

Write the Pseudocode and Flowchart for the problem statements mentioned below:

1. Smart Home Temperature Control

Problem Statement:

Design a temperature control system for a smart home. The system should read the current temperature from a sensor every minute and compare it to a user-defined setpoint.

Requirements:

- If the current temperature is above the setpoint, activate the cooling system.
- If the current temperature is below the setpoint, activate the heating system.
- Display the current temperature and setpoint on an LCD screen.
- Include error handling for sensor failures.

Answer

Initialize setpoint,sensor,lcd

temp=read_sensor()

while(1)

{
if(temp==error_condition)

Print error

Else

if(temp > setpoint)

{
cooling_activate()

heating_deactivate()

}

Else if(temp<setpoint)

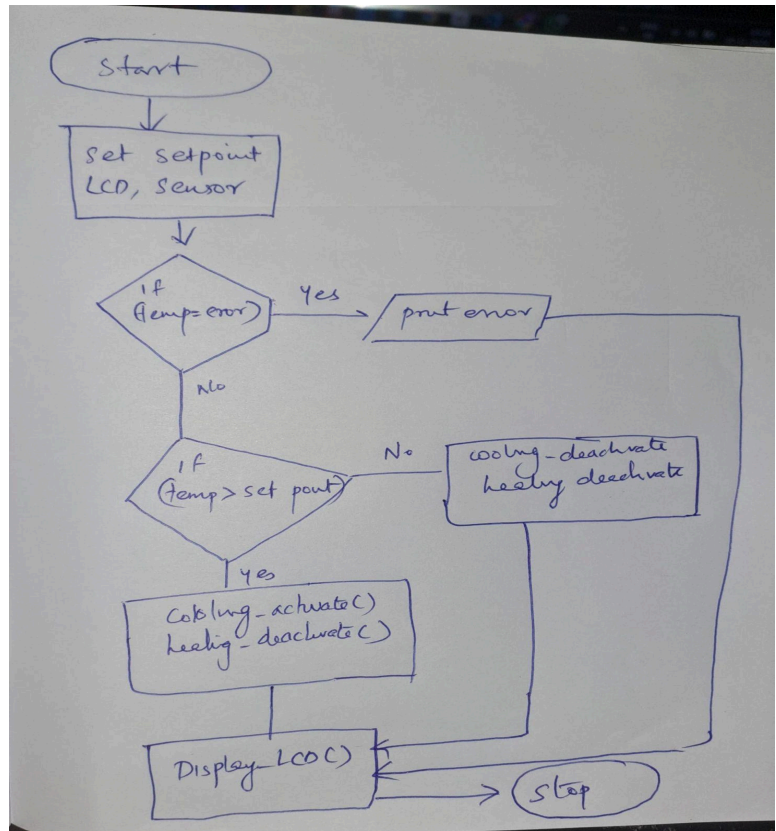
{
heating_deactivate()

cooling_deactivate()

}

}

lcd_display(temp,setpoint)



2. Automated Plant Watering System

Problem Statement:

Create an automated watering system for plant that checks soil moisture levels and waters the plants accordingly.

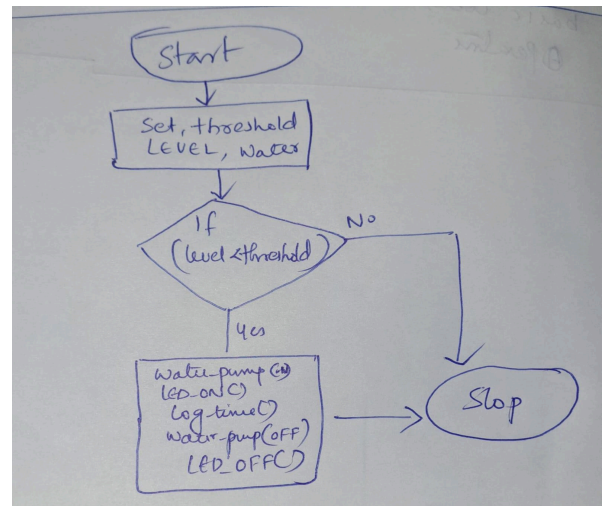
Requirements:

- Read soil moisture level from a sensor every hour.
- If moisture level is below a defined threshold, activate the water pump for a specified duration.
- Log the watering events with timestamps to an SD card.
- Provide feedback through an LED indicator (e.g., LED ON when watering).

Answer

Initialise
threshold
Level
water_duration

```
if(level < threshold)
{
    water_pump(ON)
    LED_ON()
    log_time()
    water_pump(OFF)
    LED_OFF()
}
```



3. Motion Detection Alarm System

Problem Statement:

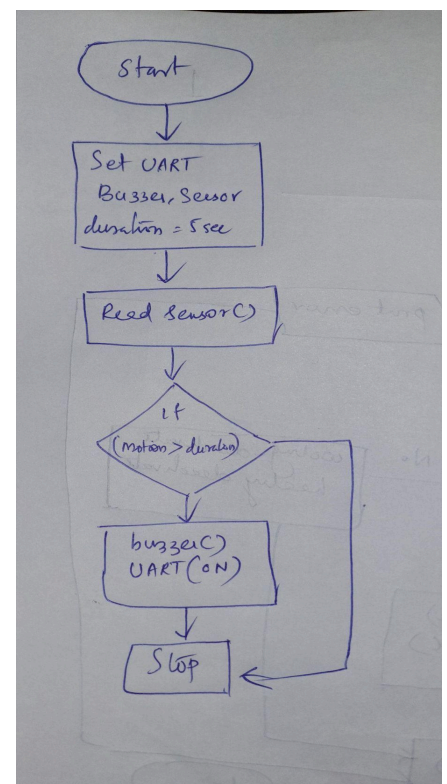
Develop a security alarm system that detects motion using a PIR sensor.

Requirements:

- Continuously monitor motion detection status.
- If motion is detected for more than 5 seconds, trigger an alarm (buzzer).
- Send a notification to a mobile device via UART communication.
- Include a reset mechanism to deactivate the alarm.

Answer

```
Set UART, Buzzer, sensor
Set duration = 5sec
while(1)
{
```



```

read_sensor()
if(motion_detected > duration)
{
  buzzer()
  UART("Motion detected")
}
}

```

4. Heart Rate Monitor

Problem Statement:

Implement a heart rate monitoring application that reads data from a heart rate sensor.

Requirements:

- Sample heart rate data every second and calculate the average heart rate over one minute.
- If the heart rate exceeds 100 beats per minute, trigger an alert (buzzer).
- Display current heart rate and average heart rate on an LCD screen.
- Log heart rate data to an SD card for later analysis.

Answer

Initialise heart_rate, buzzer, lcd, Sd_card

read_heart()

find_heart_rate()

Average()

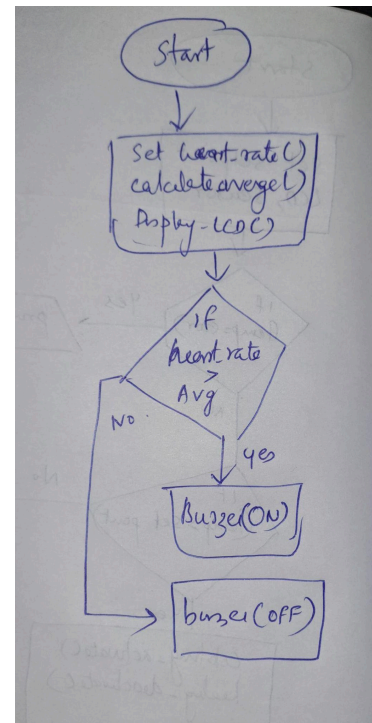
Display_LCD()

if(heart_rate > average)

buzzer(ON)

Else

buzzer(OFF)



5. LED Control Based on Light Sensor

Problem Statement:

Create an embedded application that controls an LED based on ambient light levels detected by a light sensor.

Requirements:

- Read light intensity from the sensor every minute.
- If light intensity is below a certain threshold, turn ON the LED; otherwise, turn it OFF.
- Include a manual override switch that allows users to control the LED regardless of sensor input.
- Provide status feedback through another LED (e.g., blinking when in manual mode).

Answer

Set

Threshold

LED

Sensor

Switch

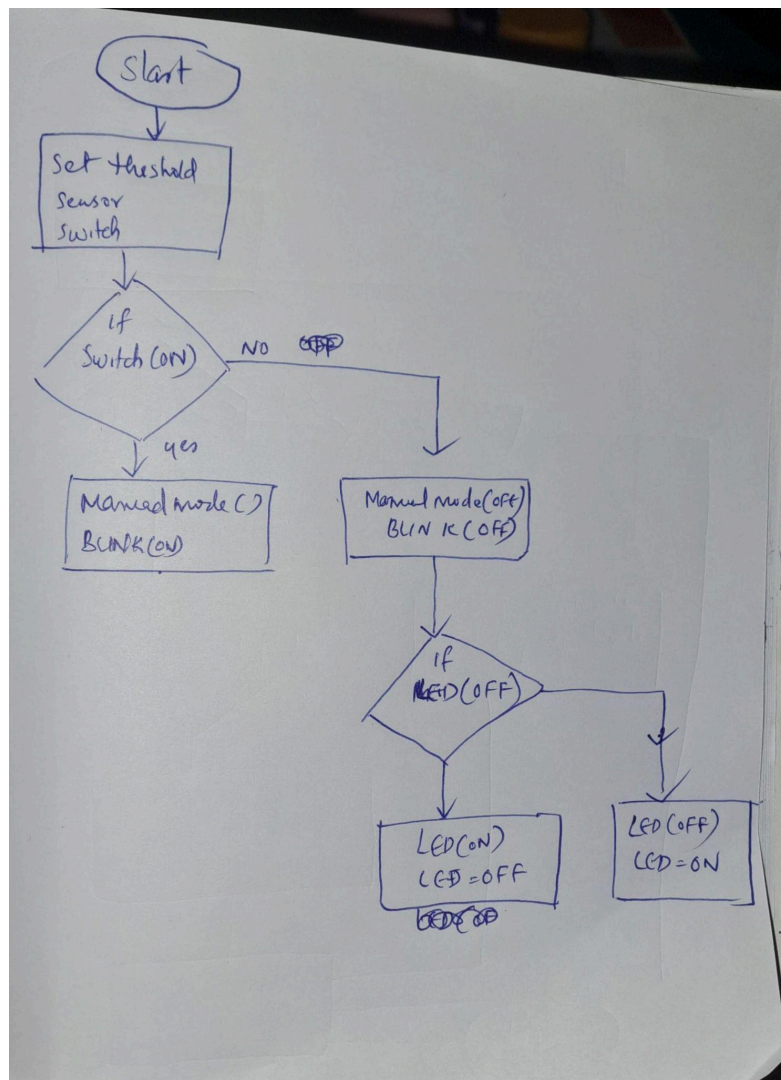
Intensity

manual_mode()

```
if(switch(ON))  
{  
  manual_mode(ON)  
  Blink(ON)  
}
```

```
Else if(switch(OFF))  
{  
  manual_mode(OFF)  
  if(LED(OFF))  
  {  
    LED_ON()  
    LED=OFF;  
  }  
  Else if(LED(ON))  
  {  
    LED_OFF()  
    LED=ON;  
  }  
}
```

```
if(intensity<threshold)  
  LED_ON()  
Else  
  LED_OFF()  
}
```



6. Digital Stopwatch

Problem Statement:

Design a digital stopwatch application that can start, stop, and reset using button inputs.

Requirements:

- Use buttons for Start, Stop, and Reset functionalities.
- Display elapsed time on an LCD screen in hours, minutes, and seconds format.
- Include functionality to pause and resume timing without resetting.
- Log start and stop times to an SD card when stopped.

Set button state

Timer lcd

set hours,minutes,seconds

start ,stop,reset

Sd-card

if(button(start)&& not running)

if(pause)

running=1

pause=0;

Else

hour=0

minute=0

second=0

running=1

LOG(time)

if(button(stop)&& running)

if(pause)

running=0

pause=1

LOG(time)

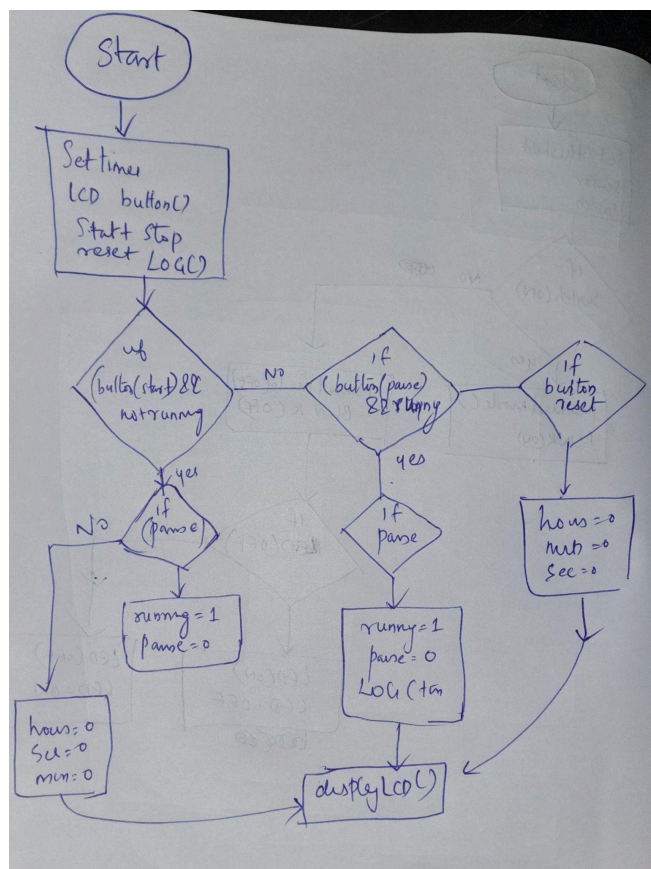
if(button(reset))

hours=0

minutes=0

seconds=0

displaylcd()



7. Temperature Logging System

Problem Statement:

Implement a temperature logging system that records temperature data at regular intervals.

Requirements:

- Read temperature from a sensor every 10 minutes.
- Store each reading along with its timestamp in an array or log file.
- Provide functionality to retrieve and display historical data upon request.
- Include error handling for sensor read failures.

Set temp, LOG interval

temp=read_temp()

if(temp==error)

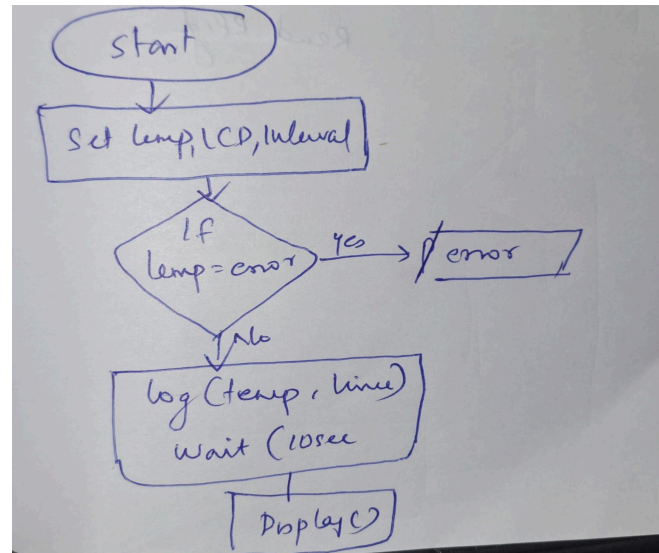
Print error

Else

log(temp, time)

wait(10sec)

Display entry()

**8. Bluetooth Controlled Robot****Problem Statement:**

Create an embedded application for controlling a robot via Bluetooth commands.

Requirements:

- Establish Bluetooth communication with a mobile device.
- Implement commands for moving forward, backward, left, and right.
- Include speed control functionality based on received commands.
- Provide feedback through LEDs indicating the current state (e.g., moving or stopped).

Answer

Command

Speed

comand=read_command()

switch(command)

{

Case forward:

{

Set direction(forward)

set_speed()

status=ON

}

Case backward:

```
{  
Set direction(backward)  
set_speed()  
status=ON  
}
```

Case left:

```
{  
Set direction(left)  
set_speed()  
status=ON  
}
```

Case forward:

```
{  
Set direction(right)  
set_speed()  
status=ON  
}
```

Case stop:

```
{  
Set direction(stop)  
set_speed()  
status=ON  
}
```

Default:

Print invalid

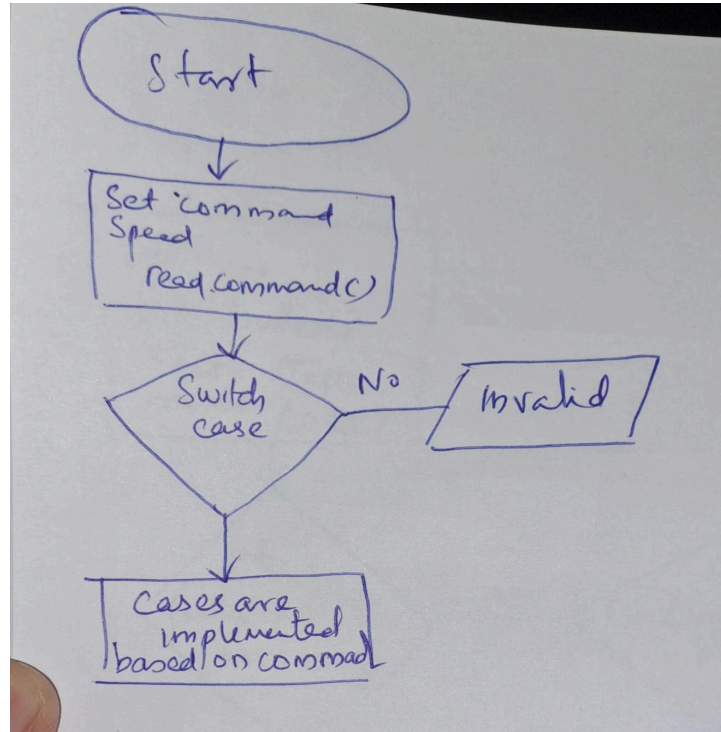
```
}
```

If (status==ON)

led_on()

Else

led_off()



9. Battery Monitoring System

Problem Statement:

Develop a battery monitoring system that checks battery voltage levels periodically and alerts if voltage drops below a safe threshold.

Requirements:

- Measure battery voltage every minute using an ADC (Analog-to-Digital Converter).
- If voltage falls below 11V, trigger an alert (buzzer) and log the event to memory.
- Display current voltage on an LCD screen continuously.
- Implement power-saving features to reduce energy consumption during idle periods.

Set threshold
ADC ICD buzzer memory

```
voltage=read_ADC()
display(VOLTAGE)
wait()
if(voltage<thershold && no alert)
{
  alert_buzzer()
  log(,voltage)
}
Else if(voltage>threshold && alert)
{
  buzzer_off()
}

if(alert_on)
{
  power-saving()
}
```

10. RFID-Based Access Control System

Problem Statement:

Design an access control system using RFID technology to grant or deny access based on scanned RFID tags.

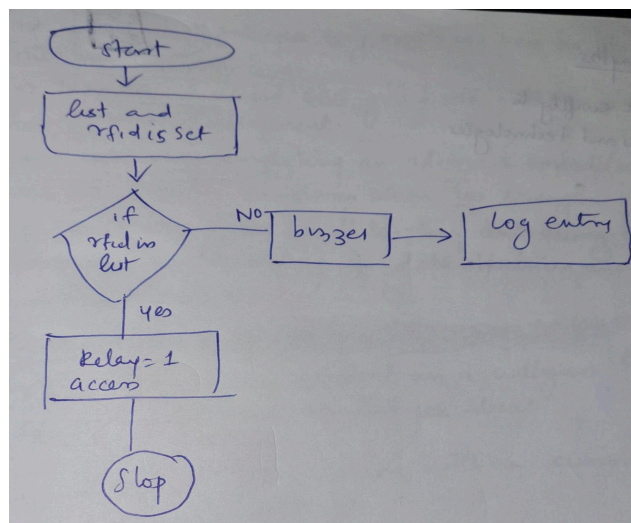
Requirements:

- Continuously monitor for RFID tag scans using an RFID reader.
- Compare scanned tags against an authorized list stored in memory.
- Grant access by activating a relay if the tag is authorized; otherwise, deny access with an alert (buzzer).
- Log access attempts (successful and unsuccessful) with timestamps to an SD card

ANSWER

Set authorized list

```
read_rfid()
if(rfid present in list)
  relay=1
Else
{
  alarm()
  Log_event
```



}