

KPIT

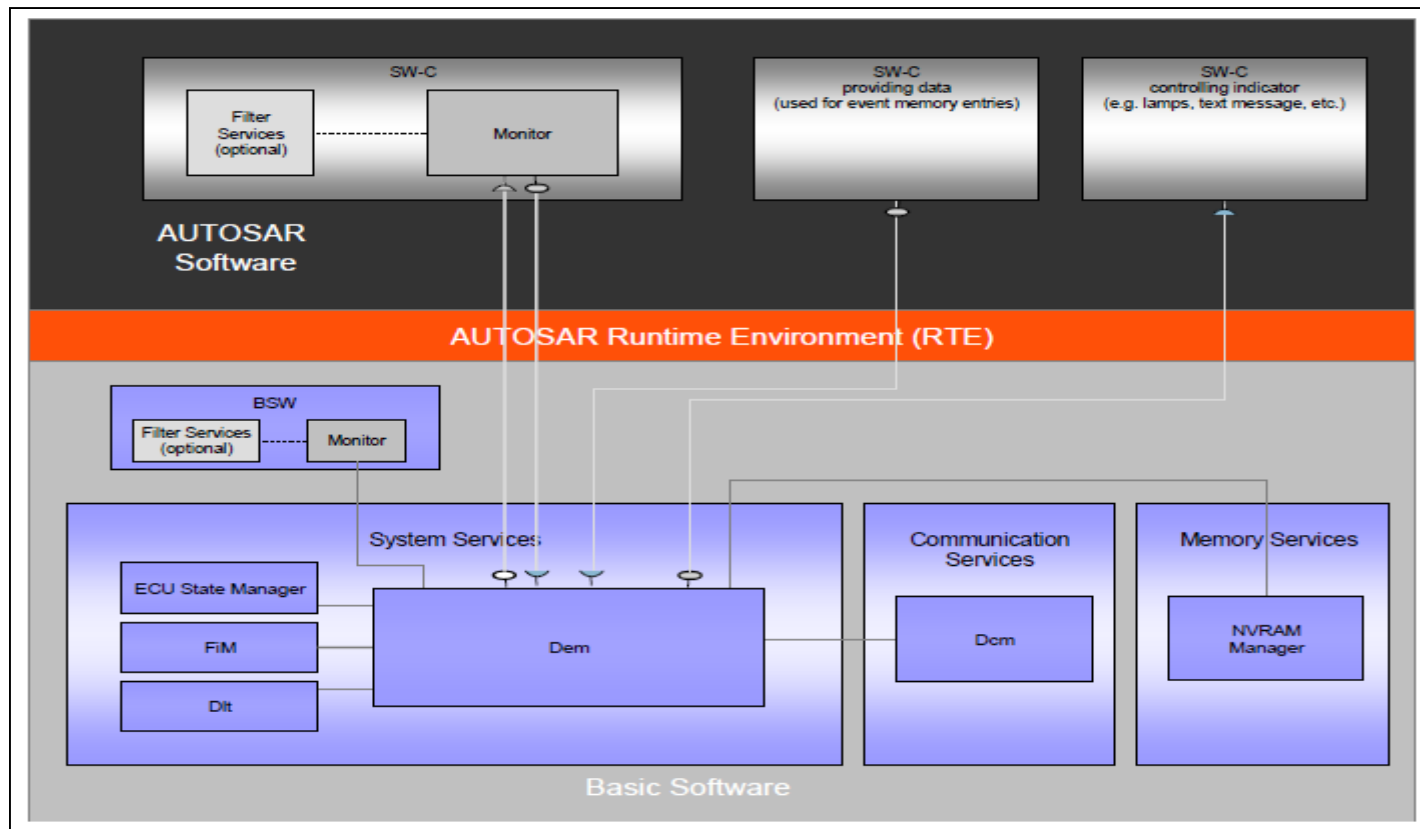
AUTOSAR DEM Overview

KPIT

Presentation on Dem

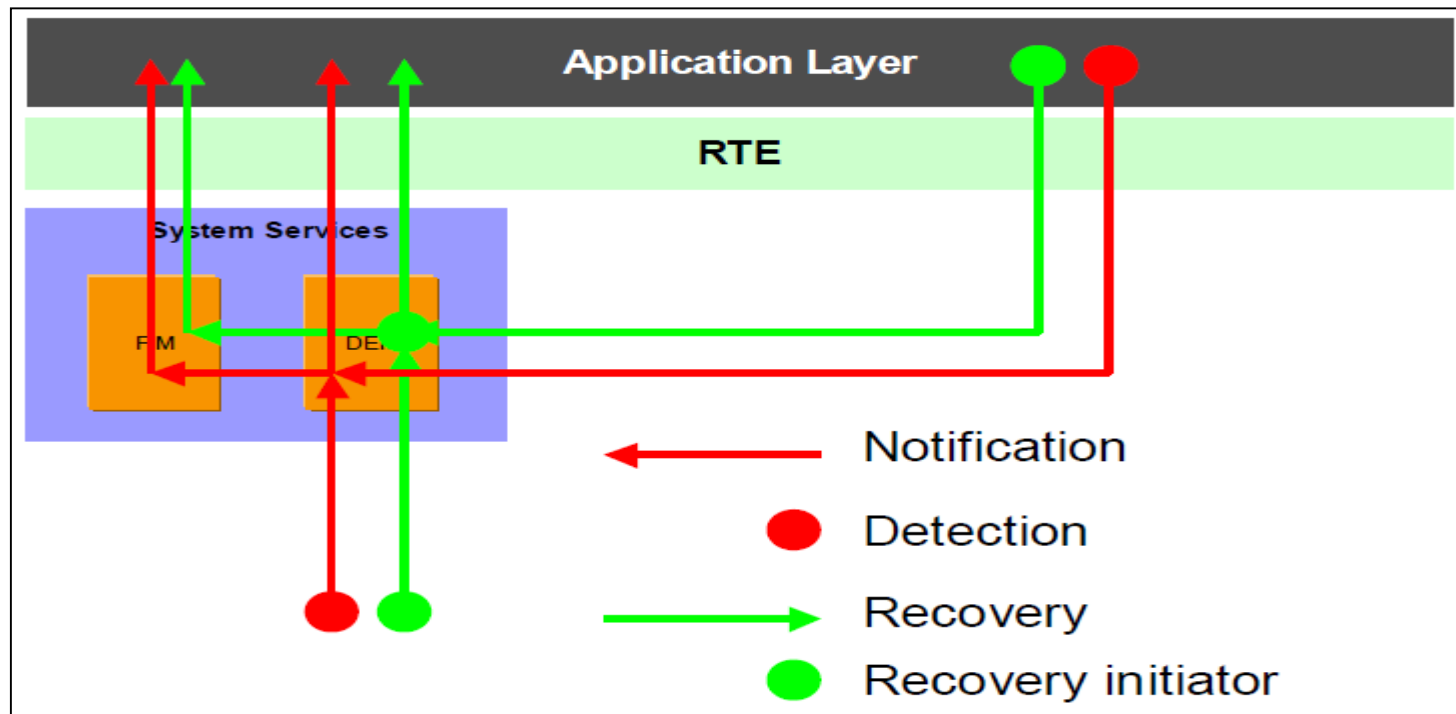
DEM – Diagnostics Event Manager

- The service component Diagnostic Event Manager (Dem) is responsible for processing and storing diagnostic events (errors) and its associated data
- The Dem offers interfaces to the application layer and to other BSW modules to make the stored event information available
- Dem provides fault information to the Dcm



Diagnostics Event

- A 'Diagnostic Event' defines the atomic unit that can be handled by the Dem module
- The status of a 'Diagnostic Event' represents the result of a monitor
- The Dem receives the result of a monitor from SW-C via the RTE or other BSW modules
- The Dem module uses the EventId to manage the status of the 'Diagnostic Event' of a system and performs the required actions for individual test results



➤ **Event priority**

- ✓ It is defined as a ranking of events based upon level of importance
- ✓ It is used to determine which fault entries may be removed from the event memory in the case of the number of stored events exceeds the maximum number of memory entries
- ✓ A priority value of 1 is the highest priority. Larger priority value shall define lower importance

➤ **Event occurrence**

- ✓ The Dem module provides an occurrence counter per event memory entry
- ✓ The Dem module increments the occurrence counter by one if the related event is already stored in the event memory and the UDS DTC status bit 0 (TestFailed) changes from 0 to 1
- ✓ If the configuration parameter *DemOccurrenceCounterProcessing* (in container DemGeneral) is DEM_PROCESS_OCCCTR_CDTC, the Dem module only increments the occurrence counter if the fault confirmation has been successfully finished

Elements of Diagnostics Event

➤ Event kind

- ✓ There are two different types of events:
 - BSW-related events (reported via C-API – Dem_ReportErrorStatus)
 - SW-C-related events (reported via RTE operation – SetEventStatus)

➤ Event significance

- ✓ There are two different significance levels of events:
 - “fault”: classifies a failure, which relates to the component/ECU itself (and requires for example a repair action)
 - “occurrence”: classifies an issue, which indicates additional information concerning insufficient system behavior (and relates for example to a condition out of the ECU’s control)

➤ Event destination

- ✓ The configuration parameter *DemEventDestination* (DemEventClass) defines the dedicated storage location(s) of the event and its related data
- ✓ The definition and use of the different memory types is OEM specific

➤ **Diagnostic monitor**

- ✓ A diagnostic monitor is a routine entity determining the proper functionality of a component
- ✓ The monitoring function identifies a specific fault type (e.g. short to ground, open load, etc.) for a monitoring path
- ✓ A monitoring path represents the physical system or a circuit, that is being monitored (e.g. sensor input). Each monitoring path is associated to exactly one diagnostic event.
- ✓ If the monitor debounces on its own, the reporting API is called only after a qualified result (passed or failed) is available
- ✓ If the monitor uses Dem-internal debouncing mechanism, the reporting API is called whenever the code with the functional check is executed

➤ Diagnostic trouble code

- ✓ There are two different kinds of DTCs:
 - non OBD-relevant DTCs (UDS DTCs)
 - OBD-relevant DTCs

➤ DTC groups

- ✓ The following DTC groups are provided:
 - emission-related DTC group (optional, fixed value = 0x000000)
 - powertrain DTC group (optional, configurable value)
 - chassis DTC group (optional, configurable value)
 - body DTC group (optional, configurable value)
 - network communication DTC group (optional, configurable value)
 - further user-defined DTC groups (optional, configurable value)
 - 'all DTCs' DTC group (mandatory, fixed value = 0xFFFFFFFF)

Operation cycle management

- ✓ The Dem module uses different operation cycles
- ✓ The cycles could either be provided by other BSW modules and SW-C or generated by the Dem module itself
- ✓ Examples of operation cycles are:
 - driving cycle
 - engine warm up cycle
 - ignition on/off cycle
 - power up/power down cycle
 - operation active/passive cycle
 - accumulated operating time
- ✓ The operation cycle management of the Dem module uses the reported state (DEM_CYCLE_STATE_START / DEM_CYCLE_STATE_END) of the API *Dem_SetOperationCycleState* (refer to chapter 8.3.3.6) to set the Dem specific operation cycle state (started / ended)

Event status management

- ✓ The 'Event Status Management' is the Dem's ability to record and retain events, event status and associated data
- ✓ The Dem module provides the capability to report the status of an event through the API ***Dem_SetEventStatus*** allowing a diagnostic monitor to inform the Dem about the result of the internal diagnostic test
- ✓ The Dem module provides the capability to reset the failed status of an event without reporting a passed result through the API ***Dem_ResetEventStatus***
- ✓ The Dem module provides the capability to retrieve the current UDS DTC status byte of a specific event through the API ***Dem_GetEventStatus***
- ✓ The Dem module provides the current event failed status through the API ***Dem_GetEventFailed***
- ✓ The Dem module provides the current event tested status through the API ***Dem_GetEventTested***

Status bit update

- ✓ In case a qualified diagnostic event (passed / failed) is reported to the Dem module, the Dem performs the event status transition immediately for the bits being relevant for fault reactions
 - Bit 0 TestFailed
 - Bit 1 TestFailedThisOperationCycle
 - Bit 4 TestNotCompletedSinceLastClear
 - Bit 5 TestFailedSinceLastClear
 - Bit 6 TestNotCompletedThisOperationCycle

Notification of status bit changes

- ✓ The Dem module triggers the event-specific callback-function ***EvenStatusChanged*** on each event status change
- ✓ The configuration container ***DemCallbackEventStatusChanged*** is used to specify one or more ports/c-callbacks per event

Debouncing of diagnostic events

- ✓ If the Dem module is configured to implement the debounce algorithm for a specific event, one of the following debounce algorithms are to be performed Dem-internally
 - ❖ *Counter based debounce algorithm*
 - ❖ *Time based debounce algorithm*
 - ❖ *Further specific debounce algorithms*
 - ❖ *Monitor internal debounce algorithm*
- ✓ The Dem module supports the event-specific configuration of debounce algorithms by using the configuration container *DemDebounceAlgorithmClass*

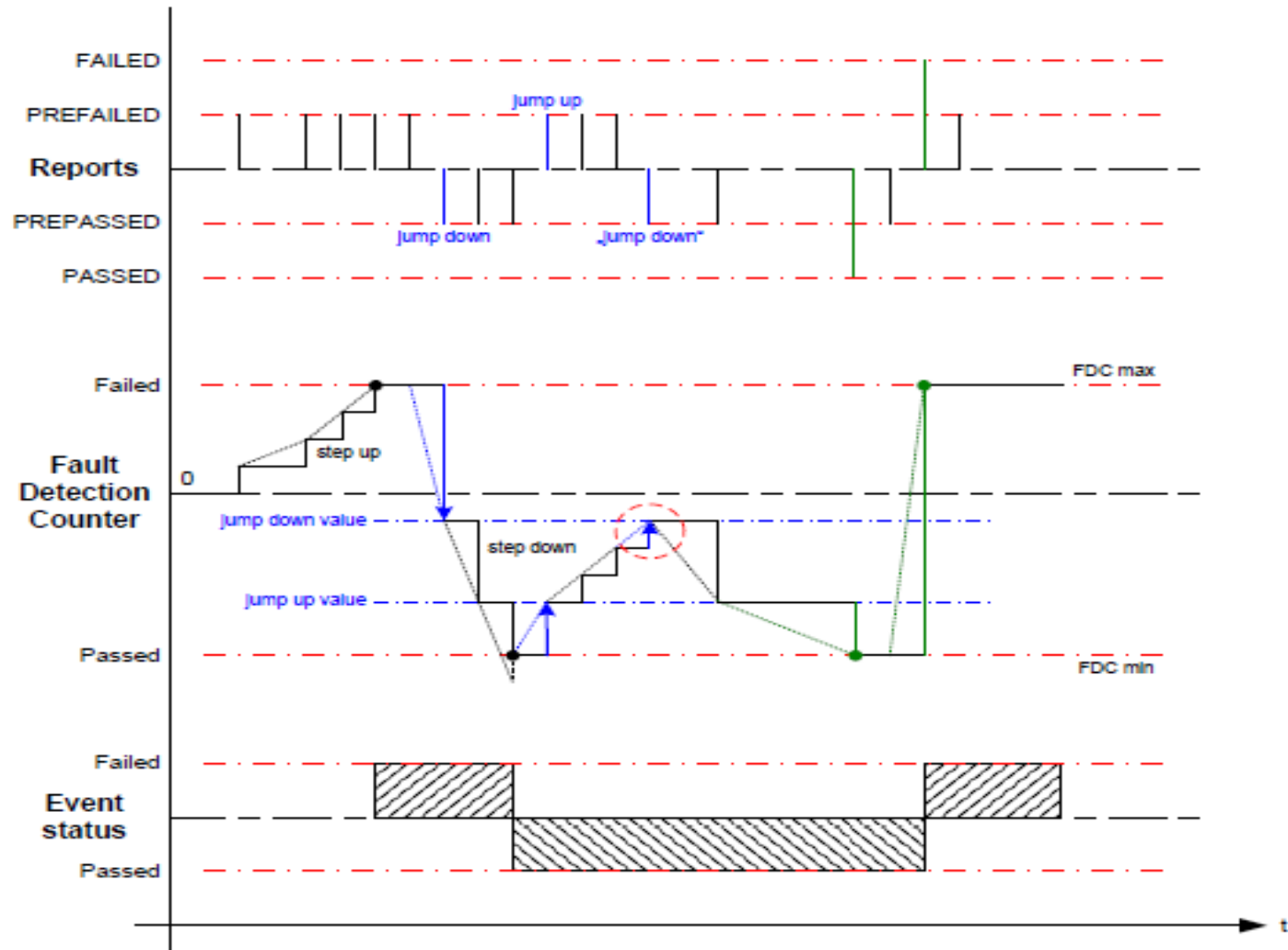
Counter based debounce algorithm

- ✓ The Dem module calculates the fault detection counter (-128 ...+127 according to UDS) based on the value and range of the internal debounce counter to map the internal counter values linearly to the external values
- ✓ *DemDebounceCounterFailedThreshold* defines the event-specific limit indicating the failed status (active)
- ✓ *DemDebounceCounterPassedThreshold* defines the event-specific limit indicating the passed status (passive)
- ✓ The Dem module increments the internal debounce counter with its configured step-size (*DemDebounceCounterIncrementStepSize*), when the monitor reports DEM_EVENT_STATUS_PREFAILED
- ✓ The Dem module decrements the internal debounce counter with its configured step-size (*DemDebounceCounterDecrementStepSize*), when the monitor reports DEM_EVENT_STATUS_PREPASSED
- ✓ If the monitor reports DEM_EVENT_STATUS_FAILED, the Dem module sets the internal debounce counter value to its configured threshold being the failed criteria
- ✓ If the monitor reports DEM_EVENT_STATUS_PASSED, the Dem module sets the internal debounce counter value to its configured threshold being the passed criteria

Counter based debounce algorithm contd.

- ✓ The Dem module provides the configuration parameter *DemDebounceCounterJumpDown* for activating or deactivating the jump down behavior
- ✓ If the jump down behavior is active, the Dem module provides the configuration parameter *DemDebounceCounterJumpDownValue* defining the new internal debounce counter init value if the counting direction changes from incrementing to decrementing
- ✓ The Dem module provides the configuration parameter *DemDebounceCounterJumpUp* for activating or deactivating the jump up behavior
- ✓ If the jump up behavior is active, the Dem module provides the configuration parameter *DemDebounceCounterJumpUpValue* defining the new internal debounce counter init value if the counting direction changes from decrementing to incrementing

Counter based debounce algorithm Example



Startup behavior

- ✓ The Dem module has a pre-initialization mode and a full-initialized mode (operation mode)
- ✓ The function Dem_PreInit initializes the internal states of the Dem module necessary to process events reported by BSW modules by using Dem_ReportErrorStatus
- ✓ The function Dem_Init is called during the startup phase of the ECU, after the NVRAM Manager has finished the restore of NVRAM data. SW-Components including monitors are initialized afterwards.
- ✓ The function Dem_Init reinitializes the Dem module after the Dem_Shutdown was called
- ✓ The Dem module provides the interface InitMonitorForEvent to trigger the initialization of a diagnostic monitor

- ✓ Beside application software components also the basic software (BSW) can detect errors (e.g. hardware driver faults), especially during startup
- ✓ Dem module provides the interface Dem_ReportErrorStatus to the BSW modules, to report BSW events
- ✓ Dem_ReportErrorStatus is used by BSW modules to report errors from the point in time when the Dem module is pre-initialized
- ✓ Within Dem_Init, the queued events are processed
- ✓ During normal operation (after full initialization), the queuing mechanism of the API Dem_ReportErrorStatus is necessary to process the reported fault within the main function of the Dem module
- ✓ Additional aspects
 - Errors can be detected before Dem is fully initialized
 - Errors can be reported during startup, information needs to be buffered until Dem is fully available
 - Errors can be reported between startup and shutdown, information needs to be buffered and need to be processed by the Dem main function (RTE related call tree requirement)
 - Entries in the event memory can have a different configuration (e.g. no emphasis on freeze frame data for the workshop)



Questions

Thank you

www.kpit.com

