
# Real Time Systems using FreeRTOS – Final Project

# Seat Heater Control System

Implementation of a seat heater control system for the front two seats in the car (driver seat and passenger seat).

Each seat shall consist of:

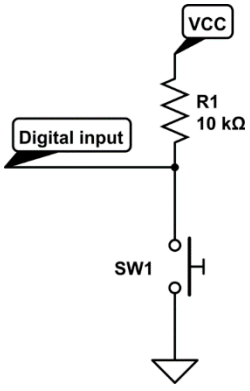
1. The button that is used to take the input level required to set the seat temperature.
2. Temperature sensor to monitor the temperature value.
3. Heating element to control the temperature value based on the desired level using a variable intensity power as input.
4. Shared screen between both seats to show the current state of the system.

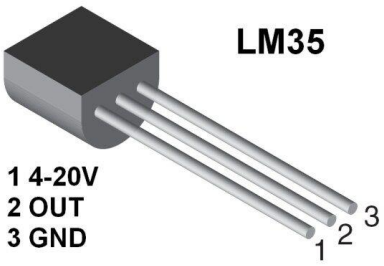
## System features:

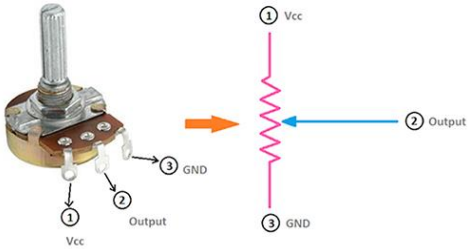
1. Heating level shall have one value of the following:
    a. Off: The feature is off, and the temperature is not controlled.
    b. Low: the desired temperature is 25°C.
    c. Medium: the desired temperature is 30°C.
    d. High: the desired temperature is 35°C.
2. System shall control the temperature to be set within the range of the desired temperature +/- 3°C.
3. Diagnostics shall be present in case of failure of the temperature sensor.
    a. If the temperature sensor gives a value out of its range this is considered as a failure in the sensor.
    b. The range of the temperature sensor shall be 5°C-40°C.
    c. Such issues shall be logged in the memory according to the requirements.
4. All the data required to be presented to the user shall be shown on the screen of the car.
5. System contains 3 different buttons as shown in the images:
    a. 2 buttons in the car's middle console where each button is used to control one of the two seat heaters.
    b. 1 extra button in the steering wheel to control the driver seat heater to make the usage of the heater easier for the driver.


## Functionality:

1. For each seat the user shall input the required level of heating (off, low, medium, or high) where initially off is set and each press goes one step further (from off to low, from low to medium, from medium to high, and from high to off once again).
2. The heater will be driven from a signal to control its intensity:
    a. If the current temperature is less than the desired temperature by 10°C or more the heater should be enabled with the high intensity.
    b. If the current temperature is less than the desired temperature by 5°C to 10°C the heater should be enabled with a medium intensity.
    c. If the current temperature is less than the desired temperature by 2°C to 5°C the heater should be enabled with a low intensity.
    d. If the current temperature is more than the desired temperature the heater should be disabled.
    e. The heater shall be enabled once again if the temperature becomes less than the desired temperature by 3°C.
    f. Note that for the purpose of testing only the heater level shall control the LED:
        i. Green color indicates low intensity.
        ii. Blue color indicates medium intensity.
        iii. Cyan color indicates high intensity.
3. The temperature sensor shall be connected to the ADC so that the current temperature is measured correctly.
    a. You can use an LM35 temperature sensor to measure the temperature.
        i. Refer to the datasheet of the sensor for its specifications.
    b. For the purpose of testing only, you can use a potentiometer connected to the ADC instead of the temperature sensor in order to take the temperature as input from the user with the following specifications:
        i. 0v-3.3v is mapped to the range 0°C-45°C.
        ii. Note that only the range 5°C-40°C shall be treated as the valid range.
4. The current temperature, the heating level, and the heater state should be displayed on the screen by sending it through the UART.
5. If failure was detected in the temperature sensor the seat assigned to such sensor shall stop from controlling the temperature and the red LED shall be ON to inform the user that there is a problem with the sensor.



## Requirements:

1. It is required to implement the controller unit using Tiva C.
2. Software shall use FreeRTOS for handling different tasks and objects.
    a. The system shall be well-designed with at least 6 tasks.
    b. Task implementation shall be reused for two different instances (one for the driver seat and another for the passenger seat) if required.
    c. Shared resources must be handled such that exclusive access to the resource is achieved.
    d. Signals and events must be correctly handled between different tasks.
    e. Data sharing between different tasks must be handled properly without any loss of data.
    f. Responsiveness to buttons shall be as high as possible.
3. GPIO, UART, GPTM, and ADC modules shall be part of the MCAL modules and used in the software.
4. Diagnostics shall be implemented where each of the following will be stored in the RAM
    a. The failure along with the timestamp (using GPTM) at which the failure occurred.
    b. The last heating level set by the user (off, low, medium, or high) with its timestamp shall be recorded.
5. Runtime measurements shall be made using the GPTM for the system to monitor the following parameters:
    a. Execution time for each task.
    b. CPU load.
    c. Resource lock time per task for each resource.

## Testing Scenario:

1. The user set the heating level to high while the temperature was 10°C.
2. Initially the heater will be driven with high intensity.
3. When the temperature reaches 25°C the heater will be driven with medium intensity.
4. When the temperature reaches 30°C the heater will be driven with low intensity.
5. When the temperature reaches 35°C or more the heater will be disabled.
6. Whenever the temperature goes down to 32°C, the heater will be reenabled with low intensity.
7. The required data will be sent by UART to be displayed on the screen.
8. If the temperature was sensed to be with a value less than 5°C or more than 40°C the controlling of the temperature shall be disabled, and the user shall be informed by enabling the LED according to the requirements.

## Bonus:

Implement non-volatile memory for the diagnostics part:

1. The last heating level set by the user (off, low, medium, or high) with its timestamp shall be saved in the non-volatile memory.
2. The diagnostics issues logged in the system shall be saved in the non-volatile memory as well.

Deliverables:

1. Application source code including the following:
    a. Application tasks source code.
    b. FreeRTOS configuration file.
    c. MCAL modules source code.
2. Tested and working elf.
3. Simso simulation project.
4. Document contains the following:
    a. Diagrams for the design of the system containing all the details of the tasks.
    b. Screenshots for the output of the system (UART messages).
    c. Simulation results using Simso.
    d. Run time measurements results.

# Thank You

# Edges For Training Team