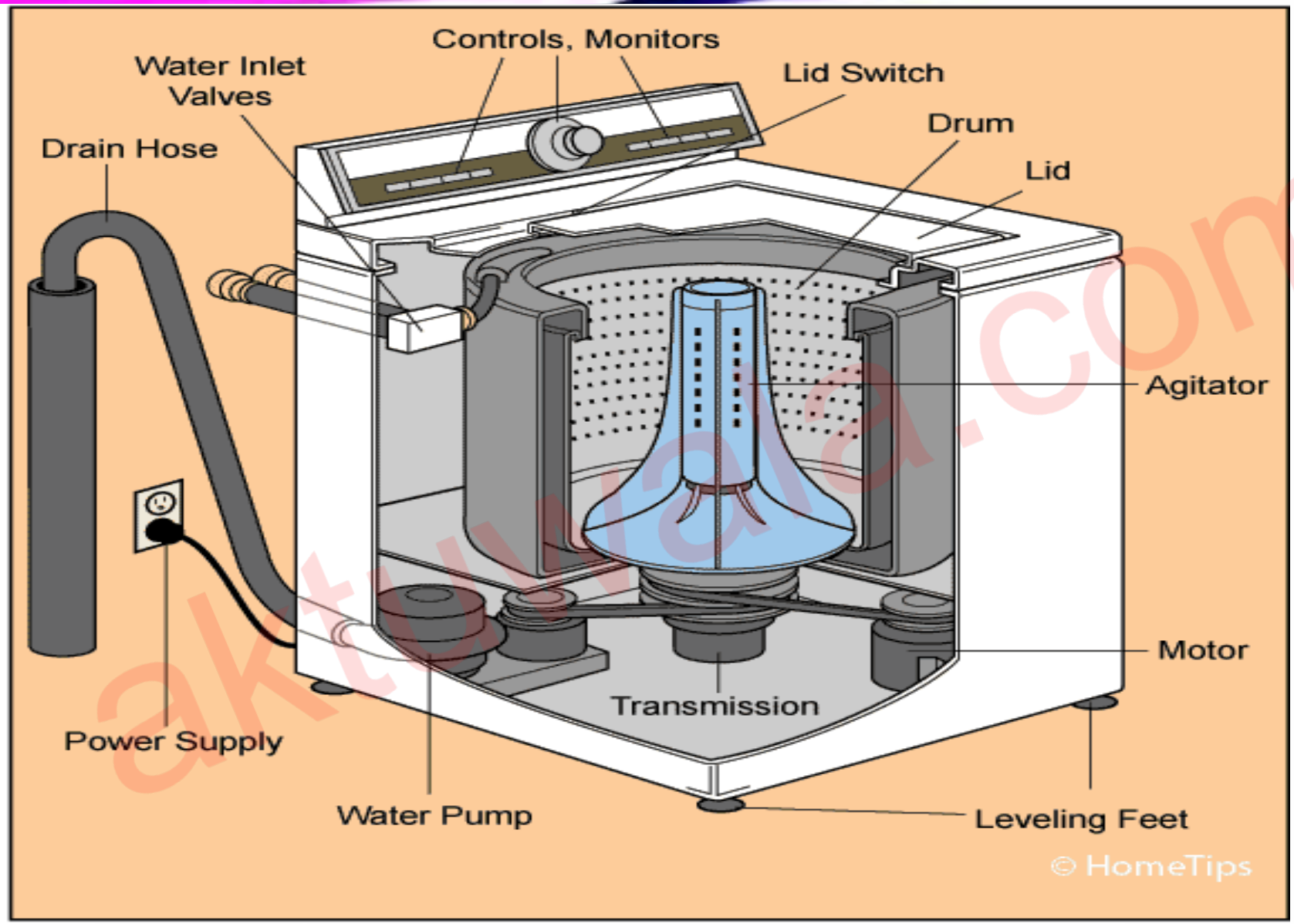
- Fault-Tolerance: this is the capability of an embedded system to survive in the presence of faults. Faults can be caused by power cuts, hardware damaged parts, overwarming, and more The embedded software must be capable of detecting faults and make decisions according to how critical a fault is. As an example, an embedded system working inside an airplane must be capable of identifying any possible fault and make decisions in order to keep the aircrew safe. Decisions can be as simple as sending an alert or as complex as performing changes in the command control.
- Portability: this is the measure of how easy it is to use the same embedded software in multiple environments. It requires generalized abstractions between the application program logic and the low-level system interfaces. As an example, embedded devices used in domotic applications must be adjustable no matter the place where they have to be installed.
- Reliability: this is the survival probability of the system when the function is critical during the run time. As an example, embedded systems used in self-driven cars must be able to make decisions in runtime, where many tasks are critical to keep passengers and pedestrians safe. In this case, reliability must be as high as possible.

WASHING MACHINE

- Washing Machine is an typical example of an embedded system providing extensive support in home automation application.

- An Embedded System contains:

a. Sensors
b. Actuators
c. Control Unit and
d. Application specific user interfaces like keyboard display units , etc.



PROCESS

- The actuator part of washing machine consists of a motorized agitator , tumble tub.
- Water drawing pump and inlet valve to control the flow of water into the unit.
- The sensor part consist of the water temperature sensor , level sensor ,etc
- The control part contains a microprocessor/controller based board with interfaces to the sensors and actuators.
- The sensor data is fed back to the control unit.
- Control unit generates the necessary actuator outputs.

- The Control Unit also provides connectivity to user interfaces like keypad for setting the washing machine time , selecting the type of material to be washed like medium , light , heavy duty , etc.
- User Feedback is reflected through the display unit and LED's connected to the control board.
- The integrated control panel consists of a microprocessor/controller based board with I/O interfaces and a control algorithm running in it.
- Input interface includes the keyboard which consists of wash type selector namely Wash, Spin and Rinse, cloth type selector namely Light, medium , heavy duty and washing time setting, etc.

- The output interface consists of LED/LCD displays, status indication LEDs, etc. connected to the I/O bus of the controller.
- The other types of I/O interfaces which are invisible to the end user are different kinds of sensor interfaces, namely, Water temperature sensor, Water level sensor.
- Actuator interface including motor control for agitator and tub movement control, inlet water flow, etc.

ADVANTAGES

- Easy to manage.
- Faster to load.
- More specific to one task.
- Performance is good.

DISADVANTAGES

- High cost.
- Hardware is limited.
- Difficult to upgrade.

What is Smart Card?

- Plastic card with embedded microprocessor chip , electronic memory and a battery.
- It is used for information storage , authentication and management.
- There are several types of smart card :Contactless(do not require to be swipe through magnetic stripe reader whereas some are contact (require to be swipe through magnetic stripe reader).

Construction of Smart cards

- Designing: It is the first step to construct smart card consisting of following specifications such as clock speed , memory size , type of Operating System , additional functionalities.
- Chip Fabrication : It is the second step which includes silicon mounting.
- Loading the code into memory is the third step by using commands.
- Data loading into memory is the fourth step which stores the data.

Basic working of Smart Card

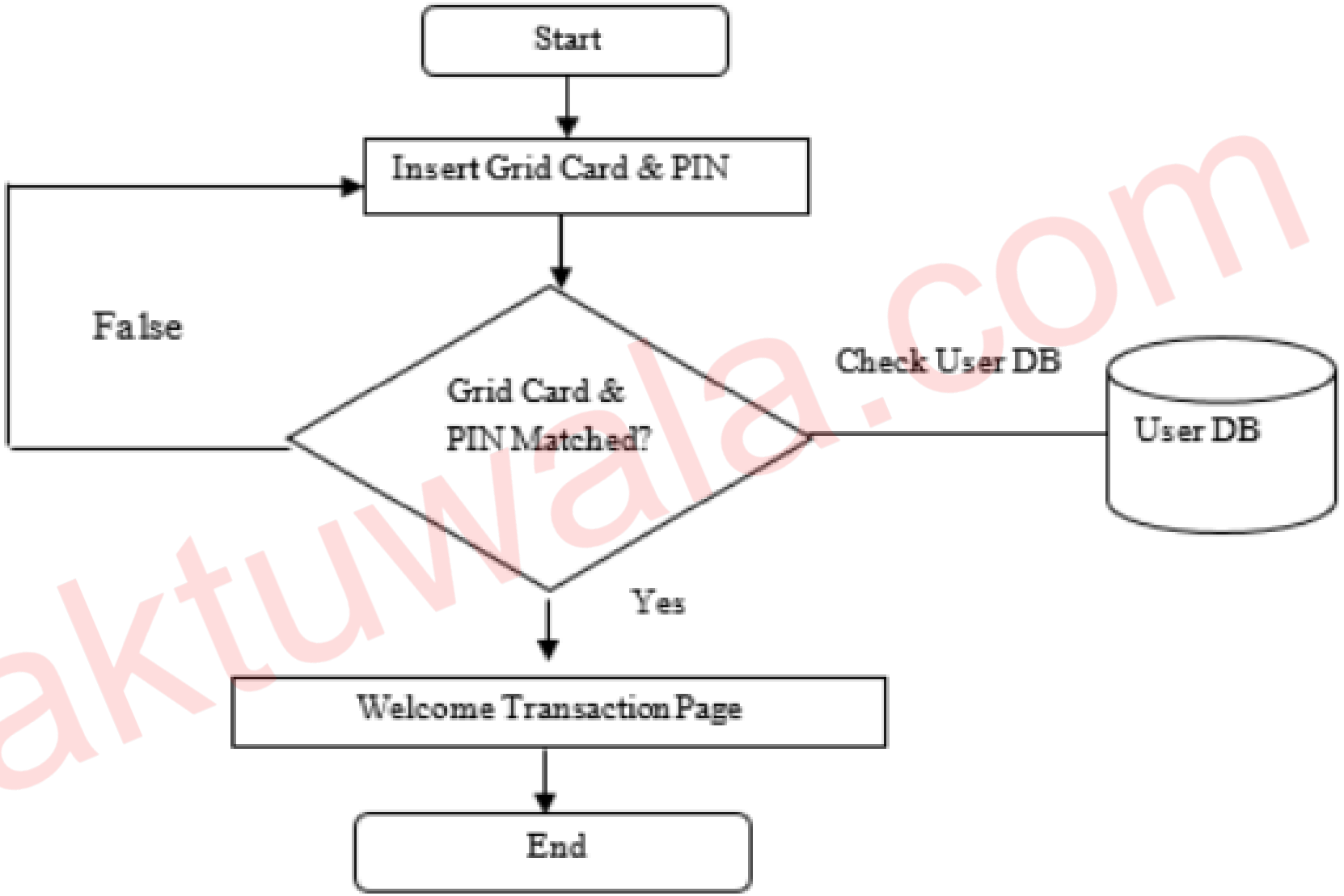
- A smart card is connected to the host computer or controller .
- They are connected through a card reader which gets the information from the smart card.
- And accordingly passes the information to the host computer or controller.

- The system consists of 4 main parts:
- A smart Card which is generally a contact memory smart card which contains the information about the individual.
- A smart card reader which is generally a contact smart card reader and is used to read information from the card.
- A controller which receives data from the smart card reader via the RS232 interface.
- A load which is a relay in this case, used to drive a motor and connected to the controller via the relay driver IC.

- The individual inserts his/her card in the card reader.
- The card reader sends the data to the MAX 232 IC through the DB9 connector.
- The Microcontroller receives the data from the MAX 232 and is accordingly programmed to compare the obtained information with the stored information in the database.
- If the data matches, the Microcontroller develops logic high at its output pin, connected to the input pin of the relay driver.
- The relay driver IC accordingly develops a low logic at its output and energizes the relay.
- The common contact of the relay is now connected to the normally open contact and the motor connected in series with the relay contacts is rotated such that the door is opened.
- In case the data doesn't matches, the microcontroller is programmed to develop logic low at its output pin and the relay accordingly doesn't get energized, keeping the door shut.
- The obtained output is accordingly displayed on the LCD which shows whether the data is matched or not.

Embedded Hardware Components

- Microcontroller
- RAM for temporary variables and stack.
- OTP ROM for application codes and RTOS codes for scheduling the tasks.
- Flash for storing user data , user address , user identification codes , card number , and expiry date.
- Timer and interrupt control
- Interfacing circuit for IOs.
- Charge Pumps for delivering power to the antenna for transmission.



Advantages

- Might be promptly reconfigured
- Reusable
- Secure transactions
- Gives more security
- More tough and dependable
- Permit numerous provisions to be saved in one card

Applications

- E-Ticketing : Bus , Event , Tourist
- Transportations : Driving Card
- Education : Library , Laboratory
- Health Care: Insurance card
- Retail : e-purchase

Made By :- AKTU WALA (Satyam Sahu)

- Website :- Extramarkslibrary.com
  :- Aktuwala.com