Vehicle Seat Heater System Using RTOS

Project Documentation

Prepared by:

Eng. Yousouf Soliman

1. Project Brief:

This project is a seat heater controller system designed for the front-row seats in a passenger vehicle. The main functionality of the controller is to get temperature readings from both the driver and passenger seats individually and then act accordingly. The action that the controller takes is setting heater intensity. There are some other supporting functions offered by the system such as logging any sensor malfunctions, logging the user's choice, controlling the LED indicators and showing the state of each seat and its heater via UART.

2. System Design:

This system consists of 6 main functionalities represented in 10 tasks. They are broken down as follows:

- Run-time measurements task
- Sensor Error Hook
- Driver and passenger input tasks (3 tasks)
- Driver and passenger seat temperature reading tasks (2 tasks)
- Driver and passenger set heater state tasks (2 tasks)
- Display state via UART task

3. Task Description:

3.1 vRunTimeMeasurementsTask:

Description: Calculates the task execution time and CPU load over a period of 2.2 seconds.

Priority: 2

Type: Periodic

Periodicity: 4.5 seconds

3.2 vSetDriverSeatHeaterState:

Description: Sets the state of the driver seat's heater and is part of the system's main loop.

Priority: 2

Type: Periodic

Periodicity: 150 ms

3.3 vSetPassengerSeatHeaterState:

Description: Sets the state of the passenger seat's heater and is part of the system's main loop.

Priority: 2

Type: Periodic

Periodicity: 150 ms

3.4 vGetDriverSeatTemp:

Description: Gets the driver seat temperature and updates it in the Driver Info structure, detects any malfunctions in the driver seat sensor and logs it in the EEPROM, also it gives the semaphore for task 3.10.

Priority: 3

Type: Periodic

Periodicity: 500 ms

3.5 vGetPassengerSeatTemp:
Description: Gets the passenger seat temperature and updates it in the passenger Info structure, detects any malfunctions in the passenger seat sensor and logs it in the EEPROM, also it gives the semaphore for task 3.10.
Priority: 3
Type: Periodic
Periodicity: 500 ms

3.6 vDisplaySystemState:
Description: Displays the system state of both seats via UART.
Priority: 3
Type: Periodic
Periodicity: 1 second

3.7 vGetDriverInput:
Description: Waits for an edge triggered interrupt on the driver's selection button to set the correct level of temperature chosen by the driver, also it logs the driver's selection on EEPROM.
Priority: 4
Type: Event Based

3.8 vGetDriverInputFromSteering:
Description: Waits for an edge triggered interrupt on the steering wheel selection button to set the correct level of temperature chosen by the driver, also it logs the driver's selection on EEPROM.
Priority: 4
Type: Event Based

3.9 vGetPassengerInput:

Description: Waits for an edge triggered interrupt on the passenger's selection button to set the correct level of temperature chosen by the passenger, also it logs the passenger's selection on EEPROM.

Priority: 4

Type: Event Based

3.10 vSensorErrorHook:

Description: Waits on semaphore that is given by the Get Temperature tasks to catch any sensor failures, turns off the heater completely, stops any input from being recorded, logs the error to the EEPROM and turns on the corresponding error LED.

Priority: 5

Type: Event Based (Depends on periodic tasks with periodicity of 500 ms)

4. Resource and Event Management:

4.1 Shared Resources:

4.1.1 Temperature Sensor:

Description: We use the same ADC port to convert the readings of each sensor (driver and passenger) therefore we can only access one at a time

Handled using: xLM35GetTempMutex

Used in: vGetDriverSeatTemp and vGetPassengerSeatTemp

4.1.2 Driver's Information Structure:

Description: A structure containing driver's seat temperature, driver's seat heater state, and diver's desired temperature.
Handled using: xDriverInfoMutex
Used in: vSetDriverSeatHeaterState, vGetDriverInput, and vDisplaySystemState

4.1.3 Passenger's Information Structure:

Description: A structure containing passenger's seat temperature, passenger's seat heater state, and passenger's desired temperature.
Handled using: xPassengerInfoMutex
Used in: vSetPassengerSeatHeaterState, vGetPassengerInput, and vDisplaySystemState

4.1.4 EEPROM:

Description: While reporting errors, we access close EEPROM segments therefore we need exclusive access to EEPROM.
Handled using: xErrorReportingMutex
Used in: vSensorErrorHook

4.2 Events:

4.2.1 Diver's button press:

Description: When the diver presses their middle console button, they trigger an edge triggered interrupt that is handled by the appropriate Handler (in our case, GPIOPortFHandler) which gives the binary semaphore to allow task vGetDriverInput to execute.
Handled using: xDriverInputSemaphore

4.2.2 Steering Wheel button press:

Description: When the diver presses their steering wheel button, they trigger an edge triggered interrupt that is handled by the appropriate Handler (in our case, GPIOPortFHandler) which gives the binary semaphore to allow task vGetDriverInput to execute.

Handled using: xDriverInputSemaphoreFromSteering

4.2.3 Passenger's button press:

Description: When the passenger presses their middle console button, they trigger an edge triggered interrupt that is handled by the appropriate Handler (in our case, GPIOPortFHandler) which gives the binary semaphore to allow task vGetPassengerInput to execute.

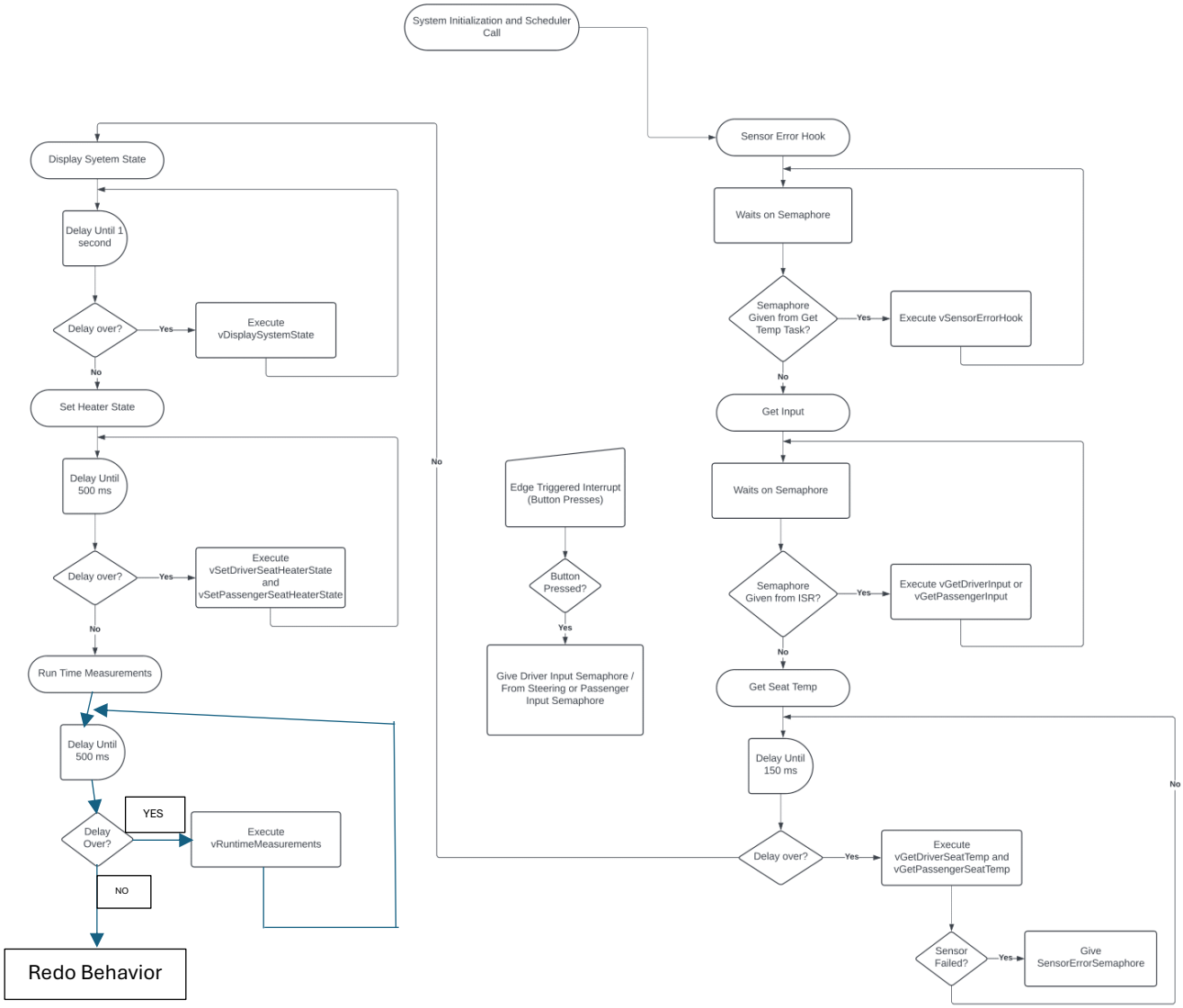
Handled using: xPassengerInputSemaphore

4.2.4 Sensor Reading Error:

Description: When any of the temperature sensor malfunctions, they give false readings, these malfunctions are detected by vGetDriverSeatTemp and vGetPassengerSeatTemp tasks which in return give the appropriate binary semaphore to allow vSensorErrorHook to execute. They also set their corresponding flags DRIVER_SENSOR_ERROR and PASSSENGER_SENSOR_ERROR to indicate exactly which sensor malfunctioned.

Handled using: xSensorErrorHookSemaphore

5. System Diagram:



6. UART messages:

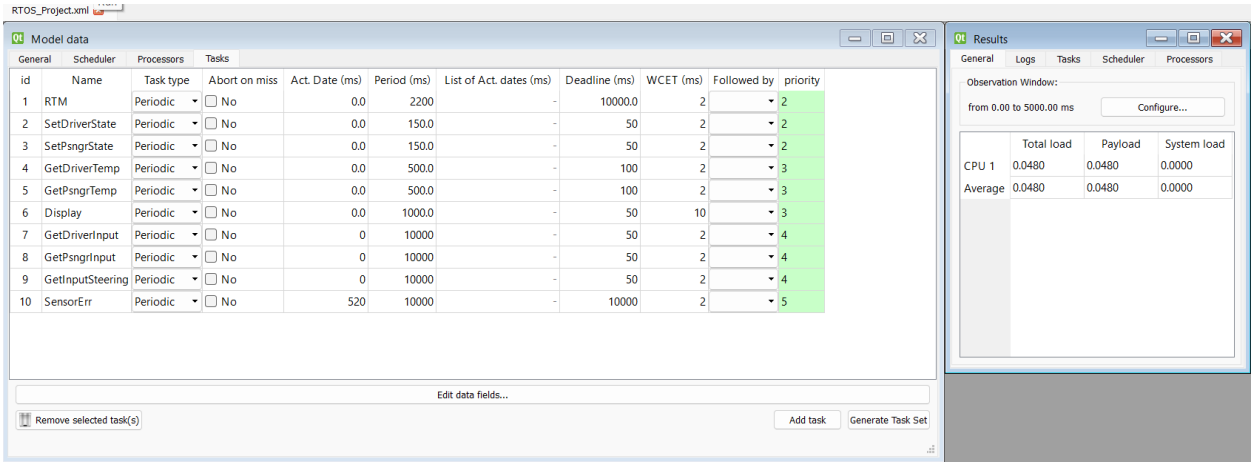
```
*****Driver Seat*****
Current Temperature: 12
Desired Heating Level: LOW
Heater Intensity: HIGH
*****Passenger Seat*****
Current Temperature: 20
Desired Heating Level: OFF
Heater Intensity: OFF
```

7. Run-time measurements results:

```
Task of tag 1 execution time is: 5 msec
Task of tag 2 execution time is: 5 msec
Task of tag 3 execution time is: 3 msec
Task of tag 4 execution time is: 3 msec
Task of tag 5 execution time is: 20 msec
Task of tag 6 execution time is: 5 msec
Task of tag 7 execution time is: 5 msec
Task of tag 8 execution time is: 5 msec
Task of tag 9 execution time is: 10 msec
Task of tag 10 execution time is: 20 msec
CPU Load is 9%
```

8. SimSo results:

8.1 Configurations* and simulated CPU load:



*Note: Tasks 7-10 are sporadic tasks, they are simulated as periodic with activation dates to simulate a sporadic event-trigger, this is due to the lack of capabilities in SimSo.

8.2 Gantt Chart: