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.

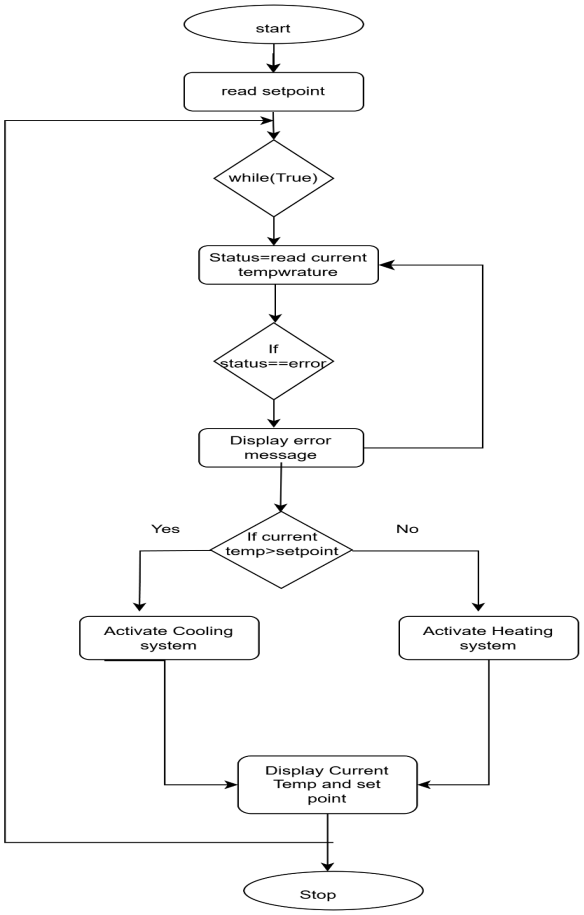
Requirements:

- 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.

Psuedocode

```
Enter setpoint
while(True):
status=current temperature
if(status==error)
    Display error message in lcd
    Continue next iteration
If current temp>setpoint THEN
     Activate cooling system
ELSE
    Activate heating system

Display current temp and setpoint
```



2. Automated Plant Watering System

Problem Statement:
Create an automated watering system for plants 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).

Psuedocode

Set threshold Value(T)

Set timestamp as empty

while(true):

Wait for one hr

M=Read soil moisture level

if(M<T} THEN

    Activate Water pump

    LED ON

    Update timestamp value

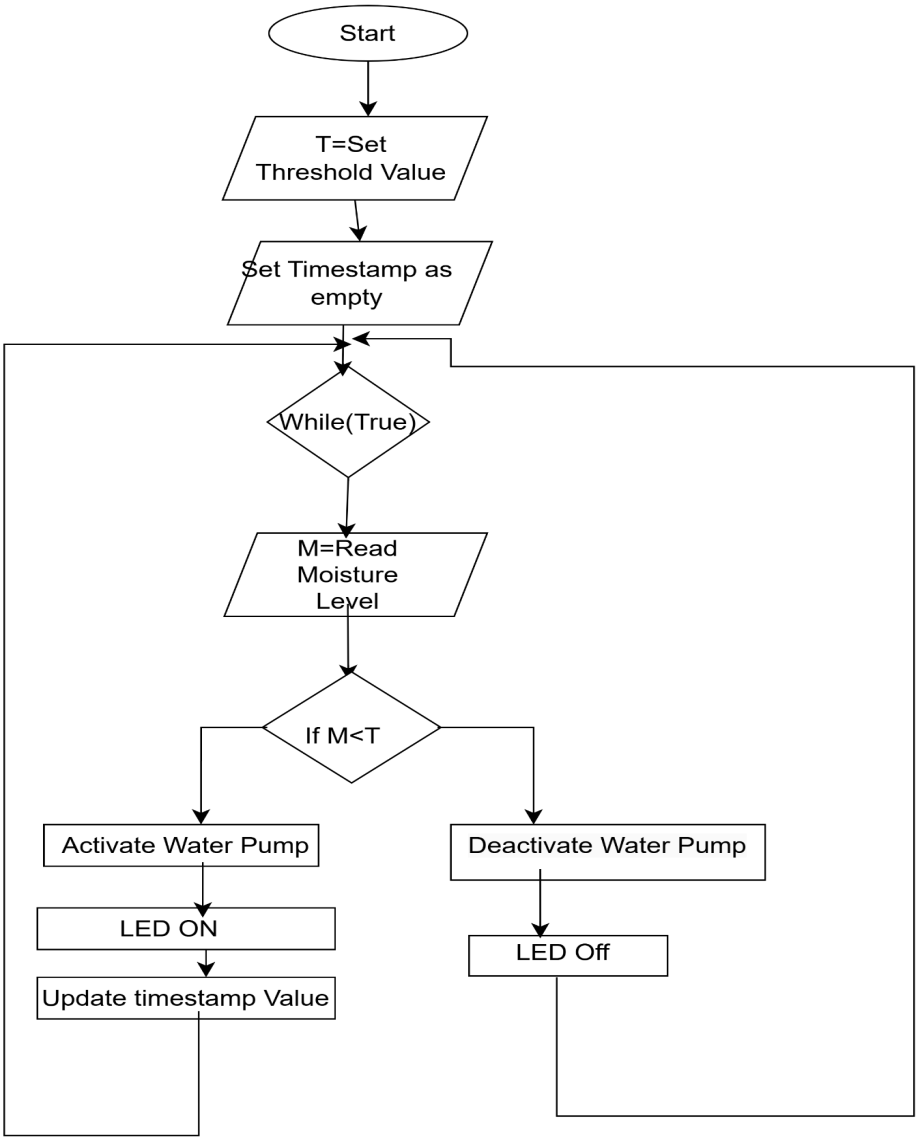
ELSE

   Deactivate Water pump

    LED OFF

End IF

End While


Start
T=Set Threshold Value
Set Timestamp as empty
While(True)
M=Read Moisture Level
If M<T
Activate Water Pump
LED ON
Update timestamp Value
Deactivate Water Pump
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). •

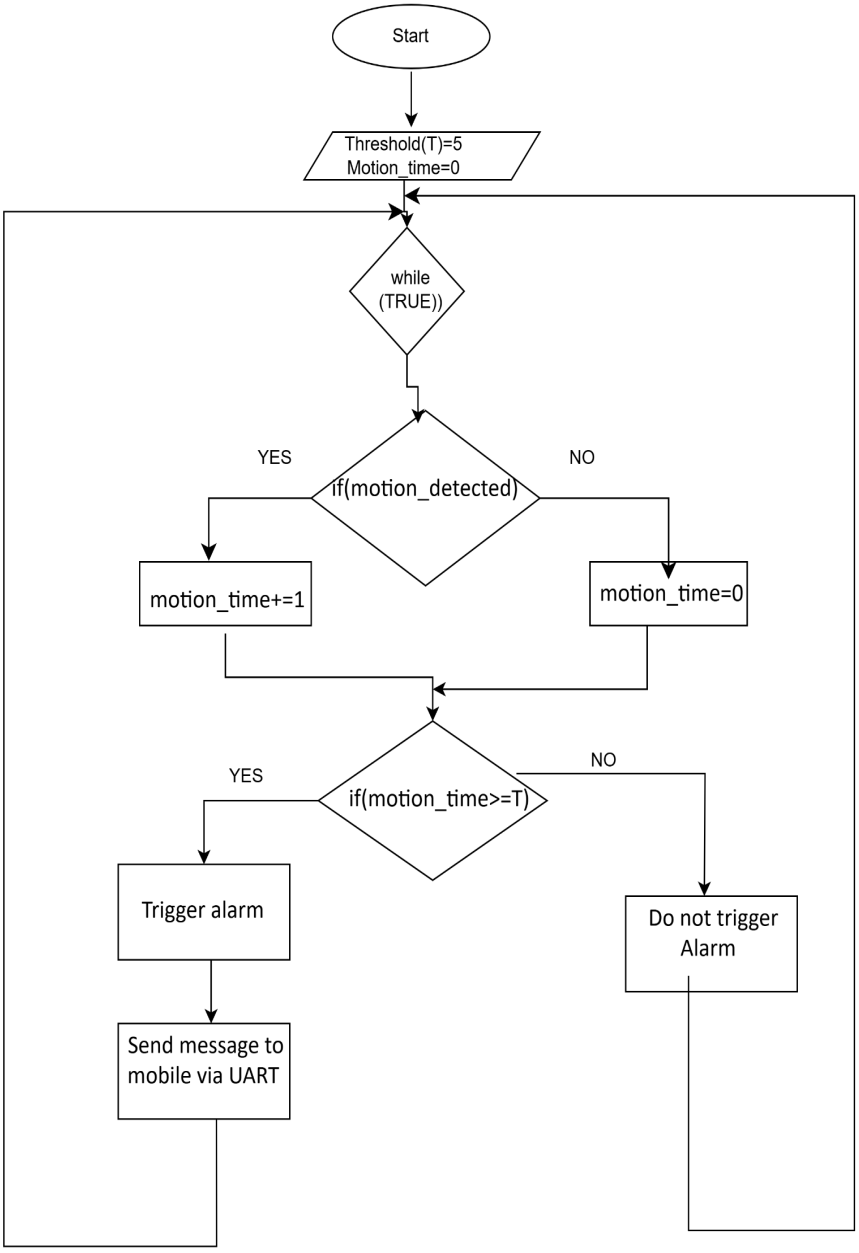
Send a notification to a mobile device via UART communication.

• Include a reset mechanism to deactivate the alarm.

Psuedocode

```
Set threshold(T)=5
Set motion_time=0
while(True):
if(motion_detected) THEN
   motion_time+=1
Else
    motion_time=0

if(motion_time>=T)
    Trigger alarm
    Send message to mobile via UART
Else
    Do not trigger Alarm
ENDIF
END While
```



4. Heart Rate Monitor

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

Requirements:

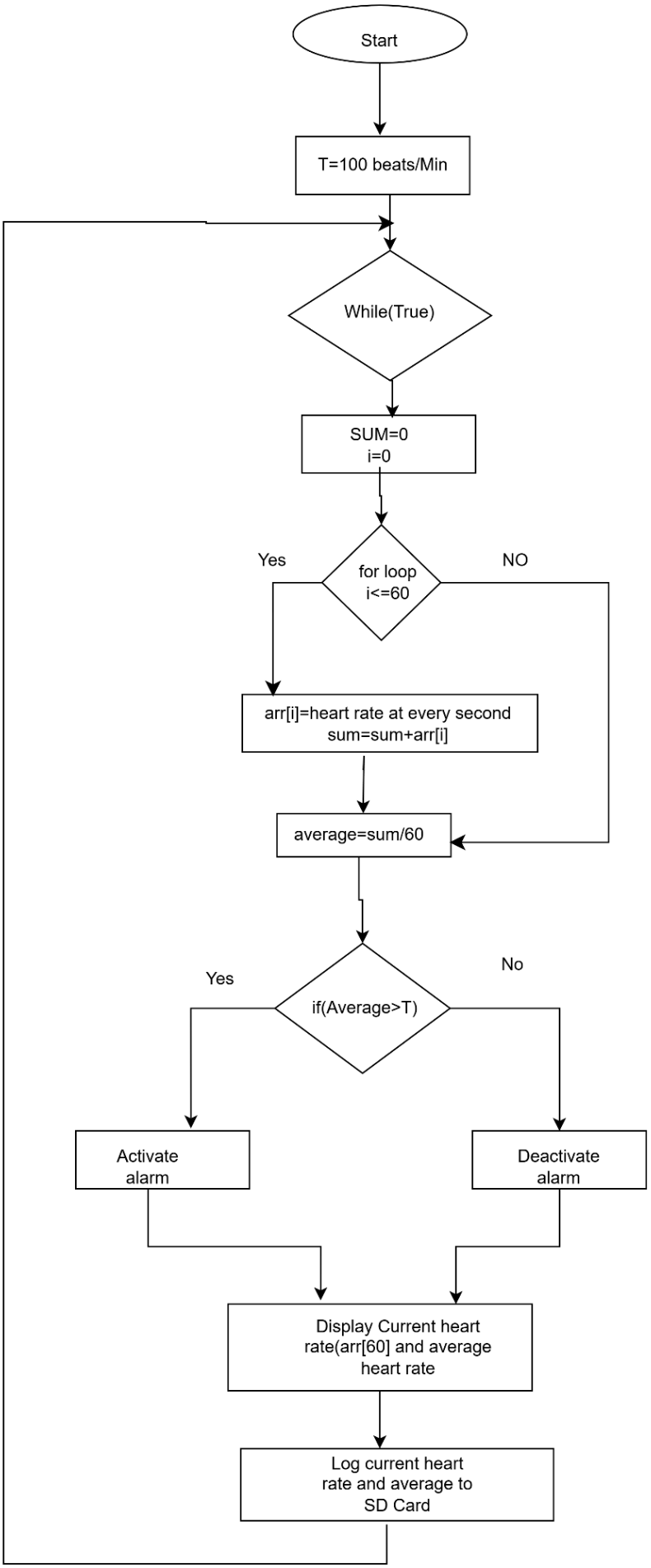
- Sample heart rate data every second and calculate the average heart rate over one minute. •

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.

PSUEDOCODE

```
Set threshold (T) = 100 beats/min
while (TRUE):
        Set sum = 0
        For i = 1 to 60:
                arr[i] = Read heart rate for the current second and store in array
                sum = sum + arr[i]
        End for
        average = sum / 60
        if (average > T):
                Activate alarm
        Else:
                Deactivate alarm
        Endif
        Display current heart rate (arr[60]) and average heart rate on LCD
        Log current heart rate and average heart rate to SD Card
End while
```


Start
T=100 beats/Min
While(True)
SUM=0
i=0
Yes
for loop
i<=60
NO
arr[i]=heart rate at every second
sum=sum+arr[i]
average=sum/60
Yes
if(Average>T)
No
Activate
alarm
Deactivate
alarm
Display Current heart
rate(arr[60] and average
heart rate
Log current heart
rate and average to
SD Card

5. LED Control Based on Light Sensor

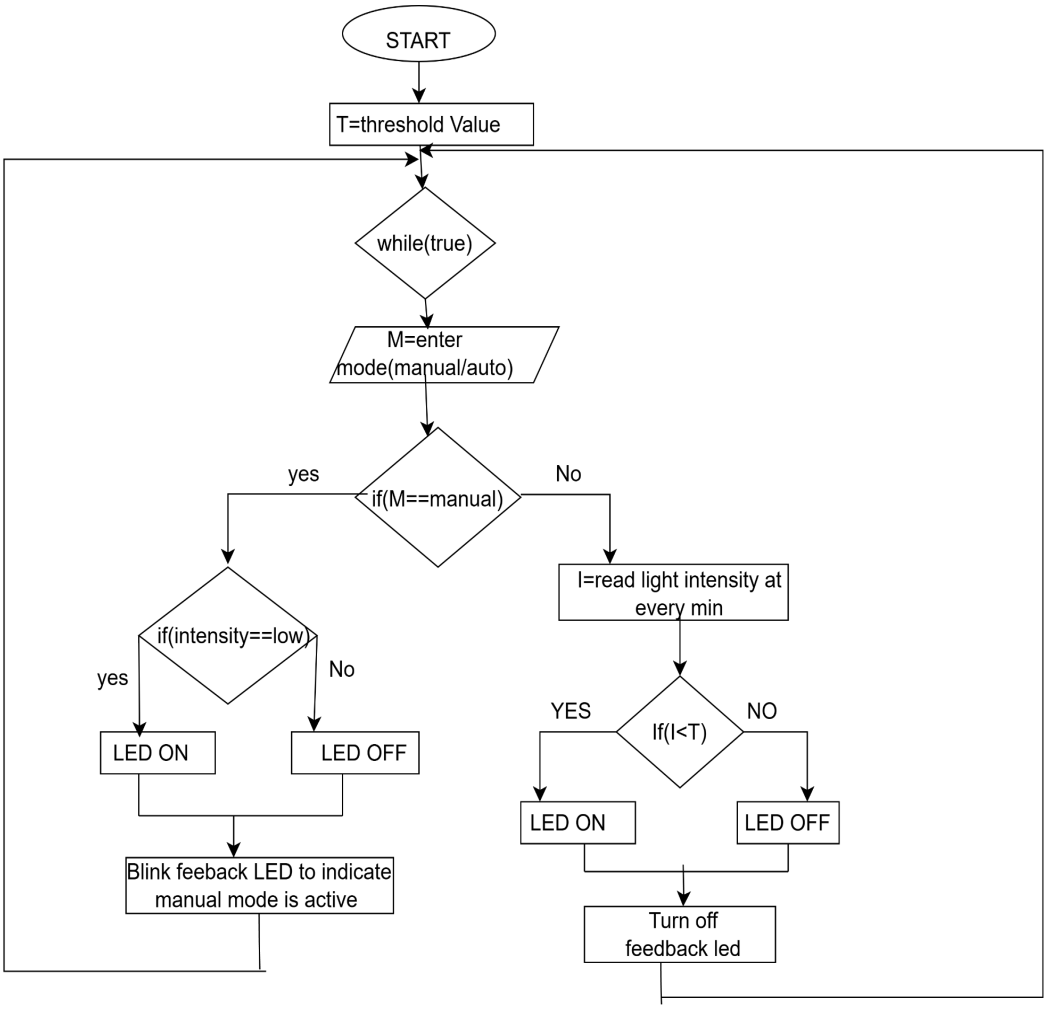
Problem Statement:
Create an embedded application that controls an LED based on ambient light levels detected by a light sensor.

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).

Psuedocode

```
Set threshold value(T)
while(true):
        M=Enter mode(manual/auto)
        IF(M==manual) THEN
                IF(intensity is low) THEN
                    LED ON
                 ELSE
                     LED OFF
                 ENDIF
                 Blink feedback LED to indicate manual mode is active
        ELSE
                I=Read light intensity at every minute by sensor
                 IF(I<T) THEN
                        Turn on LED
                 ELSE
                        Turn OFF LED
                 ENDIF
                 Turn OFF feedback LED
        ENDIF
End While
```



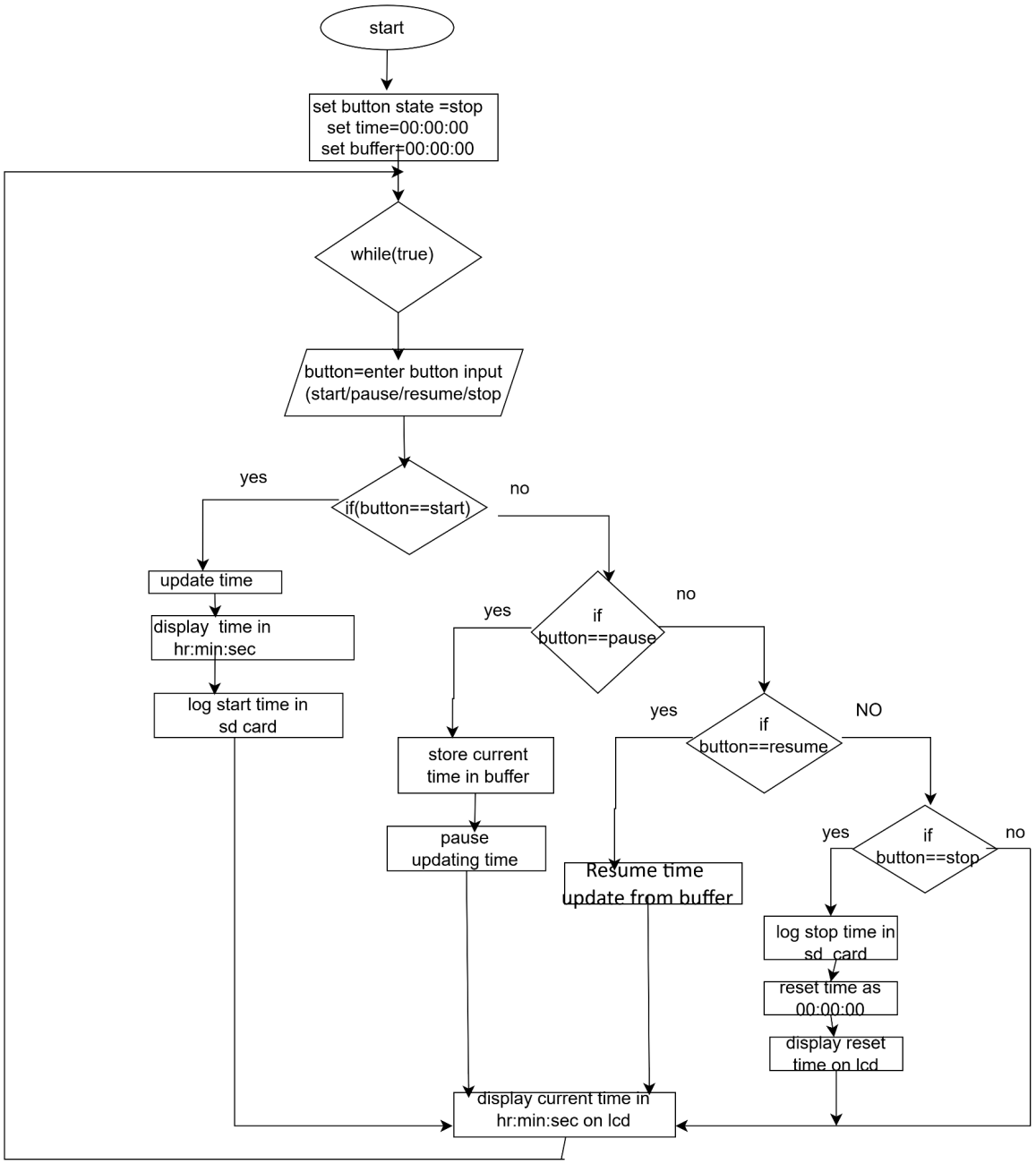
6. Digital Stopwatch

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.

Psuedocode

```
Intialise button state as stop
Intialise time as 00:00:00
Intialise buffer as 00:00:00
while(true):
        Read button input(start/pause/resume/stop)
        if(button==start)
            start Updating time
            Display time in hr-min-sec in lcd screen
            Log start time in sd card
        Else if(button==pause)
            Store current time in buffer
            Pause updating time
        Else if(button==resume)
            Resume time update from buffer value
        Else if(button==stop)
            Log stop time in sd card
            Reset time as 00:00:00
            Display reset time on LCD
        Else
            Display current time in hr:min:sec in lcd screen
        endif
End while
```



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. •

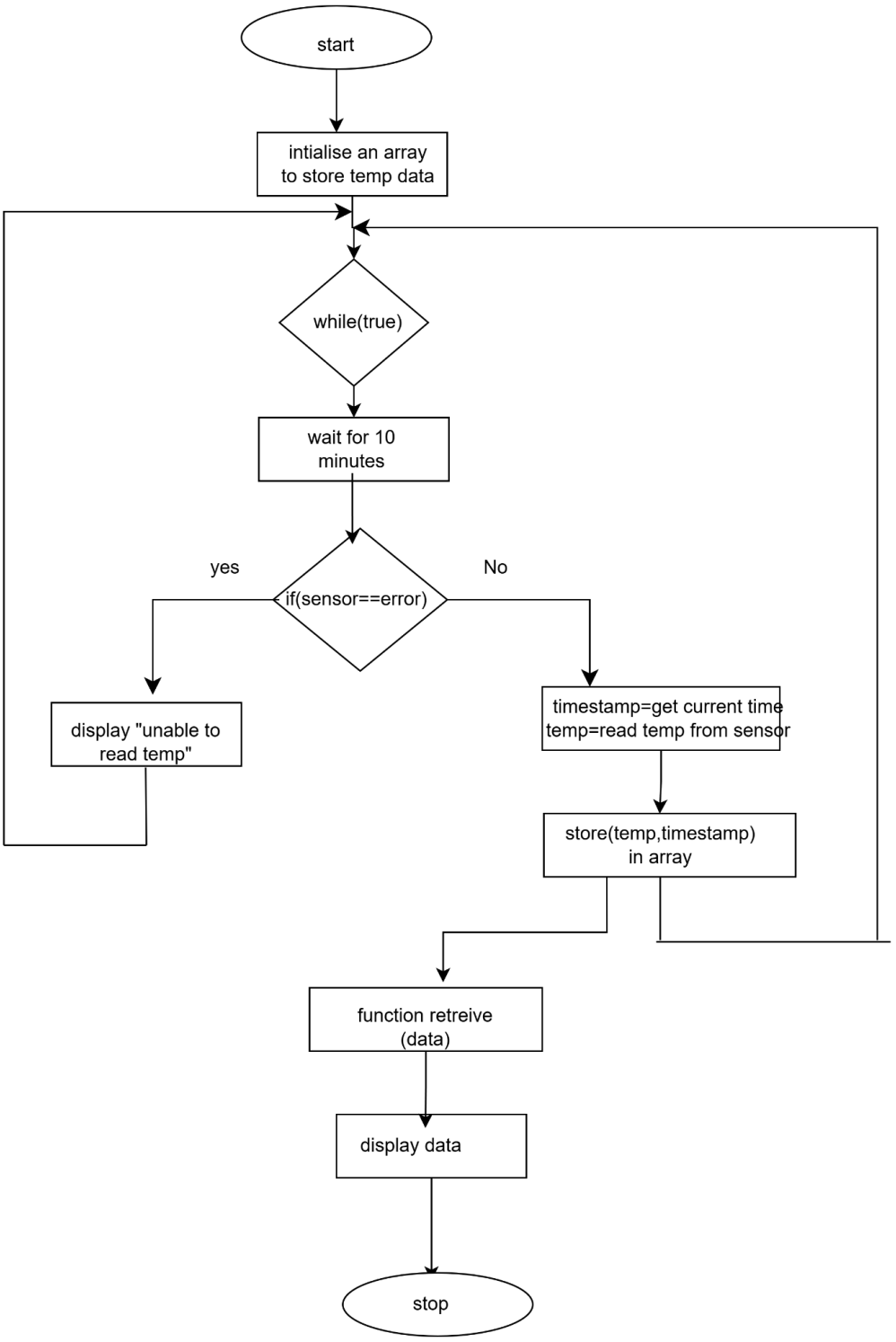
Include error handling for sensor read failures.

Psueocode

```
Set array or log file to store temperature data
while (true):
    Wait for 10 minutes
  IF (sensor error detected):
     Display "Error: Unable to read temperature"
  ELSE:
     timestamp = Get current time

     temperature = Read temperature from sensor

     Store (temperature, timestamp) in data array or log file
  ENDIF
END while
Function retrieveHistoricalData():
  Display data array or log file content
```



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).

Psuedocode

```
Initialize Bluetooth = OFF
Initialize currentState = 'STOP'
Initialize command = 'STATIONARY'
Initialize forward = 0, backward = 0, left = 0, right = 0
Initialize speed = 0

while (true):
   IF (Bluetooth == OFF):
            currentState = 'STOP'
            Set Feedback LED = OFF
         ELSE:
                  IF (command == 'FORWARD'):
                        forward += 1
                        currentState = 'MOVING_FORWARD'
                        Set Feedback LED = ON
                     ELSE IF (command == 'BACKWARD'):
```

```
            backward += 1
            currentState = 'MOVING_BACKWARD'
            Set Feedback LED = ON
        ELSE IF (command == 'LEFT'):
                left += 1
                currentState = 'MOVING_LEFT'
                Set Feedback LED = ON
        ELSE IF (command == 'RIGHT'):
                right += 1
                currentState = 'MOVING_RIGHT'
                Set Feedback LED = ON
        ELSE IF (command == 'STOP'):
                currentState = 'STOP'
                Set Feedback LED = OFF

        ENDIF
    ENDIF
END while
```

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.

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.