Session 9 - What is AUTOSAR, More on Why AUTOSAR for Automotive

AUTOSAR stands for AUTOmotive Open System ARchitecture. It's a worldwide development partnership of automotive OEMs (Original Equipment Manufacturers), Tier-1 suppliers, and software/hardware vendors. Its primary goal is to standardize the software architecture of Electronic Control Units (ECUs) in vehicles.

The automotive industry has faced significant challenges due to the increasing complexity of vehicle electronics and software. Here's a deeper dive into "Why AUTOSAR":

1. **Complexity Management**
2. **Increased Reusability**
3. **Improved Software Quality and Reliability**
4. **Enhanced Scalability**
5. **Reduced Cost**
6. **Better Interoperability**
7. **Support for Future Technologies**

→Keilumicrovision CODE

```
// MyWiperControlSWC.c - Example Application Software Component (SWC) for Wiper Control


#include "Rte_MyWiperControlSWC.h" // Generated RTE header for this SWC
#include "SchM_MyWiperControlSWC.h" // Generated Scheduler Manager header (for exclusive areas if needed)


// Internal state variables for the wiper control logic
static uint8 currentWiperMode = RTE_MODE_WiperMode_OFF; // Example enum from RTE
static uint16 intermittentCounter = 0;
static const uint16 INTERMITTENT_THRESHOLD = 3000; // 3 seconds in ms


// Runnable executed periodically (e.g., every 10ms) by the RTE
FUNC(void, MyWiperControlSWC_CODE) MyWiperControlSWC_MainFunction(void)
{
```

```
  // Read input from the user via the RTE port
  Rte_ModeType_WiperMode requestedMode;
  (void)Rte_Switch_RP_WiperModeRequest_WiperMode(&requestedMode); // Example: Reading a mode request

  // Read vehicle speed via RTE port
  uint16 vehicleSpeed;
  Std_ReturnType status = Rte_Read_RP_VehicleSpeed_VehicleSpeed(&vehicleSpeed);
  if (status != E_OK) {
    // Handle read error, e.g., use a default value or report diagnostic event
    vehicleSpeed = 0;
  }

  // --- Wiper Control Logic ---

  // Transition logic based on requestedMode
  if (requestedMode != currentWiperMode) {
    currentWiperMode = requestedMode;
    intermittentCounter = 0; // Reset counter on mode change
  }

  switch (currentWiperMode) {
    case RTE_MODE_WiperMode_OFF:
      // Set output port to OFF
      (void)Rte_Write_PP_WiperMotorControl_WiperSpeed(RTE_WIPER_SPEED_OFF);
      break;

    case RTE_MODE_WiperMode_INTERMITTENT:
      intermittentCounter += 10; // Assuming this runnable runs every 10ms
      if (intermittentCounter >= INTERMITTENT_THRESHOLD) {
```

```
            // Trigger a single wipe (this could be another runnable or a direct control)
            // For simplicity, we directly control here, but in a real system,
            // a "single wipe" might be another state or a function call
            (void)Rte_Write_PP_WiperMotorControl_WiperSpeed(RTE_WIPER_SPEED_LOW);
            // After some time, the motor control BSW or another SWC would detect park
            // and set the speed back to OFF.
            // For this example, we immediately schedule it off for simulation.
            // In a real system, the park sensor feedback would be critical here.
             //
(void)Rte_Write_PP_WiperMotorControl_WiperSpeed(RTE_WIPER_SPEED_OFF); // Will
be handled by park logic
            intermittentCounter = 0; // Reset counter
        }
        break;

    case RTE_MODE_WiperMode_LOW:
        (void)Rte_Write_PP_WiperMotorControl_WiperSpeed(RTE_WIPER_SPEED_LOW);
        break;

    case RTE_MODE_WiperMode_HIGH:
        // Example: Adjust high speed based on vehicle speed (simplified)
        if (vehicleSpeed > 100) { // If car is fast

(void)Rte_Write_PP_WiperMotorControl_WiperSpeed(RTE_WIPER_SPEED_HIGH_FAST); //
An even faster mode
        } else {
(void)Rte_Write_PP_WiperMotorControl_WiperSpeed(RTE_WIPER_SPEED_HIGH);
        }
        break;
    case RTE_MODE_WiperMode_MIST:
```

```
        // Similar to intermittent, but just one sequence
(void)Rte_Write_PP_WiperMotorControl_WiperSpeed(RTE_WIPER_SPEED_LOW);
        // After one cycle, return to OFF
        // (In a real system, this would involve a state machine or monitoring the park
sensor)
        currentWiperMode = RTE_MODE_WiperMode_OFF; // For simple demo
        break;
      // ... other modes like WASHER would have their own logic
      default:
        (void)Rte_Write_PP_WiperMotorControl_WiperSpeed(RTE_WIPER_SPEED_OFF);
        break;
  }
  // Potentially call a diagnostic service if an error condition is met
  // (void)Dem_SetEventStatus(DemConf_DemEventParameter_WiperMotorStall,
DEM_EVENT_STATUS_FAILED);
}
// Another Runnable for handling a direct Washer activation (e.g., short press)
FUNC(void, MyWiperControlSWC_CODE) MyWiperControlSWC_WashRequest(void)
{
  // This Runnable might be triggered by a specific event or data received via RTE
  // (void)Rte_Call_RP_WasherPumpControl_ActivatePump(); // Call a server port to
activate pump
  // (void)Rte_Write_PP_WiperMotorControl_WiperSpeed(RTE_WIPER_SPEED_LOW); //
Start post-wash wipes
  // ... logic for fixed number of wipes after wash
}
```