BODY CONTROL MODULE

- SEATING CONTROL
- INTERIOR LIGHTINGS
- HVAC SYSTEM

-Aditi Sarkar

INTRODUCTION

In automotive electronics, Body Control Module (BCM) is used to define an electronic control unit responsible for monitoring and controlling various electronic accessories in a vehicle's body. The body control module (BCM) is a processor-based power distribution center that supervises and controls functions related to the car body such as lights, windows, security, door locks and access control, and various comfort controls. The central BCM also operates as a gateway for bus and network interfaces to interact with remote electronic control units (ECU) for other systems. The BCM does not control any engine-related functions.

Body Control module links together all the electronic dashboard vehicle features and keeps them under central surveillance to make it easier for our automobile to operate them. In order to accommodate the increasing number of functions on a shrinking central BCM, designers need to implement a higher level of integration with a scalable and flexible architecture to fulfill a broad spectrum of requirements. The automotive ECUs plays a major role. They are responsible for taking the input from the user and transfer the signal first to the car battery and then to the target component.

The goal of our project is to implement various Body Control Module's subsystems such as Heating Ventilation and Air Condition (HVAC), Car Interior Lighting, Seat Control Module.

The body control module receives data from the input devices and controls the output devices based on this data. There are innumerable body control functions that need to be performed smoothly and reliably. Controlling these parts would also be possible without a BCM unit, but that would be an additional burden of wires going around the insides of the vehicle.

A body control module not only eliminates the need for wires clogging the car but also manages the flow of power so that the electrical module of the car does not get over-burdened if multiple functions are being carried out at the same time.

FEATURES:

- **Seating Control Module**

1. Using weight sensors under seats to monitor seat occupancy to control HVAC.
2. Seat belt monitoring- After switching of the engine, if the seat is occupied and the seat belt is NOT put ON i.e. slide switch is OFF, LED indicator will be ON and buzzer should buzz and will be vice-versa when the seat belt is put ON.
3. Seating temperature- In manual mode, according to user input through knob seat temperature can be adjusted through the heater and in automatic mode the same functionality is carried by the temperature sensor, as per the user requirements.

- **InCabin Light Module**

1. Depending on Door lock status and Ignition status user will be allowed to adjust the brightness of the InCabin light by choosing the mode of operation.
2. Manual Mode- This mode will be allowed when both doors are locked and ignition is ON. User can adjust the brightness of lights using the Manual knob present on the panel.
3. Automatic Mode- This mode uses Light sensor which provides the light intensity data of outside, in this mode user has to switch on the automatic switch for Lights and the brightness inside the cabin will be adjusted accordingly.

- **HVAC Control System**

1. In-Cabin temperature is controlled by various modes that can be selected by the passenger on the dashboard of the vehicle.
2. Manual Mode- In this mode, the passenger selects the Manual Mode on the dashboard and adjust the manual knob to set Blower speed in Low, Medium and High Speed of Blower.
3. Automatic Mode- In this mode, the passenger selects the Automatic Mode on the dashboard and the temperature sensor will sense the current cabin temperature and set the blower speed according to it.
4. Seatbelt Mode- In this mode when the passenger has put his seatbelt then automatically the automatic operation starts if neither manual nor automatic mode is selected. The temperature sensor will sense the current cabin temperature and set the blower speed according to it.
5. Defrost Mode- In this mode the passenger selects the Defrost Mode on the dashboard when the humidity value inside the cabin is increased and the RH sensor will sense the current Relative Humidity Percentage and set the blower speed according to it.

STATE OF ART:

- **Ageing-**

The innovation in automotive over the years has been phenomenal. Vehicles have become safer, speedier and more comfortable. During the first phase of the evolution of the automobiles, more emphasis was given on the engine, powertrain, and chassis. Simultaneously, electronics was also being introduced in the vehicles to make the control units and systems more safe and intelligent.

The current changes in car's Seating Systems are as follows-

- Customizable Multi-way Adjustable Seats
- Stress & Fatigue Detecting Seats
- Smartphone Connected Seats
- Customized Massage
- Memory Seats

The future of the seating solutions in cars will be personal rather than generic. With Artificial Intelligence teaming up with electronics, the seating control solutions will be designed to be smart enough to recognize the person sitting in the car and adjust to his or her preferences.

- **SEGMENTS-**

Seat belt control offers the possibility to fine tune the . All the seat adjustments can be stored and retrieved at will. In these seating control solution the automotive ECUs have a major role to play. They are responsible for taking the input from the user and transfer the signal first to the car battery and then to the target component. These functionalities generally comes in medium to high cost passenger cars and commercial vehicles.

- **4W and 1H-**
  - **WHAT-**
    In seating control, the basic idea is to provide comfortable seating not only to the driver but to other passengers also by minimizing the need of manual control and increasing safety of the passengers by monitoring seat belt functioning

  - **WHY-**
    An advanced seating control ensures comfort and safety of the passengers.

  - **WHERE-**

Sensors like temperature sensor are attached to the seats and ECU which on receiving input from user start functioning. The monitoring of seat belt will help in monitoring a number of accidents. The challenge is make sure that all the loads including a range of electric motors used to finely tune seat morphology are efficiently and reliably driven and controlled.

- **WHEN-**
  When a person seats on the weight sensors will send electrical signal which will activate the in-cabin lights. When the engine starts and the seat belt is not put ON it will give a reminder to put the seat belt ON. According to user input the seating temperature will change and movement of seats can be controlled using Bluetooth enabled device.

- **HOW-**
  By c In seating control solution, the automotive ECUs play a major role. They are responsible for taking the input from user and transfer the signal first to
  In the seating control solution, the automotive ECUs have a major role to play. They are responsible for taking the input from the user and transfer the signal first to the car battery and then to the target component. A specialized BCM ECU called body control module is responsible for activating any kind of mechanical action that need to be performed to control any component of the seating system.

- **SWOT-**

- **STRENGTH-**

In-cabin lighting and advanced seating control can be used whenever theirs a need for it, using the ambient lighting and automotive seating control could improve not only the safety but also mood and perception of the driver or passengers. HVAC system can be used as per our convenience for comfort in extreme weather conditions, to maintain in-cabin temperature. Whenever the car is parked out for long time in the sun, the HVAC system automatically sets the cooling blower at high speed when the engine is on.

- **WEAKNESS-**

Interior Lighting and these advanced seating control is one of the features which most high-end brands have, bringing them into lower-end car brands is difficult. Monitoring the air-quality coming from the outside vents is a great challenge to maintain in a HVAC System.

- **OPPORTUNITIES-**

Theirs a scope of improvement in the lighting module as we know of now. We can target the safety of the vehicle by using the lighting module with certain parts of ADAS. A new opportunity in seating control module is memory seats. These automotive seating system remember the settings for a particular user and implements it upon the user's request. HVAC system can be improved further by maintaining the in-cabin temperature for a specific passenger seat and driver seat and efficiency improvement of the ventilation system in polluted areas.

- **THREATS-**

Framing some idea into an actual working module needs a huge amount of research and time, by this time in the market some others might beat you in the same field giving results faster and better. Along with that incorrect seating position can result into fatal accidents.

LITERATURE REVIEW:

Title- Automotive HVAC Dual Unit System Cool-Down Optimization Using a DFSS Approach

Published Date- April 2, 2019

Publisher- SAE International in United States

Analysis- This paper deals with the optimal configuration of an air-conditioning system which helps in achieving the maximum cool down by investigating the effect of parameters such as distribution of air between the front and the rear unit, front and rear evaporator. In this paper, they used a Design for Six Sigma (DFSS) approach to identify the best combination of system components and air distributions that achieve the goal of optimized AC performance.

Title- A Review on Automotive Seat Comfort Design

Author and Publication year- Dama, Kiran & Babu, V & Rao, R & Rao, Madhava. (2015).

Journal- International Journal of Engineering Research. (4. 678. 10.17577/IJERTV4IS040897.)

Analysis- Seats are one of the most important components of automotive. This literature review incorporates previous studies related to automotive seating comfort and discomfort. It reviews key writings related to seating design and criteria and reviews comfort issues in automotive seating. This paper designate the measurement methods that are used to improve the physiological comfort of an automotive driver's seating. It describe the background of sitting comfort besides discomfort. In the end, they propose a methodology for the development of comfortable driver's seats.

Requirements Analysis:

Stakeholders Requirements	System requirements	Software Requirements	Test Cases for the corresponding Software Requirements

Checking the status of doors and the condition of the car (standstill or running). Also checking the obstacle in the car's way. the	1. Arduino UNO 2. Push Buttons 3. Light sensor 4. Weight sensor 5. Pressure sensor 6. Radar sensor 7. LEDs	1. Arduino 2. Tinkercad	1. Check the signals by switches, check the seatbelt, weight sensor data. 2. Check the working of LEDs depending on the data fed by pressure, weight sensor 3. Check the working of LEDs depending on data fed by the light sensor 4. Flickering of LEDs when obstacle by radar sensor is detected.

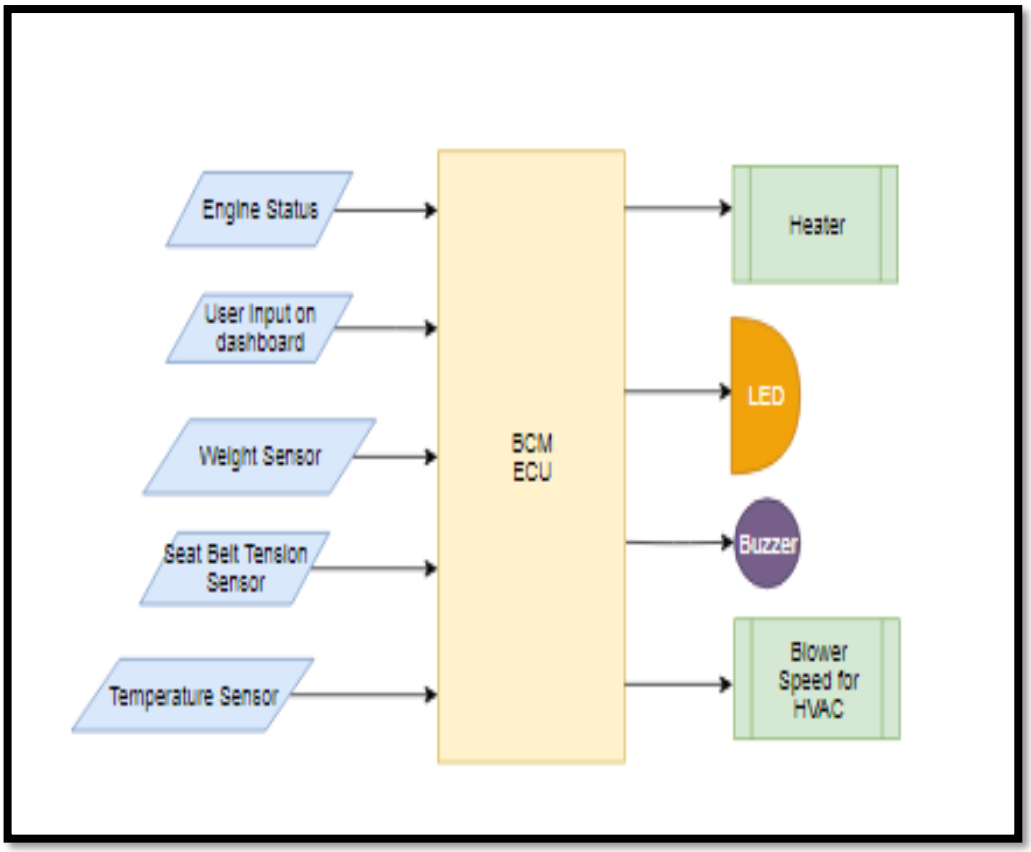
Monitoring weight sensor data and seatbelt status to give command over HVAC and lighting module to operate them in manual or automatic mode. Also adjusting the temperature of the seat	1. Arduino UNO 2. Weight sensor 3. Bluetooth module 4. Temperature sensor 5. Blower 6. Switch(seatbelt buckle) 7. Buzzer 8. LED3	1. Microcontroller 2. Tinkercard	1. Check the signals by switches, check the seatbelt, weight sensor data. 2. Check the working of LED3 on the data fed by weight sensor & status of Slideswitch. 3. LED3 glows when driver isn't using seatbelt,buzzer buzzes. 4. If driver puts on seatbelt, LED3 will stop glowing and buzzer stops too. 5. Checks for user's preferred temp. and adjusts the seating temp.

Improved automatic HVAC Control system which detects the driver has worn seatbelt or not and also by measuring the weight of the driver using weight sensor.	1. Arduino UNO R3 2. Breadboard 3. Push Buttons 4. TMP36 Temperature Sensor 5. Weight sensor 6. Servo Motor	1. Tinkercad 2. Arduino program	1. Check the signal coming from push button, temperature sensor and weight sensor. 2. Check the driver is wearing seatbelt or not. 3. Check the Ac blower motor is increasing the speed of motor according to the in-cabin temperature sensed by sensor.

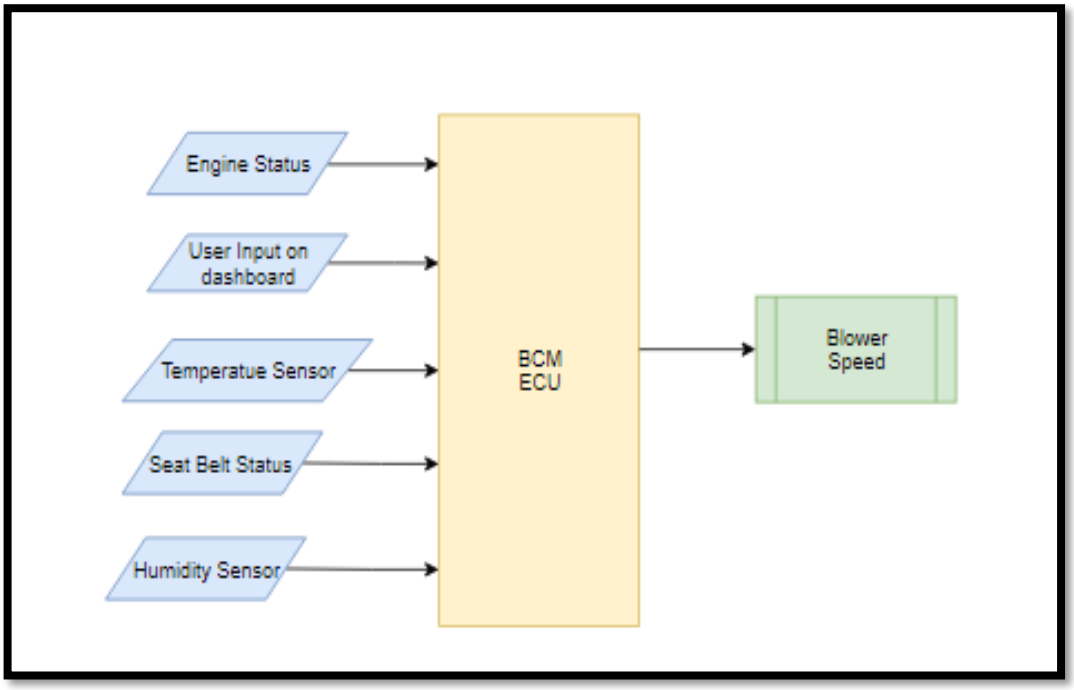
When the interior car window gets foggy due to the contact of hot and humid air on cool window, the moisture condenses to fog. In order to reduce fog, addition of humidity in the air is required, which can be done by lowering the in-cabin temperature using heater.	1. Arduino UNO R3 2. Breadboard 3. Push Buttons 4. TMP36 Temperature Sensor 5. DHT11 Humidity Sensor 6. Servo Motor	1. Tinkercad 2. Arduino program	1. Check the signal coming from temperature sensor and humidity sensor. 2. Check the current speed of the motor of blower. 3. Check whether the blower motor speed should be increased or not to reduce the in-cabin temperature.

STRUCTURAL UML DIAGRAMS (BLOCK DIAGRAMS):

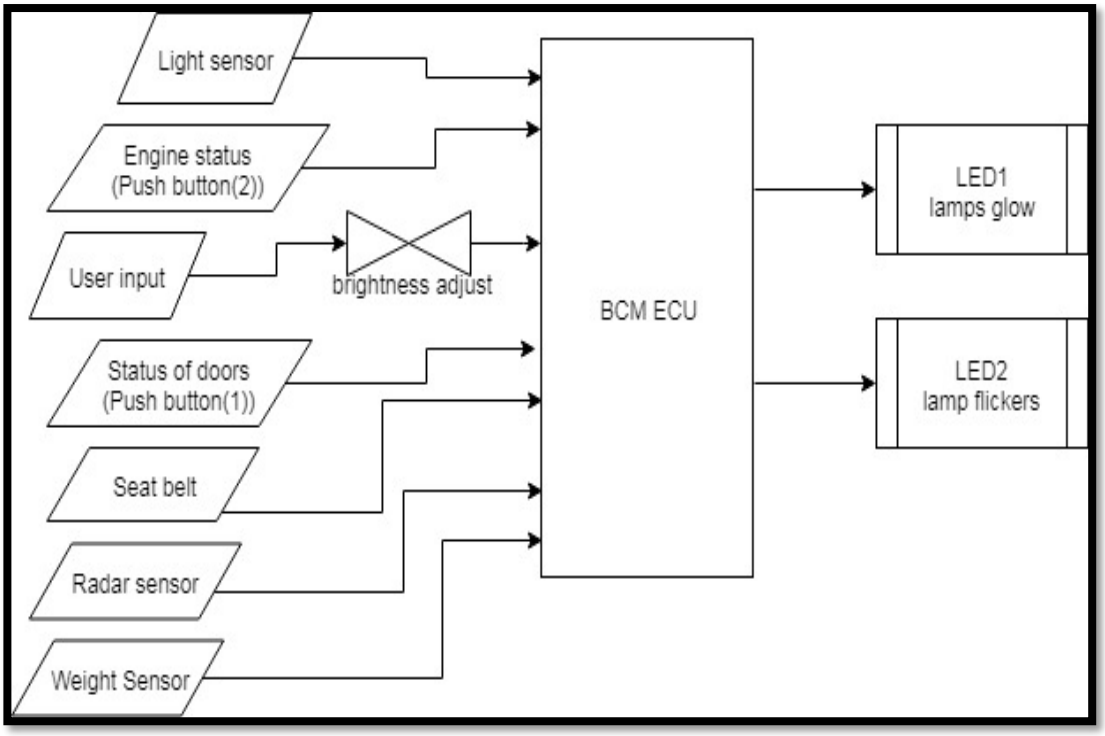
Seating Control Module:



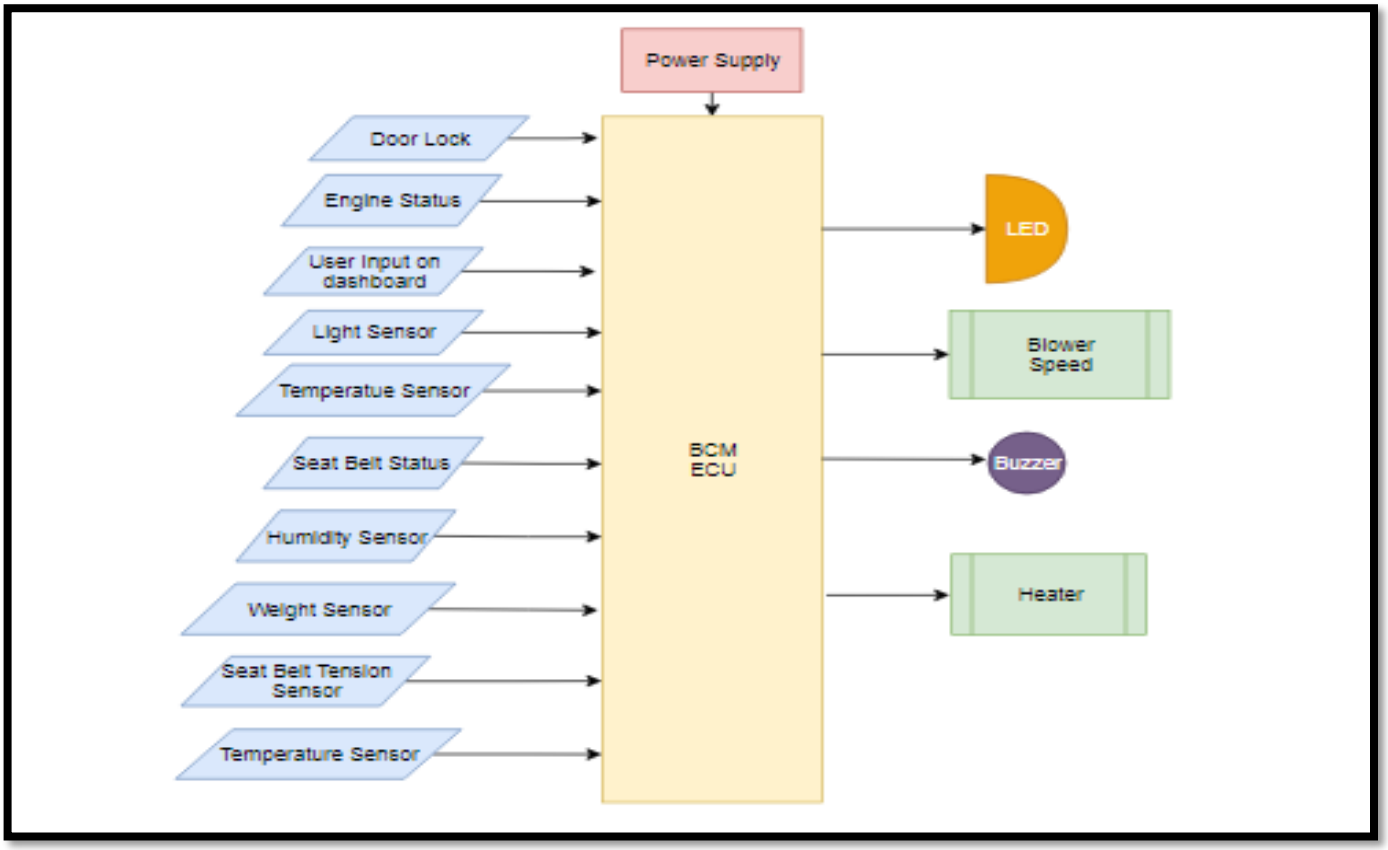
HVAC Control System



Interior Lighting System

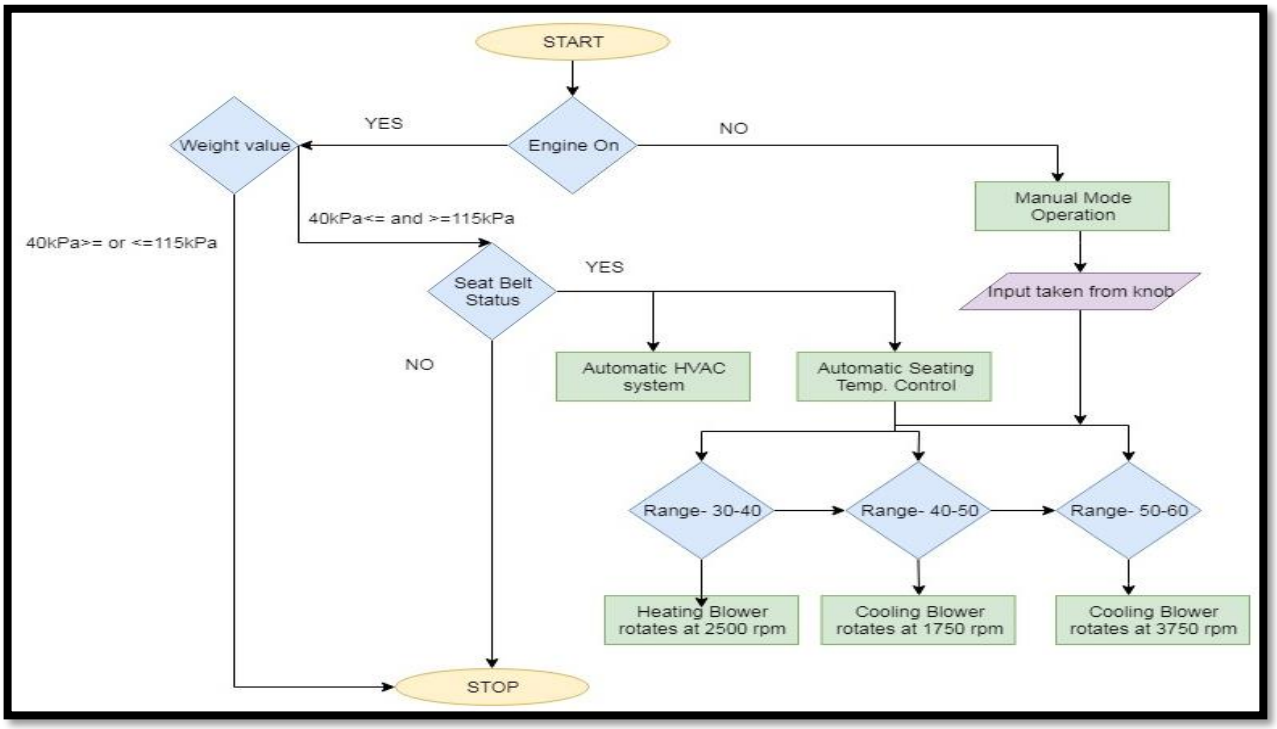


INTEGRATED BLOCK DIAGRAM:

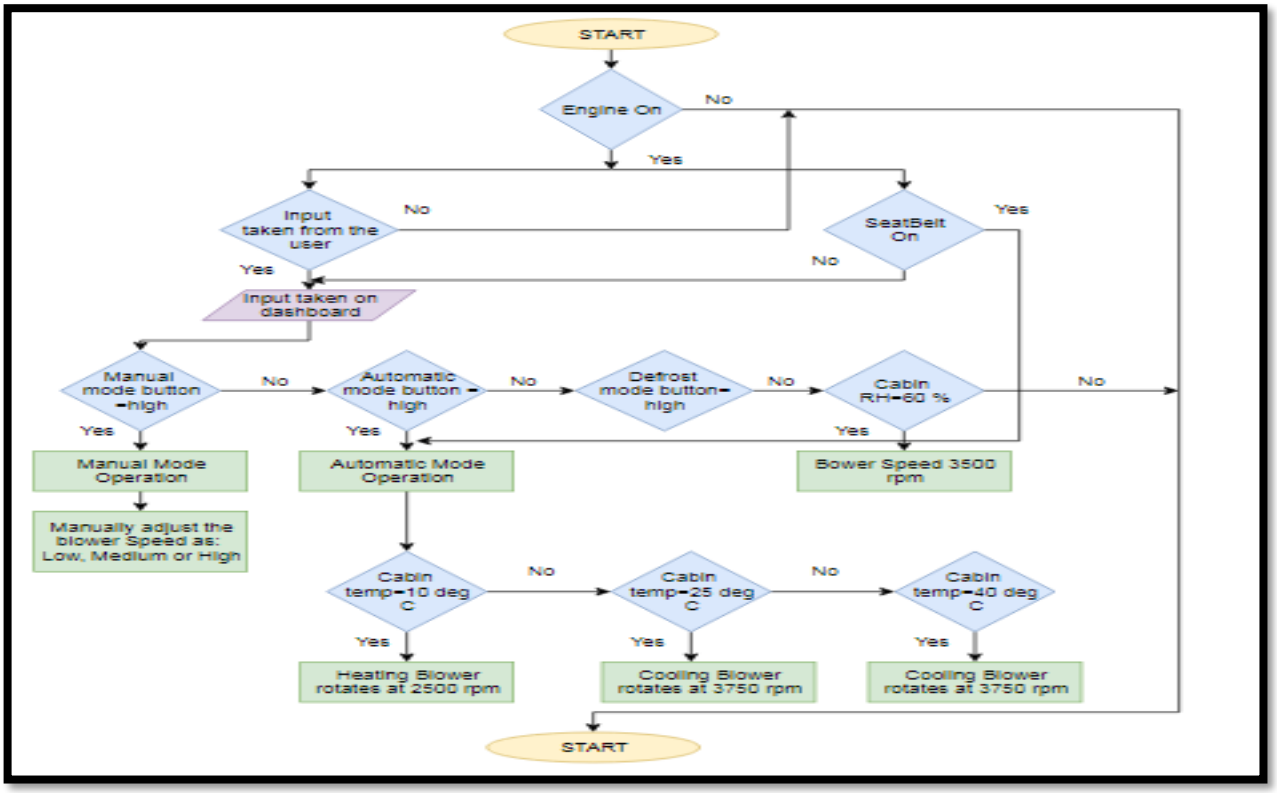


BEHAVIORAL UML DIAGRAMS (FLOW CHART):

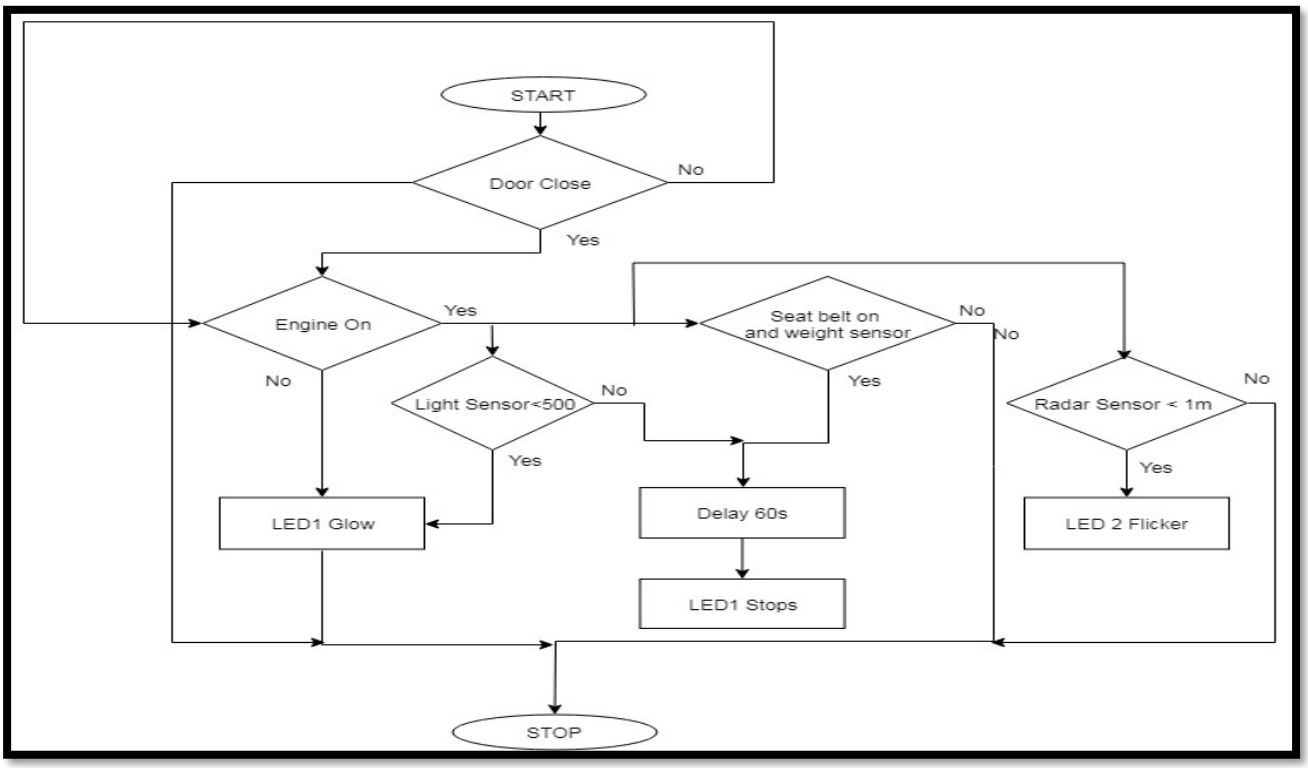
Seating Control Module:



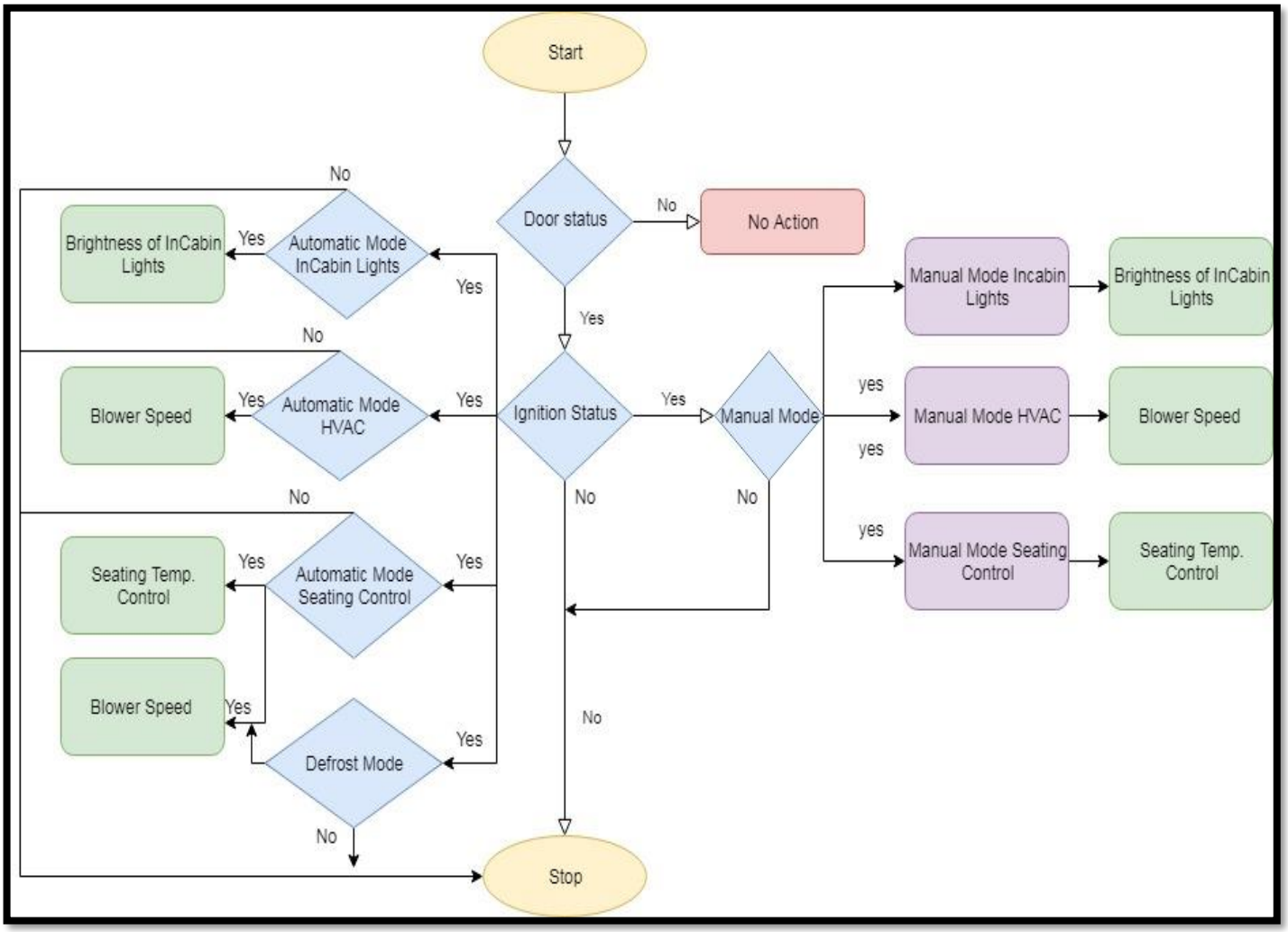
HVAC Control System



Interior Lighting System



INTEGRATED FLOWCHART:



CANoe DATABASE DESCRIPTION:

1) **Network nodes used:**
   1. ECU_BCM: Depending upon the messages from sensor_1 and sensor_2 all major operations are done.
   2. Sensor_1: This Node will be used for sensing weight values and temperature values and for activation of Automatic and manual mode.
   3. Sensor_2: This Node will be used for sensing light intensity for automatic mode and defrost mode for HVAC.

- Functionalities:-

Manual mode (In cabin lights)-

SNO	Manual Knob	Brightness level (%)	Incabin lights(ON=1/OFF=0)
1.	0-45	35	1
2.	45-85	50	1
3.	85-100	75	1
4.	>100	100	1

Automatic mode (In cabin lights)-

SNO	Light sensor data	Brightness level (%)	Incabin lights(ON=1/OFF=0)
1.	0-45	35	1
2.	45-85	50	1
3.	85-100	75	1
4.	>100	100	1

Manual mode (HVAC)-

SNO	Manual Knob	Blower Speed (rpm)	Blower Speed Range
1.	0-45	1250	Low
2.	45-85	2500	Medium
3.	85-100	3750	High

Automatic mode (HVAC)-

SNO	Temperature Sensor data(◦C)	Blower Speed (rpm)
1.	0-10	3750
2.	10-20	2500
3.	20-30	1750
4.	30-40	2500
5.	40-50	1750

Seatbelt mode (HVAC)-

SNO	Temperature Sensor data(◦C)	Blower Speed (rpm)
1.	0-10	3750
2.	10-20	2500
3.	20-30	1750
4.	30-40	2500
5.	40-50	1750

Defrost mode (HVAC)-

SNO	RH Sensor data(%)	Blower Speed (rpm)
1.	0-20	1500
2.	20-40	1750
3.	40-60	2500
4.	60-80	3500
5.	80-100	3750

Manual mode (Seating control)-

SNO	Manual Knob Position	Heater (in Watt)	Heater Mode
1.	0-30	0	OFF
2.	30-40	30	LOW
3.	40-50	60	MEDIUM
4.	50-60	160	HIGH
5.	>60	0	OFF

Automatic mode (Seating control)-

SNO	Temperature Sensor data(◦C)	Heater (in Watt)	Heater Mode
1.	0-30	0	OFF
2.	30-40	30	LOW
3.	40-50	60	MEDIUM
4.	50-60	160	HIGH
5.	>60	0	OFF

2). Messages:

1. Sensor_1 data (0x1): This message is mapped to weight value, Temperature value for HVAC and Seat Temperature I/p for Automatic seating temperature control.
2. Sensor_2 data (0x2): This message is mapped to Relative Humidity and light sensor signals.
3. Doorlock (0x3): This message is mapped to Doorlock, Manual_knb Incabin lights, Light_Intensity, Manual_knb_BlowerSpd, Manual_knb_Seattemp, Seat_Heater.
4. BlowerSpeed (0x0): This message is mapped to current blower speed.
5. SeatHeater (0x4): This message is mapped to Seat_Temp_Heater.

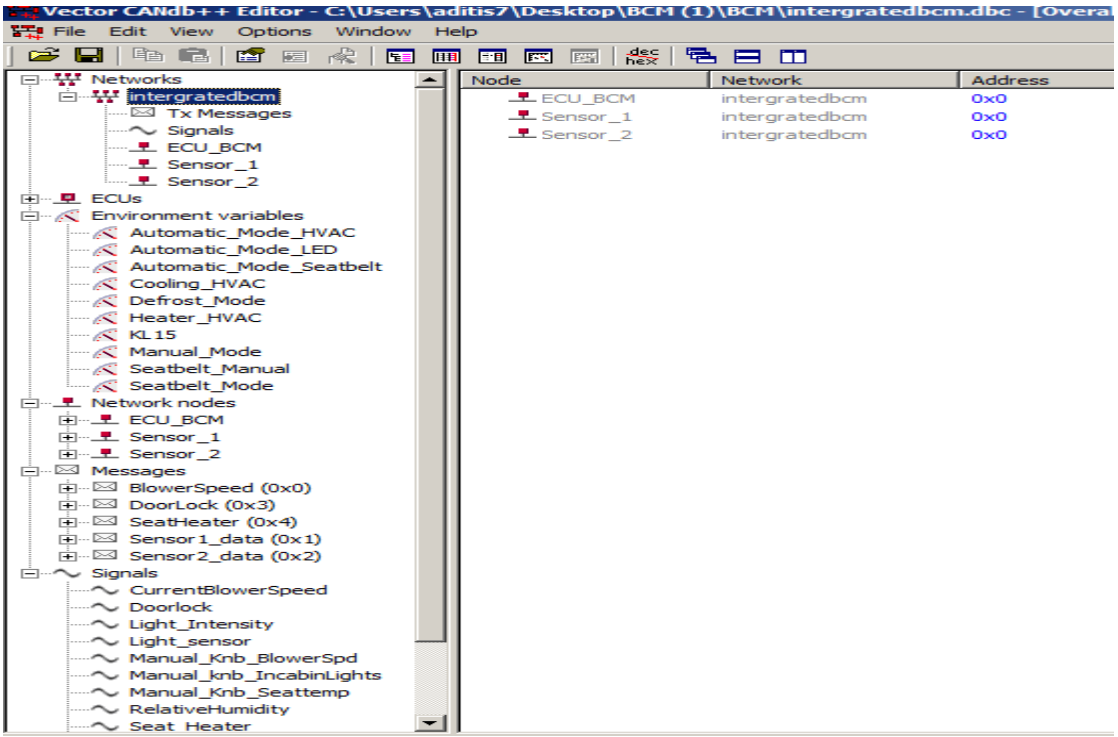
3). Signals:

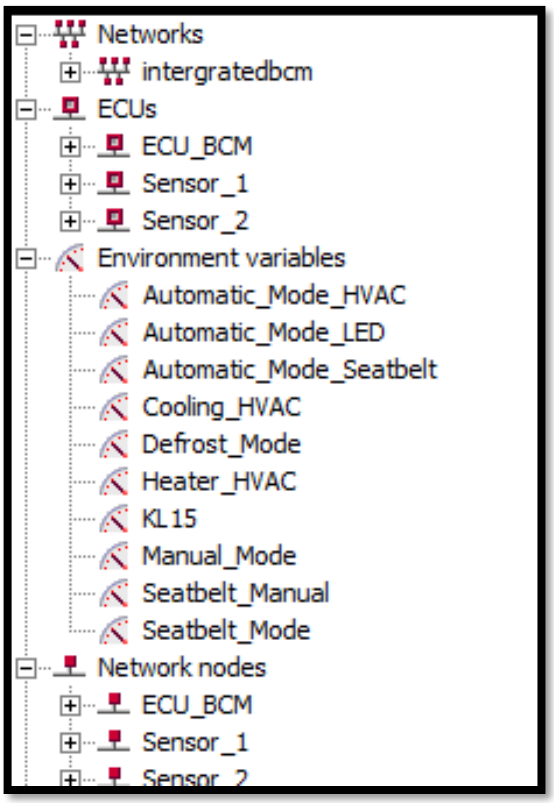
1. CurrentBlowerSpeed
2. Doorlock
3. Light_Intensity
4. Light_sensor
5. Manual_knb_BlowerSpd
6. Manual_knb_InCabinLights
7. Manual_knb_Seattemp
8. RelativeHumidity
9. Seat_Heater
10. Seat_Temp_Heater
11. Seat_temp_ip
12. Temperature
13. WeightValue

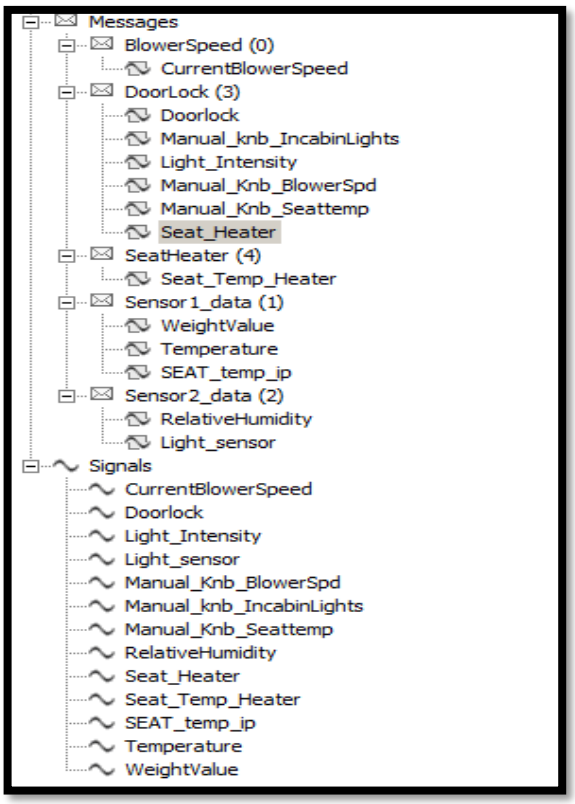
4). Environment Variable

1. KL15- To check the Ignition Status whether the system is ON/OFF.
2. Automatic_Mode_HVAC- To switch ON the Automatic mode for HVAC.
3. Automatic_Mode_LED - To switch ON the Automatic mode for InCabin Lights.
4. Automatic_Mode_Seatbelt- To switch ON the Automatic mode for the Seating control module.
5. Manual_Mode- To enable the Manual mode for all the functionalities.
6. Seatbelt_Mode- To monitor the Seat Belt status and activation of seating temperature and HVAC system.
7. Cooling_HVAC- Status bit LED for HVAC in cooling mode.
8. Defrost_Mode- To enable the defrost mode.
9. Heater_HVAC- Status bit LED for HVAC in heating mode.

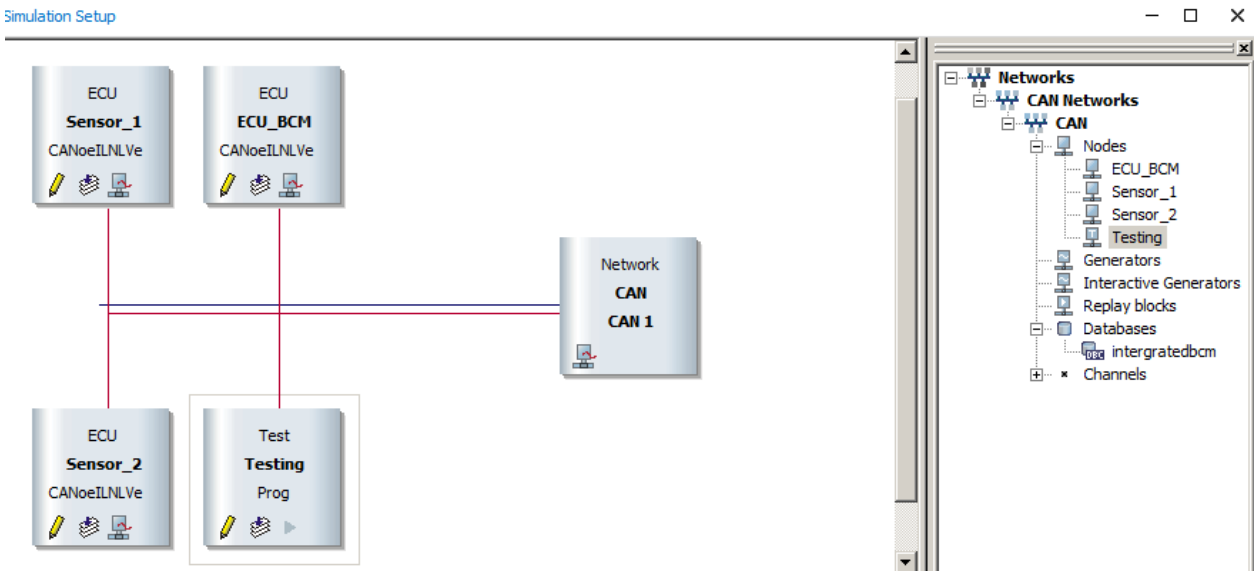
DBC File-


Vector CANdb++ Editor - C:\Users\aditis7\Desktop\BCM (1)\BCM\intergratedbcm.dbc - [Overa
File Edit View Options Window Help
Networks
intergratedbcm
Tx Messages
Signals
ECU_BCM
Sensor_1
Sensor_2
ECUs
Environment variables
Automatic_Mode_HVAC
Automatic_Mode_LED
Automatic_Mode_Seatbelt
Cooling_HVAC
Defrost_Mode
Heater_HVAC
KL15
Manual_Mode
Seatbelt_Manual
Seatbelt_Mode
Network nodes
ECU_BCM
Sensor_1
Sensor_2
Messages
BlowerSpeed (0x0)
DoorLock (0x3)
SeatHeater (0x4)
Sensor1_data (0x1)
Sensor2_data (0x2)
Signals
CurrentBlowerSpeed
Doorlock
Light_Intensity
Light_sensor
Manual_Knb_BlowerSpd
Manual_knb_IncabinLights
Manual_Knb_Seattemp
RelativeHumidity
Seat_Heater
Node
Network
Address
ECU_BCM intergratedbcm 0x0
Sensor_1 intergratedbcm 0x0
Sensor_2 intergratedbcm 0x0

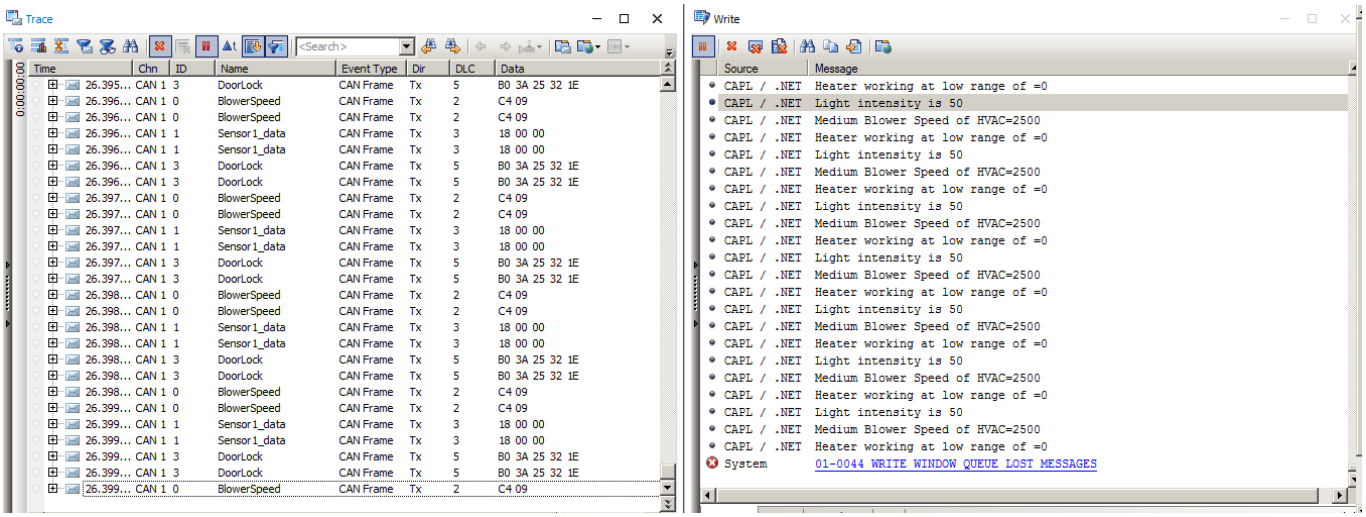

Networks
intergratedbcm
ECUs
ECU_BCM
Sensor_1
Sensor_2
Environment variables
Automatic_Mode_HVAC
Automatic_Mode_LED
Automatic_Mode_Seatbelt
Cooling_HVAC
Defrost_Mode
Heater_HVAC
KL15
Manual_Mode
Seatbelt_Manual
Seatbelt_Mode
Network nodes
ECU_BCM
Sensor_1
Sensor_2


Messages
BlowerSpeed (0)
CurrentBlowerSpeed
DoorLock (3)
Doorlock
Manual_knb_IncabinLights
Light_Intensity
Manual_Knb_BlowerSpd
Manual_Knb_Seattemp
Seat_Heater
SeatHeater (4)
Seat_Temp_Heater
Sensor1_data (1)
WeightValue
Temperature
SEAT_temp_ip
Sensor2_data (2)
RelativeHumidity
Light_sensor
Signals
CurrentBlowerSpeed
Doorlock
Light_Intensity
Light_sensor
Manual_Knb_BlowerSpd
Manual_knb_IncabinLights
Manual_Knb_Seattemp
RelativeHumidity
Seat_Heater
Seat_Temp_Heater
SEAT_temp_ip
Temperature
WeightValue

CONFIGURATION-

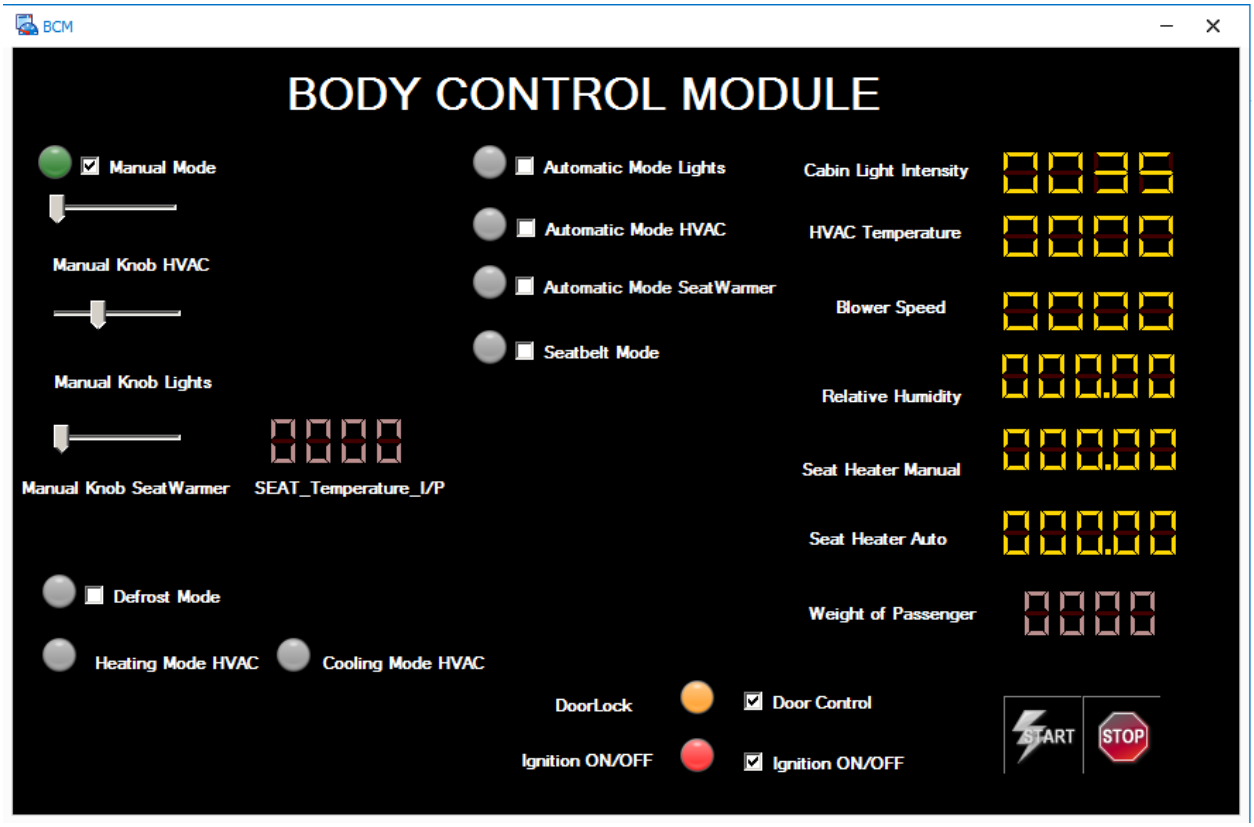


Write and Trace Window

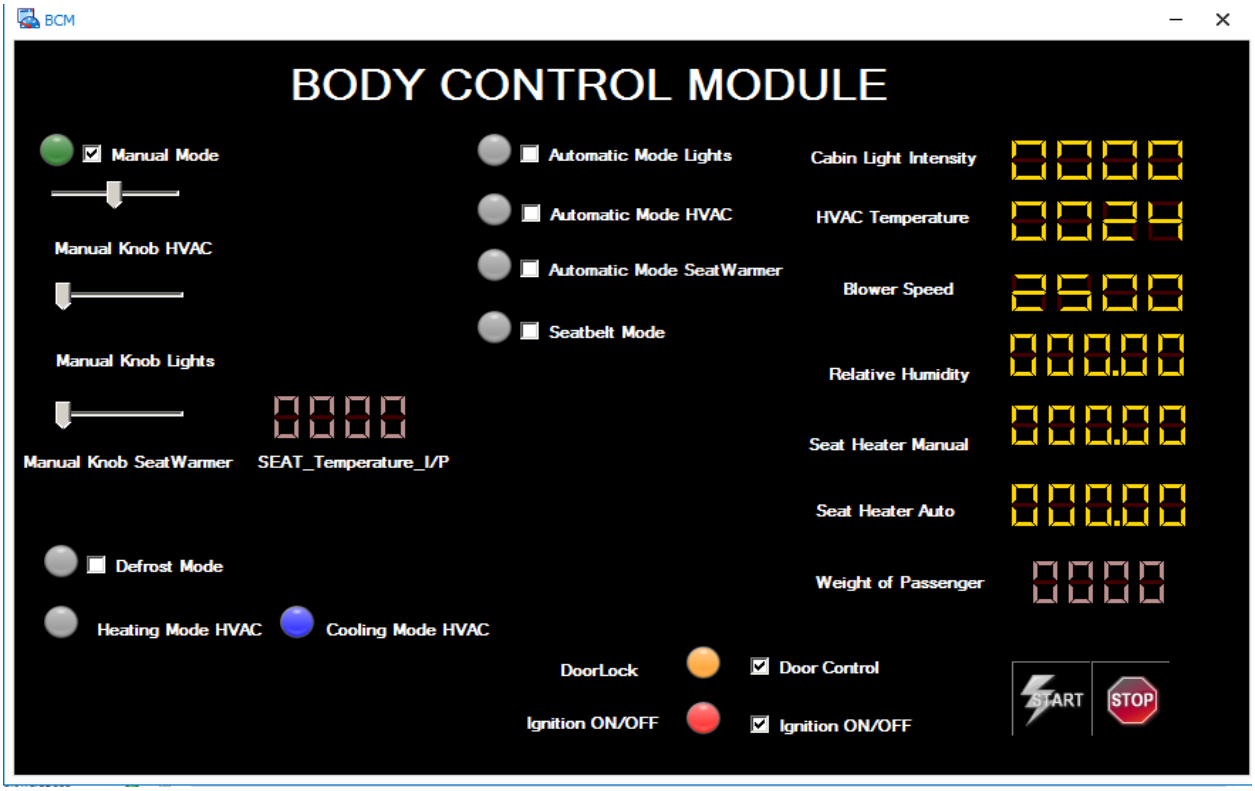


PANEL-

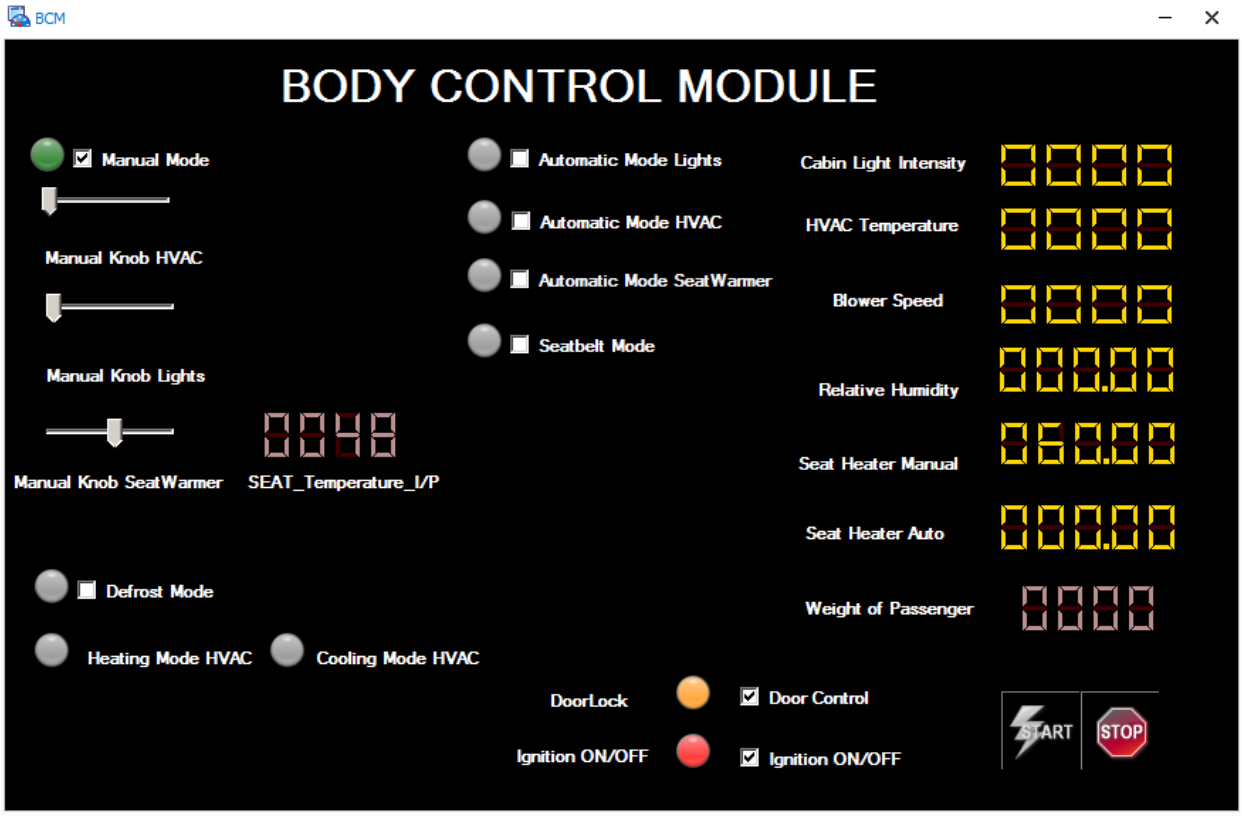
(1) Manual Mode In cabin Lights



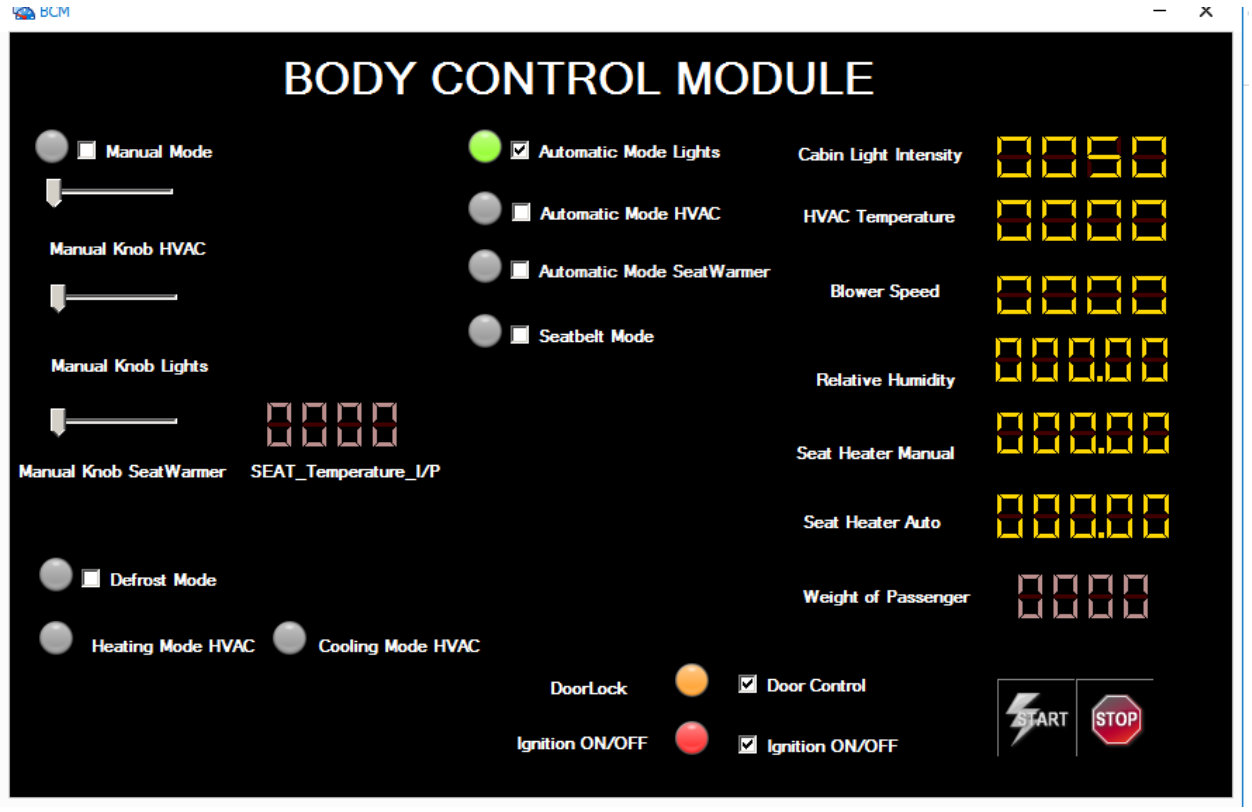
(2)Manual Mode HVAC



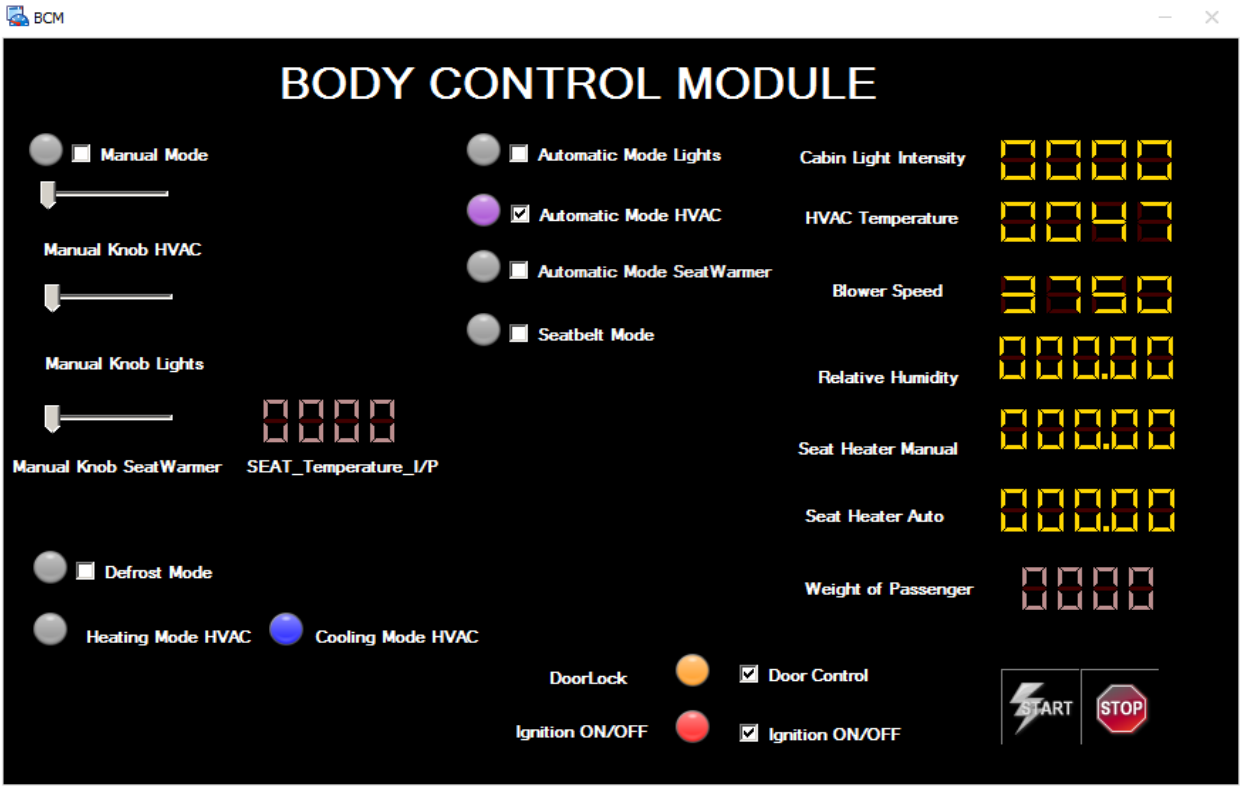
(3)Manual Mode Seating Temperature



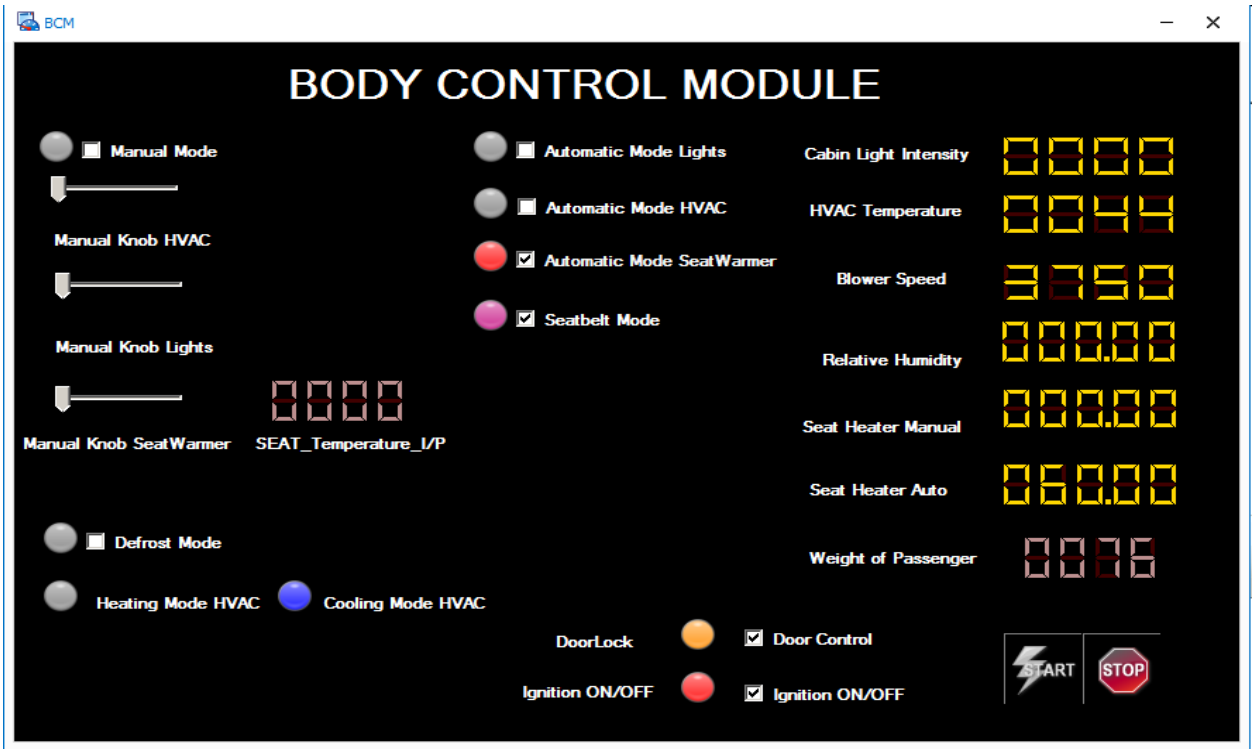
(4)Automatic Mode In cabin Lights



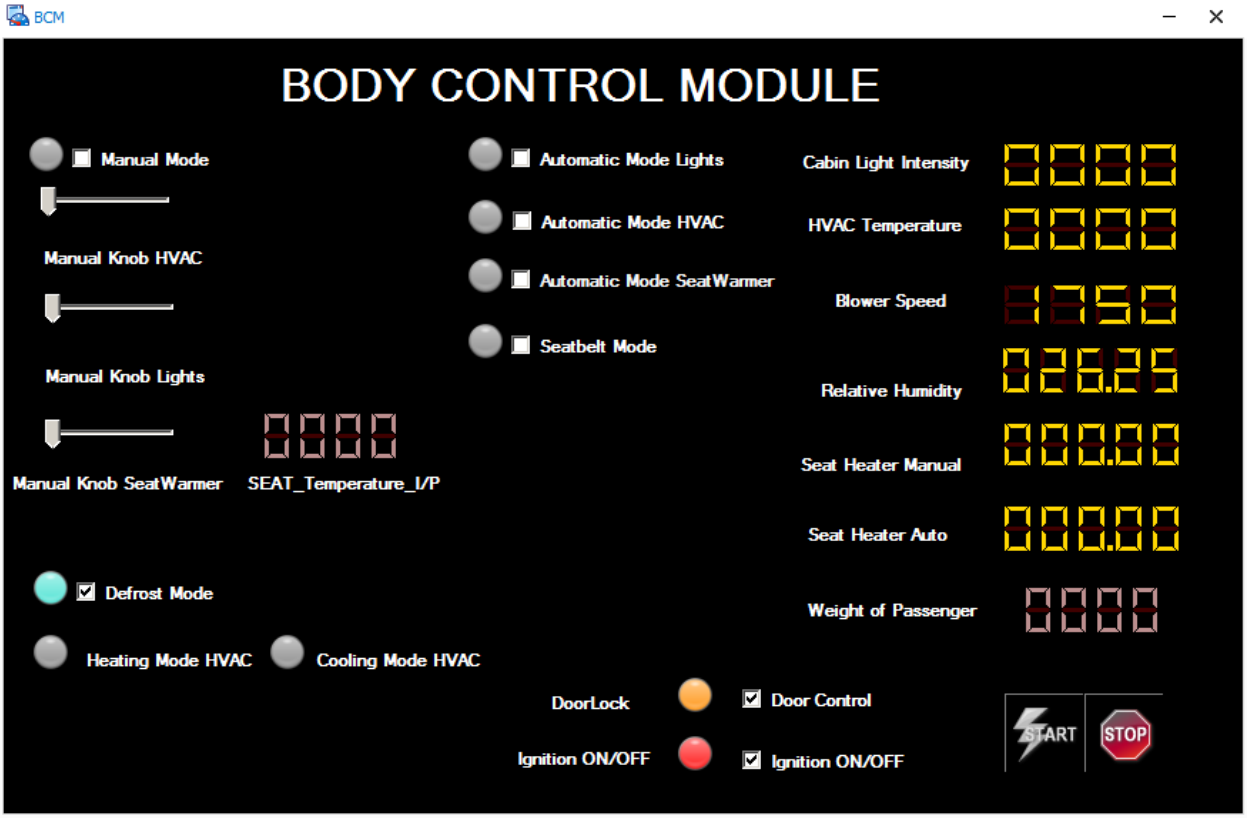
(5)Automatic Mode HVAC



(6)Seatbelt Monitoring, Automatic Temperature Control of Seat & Cabin

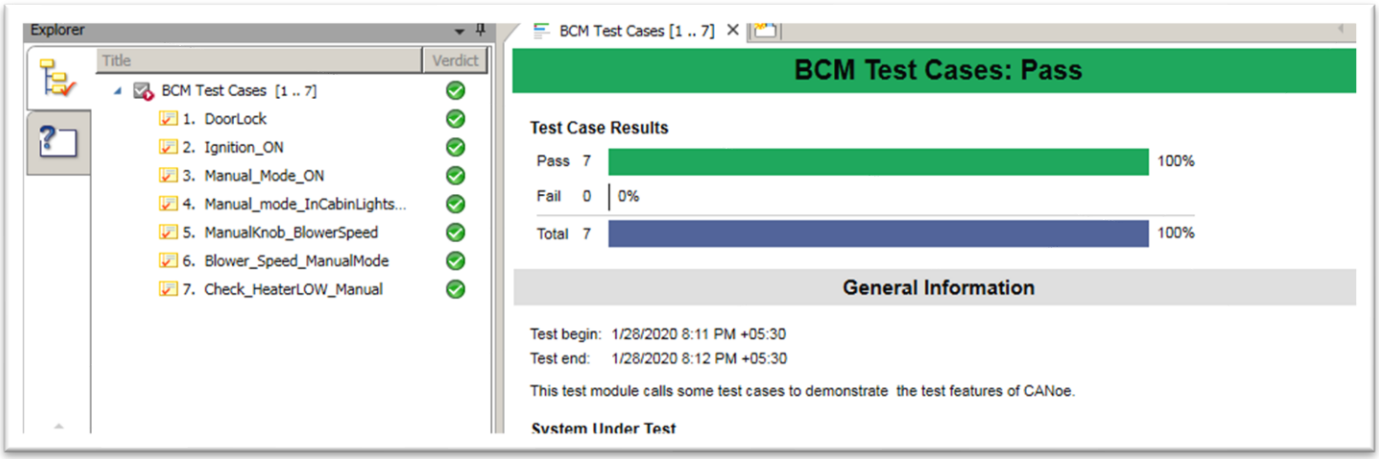


(7) Defrost Mode HVAC

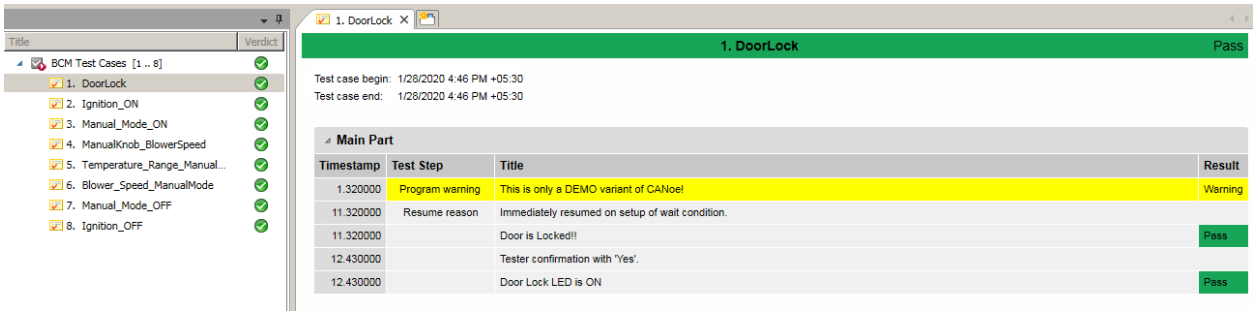


Test Report-

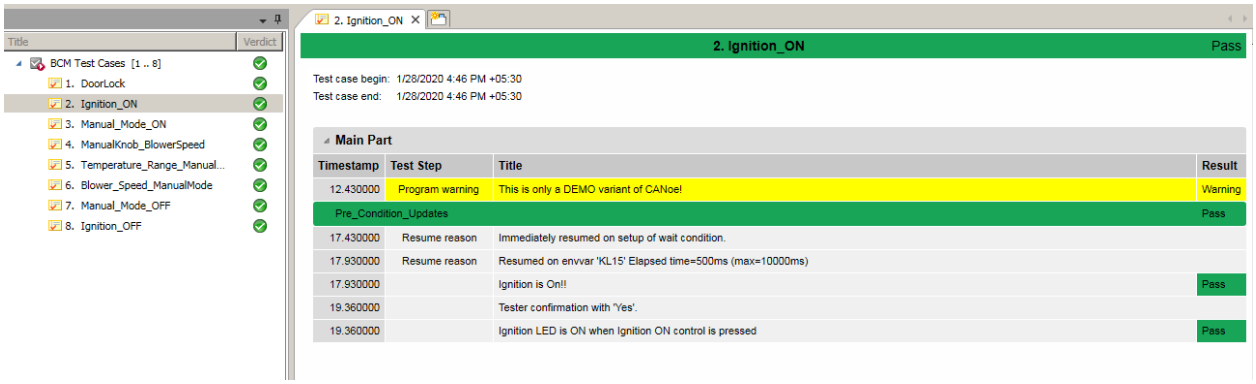
Overview-



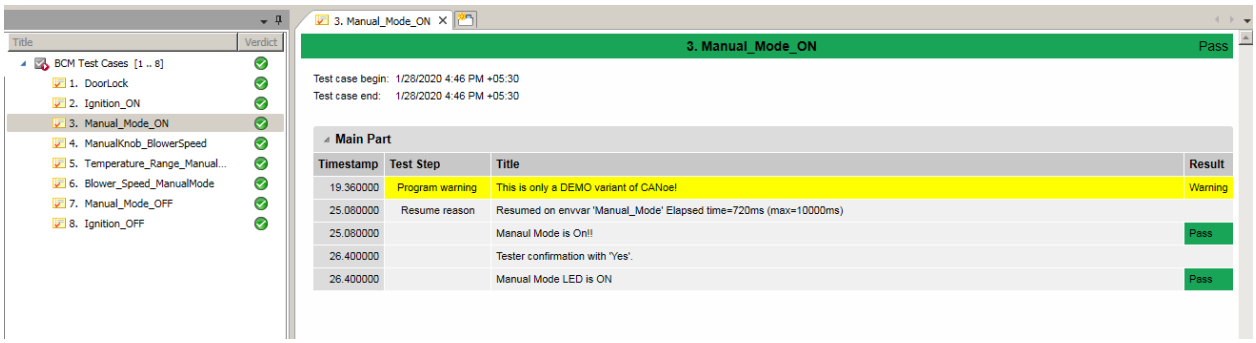
(1) Test Case 1-



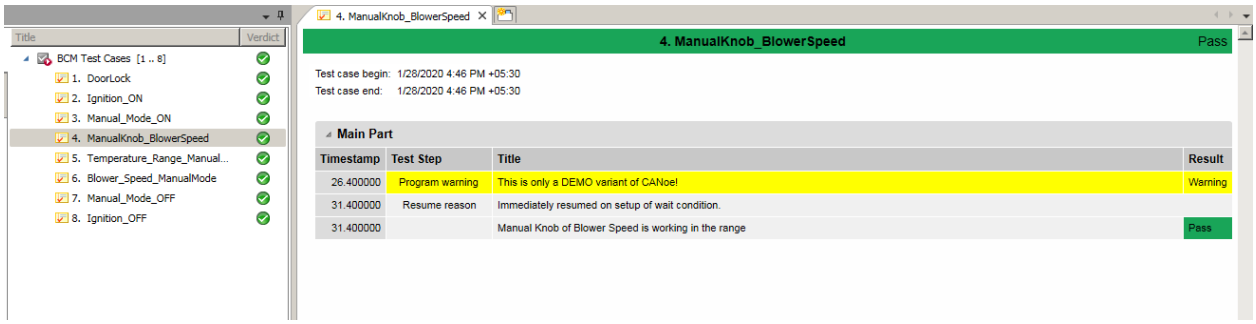
(2) Test Case 2-



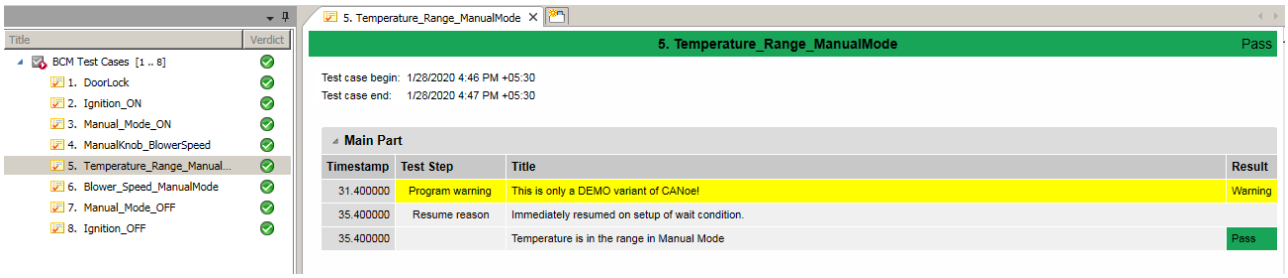
(3) Test Case 3-



(4) Test Case 4-



(5) Test Case 5-



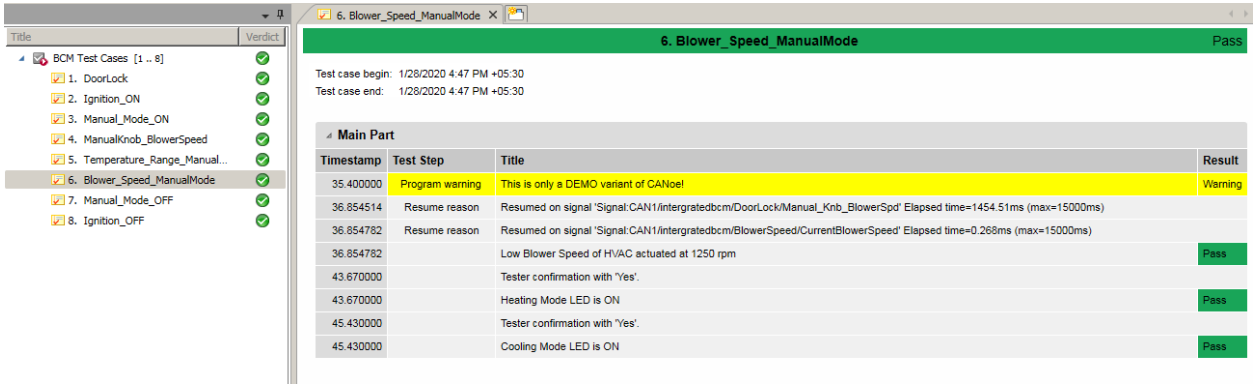
5. Temperature_Range_ManualMode — Pass

Test case begin: 1/28/2020 4:46 PM +05:30
Test case end: 1/28/2020 4:47 PM +05:30

Main Part

Timestamp	Test Step	Title	Result
31.400000	Program warning	This is only a DEMO variant of CANoe!	Warning
35.400000	Resume reason	Immediately resumed on setup of wait condition.	
35.400000		Temperature is in the range in Manual Mode	Pass

(6) Test Case 6-



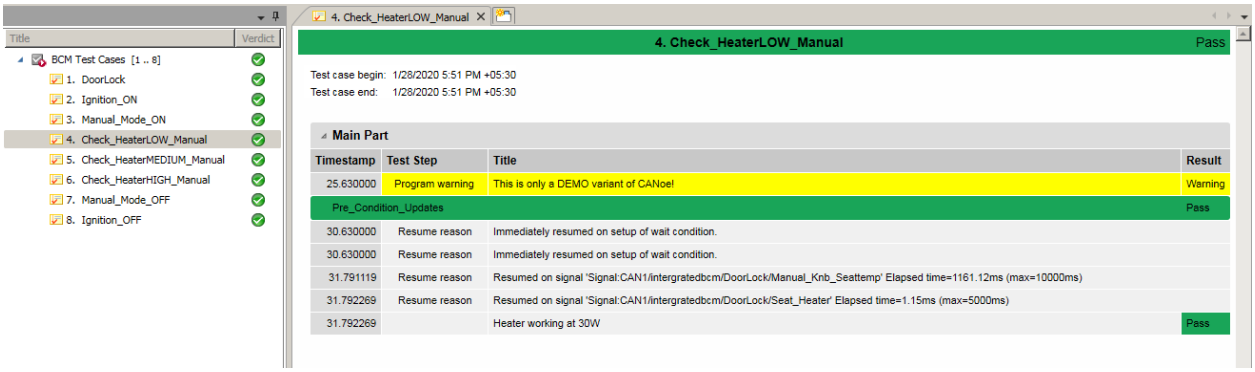
6. Blower_Speed_ManualMode — Pass

Test case begin: 1/28/2020 4:47 PM +05:30
Test case end: 1/28/2020 4:47 PM +05:30

Main Part

Timestamp	Test Step	Title	Result
35.400000	Program warning	This is only a DEMO variant of CANoe!	Warning
36.854514	Resume reason	Resumed on signal 'Signal:CAN1/intergratedbcm/DoorLock/Manual_Knb_BlowerSpd' Elapsed time=1454.51ms (max=15000ms)	
36.854782	Resume reason	Resumed on signal 'Signal:CAN1/intergratedbcm/BlowerSpeed/CurrentBlowerSpeed' Elapsed time=0.268ms (max=15000ms)	
36.854782		Low Blower Speed of HVAC actuated at 1250 rpm	Pass
43.670000		Tester confirmation with 'Yes'.	
43.670000		Heating Mode LED is ON	Pass
45.430000		Tester confirmation with 'Yes'.	
45.430000		Cooling Mode LED is ON	Pass

(7) Test Case 7-



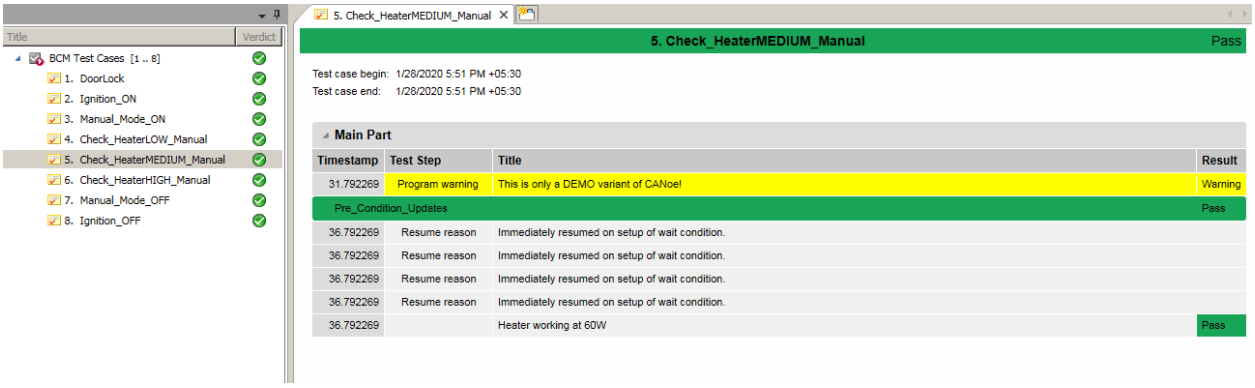
4. Check_HeaterLOW_Manual — Pass

Test case begin: 1/28/2020 5:51 PM +05:30
Test case end: 1/28/2020 5:51 PM +05:30

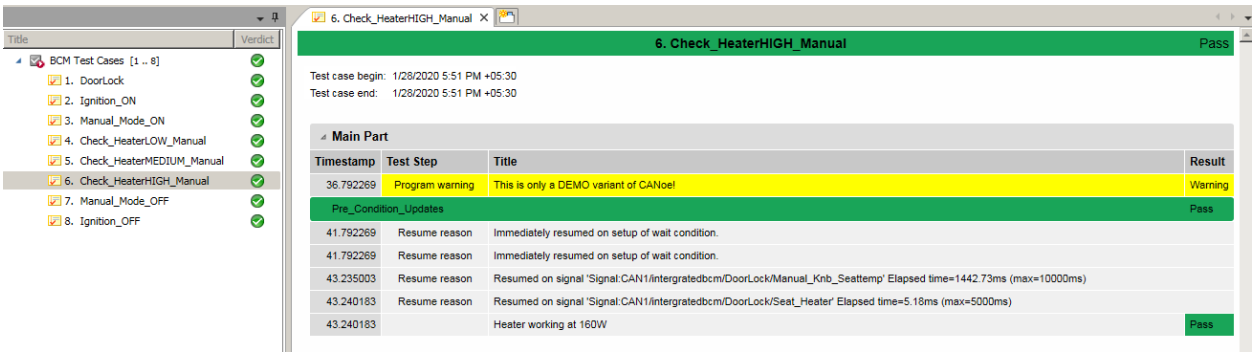
Main Part

Timestamp	Test Step	Title	Result
25.630000	Program warning	This is only a DEMO variant of CANoe!	Warning
Pre_Condition_Updates			Pass
30.630000	Resume reason	Immediately resumed on setup of wait condition.	
30.630000	Resume reason	Immediately resumed on setup of wait condition.	
31.791119	Resume reason	Resumed on signal 'Signal:CAN1/intergratedbcm/DoorLock/Manual_Knb_Seattemp' Elapsed time=1161.12ms (max=10000ms)	
31.792269	Resume reason	Resumed on signal 'Signal:CAN1/intergratedbcm/DoorLock/Seat_Heater' Elapsed time=1.15ms (max=5000ms)	
31.792269		Heater working at 30W	Pass

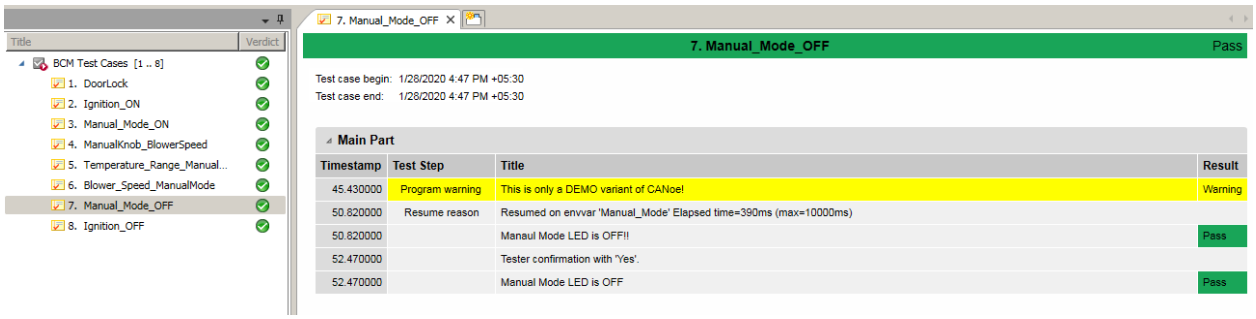
(8) Test Case 8-



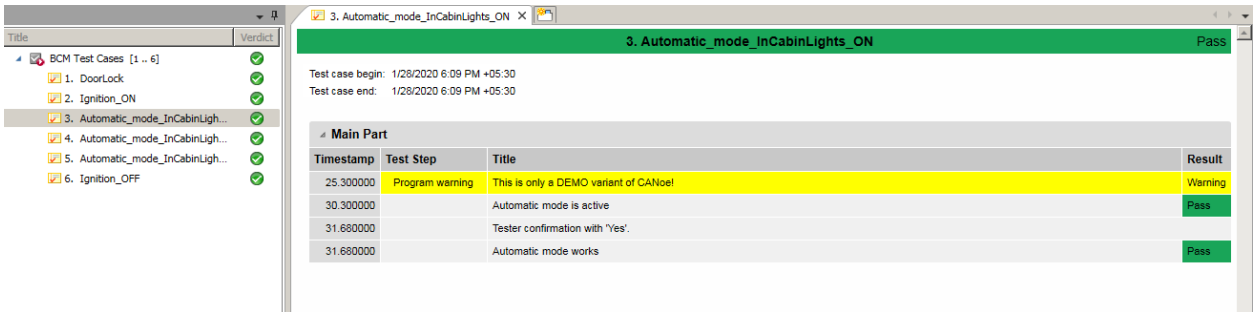
(9) Test Case 9-



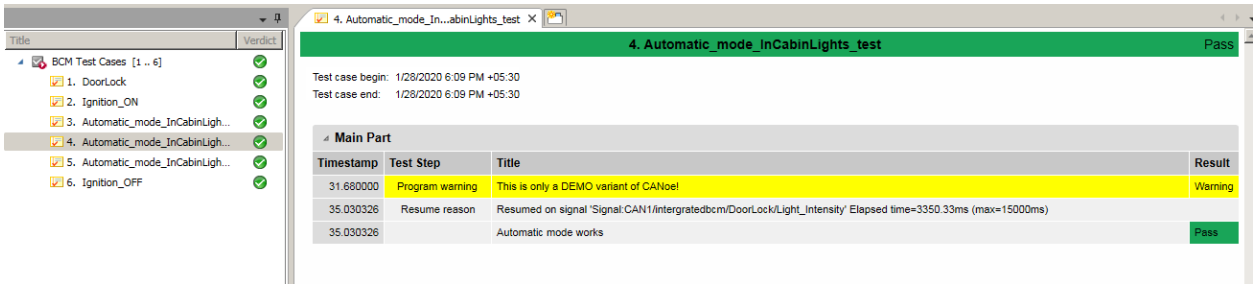
(10) Test Case 10-



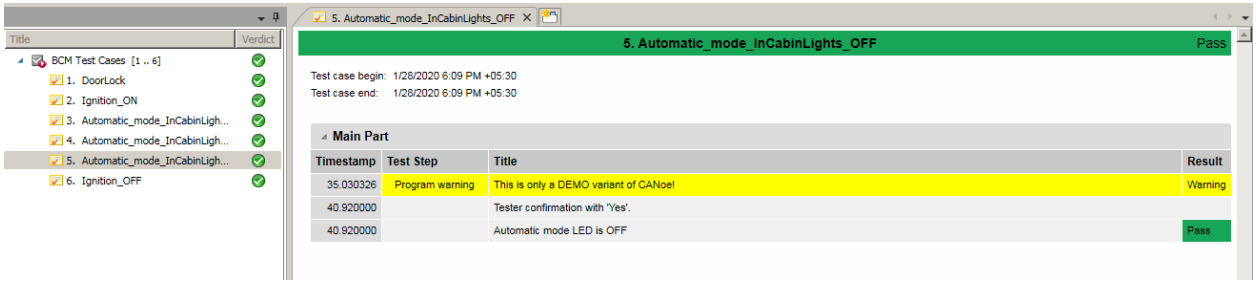
(11) Test Case 11-



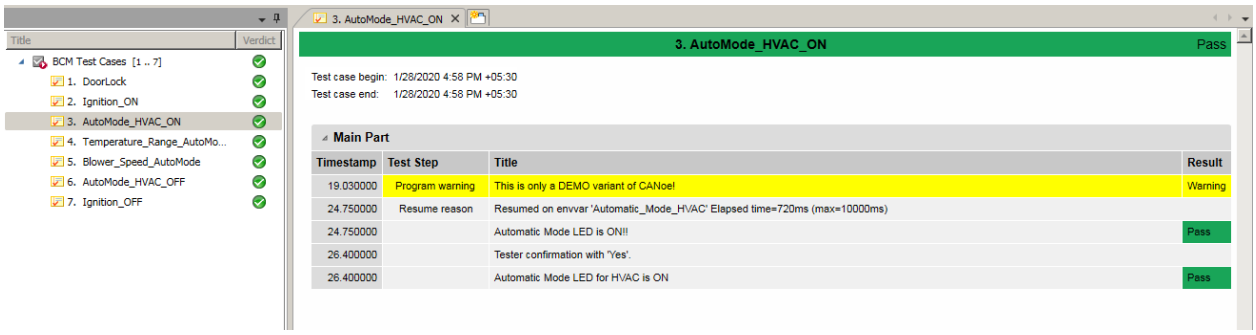
(12) Test Case 12-



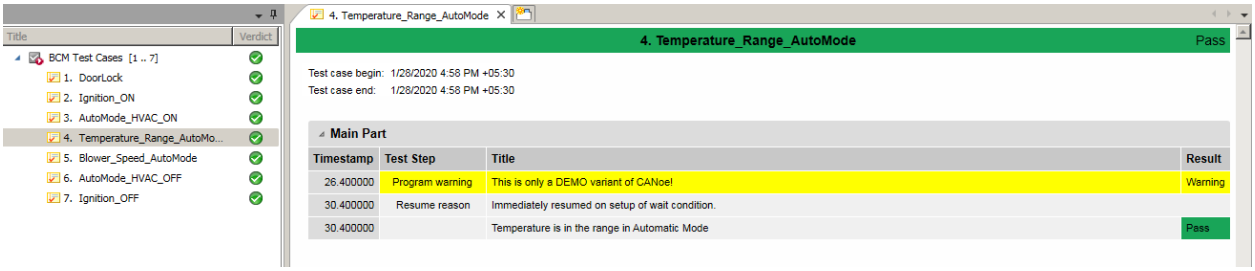
(13) Test Case 13-



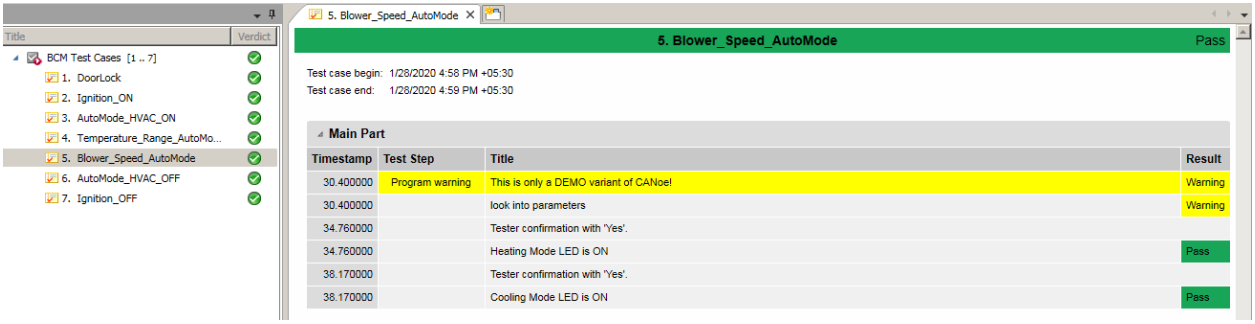
(14) Test Case 14-



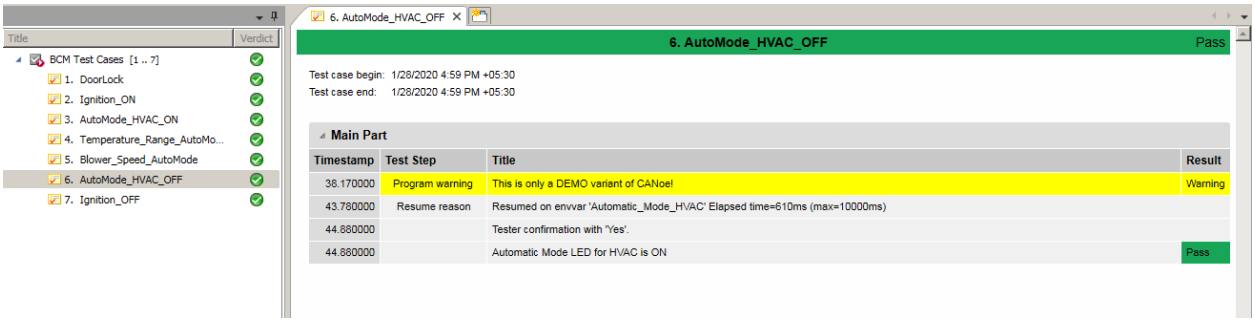
(15) Test Case 15-



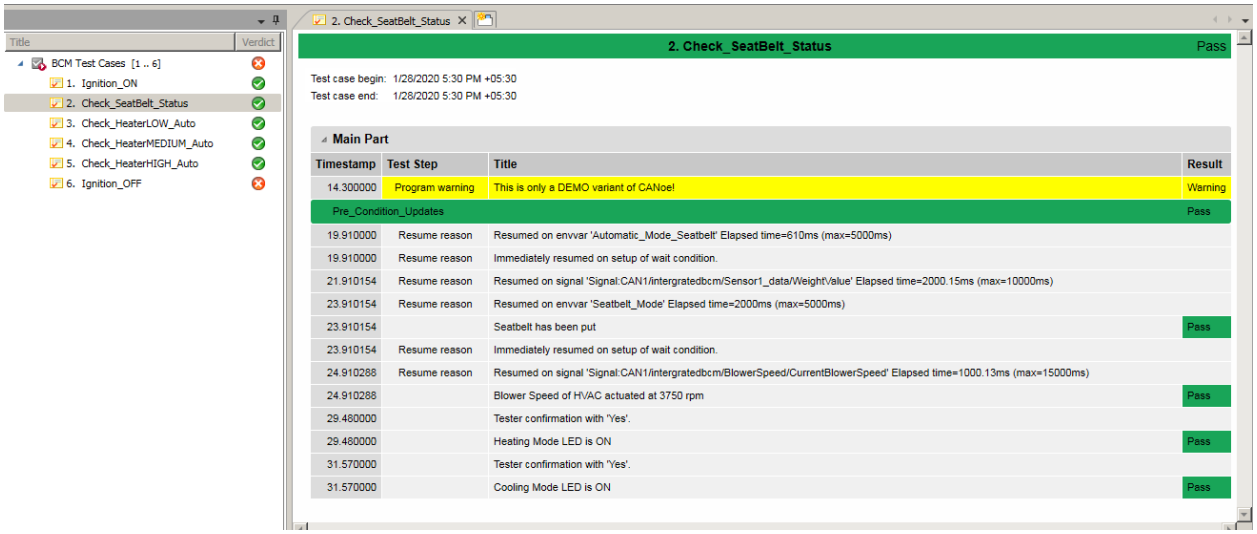
(16) Test Case 16-



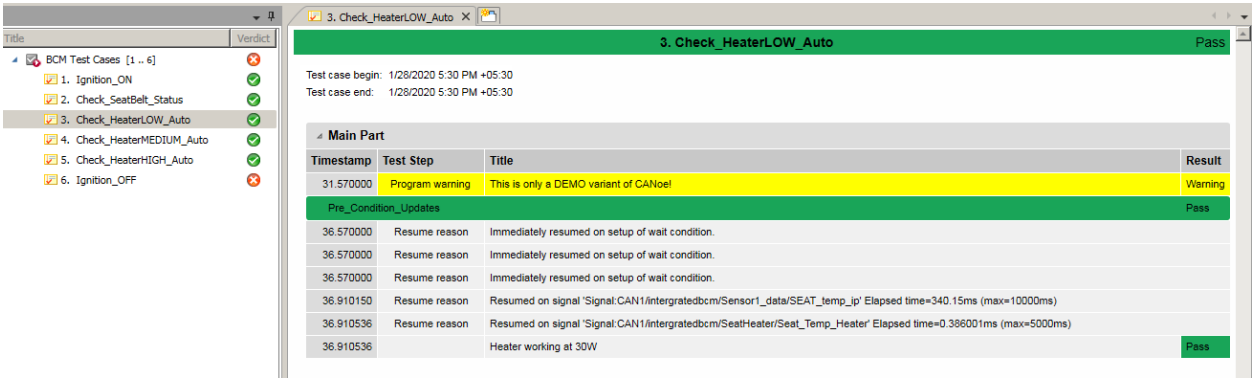
(17) Test Case 17-



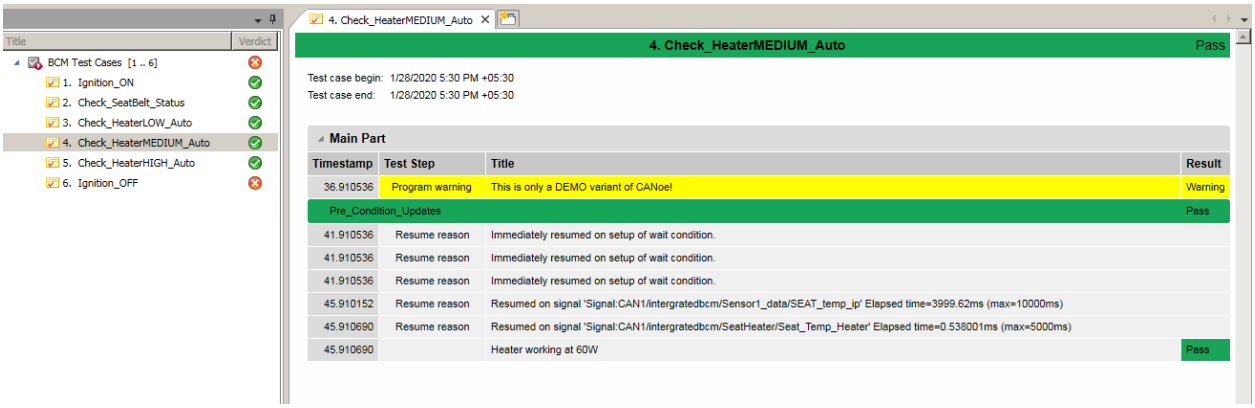
(18) Test Case 18-



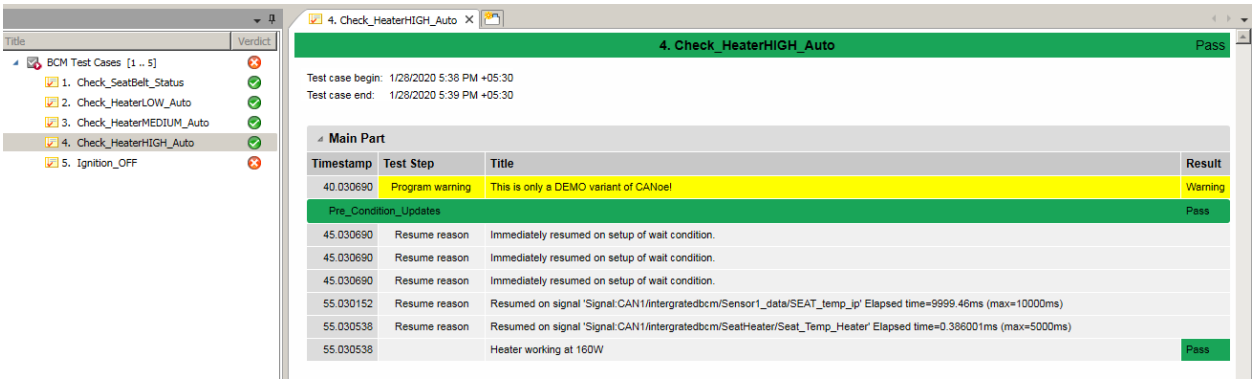
(19) Test Case 19-



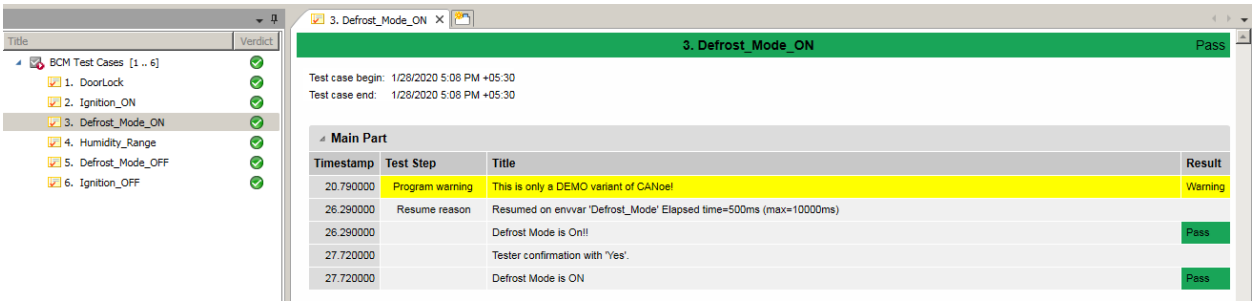
(20)Test Case 20-



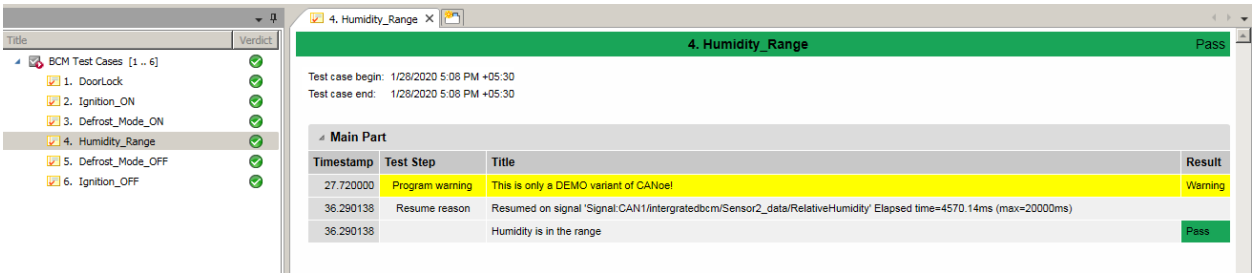
(20) Test Case 21-



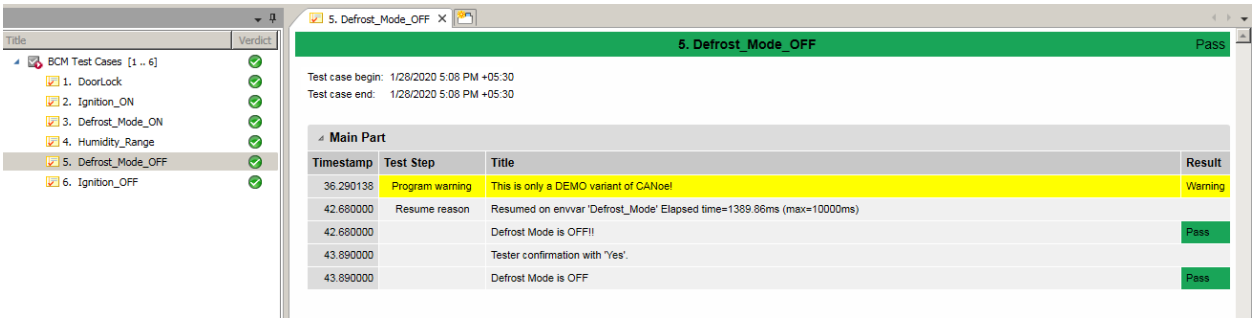
(22) Test Case 22-



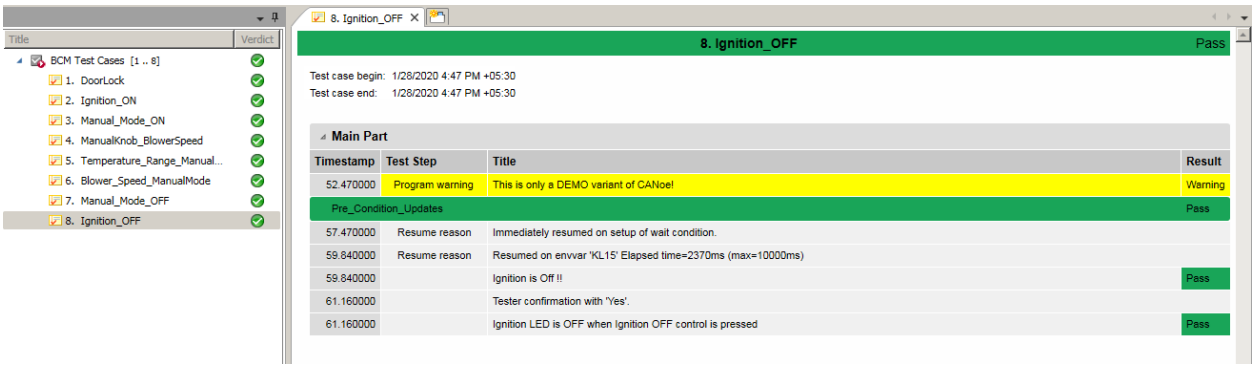
(23) Test Case 23-



(24) Test Case 24-



(25) Test Case 25-



APPENDIX-

Code for Sensor_1

```
Includes
{

}

Variables
{
 Timer TIMER1;
 Timer TIMER2;
 int Weight_Loc;

  int temp;
}
on start
{
 write("Temperature Sensor is processing!!");
}
on envVar KL15
{
  if(@KL15==1)
  {
    write("Ignition is On");
  }
  else if(@KL15==0)
  {
    write("Ignition is Off");
    $Temperature=0;
    cancelTimer(TIMER1);
    }
  }
on envVar Automatic_Mode_Seatbelt{
  if(@Automatic_Mode_Seatbelt==1){
write("Automatic mode for seat is ON");

  setTimer(TIMER2,2);
}else if(@Automatic_Mode_Seatbelt==0){
  write("Automatic mode for seat is OFF");
cancelTimer(TIMER2);
}
}
on envVar Automatic_Mode_HVAC{
if(@Automatic_Mode_HVAC==1){
write("Automatic mode for HVAC is ON");
  setTimer(TIMER1,2);
}else if(@Automatic_Mode_HVAC==0){
  write("Automatic mode for HVAC is OFF");
cancelTimer(TIMER1);
}
}
on envVar Seatbelt_Mode{
if(@Seatbelt_Mode==1){
write("Automatic mode for HVAC is ON");
  setTimer(TIMER1,2);
}else if(@Seatbelt_Mode==0){
  write("Automatic mode for HVAC is OFF");
cancelTimer(TIMER1);
```

```
}
}
on timer TIMER1
{
  if($Doorlock==1 && @Manual_Mode==0){
    if(@KL15==1 && (@Automatic_Mode_HVAC==1 || @Seatbelt_Mode==1))
  {
    write("timer 1 has started");
    $Temperature=random(50);
    temp=$Temperature;
    write("Temperature=%d",temp);
    setTimer(TIMER1,2);
  }
  }
  else if( $Doorlock==0 || @KL15==0 || @Automatic_Mode_HVAC==0)
  {
    $Temperature=0;
    cancelTimer(TIMER1);
  }
}
on timer TIMER2
{
  if($Doorlock==1 && @Manual_Mode==0){
    if(@KL15==1 && @Automatic_Mode_Seatbelt==1)
  {
    write("timer 2 has started");
    $WeightValue=40+random(60);
    Weight_Loc=$WeightValue;
    write("Weight= %d",Weight_Loc);
    $SEAT_temp_ip=30+random(30);

    //setTimer(TIMER2,2);
  }
  }
  else if( $Doorlock==0 || @KL15==0 || @Automatic_Mode_Seatbelt==0)
  {
    $WeightValue=0;
    cancelTimer(TIMER2);
    $SEAT_temp_ip=30+random(30);

  }
  setTimer(TIMER2,3);
}

on stopMeasurement
{
  $Temperature=0;
  $WeightValue=0;
  cancelTimer(TIMER1);
  cancelTimer(TIMER2);
}
```

Code for Sensor_2-

```
includes
{

}

variables
{
  timer TIMER2;
  timer TIMER1;
  int rh;
}
on start
{
 write("Relative Humidity Sensor is processing!!");
}

on envVar KL15
{
  if(@KL15==1)
  {
    write("Ignition is On");
  }
  else
  {
    write("Ignition is Off");
    $Temperature=0;
    cancelTimer(TIMER1);
    cancelTimer(TIMER2);
  }
}
on envVar Automatic_Mode_LED{
if(@Automatic_Mode_LED==1){
 write("Automatic mode for INCabin Lights is ON");
  setTimer(TIMER1,5);
}else if(@Automatic_Mode_LED==0){
  write("Automatic mode for InCabin Lights is OFF");
cancelTimer(TIMER1);
}
}
on envVar Defrost_Mode{
if(@Defrost_Mode==1){
write("Automatic mod for HVAC is ON");
  setTimer(TIMER2,5);
}else if(@Defrost_Mode==0){
  write("Automatic mod for HVAC is OFF");
cancelTimer(TIMER2);
}
}
on timer TIMER1{
if($Doorlock==1)
  {
  if((@KL15==1)&&(@Automatic_Mode_LED==1))// If Ignition is On and Defrost Mode is
On
  {
    write("timer 1 has started");
    $Light_sensor=random(100);
    rh=$Light_sensor;
    write("Brightness level is=%d",rh);
    setTimer(TIMER1,5);//recursive timer
```

```
  }
  else if( $Doorlock==0 || @KL15==0 || @Automatic_Mode_LED==0)
  {
    $Light_sensor=0;
  }
}
}
on timer TIMER2
{
  if($Doorlock==1)
  {
  if((@KL15==1)&&(@Defrost_Mode==1))// If Ignition is On and Defrost Mode is On
  {
    write("timer 1 has started");
    $RelativeHumidity=random(70);
    rh=$RelativeHumidity;
    write("Relative Humidity=%d",rh);
    setTimer(TIMER2,5);//recursive timer
  }
  else if( $Doorlock==0 || @KL15==0 || @Defrost_Mode==0)
  {
    $RelativeHumidity=0;
  }
}
}
on stopMeasurement
{
  $RelativeHumidity=0;
  $Light_Intensity=0;
  cancelTimer(TIMER2);
  cancelTimer(TIMER1);
}
```

Code for ECU_BCM-

```
includes
{

}

variables
{
  message DoorLock Doors;
  message Sensor1_data Sens1;
  int var_LED;
  int var_HVAC;
  int var_SEAT;
  int var;
  int blower;
  int Weight_Loc;
  int Temp_Loc;
  timer Tmr;

}
on start{
write("CANoe has started");
}
on message DoorLock{
  if($Doorlock==1){
```

```
    //write("Doors are locked");
    if(@KL15==1){
    //write("Ignition is ON");
      if(@Manual_Mode==1){
        var_LED=$Manual_knb_IncabinLights;
        var_HVAC=$Manual_Knb_BlowerSpd;
        var_SEAT=$Manual_Knb_Seattemp;
        if(var_LED>0 && var_LED<45){
        write("Light intensity is 35%");
          setSignal(Light_Intensity,35);
        }else if(var_LED>=45 && var_LED<85){
        write("Light intensity is 50%");
          setSignal(Light_Intensity,50);
        }else if(var_LED>=85 && var_LED<100){
        write("Light intensity is 75%");
          setSignal(Light_Intensity,75);
        }else if(var_LED>=100){
          write("Light intensity is 100%");
          setSignal(Light_Intensity,100);
        }
        if(var_HVAC>0 && var_HVAC<45){
          $CurrentBlowerSpeed=1250;
          var=$CurrentBlowerSpeed;
          $Temperature=30;
          @Cooling_HVAC=1;
          @Heater_HVAC=0;
        write("Low Blower Speed of HVAC=%d",var);
        }else if(var_HVAC>=45 && var_HVAC<85){
          $CurrentBlowerSpeed=2500;
          var=$CurrentBlowerSpeed;
          $Temperature=24;
          @Cooling_HVAC=1;
          @Heater_HVAC=0;
        write("Medium Blower Speed of HVAC=%d",var);
        }else if(var_HVAC>=85 && var_HVAC<100){
          $CurrentBlowerSpeed=3750;
          var=$CurrentBlowerSpeed;
          $Temperature=16;
          @Cooling_HVAC=0;
          @Heater_HVAC=1;
        write("High Blower Speed of HVAC=%d",var);
        }
        if(var_SEAT>=30 && var_SEAT<40){
          $Seat_Heater=30;
          var=$Seat_Temp_Heater;
        write("Heater working at low range of =%d",var);
        }else if(var_SEAT>=40 && var_SEAT<50){
          $Seat_Heater=60;
          var=$Seat_Temp_Heater;
        write("Heater working at mid range of =%d",var);
        }else if(var_SEAT>=50 && var_SEAT<60){
          $Seat_Temp_Heater=160;
          var=$Seat_Heater;
        write("Heater working at high range of = %d",var);
        }else{
        write("This range of temperature is not permitted");
          $Seat_Heater=0;
        }
    }
}
```

```
}
}
on message Sensor1_data{
  if($Doorlock==1 && @Manual_Mode==0){
if(@KL15==1 && @Automatic_Mode_HVAC==1)
    {
      if($Temperature>=0 && $Temperature<10)
       {
        $CurrentBlowerSpeed=3750;
        blower=$CurrentBlowerSpeed;
        write("Blower Speed=%d",blower);
        @Heater_HVAC=1;
        @Cooling_HVAC=0;
       }
      if($Temperature>=10 && $Temperature<20)
       {
        $CurrentBlowerSpeed=2500;
        blower=$CurrentBlowerSpeed;
        write("Blower Speed=%d",blower);
        @Heater_HVAC=1;
        @Cooling_HVAC=0;
       }
      if($Temperature>=20 && $Temperature<30)
       {
        $CurrentBlowerSpeed=1750;
        blower=$CurrentBlowerSpeed;
        write("Blower Speed=%d",blower);
        @Cooling_HVAC=1;
        @Heater_HVAC=0;
       }
      if($Temperature>=30 && $Temperature<40)
        {
        $CurrentBlowerSpeed=2500;
        blower=$CurrentBlowerSpeed;
        write("Blower Speed=%d",blower);
         @Cooling_HVAC=1;
          @Heater_HVAC=0;
        }
      if($Temperature>=40 && $Temperature<=50)
       {
        $CurrentBlowerSpeed=3750;
        blower=$CurrentBlowerSpeed;
        write("Blower Speed=%d",blower);
        @Cooling_HVAC=1;
        @Heater_HVAC=0;
      }
    }
    //
   if((@KL15==1)&&(@Automatic_Mode_Seatbelt==1))
   {
    write("Monitoring seat occupancy"); //checks for seat occupancy
    Weight_Loc=$WeightValue;
    Temp_Loc= $SEAT_temp_ip;
  //Temp_Loc= $SEAT_temp_ip
    if ((Weight_Loc >=40 && Weight_Loc<= 115))
    {
      write("Seat is occupied");

      setTimer(Tmr,2);
```

```
      //@Seatbelt_Mode=1;
      if(@Seatbelt_Mode==1){
       if($Temperature>=0 && $Temperature<10)
       {
        $CurrentBlowerSpeed=3750;
        blower=$CurrentBlowerSpeed;
        write("Blower Speed=%d",blower);
        @Heater_HVAC=1;
        @Cooling_HVAC=0;

       }
      if($Temperature>=10 && $Temperature<20)
       {
        $CurrentBlowerSpeed=2500;
        blower=$CurrentBlowerSpeed;
        write("Blower Speed=%d",blower);
        @Heater_HVAC=1;
        @Cooling_HVAC=0;
       }
      if($Temperature>=20 && $Temperature<30)
       {
        $CurrentBlowerSpeed=1750;
        blower=$CurrentBlowerSpeed;
        write("Blower Speed=%d",blower);
        @Cooling_HVAC=1;
        @Heater_HVAC=0;
       }
      if($Temperature>=30 && $Temperature<40)
        {
        $CurrentBlowerSpeed=2500;
        blower=$CurrentBlowerSpeed;
        write("Blower Speed=%d",blower);
         @Cooling_HVAC=1;
          @Heater_HVAC=0;
        }
      if($Temperature>=40 && $Temperature<=50)
      {
        $CurrentBlowerSpeed=3750;
        blower=$CurrentBlowerSpeed;
        write("Blower Speed=%d",blower);
        @Cooling_HVAC=1;
        @Heater_HVAC=0;
      }
      }
      //write("Seat Belt can be put on");  //Seat Belt status monitoring
      write("Taking desired temperature range for temperature control");
      //$SEAT_temp_ip=30+random(30);
      //Temp_Loc= $SEAT_temp_ip;
      if ((Temp_Loc >30) && (Temp_Loc<=40))
    {
      write("Heater working at lower range of 30W");  //Heater working at low range
      $Seat_Temp_Heater=30;

    }
    else if((Temp_Loc >40) && (Temp_Loc<=50))
    {
      write("Heater working at medium range of 60W");  //Heater working at medium
range
      $Seat_Temp_Heater=60;
    }
```

```
    else if((Temp_Loc >50) && (Temp_Loc<=60))
    {
      write("Heater working at high range of 160W");  //Heater working at high range
      $Seat_Temp_Heater=160;
    }
    else
    {
      write("This range of temperature is not permitted");
      $Seat_Temp_Heater=0;
    }

    }
   else if((Weight_Loc <40) && (Weight_Loc> 115))
    {
      write("Seat is not occupied or weight more than human");
      @Seatbelt_Mode=0;
    }
  }
  else
  {
    write("Error in reading since ignition is OFF");
    @Seatbelt_Mode=0;
  }
  }
  //

}



on timer Tmr
{
  Weight_Loc=$WeightValue;
  if((@KL15 == 1) && (Weight_Loc >=40 && Weight_Loc<= 115))
  {
    @Seatbelt_Mode =1;

    write("Seatbelt can be put on");
  }
  else
  {
    write("Ignition is OFF");
    @Seatbelt_Mode =0;

  }
}


on message Sensor2_data
{
  if($Doorlock==1 && @KL15==1 && @Automatic_Mode_LED==1){
  if($Light_sensor>0 && $Light_sensor<45){
        write("Light intensity is 35%");
          setSignal(Light_Intensity,35);
        }else if($Light_sensor>=45 && $Light_sensor<85){
        write("Light intensity is 50%");
          setSignal(Light_Intensity,50);
        }else if($Light_sensor>=85 && $Light_sensor<100){
        write("Light intensity is 75%");
          setSignal(Light_Intensity,75);
```

```
        }else if($Light_sensor>=100){
          write("Light intensity is 100%");
          setSignal(Light_Intensity,100);
        }
  }
  if($Doorlock==1 && @Manual_Mode==0 && @Automatic_Mode_HVAC==0)
  {
    if(@KL15==1 && @Defrost_Mode==1)
    {
      if($RelativeHumidity>=0 && $RelativeHumidity<20)
       {
        $CurrentBlowerSpeed=1500;
        blower=$CurrentBlowerSpeed;
        write("Blower Speed=%d",blower);
        @Heater_HVAC=0;
        @Cooling_HVAC=0;
       }
      if($RelativeHumidity>=20 && $RelativeHumidity<40)
       {
        $CurrentBlowerSpeed=1750;
        blower=$CurrentBlowerSpeed;
        write("Blower Speed=%d",blower);
        @Heater_HVAC=0;
        @Cooling_HVAC=0;
       }
      if($RelativeHumidity>=40 && $RelativeHumidity<60)
       {
        $CurrentBlowerSpeed=2500;
        blower=$CurrentBlowerSpeed;
        write("Blower Speed=%d",blower);
        @Heater_HVAC=0;
        @Cooling_HVAC=0;
       }
      if($RelativeHumidity>=60 && $RelativeHumidity<80)
        {
        $CurrentBlowerSpeed=3500;
        blower=$CurrentBlowerSpeed;
        write("Blower Speed=%d",blower);
        @Heater_HVAC=0;
        @Cooling_HVAC=0;
        }
      if($RelativeHumidity>=80 && $RelativeHumidity<=100)
      {
        $CurrentBlowerSpeed=3750;
        blower=$CurrentBlowerSpeed;
        write("Blower Speed=%d",blower);
        @Heater_HVAC=0;
        @Cooling_HVAC=0;
      }
  }
}
}

on stopMeasurement{
write("CANoe has stopped");
}
```

TEST CODE-

```
includes
{

}

variables
{
  int rVal;
  int rVal1;
  int rVal2;
  int var;
  int var1;
  int var2;
  int flag=0;
  int signalvalue;
}
testfunction Pre_Condition_Updates()
{
switch(flag){
    case 1:
      @KL15=0;//Ignition OFF
      break;
    case 2:
      @KL15=1;//Ignition ON
      break;
}
}
testcase DoorLock(){
  testWaitForTimeout(5000);
   write("In test case Door Lock ");
  //data verification of Door Lock ON LED
  testWaitForTimeout(5000);
  var=testWaitForSignalMatch(Doorlock,1,10000);
  if(var==1)
  {
    if($Doorlock==1)
    {
      testStepPass("Door is Locked!!");
    }
    else if($Doorlock==0)
    {
      testStepFail("Door is not Locked!!");
    }
  }
  else
  {
    testStepWarning("Check the Timing parameter");
  }
  //Visual Confirmation of Door Lock ON LED
  rVal2=testWaitForTesterConfirmation("Door Lock ON LED:Is Door Lock LED ON?");
  if(rVal2==1)//if yes is clicked!
  {
    testStepPass("Door Lock LED is ON");
  }
  else if(rVal2 == 2)//if No is clicked!
  {
    testStepFail("Door Lock LED is OFF!!");
  }
  else if (rVal2 == 0)//Time out has occured!!
```

```
  {
    testStepWarning("No verification is done from the tester");
  }

}
testcase Ignition_ON()//To check Ignition On
{
  flag++;
  if(flag==1)
  {
    Pre_Condition_Updates();
  write("In test case Ignition_ON");
  //data verification of Ignition ON LED
  testWaitForTimeout(5000);
  rVal=testWaitForSignalMatch(KL15,0,10000);
  if(rVal==1)
    rVal=testWaitForEnvVar(KL15,10000);
  if(rVal==1)
  {
    if(@KL15==1)
    {
      testStepPass("Ignition is On!!");
    }
    else if(@KL15==0)
    {
      testStepFail("Ignition is Off!!");
    }
  }
  else
  {
    testStepWarning("Check the Timing parameter");
  }
  //Visual Confirmation of Ignition ON LED
  rVal2=testWaitForTesterConfirmation("Ignition ON LED:Is Ignition LED ON?");
  if(rVal2==1)//if yes is clicked!
  {
    testStepPass("Ignition LED is ON when Ignition ON control is pressed");
  }
  else if(rVal2 == 2)//if No is clicked!
  {
    testStepFail("Ignition LED remained OFF!!");
  }
  else if (rVal2 == 0)//Time out has occured!!
  {
    testStepWarning("No verification is done from the tester");
  }
}
}
testcase Manual_Mode_ON()//To check Manual Mode for ALL
{
  write("In test case Manual Mode ");
  //data verification of Manual Mode ON LED
  testWaitForTimeout(5000);
  var=testWaitForEnvVar(Manual_Mode,10000);
  if(var==1)
  {
    if((@KL15==1) && ($Doorlock==1) && (@Manual_Mode==1))
    {
      testStepPass("Manaul Mode is On!!");
    }
```

```
    else if((@KL15==1) && ($Doorlock==1) && (@Manual_Mode==0))
    {
      testStepFail("Manual Mode is OFF!!");
    }
    else if(@KL15==0)
    {
      testStepFail("Ignition is OFF");
    }
    else if($Doorlock==0)
    {
      testStepFail("Door is not Locked");
    }
  }
  else
  {
    testStepWarning("Check the Timing parameter");
  }
  //Visual Confirmation of Manual Mode ON LED
  rVal2=testWaitForTesterConfirmation("Manual Mode ON LED:Is Manual Mode LED ON?");
  if(rVal2==1)//if yes is clicked!
  {
    testStepPass("Manual Mode LED is ON");
  }
  else if(rVal2 == 2)//if No is clicked!
  {
    testStepFail("Manual Mode LED is still OFF!!");
  }
  else if (rVal2 == 0)//Time out has occured!!
  {
    testStepWarning("No verification is done from the tester");
  }
}
testcase Manual_mode_InCabinLights_test(){
testWaitForTimeout(5000);
  if(@KL15==1 && $Doorlock==1){
    if(@Manual_Mode==1){
    if(testWaitForSignalInRange(Light_Intensity,10,90,2000)){
    testStepPass("Manual mode_knob works");
    }
    else if($Light_Intensity<0 && $Light_Intensity>1000){
    teststepfail("manual mode_knob isn't working");
    }
    else if($Doorlock==0)
    {
      testStepFail("Doors are not Locked");
    }
  }else{
    testStepWarning("look into parameters");
  }
  }
}
testcase Automatic_mode_InCabinLights_ON(){
  if(@KL15==1 && $Doorlock==1){
    testWaitForTimeout(5000);
    if(@Automatic_Mode_LED==1){
    testStepPass("Automatic mode is active");
    }
  }
 else if(@Automatic_Mode_LED==0 || @KL15==0 || $Doorlock==0){
  testStepFail("Automatic mode is inactive");
```

```
  }else{
  teststepwarning("look into parameters");
  }
  rVal = testWaitForTesterConfirmation("Automatic mode Verification: Automatic mode
is active or not",10000);
  if(rVal == 1)//yes clicked
  {
    testStepPass("Automatic mode works");
  }
  else if(rVal == 2)//No clicked
  {
    testStepFail("Automatic mode is OFF!!");
  }
  else if (rVal == 0)//time out
  {
    testStepWarning("No verification from tester provided");
  }
}
testcase Automatic_mode_InCabinLights_OFF(){
  if(@KL15==1 && $Doorlock==1){
    testWaitForTimeout(5000);
    if(@Automatic_Mode_LED==0){
    testStepPass("Automatic mode is inactive");
    }
  }
 else if(@Automatic_Mode_LED==0 || @KL15==0 || $Doorlock==0){
  testStepFail("Automatic mode is inactive");
  }else{
  teststepwarning("look into parameters");
  }
  rVal = testWaitForTesterConfirmation("Automatic mode Verification: Automatic mode
LED is active or not",10000);
  if(rVal == 1)//yes clicked
  {
    testStepPass("Automatic mode LED is OFF");
  }
  else if(rVal == 2)//No clicked
  {
    testStepFail("Automatic mode LED is ON!!");
  }
  else if (rVal == 0)//time out
  {
    testStepWarning("No verification from tester provided");
  }
}

testcase Automatic_mode_InCabinLights_test(){
  if(@KL15==1 && $Doorlock==1){
    if(@Automatic_Mode_LED==1){
    if(testWaitForSignalInRange(Light_Intensity,10,90,15000)==1){
    testStepPass("Automatic mode works");
    }
    else if($Light_Intensity<0 && $Light_Intensity>1000 || @Automatic_Mode_LED==0){
    teststepfail("Automatic Mode isn't working");
  }else{
    testStepWarning("look into parameters");
  }
  }
  }
  }
```

```
testcase ManualKnob_BlowerSpeed()
{
  write("In test case Manual Knob - Blower Speed of HVAC");
  testWaitForTimeout(5000);
  //data verification
  if(@KL15==1 && $Doorlock==1)
  {
    if(testWaitForSignalInRange(Manual_Knb_BlowerSpd,0,100,2000))
    {
    testStepPass("Manual Knob of Blower Speed is working in the range");
    }
  }
  else if($Manual_Knb_BlowerSpd<0 && $Manual_Knb_BlowerSpd>100)
  {
    teststepfail("Manual Knob of Blower Speed is not working in the range");
  }
  else
  {
    testStepWarning("Check the timing parameters");
  }
}
testcase Temperature_Range_ManualMode()
{
  write("In testcase Temperature Range for Manual Mode");
  testWaitForTimeout(4000);
  rVal = testWaitForSignalInRange(Temperature,0,50,20000);
  if(rVal== 1)
  {
    testStepPass("Temperature is in the range in Manual Mode");
   }
  else if(rVal ==0)
  {
    testStepFail("Temperature is not in the rangem in Manual Mode");
  }
  else
  {
    testStepWarning("Check the timing parameters");
  }
}
testcase Blower_Speed_ManualMode()
{
  if(@KL15==1 && $Doorlock==1)
  {
    if(@Manual_Mode==1)
    {
   if(testWaitForSignalInRange(Manual_Knb_BlowerSpd,0,44,15000))
    {
      if(testWaitForSignalMatch(CurrentBlowerSpeed,1250,15000))
      {
       testStepPass("Low Blower Speed of HVAC actuated at 1250 rpm");
      }
      else
      {
        testStepFail("Low Blower Speed of HVAC is not at desired rpm");
      }
    }
    else if(testWaitForSignalInRange(Manual_Knb_BlowerSpd,45,84,15000))
    {
      if(testWaitForSignalMatch(CurrentBlowerSpeed,2500,15000))
      {
```

```
        testStepPass("Medium Blower Speed of HVAC actuated at 2500 rpm");
       }
       else
       {
         testStepFail("Medium Blower Speed of HVAC is not at desired rpm");
       }
     }
     else if(testWaitForSignalInRange(Manual_Knb_BlowerSpd,85,100,15000))
     {
       if(testWaitForSignalMatch(CurrentBlowerSpeed,3750,15000))
       {
        testStepPass("High Blower Speed of HVAC actuated at 3750 rpm");
       }
       else
       {
         testStepFail("High Blower Speed of HVAC is not at desired rpm");
       }
     }
   else
     {
       testStepFail("Manual Knob of Blower is not in range hence blower is not
working as desired");
     }
   }
   else
   {
     testStepWarning("look into parameters");
   }
   //Visual Confirmation of Heating Mode HVAC LED
   rVal2=testWaitForTesterConfirmation("Heating Mode HVAC LED:Is Heating Mode HVAC
LED ON?");
   if(rVal2==1)//if yes is clicked!
   {
     testStepPass("Heating Mode LED is ON");
   }
   else if(rVal2 == 2)//if No is clicked!
   {
     testStepFail("Heating Mode LED is OFF!!");
   }
   else if (rVal2 == 0)//Time out has occured!!
   {
     testStepWarning("No verification is done from the tester");
   }
   //Visual Confirmation of Cooling Mode HVAC LED
   rVal2=testWaitForTesterConfirmation("Cooling Mode HVAC LED:Is Cooling Mode HVAC
LED ON?");
   if(rVal2==1)//if yes is clicked!
   {
     testStepPass("Cooling Mode LED is ON");
   }
   else if(rVal2 == 2)//if No is clicked!
   {
     testStepFail("Cooling Mode LED is OFF!!");
   }
   else if (rVal2 == 0)//Time out has occured!!
   {
     testStepWarning("No verification is done from the tester");
   }
   }
}
```

```
testcase Check_HeaterLOW_Manual()
{
 flag=2;
  if(flag==2){
    Pre_Condition_Updates();
  }
  write("In test case of Manual seat warmer ");

  testWaitForTimeout(5000);
  rVal1=testWaitForSignalMatch(Manual_Mode,1,5000);
  rVal=testWaitForSignalMatch(KL15,1,5000);

  if((rVal==1)&&(rVal1==1)){
  if((@KL15==1)&&(@Manual_Mode==1))
  {
      var1=testWaitForSignalInRange(Manual_Knb_Seattemp,30,40,10000);
      if(var1==1)
      {
        var2=testWaitForSignalMatch(Seat_Heater,30,5000);
        if(var2==1)
        {
          testStepPass("Heater working at 30W");
        }
        else if(var2 ==0)
        {
          testStepFail("Heater not working properly");
        }
      }
    }

  }
  else{
    testStepWarning("look into parameters");
  }

}
testcase Check_HeaterMEDIUM_Manual()
{
 flag=2;
  if(flag==2){
    Pre_Condition_Updates();
  }
  write("In test case of Manual seat warmer ");

  testWaitForTimeout(5000);
  rVal1=testWaitForSignalMatch(Manual_Mode,1,5000);
  rVal=testWaitForSignalMatch(KL15,1,5000);

  if((rVal==1)&&(rVal1==1)){
  if((@KL15==1)&&(@Manual_Mode==1))
  {
      var1=testWaitForSignalInRange(Manual_Knb_Seattemp,40,50,10000);
      if(var1==1)
      {
        var2=testWaitForSignalMatch(Seat_Heater,60,5000);
        if(var2==1)
        {
          testStepPass("Heater working at 60W");
        }
        else if(var2 ==0)
```

```
        {
          testStepFail("Heater not working properly");
        }
      }
    }

  }
  else{
    testStepWarning("look into parameters");
  }

}
testcase Check_HeaterHIGH_Manual()
{
 flag=2;
  if(flag==2){
    Pre_Condition_Updates();
  }
  write("In test case of Manual seat warmer ");

  testWaitForTimeout(5000);
  rVal1=testWaitForSignalMatch(Manual_Mode,1,5000);
  rVal=testWaitForSignalMatch(KL15,1,5000);

  if((rVal==1)&&(rVal1==1)){
  if((@KL15==1)&&(@Manual_Mode==1))
  {
      var1=testWaitForSignalInRange(Manual_Knb_Seattemp,50,60,10000);
      if(var1==1)
      {
        var2=testWaitForSignalMatch(Seat_Heater,160,5000);
        if(var2==1)
        {
          testStepPass("Heater working at 160W");
        }
        else if(var2 ==0)
        {
          testStepFail("Heater not working properly");
        }
      }
    }

  }
  else{
    testStepWarning("look into parameters");
  }

}
testcase Manual_Mode_OFF()//To check Manual Mode
{
  write("In test case Manual Mode ");
  //data verification of Manual Mode OFF LED
  testWaitForTimeout(5000);
//  rVal=testWaitForSignalMatch(KL15,0,10000);
//  if(rVal==1)
//    rVal=testWaitForEnvVar(KL15,10000);
 // testWaitForTimeout(5000);
  var=testWaitForEnvVar(Manual_Mode,10000);
  if(var==1)
  {
```

```
    if((@KL15==1) && ($Doorlock==1) && (@Manual_Mode==0))
    {
      testStepPass("Manaul Mode LED is OFF!!");
    }
    else if((@KL15==1) && ($Doorlock==1) && (@Manual_Mode==1))
    {
      testStepFail("Manual Mode LED is still ON!!");
    }
    else if(@KL15==0)
    {
      testStepFail("Ignition is OFF");
    }
    else if($Doorlock==0)
    {
      testStepFail("Door is not Locked");
    }
  }
  else
  {
    testStepWarning("Check the Timing parameter");
  }
  //Visual Confirmation of Manual Mode OFF LED
  rVal2=testWaitForTesterConfirmation("Manual Mode OFF LED:Is Manual Mode LED is
OFF?");
  if(rVal2==1)//if yes is clicked!
  {
    testStepPass("Manual Mode LED is OFF");
  }
  else if(rVal2 == 2)//if No is clicked!
  {
    testStepFail("Manual Mode LED is still ON!!");
  }
  else if (rVal2 == 0)//Time out has occured!!
  {
    testStepWarning("No verification is done from the tester");
  }
}

testcase AutoMode_HVAC_ON()//To check Automatic Mode for HVAC
{
  write("In test case Automatic Mode for HVAC ");
  //data verification of Automatic Mode HVAC - ON LED
  testWaitForTimeout(5000);
  var=testWaitForEnvVar(Automatic_Mode_HVAC,10000);
  if(var==1)
  {
    if(($Doorlock==1)&& (@KL15==1) && (@Automatic_Mode_HVAC==1))
    {
      testStepPass("Automatic Mode LED is ON!!");
    }
    else if(($Doorlock==1)&& (@KL15==1) && (@Automatic_Mode_HVAC==0))
    {
      testStepFail("Automatic Mode LED is still OFF!!");
    }
    else if(@KL15==0)
    {
      testStepFail("Ignition is OFF");
    }
    else if($Doorlock==0)
    {
```

```
      testStepFail("Door is not Locked");
    }
  }
  else
  {
    testStepWarning("Check the Timing parameter");
  }
  //Visual Confirmation of Automatic Mode HVAC - ON LED
  rVal2=testWaitForTesterConfirmation("Automatic Mode ON LED:Is Automatic Mode LED
is ON?");
  if(rVal2==1)//if yes is clicked!
  {
    testStepPass("Automatic Mode LED for HVAC is ON");
  }
  else if(rVal2 == 2)//if No is clicked!
  {
    testStepFail("Automatic Mode LED for HVAC is still OFF!!");
  }
  else if (rVal2 == 0)//Time out has occured!!
  {
    testStepWarning("No verification is done from the tester");
  }
}
testcase Temperature_Range_AutoMode()
{
  write("In testcase Temperature Range for Automatic Mode");
  testWaitForTimeout(4000);
  rVal = testWaitForSignalInRange(Temperature,0,50,20000);
  if(rVal== 1)
  {
    testStepPass("Temperature is in the range in Automatic Mode");
   }
  else if(rVal ==0)
  {
    testStepFail("Temperature is not in the rangem in Automatic Mode");
  }
  else
  {
    testStepWarning("Check the timing parameters");
  }
}
testcase Blower_Speed_AutoMode()
{
  if(@KL15==1 && $Doorlock==1)
  {
    if(@Manual_Mode==1)
    {
   if(testWaitForSignalInRange(Temperature,0,10,15000))
    {
      if(testWaitForSignalMatch(CurrentBlowerSpeed,3750,15000))
      {
       testStepPass("Blower Speed of HVAC actuated at 3750 rpm");
      }
      else
      {
        testStepFail("Blower Speed of HVAC is not at desired rpm");
      }
    }
    else if(testWaitForSignalInRange(Temperature,10,20,15000))
    {
```

```
      if(testWaitForSignalMatch(CurrentBlowerSpeed,2500,15000))
      {
       testStepPass("Blower Speed of HVAC actuated at 2500 rpm");
      }
      else
      {
        testStepFail("Blower Speed of HVAC is not at desired rpm");
      }
    }
    else if(testWaitForSignalInRange(Temperature,20,30,15000))
    {
      if(testWaitForSignalMatch(CurrentBlowerSpeed,1750,15000))
      {
       testStepPass("Blower Speed of HVAC actuated at 1750 rpm");
      }
      else
      {
        testStepFail("Blower Speed of HVAC is not at desired rpm");
      }
    }
     else if(testWaitForSignalInRange(Temperature,30,40,15000))
    {
      if(testWaitForSignalMatch(CurrentBlowerSpeed,2500,15000))
      {
       testStepPass("Blower Speed of HVAC actuated at 2500 rpm");
      }
      else
      {
        testStepFail("Blower Speed of HVAC is not at desired rpm");
      }
    }
    else if(testWaitForSignalInRange(Temperature,40,50,15000))
    {
      if(testWaitForSignalMatch(CurrentBlowerSpeed,3750,15000))
      {
       testStepPass("Blower Speed of HVAC actuated at 3750 rpm");
      }
      else
      {
        testStepFail("Blower Speed of HVAC is not at desired rpm");
      }
    }
  else
    {
      testStepFail("Blower speed is not in range hence blower is not working as
desired");
    }
  }
  else
  {
    testStepWarning("look into parameters");
  }
  //Visual Confirmation of Heating Mode HVAC LED
  rVal2=testWaitForTesterConfirmation("Heating Mode HVAC LED:Is Heating Mode HVAC
LED ON?");
  if(rVal2==1)//if yes is clicked!
  {
    testStepPass("Heating Mode LED is ON");
  }
  else if(rVal2 == 2)//if No is clicked!
```

```
  {
    testStepFail("Heating Mode LED is OFF!!");
  }
  else if (rVal2 == 0)//Time out has occured!!
  {
    testStepWarning("No verification is done from the tester");
  }
  //Visual Confirmation of Cooling Mode HVAC LED
  rVal2=testWaitForTesterConfirmation("Cooling Mode HVAC LED:Is Cooling Mode HVAC
LED ON?");
  if(rVal2==1)//if yes is clicked!
  {
    testStepPass("Cooling Mode LED is ON");
  }
  else if(rVal2 == 2)//if No is clicked!
  {
    testStepFail("Cooling Mode LED is OFF!!");
  }
  else if (rVal2 == 0)//Time out has occured!!
  {
    testStepWarning("No verification is done from the tester");
  }
  }
}
testcase AutoMode_HVAC_OFF()//To check Automatic Mode for HVAC
{
  write("In test case Automatic Mode for HVAC ");
  //data verification of Automatic Mode HVAC - OFF LED
  testWaitForTimeout(5000);
  var=testWaitForEnvVar(Automatic_Mode_HVAC,10000);
  if(var==1)
  {
    if(($Doorlock==1)&& (@KL15==1) && (@Automatic_Mode_HVAC=0))
    {
      testStepPass("Automatic Mode LED for HVAC is OFF!!");
    }
    else if(($Doorlock==1)&& (@KL15==1) && (@Automatic_Mode_HVAC==1))
    {
      testStepFail("Automatci Mode LED for HVAC is still ON!!");
    }
    else if(@KL15==0)
    {
      testStepFail("Ignition is OFF");
    }
    else if($Doorlock==0)
    {
      testStepFail("Door is not Locked");
    }
  }
  else
  {
    testStepWarning("Check the Timing parameter");
  }
  //Visual Confirmation of Automatic Mode HVAC - OFF LED
  rVal2=testWaitForTesterConfirmation("Automatic Mode OFF LED:Is Automatic Mode LED
is OFF?");
  if(rVal2==1)//if yes is clicked!
  {
    testStepPass("Automatic Mode LED for HVAC is ON");
  }
```

```
  else if(rVal2 == 2)//if No is clicked!
  {
    testStepFail("Automatic Mode LED for HVAC is still OFF!!");
  }
  else if (rVal2 == 0)//Time out has occured!!
  {
    testStepWarning("No verification is done from the tester");
  }
}
testcase Temperature_Range_Auto()
{
  write("In testcase Temperature Range");
  testWaitForTimeout(4000);
  rVal = testWaitForSignalInRange(Temperature,0,50,20000);
  if(rVal== 1)
  {
    testStepPass("Temperature is in the range for Manual Mode");
   }
  else if(rVal ==0)
  {
    testStepFail("Temperature is not in the range");
  }
  else
  {
    testStepWarning("Check the timing parameters");
  }
}
testcase Check_SeatBelt_Status()
{
  flag=2;
  if(flag==2){
    Pre_Condition_Updates();
  }
  write("In test case of Automatic seat warmer for activatoin of Seatbelt status and
Automatic seat warmer ");
  //data verification of Manual Mode ON LED
  testWaitForTimeout(5000);
  rVal1=testWaitForEnvVar(Automatic_Mode_Seatbelt,5000);
  rVal=testWaitForSignalMatch(KL15,1,5000);

  if((rVal==1)&&(rVal1==1)){
  if((@KL15==1)&&(@Automatic_Mode_Seatbelt==1))
  {
    var=testWaitForSignalInRange(WeightValue,40,115,10000);
    if(var==1){
      var1=testWaitForSignalMatch(Seatbelt_Mode,1,5000);
      if(var1==1)
      {
  testStepPass("Seatbelt has been put");
  if(testWaitForSignalInRange(Temperature,0,10,15000))
    {
      if(testWaitForSignalMatch(CurrentBlowerSpeed,3750,15000))
      {
       testStepPass("Blower Speed of HVAC actuated at 3750 rpm");
      }
      else
      {
        testStepFail("Blower Speed of HVAC is not at desired rpm");
      }
    }
```

```
    else if(testWaitForSignalInRange(Temperature,10,20,15000))
    {
      if(testWaitForSignalMatch(CurrentBlowerSpeed,2500,15000))
      {
       testStepPass("Blower Speed of HVAC actuated at 2500 rpm");
      }
      else
      {
        testStepFail("Blower Speed of HVAC is not at desired rpm");
      }
    }
    else if(testWaitForSignalInRange(Temperature,20,30,15000))
    {
      if(testWaitForSignalMatch(CurrentBlowerSpeed,1750,15000))
      {
       testStepPass("Blower Speed of HVAC actuated at 1750 rpm");
      }
      else
      {
        testStepFail("Blower Speed of HVAC is not at desired rpm");
      }
    }
     else if(testWaitForSignalInRange(Temperature,30,40,15000))
    {
      if(testWaitForSignalMatch(CurrentBlowerSpeed,2500,15000))
      {
       testStepPass("Blower Speed of HVAC actuated at 2500 rpm");
      }
      else
      {
        testStepFail("Blower Speed of HVAC is not at desired rpm");
      }
    }
    else if(testWaitForSignalInRange(Temperature,40,50,15000))
    {
      if(testWaitForSignalMatch(CurrentBlowerSpeed,3750,15000))
      {
       testStepPass("Blower Speed of HVAC actuated at 3750 rpm");
      }
      else
      {
        testStepFail("Blower Speed of HVAC is not at desired rpm");
      }
    }
  else
  {
    testStepWarning("look into parameters");
  }
  //Visual Confirmation of Heating Mode HVAC LED
  rVal2=testWaitForTesterConfirmation("Heating Mode HVAC LED:Is Heating Mode HVAC
LED ON?");
  if(rVal2==1)//if yes is clicked!
  {
    testStepPass("Heating Mode LED is ON");
  }
  else if(rVal2 == 2)//if No is clicked!
  {
    testStepFail("Heating Mode LED is OFF!!");
  }
  else if (rVal2 == 0)//Time out has occured!!
```

```
  {
    testStepWarning("No verification is done from the tester");
  }
  //Visual Confirmation of Cooling Mode HVAC LED
  rVal2=testWaitForTesterConfirmation("Cooling Mode HVAC LED:Is Cooling Mode HVAC
LED ON?");
  if(rVal2==1)//if yes is clicked!
  {
    testStepPass("Cooling Mode LED is ON");
  }
  else if(rVal2 == 2)//if No is clicked!
  {
    testStepFail("Cooling Mode LED is OFF!!");
  }
  else if (rVal2 == 0)//Time out has occured!!
  {
    testStepWarning("No verification is done from the tester");
  }
  }
    else if(var1==0)
      {
        testStepFail("Seatbelt has not been put ");
      }
    }
    else if(var==0){
    testStepFail("Weight is not in range");
    }
  }
}
  else{
    testStepWarning("look into parameters");
  }
}
testcase Check_HeaterLOW_Auto()
{
 flag=2;
  if(flag==2){
    Pre_Condition_Updates();
  }
  write("In test case of Automatic seat warmer ");

  testWaitForTimeout(5000);
  rVal1=testWaitForSignalMatch(Automatic_Mode_Seatbelt,1,5000);
  rVal=testWaitForSignalMatch(KL15,1,5000);

  if((rVal==1)&&(rVal1==1)){
  if((@KL15==1)&&(@Automatic_Mode_Seatbelt==1))
  {
    var=testWaitForSignalInRange(WeightValue,40,115,10000);

    if(var==1){
      var1=testWaitForSignalInRange(SEAT_temp_ip,30,40,10000);
      if(var1==1)
      {
        var2=testWaitForSignalMatch(Seat_Temp_Heater,30,5000);
        if(var2==1)
        {
          testStepPass("Heater working at 30W");
        }
        else if(var2 ==0)
```

```
        {
          testStepFail("Heater not working properly");
        }
      }
    }
    else if(var==0){
    testStepFail("Weight is not in range");
    }
  }
  else{
    testStepWarning("look into parameters");
  }
  }

}
testcase Check_HeaterMEDIUM_Auto()
{
 flag=2;
  if(flag==2){
    Pre_Condition_Updates();
  }
  write("In test case of Automatic seat warmer ");

  testWaitForTimeout(5000);
  rVal1=testWaitForSignalMatch(Automatic_Mode_Seatbelt,1,5000);
  rVal=testWaitForSignalMatch(KL15,1,5000);

  if((rVal==1)&&(rVal1==1)){
  if((@KL15==1)&&(@Automatic_Mode_Seatbelt==1))
  {
    var=testWaitForSignalInRange(WeightValue,40,115,10000);

    if(var==1){
      var1=testWaitForSignalInRange(SEAT_temp_ip,40,50,10000);
      if(var1==1)
      {
        var2=testWaitForSignalMatch(Seat_Temp_Heater,60,5000);
        if(var2==1)
        {
          testStepPass("Heater working at 60W");
        }
        else if(var2 ==0)
        {
          testStepFail("Heater not working properly");
        }
      }
    }
    else if(var==0){
    testStepFail("Weight is not in range");
    }
  }
  else{
    testStepWarning("look into parameters");
  }
  }

}
testcase Check_HeaterHIGH_Auto()
{
 flag=2;
```

```
  if(flag==2){
    Pre_Condition_Updates();
  }
  write("In test case of Automatic seat warmer ");

  testWaitForTimeout(5000);
  rVal1=testWaitForSignalMatch(Automatic_Mode_Seatbelt,1,5000);
  rVal=testWaitForSignalMatch(KL15,1,5000);

  if((rVal==1)&&(rVal1==1)){
  if((@KL15==1)&&(@Automatic_Mode_Seatbelt==1))
  {
    var=testWaitForSignalInRange(WeightValue,40,115,10000);

    if(var==1){
      var1=testWaitForSignalInRange(SEAT_temp_ip,50,60,10000);
      if(var1==1)
      {
        var2=testWaitForSignalMatch(Seat_Temp_Heater,160,5000);
        if(var2==1)
        {
          testStepPass("Heater working at 160W");
        }
        else if(var2 ==0)
        {
          testStepFail("Heater not working properly");
        }
      }
    }
    else if(var==0){
    testStepFail("Weight is not in range");
    }
  }
  else{
    testStepWarning("look into parameters");
  }
  }
}
testcase Defrost_Mode_ON()//To check Defrost Mode
{
  write("In test case Defrost Mode ");
  //data verification of Defrost Mode ON LED
  testWaitForTimeout(5000);
  var=testWaitForEnvVar(Defrost_Mode,10000);
  if(var==1)
  {
    if((@KL15==1) && ($Doorlock==1) && (@Defrost_Mode==1))
    {
      testStepPass("Defrost Mode is On!!");
    }
    else if((@KL15==1) && ($Doorlock==1) && (@Defrost_Mode==0))
    {
      testStepFail("Defrost Mode is OFF!!");
    }
    else if(@KL15==0)
    {
      testStepFail("Ignition is OFF");
    }
    else if($Doorlock==0)
    {
```

```
      testStepFail("Door is not Locked");
    }
  }
  else
  {
    testStepWarning("Check the Timing parameter");
  }
  //Visual Confirmation of Defrost Mode ON LED
  rVal2=testWaitForTesterConfirmation("Defrost Mode ON LED:Is Defrost Mode ON?");
  if(rVal2==1)//if yes is clicked!
  {
    testStepPass("Defrost Mode is ON");
  }
  else if(rVal2 == 2)//if No is clicked!
  {
    testStepFail("Defrost Mode is still OFF!!");
  }
  else if (rVal2 == 0)//Time out has occured!!
  {
    testStepWarning("No verification is done from the tester");
  }
}
testcase Humidity_Range()
{
  write("In testcase Humidity Range");
  testWaitForTimeout(4000);
  rVal = testWaitForSignalInRange(RelativeHumidity,0,50,20000);
  if(rVal== 1)
  {
    testStepPass("Humidity is in the range");
   }
  else
  {
    testStepFail("Humidity is not in the range");
  }

}
testcase Defrost_Mode_OFF()//To check Defrost Mode
{
  write("In test case Defrost Mode ");
  //data verification of Defrost Mode OFF LED
  testWaitForTimeout(5000);
  var=testWaitForEnvVar(Defrost_Mode,10000);
  if(var==1)
  {
    if((@KL15==1) && ($Doorlock==1) && (@Defrost_Mode==0))
    {
      testStepPass("Defrost Mode is OFF!!");
    }
    else if((@KL15==1) && (@Defrost_Mode==1))
    {
      testStepFail("Defrost Mode is still ON!!");
    }
    else if(@KL15==0)
    {
      testStepFail("Ignition is OFF");
    }
    else if($Doorlock==0)
    {
      testStepFail("Door is not Locked");
```

```
    }
  }
  else
  {
    testStepWarning("Check the Timing parameter");
  }
  //Visual Confirmation of Defrost Mode OFF LED
  rVal2=testWaitForTesterConfirmation("Defrost Mode OFF LED:Is Defrost Mode OFF?");
  if(rVal2==1)//if yes is clicked!
  {
    testStepPass("Defrost Mode is OFF");
  }
  else if(rVal2 == 2)//if No is clicked!
  {
    testStepFail("Defrost Mode is still ON!!");
  }
  else if (rVal2 == 0)//Time out has occured!!
  {
    testStepWarning("No verification is done from the tester");
  }
}
testcase Ignition_OFF()//To check Ignition Off
{
  flag++;
  if(flag==2)
  {
    Pre_Condition_Updates();
  }
  write("In test case Ignition_OFF");
  //data verification of Ignition OFF LED
  testWaitForTimeout(5000);
  rVal=testWaitForSignalMatch(KL15,1,10000);
  if(rVal==1)
    rVal=testWaitForEnvVar(KL15,10000);
  if(rVal==1)
  {
    if(@KL15==0)
    {
      testStepPass("Ignition is Off !!");
    }
    else if(@KL15==1)
    {
      testStepFail("Ignition remained ON!!");
    }
  }
  else
  {
    testStepWarning("Check the Timing parameter");
  }
  //Visual Confirmation of Ignition OFF LED
  rVal2=testWaitForTesterConfirmation("Ignition OFF LED:Is Ignition LED is OFF?");
  if(rVal2==1)//if yes is clicked!
  {
    testStepPass("Ignition LED is OFF when Ignition OFF control is pressed");
  }
  else if(rVal2 == 2)//if No is clicked!
  {
    testStepFail("Ignition LED remained ON!!");
  }
  else if (rVal2 == 0)//Time out has occured!!
```

```
  {
    testStepWarning("No verification is done from the tester");
  }
}


MainTest(){
   // ----- add information to test report
  TestModuleTitle("BCM Test Cases"); // change test module title and add description
  TestModuleDescription("This test module calls some test cases to demonstrate ");
  TestModuleDescription("the test features of CANoe.");

  TestReportAddSUTInfo("Team: ", "Gamma"); // write info about SUT into report
  TestReportAddSUTInfo("Manufactured", "28-02-2020");

  TestReportAddEngineerInfo("Test Engineer", "Aditi Sarkar,Aishwarya Mishra,Anamika
Singh"); // add info about engineer to report

  //execute test cases
  DoorLock();
  Ignition_ON();
  Manual_Mode_ON();
  Manual_mode_InCabinLights_test();
//  Automatic_mode_InCabinLights_ON();
//  Automatic_mode_InCabinLights_test();
//  Automatic_mode_InCabinLights_OFF();
  ManualKnob_BlowerSpeed();
 // Temperature_Range_ManualMode();
  Blower_Speed_ManualMode();
 Check_HeaterLOW_Manual();
 Check_HeaterMEDIUM_Manual();
 Check_HeaterHIGH_Manual();
  Manual_Mode_OFF();
//  AutoMode_HVAC_ON();
//  Temperature_Range_AutoMode();
//  Blower_Speed_AutoMode();
//  AutoMode_HVAC_OFF();
//  Check_SeatBelt_Status();
//  Check_HeaterLOW_Auto();
//  Check_HeaterMEDIUM_Auto();
//  Check_HeaterHIGH_Auto();
//  Defrost_Mode_ON();
//  Humidity_Range();
//  Defrost_Mode_OFF();
 //Ignition_OFF();

}
```