Seat Control Application
(complete AUTOSAR application)

Description:

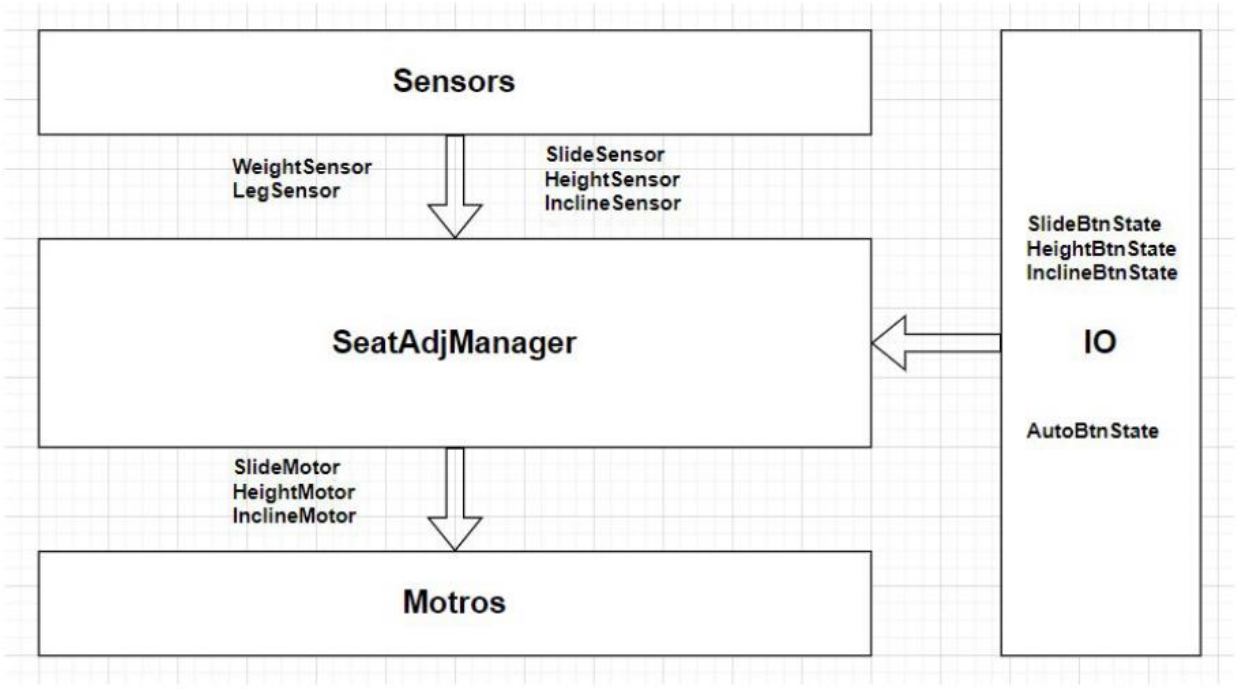
- The application is divided into several SWCs (App SWCs & Sensor/Actuator SWCs) each of whom performs an atomic functionality. These SWCs communicate through Ports; Port interfaces used are Sender/Receiver, Client/Server and Mode Switch.
- An AUTOSAR authoring tool “SAAT” is used to generate the SWCs’ templates.
- A RTE is used to connect the SWCs during runtime, and map data elements, Service Ports and events to OS tasks.
- The BSW modules (Port, Dio, Adc, Spi, Uart(CDD) and Det) are reused from an older project.

Acknowledgement:

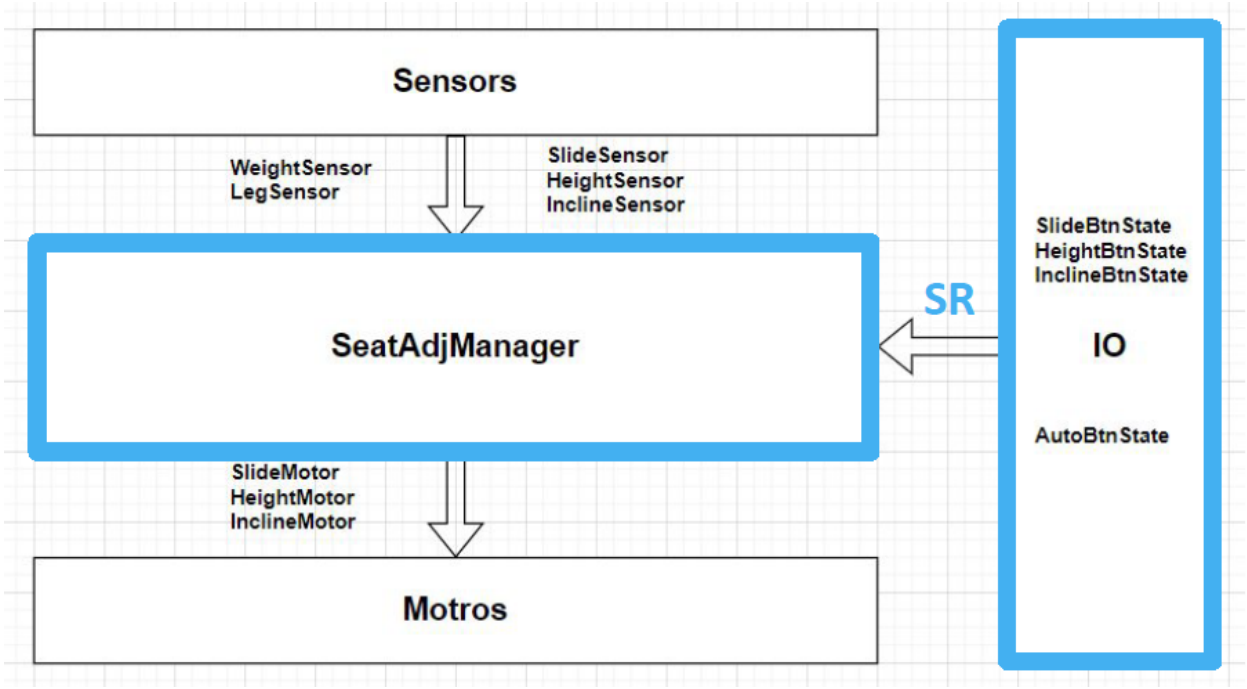
This is the final project of AUTOSAR Master Class sponsored by Sprints.

Project Specification:

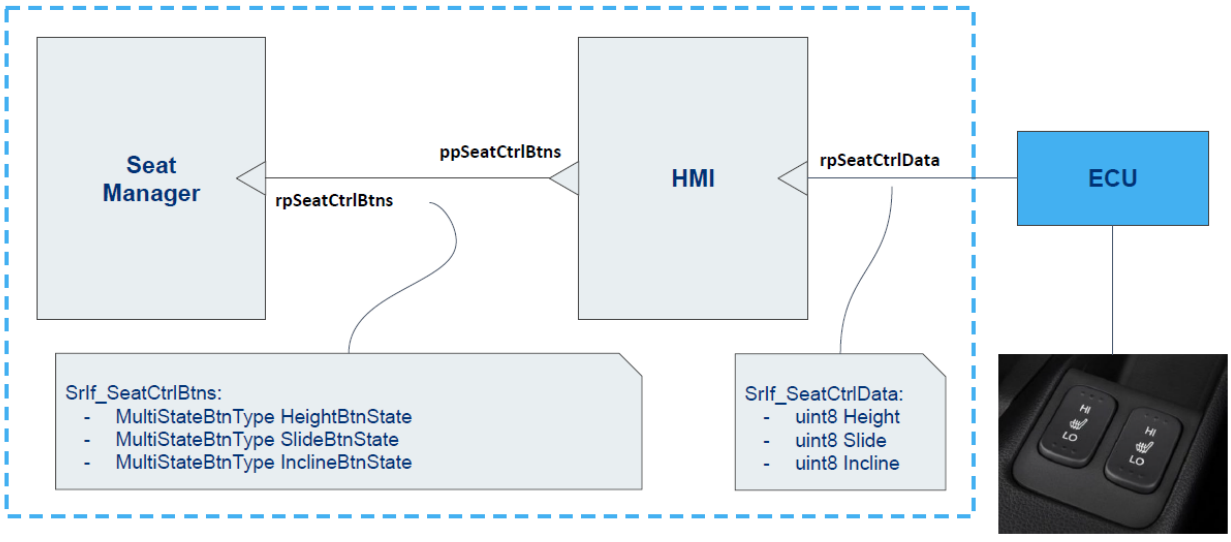
A. High-Level Application Design:



B. Sender Receiver Communication:



(1) High Level Design (Types, Port Interfaces, SWCs, Ports, Connections)



(2) Impl Level Design (Runnables, Access Points, Events)

- HMI_MainFunction shall be triggered each 100 ms
- HMI_MainFunction shall receive all buttons ctrl data
- HMI_MainFunction shall send all buttons states

- SeatManager_SetHeight shall be triggered when HeightBtnState is received
- SeatManager_SetSlide shall be triggered when SlideBtnState is received
- SeatManager_SetIncline shall be triggered when InclineBtnState is received

(3) Software Logic

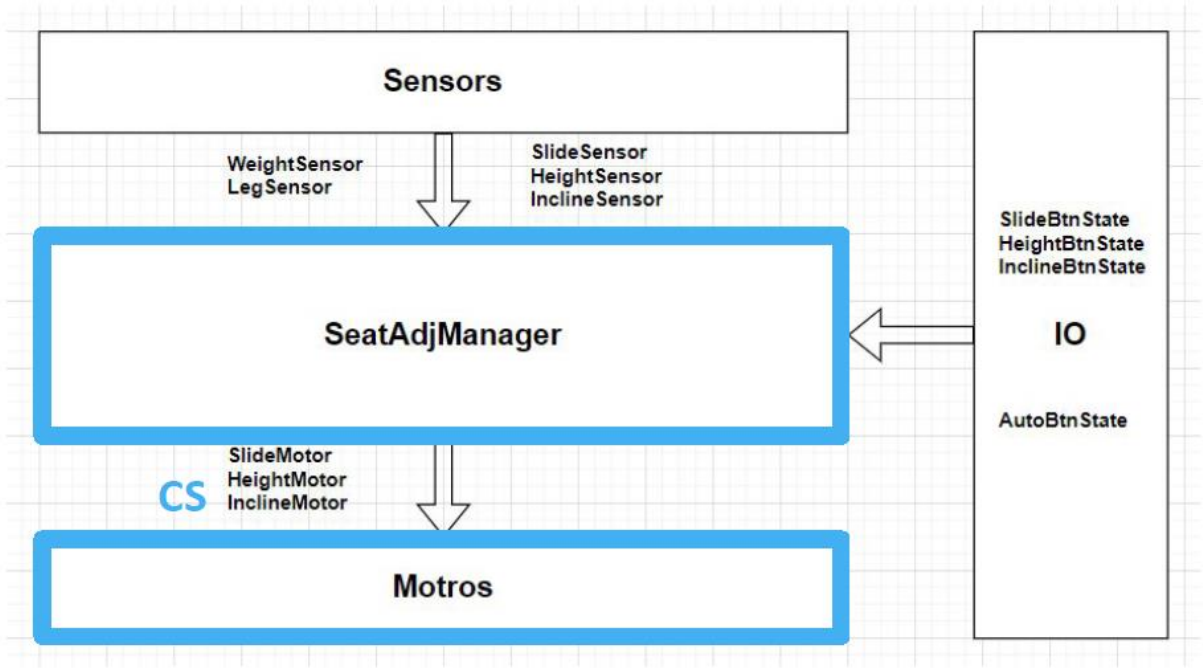
- HMI_MainFunction shall read all the button raw data and set the corresponding button states
  if(Timeout || Not Updated || Data == 0) => BtnState = MULTI_STATE_BTN_INIT
  if(Data == 1) => BtnState = MULTI_STATE_BTN_MINUS
  if(Data == 2) => BtnState = MULTI_STATE_BTN_PLUS

- SeatManager_SetHeight shall read HeightBtnState and drive Actuator (Motor) to increase/decrease height
  Actuators to be developed, for now call Rte_Call_rpHeightMotor_Move(sint8 Step) to increase/decrease the height

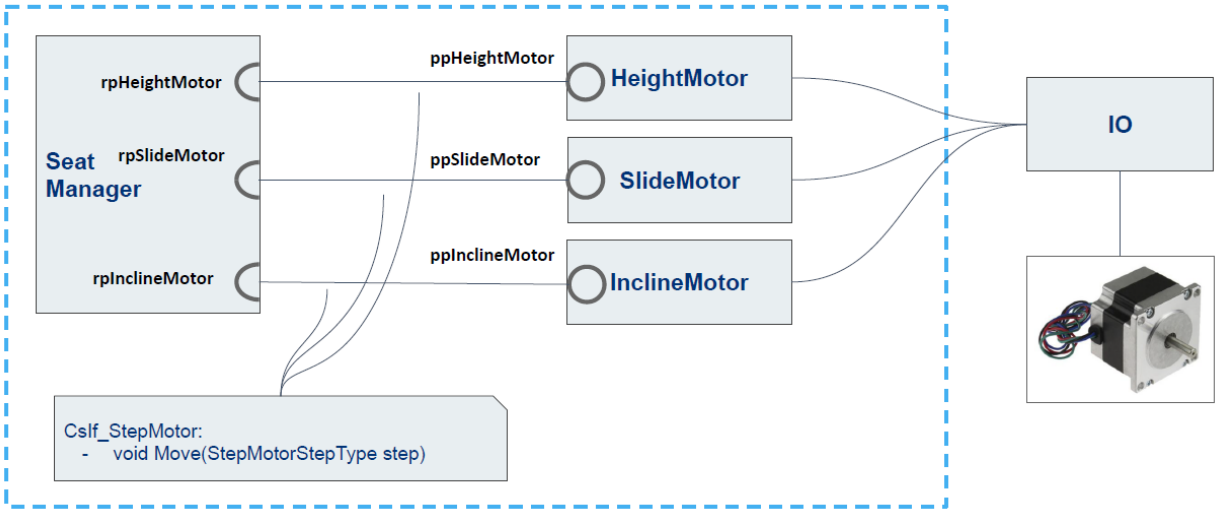
  if(HeightBtnState == MULTI_STATE_BTN_MINUS) => Rte_Call_rpHeightMotor_Move(MOTOR_STEP_MINUS)
  if(HeightBtnState == MULTI_STATE_BTN_PLUS) => Rte_Call_rpHeightMotor_Move(MOTOR_STEP_PLUS)

- The same logic is applied for SeatManager_SetSlide and SeatManager_SetIncline

C. Client Server Communication:



(1) High Level Design (Types, Port Interfaces, SWCs, Ports, Connections)



(2) Impl Level Design (Runnables, Access Points, Events)

- SeatManager_SetHeight shall request Move operation from HeightMotor SWC
- SeatManager_SetSlide shall request Move operation from SlideMotor SWC
- SeatManager_SetIncline shall request Move operation from InclineMotor SWC

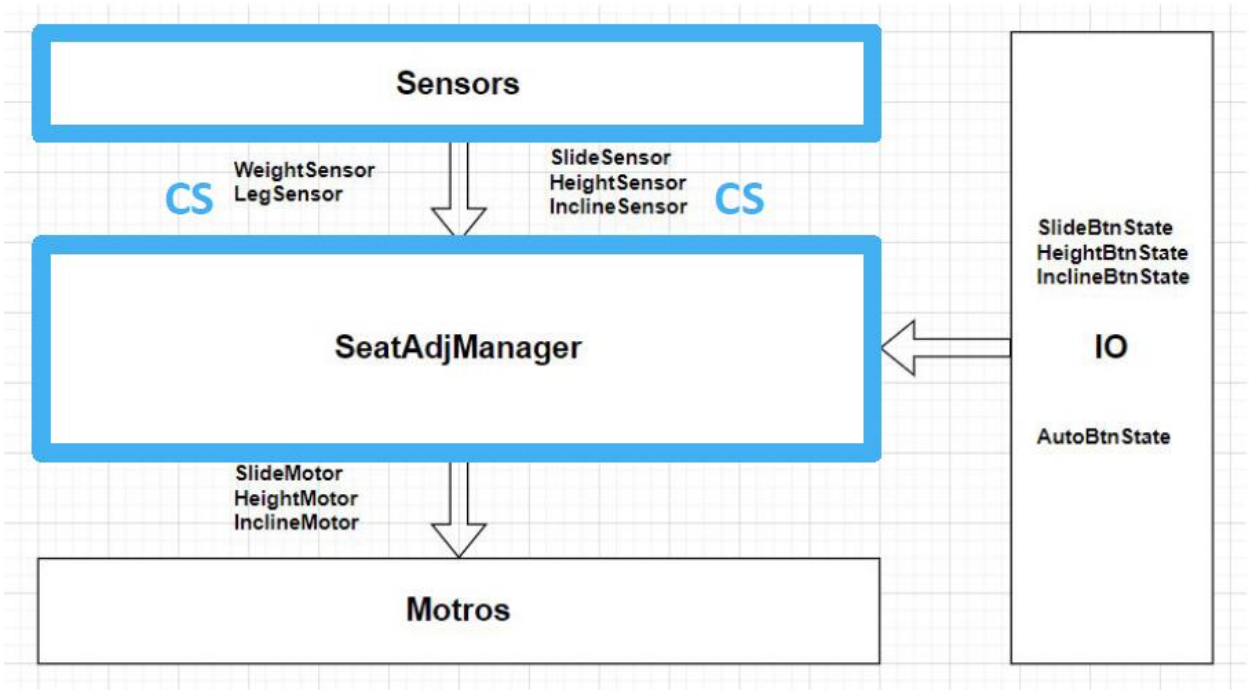
- HeightMotor_Move shall serve Move operation in HeightMotor SWC
- SlideMotor_Move shall serve Move operation in SlideMotor SWC
- InclineMotor_Move shall serve Move operation in InclineMotor SWC

(3) Software Logic

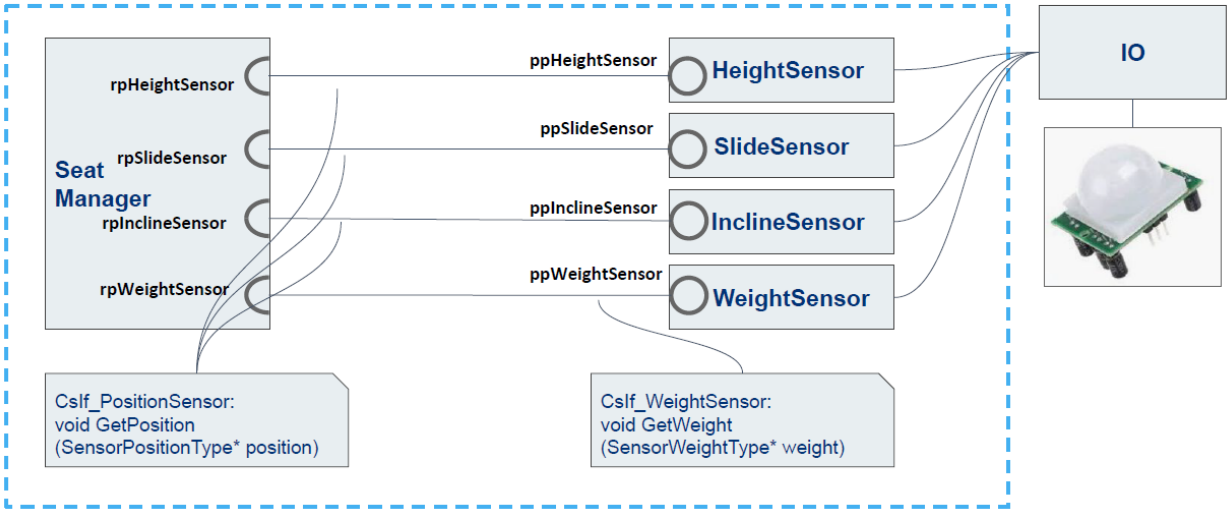
- SeatManager_SetHeight, SeatManager_SetSlide, SeatManager_SetIncline shall control the motors based on the below conditions:
  if(<Height/Slide/Incline>BtnState == MULTI_STATE_BTN_MINUS)
  => Rte_Call_rp<Height/Slide/Incline>Motor_Move(MOTOR_STEP_MINUS)
  if(<Height/Slide/Incline> == MULTI_STATE_BTN_PLUS)
  => Rte_Call_rp<Height/Slide/Incline>Motor_Move(MOTOR_STEP_PLUS)

- HeightMotor_Move, SlideMotor_Move, InclineMotor_Move shall drive the motors through Dio module
  Dio_WriteChannel (Dio_ChannelType ChannelId, Dio_LevelType Level)

D. Software Component Design:



(1) High Level Design (Types, Port Interfaces, SWCs, Ports, Connections)



(2) Impl Level Design (Runnables, Access Points, Events)

- SeatManager_AutoHeight shall control auto height setting for the Seat, each 200 ms
- SeatManager_AutoSlide shall control auto slide setting for the Seat, each 200 ms
- SeatManager_AutoIncline shall control auto incline setting for the Seat, each 200 ms
- SeatManager_AutoHeight shall request GetPosition operation from HeightSesnor SWC
- SeatManager_AutoSlide shall request GetPosition operation from SlideSensor SWC
- SeatManager_AutoIncline shall request GetPosition operation from InclineSensor SWC
- SeatManager_AutoHeight, SeatManager_AutoSlide, SeatManager_AutoIncline shall request GetWeight operation from WeightSensor SWC
- SeatManager_AutoHeight shall request Move operation from HeightMotor SWC
- SeatManager_AutoSlide shall request Move operation from SlideMotor SWC
- SeatManager_AutoIncline shall request Move operation from InclineMotor SWC

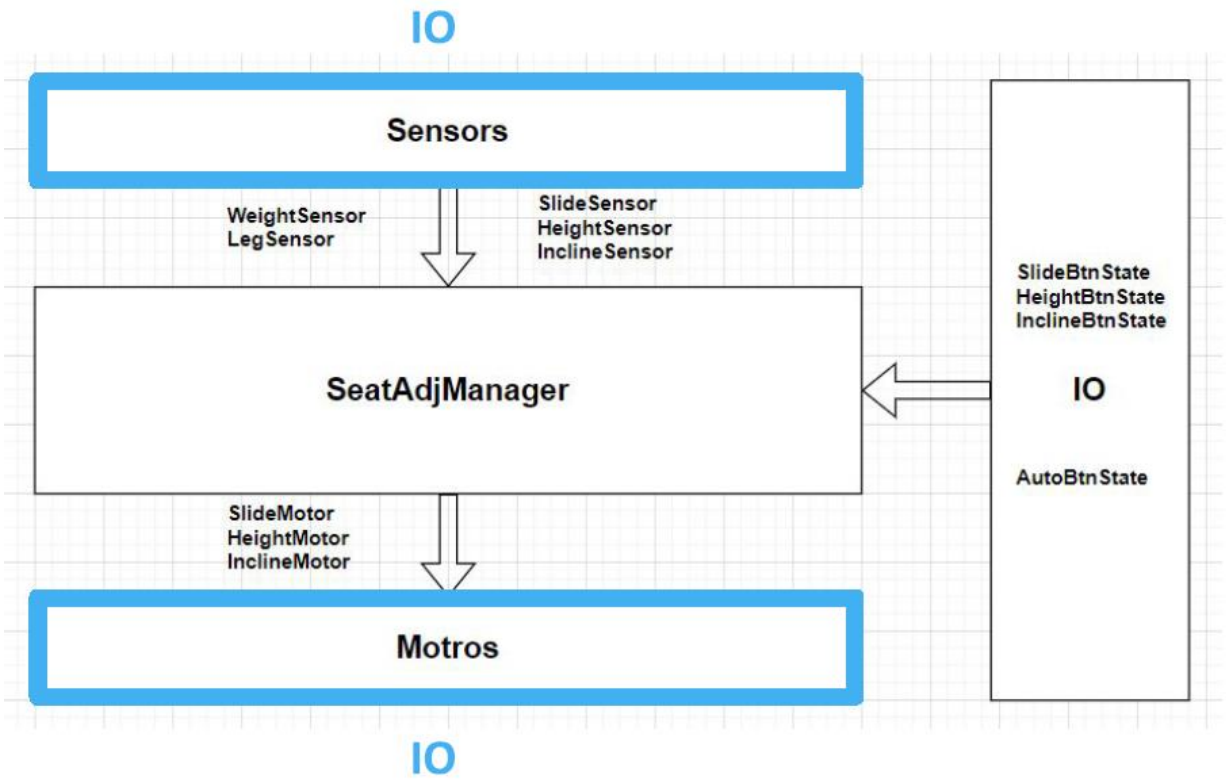
- HeightSensor_GetPosition shall serve GetPosition operation in HeightSensor SWC
- SlideSensor_GetPosition shall serve GetPosition operation in SlideSensor SWC
- InclineSensor_GetPosition shall serve GetPosition operation in InclineSensor SWC
- WeightSensor_GetWeight shall serve GetWeight operation in WeightSensor SWC

(3) Software Logic

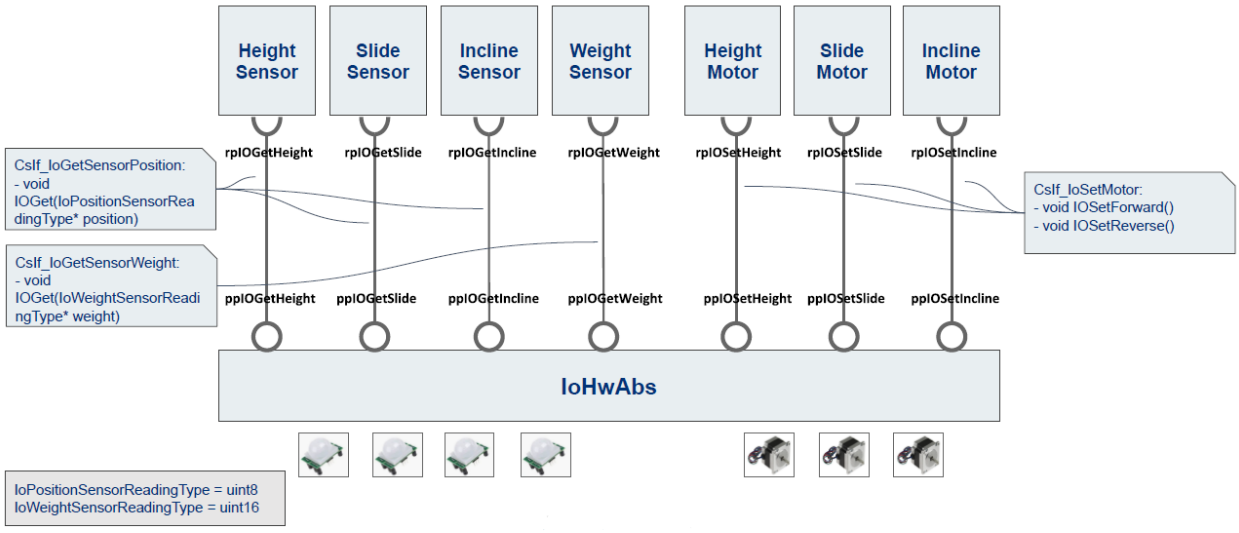
- SeatManager_AutoHeight, SeatManager_AutoSlide, SeatManager_AutoIncline shall control auto setting for Height, Slide and Incline according to the below logic:
  - if(weight > 100) => Set Increment/decrement Height/Slide/Incline till sensor position = SENSOR_POSITION_STEP_3
  - if(weight 100:80) => Set Increment/decrement Height/Slide/Incline till sensor position = SENSOR_POSITION_STEP_2
  - if(weight 80:60) => Set Increment/decrement Height/Slide/Incline till sensor position = SENSOR_POSITION_STEP_1
  - if(weight < 60) => Set Increment/decrement Height/Slide/Incline till sensor position = SENSOR_POSITION_STEP_0

- HeightSensor_GetPosition, SlideSensor_GetPosition , InclineSensor_GetPosition, WeightSensor_GetWeight shall read the sensor value through Adc Moule:
  Adc_ReadGroup(Adc_GroupType Group, Adc_ValueGroupType* DataBufferPtr)

E. **IO Hardware Abstraction Module:**



(1) High Level Design (Types, Port Interfaces, SWCs, Ports, Connections)



(2) Impl Level Design (Runnables, Access Points, Events)

- HeightSensor_GetPosition shall request IOGet operation from IoHwAbs SWC
- SlideSensor_GetPosition shall request IOGet operation from IoHwAbs SWC
- InclineSensor_GetPosition shall request IOGet operation from IoHwAbs SWC
- WeightSensor_GetWeight shall request IOGet operation from IoHwAbs SWC
- HeightMotor_Move shall request IOSetForward, IOSetReverse operations from IoHwAbs SWC
- InclineMotor_Move shall request IOSetForward, IOSetReverse operations from IoHwAbs SWC
- SlideMotor_Move shall request IOSetForward, IOSetReverse operations from IoHwAbs SWC

- IoHwAbs_ECUGet_Height, IoHwAbs_ECUGet_Slide, IoHwAbs_ECUGet_Incline, IoHwAbs_ECUGet_Weight shall serve IOGet operations in IoHwAbs SWC
- IoHwAbs_ECUSetForward_Height, IoHwAbs_ECUSetForward_Incline, IoHwAbs_ECUSetForward_Slide shall serve IOSetForward operations in IoHwAbs SWC
- IoHwAbs_ECUSetReverse_Height, IoHwAbs_ECUSetReverse_Incline, IoHwAbs_ECUSetReverse_Slide shall serve IOSetReverse operations in IoHwAbs SWC

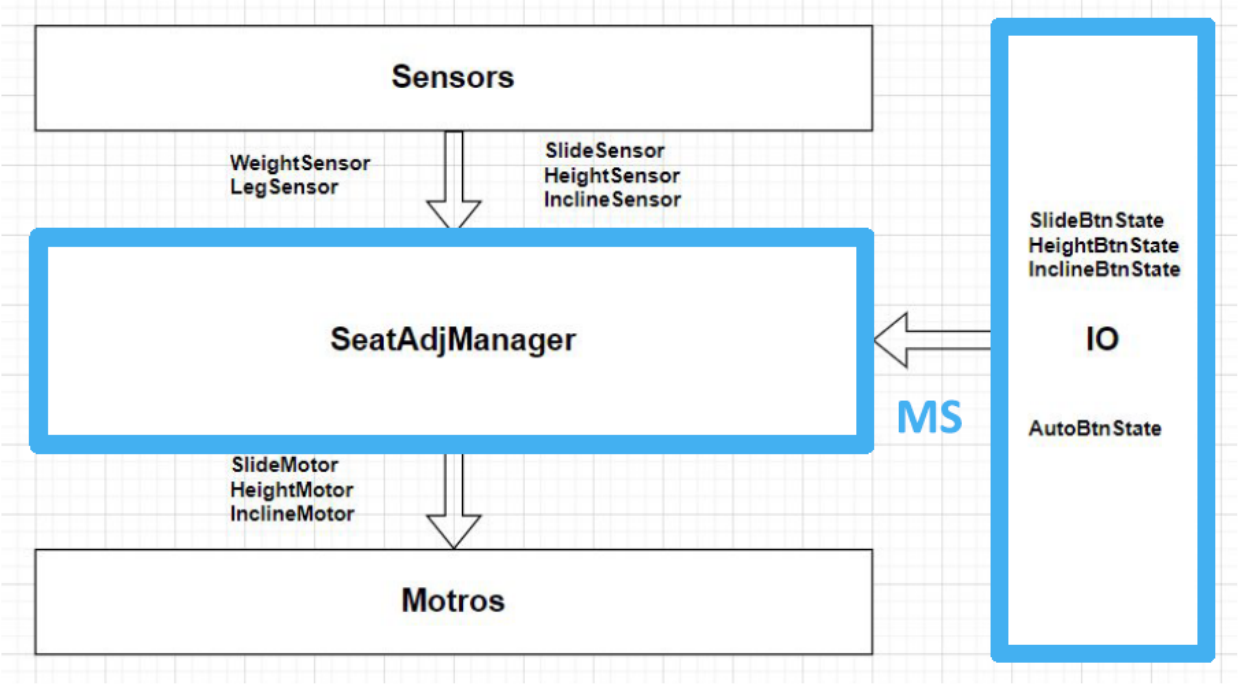
(3) Software Logic

- HeightSensor_GetPosition, SlideSensor_GetPosition, InclineSensor_GetPosition shall return position based on the following conditions:
  if(position == 0) , Position = SENSOR_POSITION_STEP_0
  if(position > 0 && position<= 64), Position = SENSOR_POSITION_STEP_1
  if(position > 64 && position<= 192), Position = SENSOR_POSITION_STEP_2
  if(position > 192 && position<=255), Position = SENSOR_POSITION_STEP_3

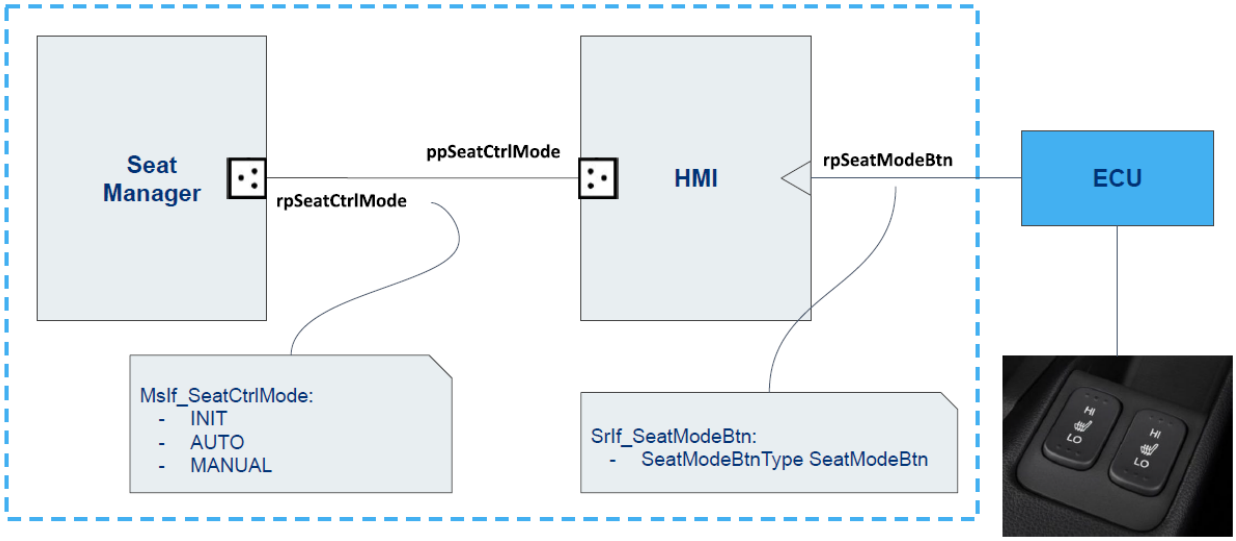
- WeightSensor_GetWeight shall return Weight as (weight/1000)

- HeightMotor_Move, InclineMotor_Move, SlideMotor_Move shall set motor forward/reverse according to Step (PLUS/MINUS)

F. Mode Switch Communication:



(1) High Level Design (Types, Port Interfaces, SWCs, Ports, Connections)



(2) Impl Level Design (Runnables, Access Points, Events)

- HMI_SeatModeChanged shall be triggered when SeatModeBtn is received
- HMI_SeatModeChanged shall receive SeatModeBtn
- HMI_SeatModeChanged shall switch SeatCtrlMode

- SeatManager_SetHeight, SeatManager_SetIncline, SeatManager_SetSlide shall be disabled if SeatCtrlMode is AUTO or INIT
- SeatManager_AutoHeight, SeatManager_AutoSlide, SeatManager_AutoIncline shall be disabled if SeatCtrlMode is MANUAL or INIT

(3) Software Logic

- HMI_SeatModeChanged shall switch SeatCtrlMode according to the below conditions:
  if(SeatModeBtn == SEAT_MODE_BTN_MANUAL) => Switch SeatCtrlMode to MANUAL
  if(SeatModeBtn == SEAT_MODE_BTN_AUTO) => Switch SeatCtrlMode to AUTO
  Otherwise => Switch SeatCtrlMode to INIT

G. RTE Generated Files:

- **Rte_Type.h**
- **Rte_<SWC>.h**
- **Rte_<SWC>_Type.h**
- **Rte_Hook.h**
- **Rte.c**