

Run #1



You are a Requirements Engineer for the "Standard Firmware" project. I will provide a set of linked System Requirements and their corresponding Software Requirements. Your task is to thoroughly evaluate whether every aspect of the system requirement is sufficiently detailed in its linked software requirements. Note that the system and software requirements are restricted to their scope system and software respectively hence can have different levels of abstraction. The software requirements can contain additional details than its linked system requirements as long as it adds more details and is not unrelated to its system requirement. If the additional condition is not at all mentioned in the system requirement, then flag it. It is acceptable that one SYSRS can be linked to multiple SWRS or vice versa. Please stick to the information provided in the requirements alone. Don't assume anything that is not explicitly mentioned. Please provide a single combined rating for each set of System Requirements (SYSRS) and their corresponding Software Requirements (SWRS).

Condition Check: Verify if the conditions such as 'After event X occurs', 'before event Y', 'IF X THEN' in the system requirements are matched in the software requirements. Only these constitute to the term 'conditions'.

Action and Response Check: Confirm if the actions and responses described in the system requirements are correctly implemented in the software requirements.

Parameter Sources Check: Ensure that all parameters mentioned in the system requirements are addressed in the software requirements including information on where these parameters can be found or how they are sourced. If the source is not mentioned, indicate that it needs to be identified.

Additional Conditions Check: Identify if there are any additional conditions in the software requirements that are not specified in the system requirements. If they are not relevant to the system requirement, then flag it. Ensure that all these checks are given equal priority to avoid oversight. Every observation is critical, be it major or minor. Use this to provide rating:

For each set, provide the following ratings and reasons:

Then use your checks to rate the quality of the link.

Meaningfulness Rating (High/Medium/Low):

Reason for Meaningfulness Rating: List all the issues.

Completeness Rating (High/Medium/Low):

Reason for Completeness Rating: List all the issues.

Overall Quality of the Link (High/Medium/Low):

Reason for Overall Quality Rating:

If any aspects are not fully addressed or require clarification, suggest specific changes or revisions to the software requirements to better align with the system requirements.



Please provide me with the set of System Requirements (SYSRS) and their corresponding Software Requirements (SWRS) so I can evaluate them according to the checks you specified.



Meaningfulness:

High: The software requirement clearly and directly supports or implements the system requirement, adding significant value and relevance. The link between them is strong and logical.

Medium: The software requirement generally supports the system requirement, but there may be a need for better alignment or more specific details.

Low: The software requirement is poorly aligned or irrelevant to the system requirement, making the link weak or unclear.

Completeness:

High: The link between the system and software requirement is fully detailed, covering all necessary aspects needed for implementation. No significant information is missing.

Medium: The link is somewhat complete but lacks some critical details or clarifications that would improve understanding or implementation.

Low: The link is significantly incomplete or lacks essential details, making it difficult to see how the software requirement fully supports the system requirement.

Formula-Based Overall Quality:

High: If both Meaningfulness and Completeness are rated as High, the Overall Quality is High.

Medium: If either Meaningfulness or Completeness is Medium while the other is High, or if both are Medium, the Overall Quality is Medium.

Low: If either Meaningfulness or Completeness is Low, the Overall Quality is Low.



Thank you for clarifying the rating criteria. Please provide the system and software requirements so I can proceed with the evaluation as requested.



SYSRS 021: Diagnostic - Retry configuration: The firmware shall allow configurability to select the retry for open and short faults independently "SWRS 023: Open/Short retry: IF the software detects an open or short error AND retry is configured as enabled in the DEFAULT OPEN SHORT RETRY value in the COMPILE TIME CONFIGURATION data, the affected channel shall be re-enabled for one TIMESLICE with the following setup at least every 200 milliseconds

SWRS 024: Diagnostic configuration items : DEFAULT OPEN SHORT RETRY"



Evaluation of SYSRS 021 and Linked SWRS 023, SWRS 024

SYSRS 021:

- **Title:** Diagnostic - Retry configuration
- **Description:** The firmware shall allow configurability to select the retry for open and short faults independently.

SWRS 023:

- **Description:** IF the software detects an open or short error AND retry is configured as enabled in the DEFAULT OPEN SHORT RETRY value in the COMPILE TIME CONFIGURATION data, the affected channel shall be re-enabled for one TIMESLICE with the following setup at least every 200 milliseconds.

SWRS 024:

- **Description:** Diagnostic configuration items : DEFAULT OPEN SHORT RETRY.

Condition Check:

- **SYSRS 021 Condition:** No explicit condition like "IF" or "AFTER" is mentioned, but it implies configurability for retry conditions (open and short faults independently).
- **SWRS 023 Condition:** The condition is present: "IF the software detects an open or short error AND retry is configured." This matches the intent of SYSRS 021 but introduces a specific condition about compile-time configuration and timeslice intervals, which is more detailed than the system requirement.

Assessment:

- The software requirement introduces an additional condition related to "compile-time configuration" and "timeslice intervals" that is not mentioned in the system requirement but may add implementation details. However, the independent configurability for open and short faults is not explicitly discussed.

Action and Response Check:

- **SYSRS 021 Action:** The firmware should allow configurability for retries on open and short faults independently.
- **SWRS 023 Action:** This describes enabling retries if a fault (open/short) is detected and retries are configured, but it doesn't clearly explain how the independent selection of open and short retries is supported.
- **SWRS 024 Action:** Refers to the "DEFAULT OPEN SHORT RETRY," which could represent some form of configurability, but it does not explicitly explain the independent nature of open and short retries, as SYSRS 021 demands.

Assessment:

- The software requirements do not clearly detail how the independent configurability for open and short retries is achieved. Instead, SWRS 023 and SWRS 024 focus on generic retry behavior and default configuration, without addressing independent selection explicitly.

Parameter Sources Check:

- **SYSRS 021 Parameters:** No explicit parameters are defined other than "retry for open and short faults."
- **SWRS 023 and SWRS 024 Parameters:** SWRS 023 refers to "DEFAULT OPEN SHORT RETRY" and "COMPILE TIME CONFIGURATION data." These represent parameters relevant to configuring retries but do not indicate where independent retry options for open and short faults come from.

Assessment:

- While the software requirements introduce a configuration parameter (DEFAULT OPEN SHORT RETRY), the source of the independent open/short fault retry is not clarified. It is unclear if there are separate configuration items for each fault type.

Additional Conditions Check:

- **SWRS 023:** The mention of retry intervals (every 200 milliseconds) and the "compile-time configuration" are additional conditions not mentioned in SYSRS 021. These additions seem relevant to the implementation but add complexity that SYSRS 021 does not hint at.
- **SWRS 024:** The description of "DEFAULT OPEN SHORT RETRY" is relevant, but it does not clarify how independent retries are handled.

Assessment:

- Additional details like retry timing and compile-time configurations are included, but no explanation for independent open and short retries is provided.

Ratings:

- **Meaningfulness Rating: Medium**
 - **Reason:** While SWRS 023 and SWRS 024 generally support retry configuration, they fail to address the independent configurability of open and short faults clearly. The software requirements add conditions like compile-time configuration and retry intervals, which are not explicitly stated in the system requirement, yet they may be important for implementation. The independent configurability, however, remains under-explained.
- **Completeness Rating: Medium**
 - **Reason:** The software requirements introduce valuable details (compile-time configuration and retry intervals) but lack clarity on how the system requirement's focus on "independent configurability" is implemented. It is unclear how the user can select open or short retries separately.
- **Overall Quality Rating: Medium**
 - **Reason:** The link between the system and software requirements is somewhat complete but misses a critical aspect of independent configurability. It could be improved by specifying how retries for open and short faults can be selected independently and clarifying whether separate parameters or configuration items are involved.

Suggested Revisions:

- **SWRS 023 and SWRS 024** should clarify the configurability of open and short faults independently. Specifically, they should address whether separate configuration parameters exist for each fault type (open and short) and how these parameters can be modified.
- If relevant, specify in the software requirements how the user can independently enable/disable retries for open or short faults, ensuring the system's configurability goal is met.



```
"SYSRS 012: BEFORE Leaving the Discovery Mode
AND IF the effective value of the parameter DSI3_physicalAddress is NOT equal to the default
value of the Parameter DSI3_physicalAddress
THEN
the Sensor IC shall
set Error: err_discPhysAddrMismatch = 0x1 ("error detected").
"
"SWRS 012 : BEFORE Leaving the Discovery Mode
AND IF the effective value of the Parameter DSI3_physicalAddress (RAM) is NOT equal to the
default value of the Parameter DSI3_physicalAddress (NVM)
THEN
the Software shall
set Error: err_discPhysAddrMismatch = 0x1 ("error detected").
OTHERWISE
the Software shall
set Error: err_discPhysAddrMismatch = 0x0 ("no error").
"
```



Evaluation of SYSRS 012 and Linked SWRS 012

SYSRS 012:

- **Title:** BEFORE Leaving the Discovery Mode, check DSI3_physicalAddress
- **Description:**
 - **Condition:** IF the effective value of the parameter `DSI3_physicalAddress` is NOT equal to the default value of `DSI3_physicalAddress`.
 - **Action:** The Sensor IC shall set `Error: err_discPhysAddrMismatch = 0x1` ("error detected").

SWRS 012:

- **Description:**
 - **Condition:** IF the effective value of the parameter `DSI3_physicalAddress` (RAM) is NOT equal to the default value of the parameter `DSI3_physicalAddress` (NVM).
 - **Action:** The Software shall:
 - Set `Error: err_discPhysAddrMismatch = 0x1` ("error detected").
 - Otherwise, set `Error: err_discPhysAddrMismatch = 0x0` ("no error").

Condition Check:

- **SYSRS 012 Condition:**
 - The system requirement checks if the effective value of `DSI3_physicalAddress` is not equal to the default value before leaving Discovery Mode.
- **SWRS 012 Condition:**
 - SWRS 012 introduces a comparison between the RAM-stored value and the NVM-stored value of the `DSI3_physicalAddress`. The condition to check the values before leaving Discovery Mode matches the system requirement, but the software requirement introduces specific details regarding RAM and NVM, which are not explicitly stated in the system requirement.

Assessment:

- The conditions match in intent, but the software requirement adds the specifics of comparing RAM and NVM values, which are not mentioned in SYSRS 012. This additional detail may be part of implementation, but the RAM/NVM distinction is an enhancement not indicated by the system requirement.

Action and Response Check:

- **SYSRS 012 Action:**
 - If the values differ, set ``err_discPhysAddrMismatch = 0x1``.
- **SWRS 012 Action:**
 - SWRS 012 correctly implements the action of setting ``err_discPhysAddrMismatch = 0x1`` if there is a mismatch. However, it adds an additional action: setting the error to ``0x0`` if no mismatch is detected, which is not mentioned in the system requirement.

Assessment:

- The main action of setting the error to ``0x1`` on a mismatch is correctly implemented, but the software requirement also includes an additional "no error" condition (``0x0``), which is not part of the system requirement. This additional behavior is logical but was not explicitly required by SYSRS 012.

Parameter Sources Check:

- **SYSRS 012 Parameters:**
 - The system requirement mentions ``DSI3_physicalAddress`` without specifying where the values are sourced from.
- **SWRS 012 Parameters:**
 - SWRS 012 clarifies the source of the parameter values by specifying RAM (effective value) and NVM (default value). This is an additional level of detail not provided in SYSRS 012.

Assessment:

- The software requirement adds valuable clarity by specifying the RAM and NVM sources for the ``DSI3_physicalAddress`` values, which enhances the understanding of how the comparison should be performed. The system requirement does not specify this.

Additional Conditions Check:

- **SWRS 012 Additional Condition:**
 - The software requirement includes an additional condition for setting ``err_discPhysAddrMismatch = 0x0`` when there is no error detected. This is not mentioned in SYSRS 012 but adds completeness to the software behavior.

Assessment:

- The additional condition for "no error" is not specified in the system requirement but logically complements the error detection process. However, since it was not mentioned in SYSRS 012, it represents a deviation that might need clarification.

Ratings:

- **Meaningfulness Rating: High**

- **Reason:** The software requirement meaningfully implements the system requirement by correctly checking for a mismatch in the ``DSI3_physicalAddress`` and setting the appropriate error flag. The additional details about RAM and NVM clarify the sources of the values, adding specificity without deviating from the intent. The additional action (setting error to ``0x0``) enhances the functionality but could be considered part of robust software design.

- **Completeness Rating: Medium**

- **Reason:** While the software requirement addresses the core action of error detection, it includes additional details (RAM vs. NVM comparison and setting the error to ``0x0``) that are not mentioned in the system requirement. The system requirement does not describe what should happen if no error is detected, which leaves a gap in understanding the full expected behavior from the system's perspective. The sources for the parameters (RAM and NVM) are also additional information not present in SYSRS 012 but help the software's completeness.

- **Overall Quality Rating: Medium**

- **Reason:** The software requirement accurately reflects the core action of the system requirement and adds important details for implementation, but the additional condition (setting ``err_discPhysAddrMismatch = 0x0``) and the RAM/NVM distinction could be seen as outside the strict scope of SYSRS 012. To achieve a higher rating, the system requirement could be expanded to reflect these additional implementation details, or the software requirement could better clarify that these are implementation-specific choices rather than necessary conditions.

Suggested Revisions:

- **SYSRS 012** could be updated to specify the behavior when no error is detected, aligning it with the software requirement's handling of both error and no-error states.
- If the RAM and NVM distinction is an important part of the design, this could also be included in SYSRS 012 to provide clearer guidance for implementation.
- Alternatively, **SWRS 012** could note that the "no error" condition is an implementation-specific enhancement, ensuring that it doesn't create confusion about the system requirement's expectations.



"SYSRS 010: If Performing Memory Area Modification
AND AFTER reception of CRM_CMD 0x9: Take Data or Send Data
THEN
the Sensor IC shall
cache the CRM_CMD 0x9 Parameter Data to the Data Cache.

SYSRS 011: If Performing Memory Area Modification
AND AFTER reception of CRM_CMD 0x9: Take Data or Send Data
THEN
the Sensor IC shall

increment the CRM_CMD 0x8 Parameter StartAddress by 1 word address.

" "SWRS 011: If Performing Memory Area Modification
AND AFTER reception of CRM_CMD 0x9: Take Data or Send Data
THEN
the Software shall store the CRM_CMD 0x9 Parameter Data to RAM until the command CRM_CMD
0xA: Perform Memory Operation with CRM_CMD 0xA Parameter MemoryOperation == 0x2 ""Write
to NVM"" is received."



Evaluation of SYSRS 010, SYSRS 011 and Linked SWRS 011

SYSRS 010:

- **Condition:** IF performing memory area modification AND after reception of CRM_CMD 0x9 (Take Data or Send Data).
- **Action:** The Sensor IC shall cache the CRM_CMD 0x9 Parameter Data to the Data Cache.

SYSRS 011:

- **Condition:** IF performing memory area modification AND after reception of CRM_CMD 0x9 (Take Data or Send Data).
- **Action:** The Sensor IC shall increment the CRM_CMD 0x8 Parameter StartAddress by 1 word address.

SWRS 011:

- **Condition:** IF performing memory area modification AND after reception of CRM_CMD 0x9 (Take Data or Send Data).
- **Action:** The Software shall store the CRM_CMD 0x9 Parameter Data to RAM until the command CRM_CMD 0xA (Perform Memory Operation) with CRM_CMD 0xA Parameter MemoryOperation == 0x2 ("Write to NVM") is received.

Condition Check:

- **SYSRS 010 Condition:**
 - The system requirement states that after receiving CRM_CMD 0x9, data must be cached.
- **SYSRS 011 Condition:**
 - Similarly, the condition for SYSRS 011 is also tied to receiving CRM_CMD 0x9.
- **SWRS 011 Condition:**
 - The condition matches, as it also checks if CRM_CMD 0x9 has been received and memory area modification is being performed.

Assessment:

- The conditions in SWRS 011 match the system requirements SYSRS 010 and SYSRS 011 in terms of performing memory area modification and reacting after CRM_CMD 0x9 is received. The conditions align properly between the system and software requirements.

Action and Response Check:

- **SYSRS 010 Action:**
 - The system requirement instructs that the CRM_CMD 0x9 parameter data should be cached in the Data Cache.
- **SYSRS 011 Action:**
 - The system requirement instructs that the CRM_CMD 0x8 Parameter StartAddress should be incremented by 1 word address.
- **SWRS 011 Action:**
 - The software requirement instructs that the CRM_CMD 0x9 parameter data should be stored in RAM until another command (CRM_CMD 0xA with MemoryOperation == 0x2) is received to perform a write to NVM.

Assessment:

- **SYSRS 010 vs. SWRS 011:**
 - SWRS 011 does not mention the "Data Cache" as specified in SYSRS 010. Instead, it refers to storing the data in RAM, which might be an implementation detail, but the use of "Data Cache" vs. "RAM" represents a deviation from the system requirement. Additionally, SWRS 011 adds a condition where the data is stored until CRM_CMD 0xA is received, which is not mentioned in SYSRS 010. This is an extra action that wasn't specified in the system requirement.
- **SYSRS 011 vs. SWRS 011:**
 - SWRS 011 does not mention incrementing CRM_CMD 0x8 Parameter StartAddress by 1 word, as required by SYSRS 011. The action described in SYSRS 011 is missing in SWRS 011, which results in an incomplete link.

Parameter Sources Check:

- **SYSRS 010 Parameters:**
 - The parameter "CRM_CMD 0x9 Parameter Data" is mentioned, and it is expected to be cached in the Data Cache.
- **SYSRS 011 Parameters:**
 - The parameter "CRM_CMD 0x8 Parameter StartAddress" is mentioned, and it must be incremented by 1 word address.
- **SWRS 011 Parameters:**
 - SWRS 011 mentions "CRM_CMD 0x9 Parameter Data" but does not mention "CRM_CMD 0x8 Parameter StartAddress," which is required by SYSRS 011. It also introduces a new parameter (CRM_CMD 0xA and MemoryOperation), which is not mentioned in either SYSRS 010 or SYSRS 011.

Assessment:

- While SWRS 011 correctly references the CRM_CMD 0x9 Parameter Data, it fails to address CRM_CMD 0x8 Parameter StartAddress. It also introduces the CRM_CMD 0xA and MemoryOperation parameters, which are not part of the system requirements. These added parameters might be relevant for implementation but are not aligned with SYSRS 010 or SYSRS 011.

Additional Conditions Check:

- **SWRS 011 Additional Condition:**

- SWRS 011 introduces an additional condition: data is stored in RAM until CRM_CMD 0xA with MemoryOperation == 0x2 ("Write to NVM") is received. This behavior is not described in SYSRS 010 or SYSRS 011.

Assessment:

- The additional condition in SWRS 011, which involves the CRM_CMD 0xA command and storing data in RAM until a specific memory operation is triggered, adds complexity not mentioned in SYSRS 010 or SYSRS 011. This could either represent an important implementation detail or an unnecessary deviation from the original system requirements.

Ratings:

- **Meaningfulness Rating: Low**
 - **Reason:** SWRS 011 diverges from both SYSRS 010 and SYSRS 011 by introducing new parameters (CRM_CMD 0xA and MemoryOperation) and actions (storing in RAM instead of Data Cache). Furthermore, it omits a critical part of SYSRS 011 (incrementing the StartAddress). The link between the system and software requirements is weak due to these discrepancies.
- **Completeness Rating: Low**
 - **Reason:** SWRS 011 does not fully implement the actions required by SYSRS 010 and SYSRS 011. Specifically, it fails to mention caching the data in the Data Cache (as per SYSRS 010) and omits the required incrementing of CRM_CMD 0x8 Parameter StartAddress (as per SYSRS 011). The addition of unrelated conditions further detracts from completeness.
- **Overall Quality Rating: Low**
 - **Reason:** The significant mismatch between the system and software requirements, both in terms of omitted actions and additional conditions, results in a low overall quality. The software requirement does not fully address the necessary system behavior and introduces complexity that is not explained in the system requirements.

Suggested Revisions:

- **SWRS 011** should:
 - Clearly specify that the CRM_CMD 0x9 Parameter Data is cached in the Data Cache, as per SYSRS 010.
 - Include the missing action of incrementing the CRM_CMD 0x8 Parameter StartAddress by 1 word address, as required by SYSRS 011.
 - Remove or clarify the reference to CRM_CMD 0xA and MemoryOperation, as these parameters are not mentioned in the system requirements. If they are necessary for implementation, they should be documented as part of the overall system design rather than adding them arbitrarily in the software requirement.



"SYSRS 020: If Performing Memory Access in Broadcast
AND AFTER reception of the CRM_FCC_CMD,

AND If Addressed with the Unicast Command
 THEN
 the Sensor IC shall
 set the parameter Status Value = 0x2 ("invalid")." "SWRS 022: If Performing Memory Access in Broadcast
 AND AFTER reception of the CRM_FCC_CMD,
 AND If Addressed with the Broadcast Command
 THEN
 the Software shall
 set the parameter Status Value = 0x2 ("invalid")."



Evaluation of SYSRS 020 and Linked SWRS 022

SYSRS 020:

- **Condition:**
 - **IF performing memory access in broadcast,**
 - **AND AFTER reception of the CRM_FCC_CMD,**
 - **AND IF addressed with the Unicast Command**
- **Action:**
 - The Sensor IC shall set the parameter `Status Value = 0x2` ("invalid").

SWRS 022:

- **Condition:**
 - **IF performing memory access in broadcast,**
 - **AND AFTER reception of the CRM_FCC_CMD,**
 - **AND IF addressed with the Broadcast Command**
- **Action:**
 - The Software shall set the parameter `Status Value = 0x2` ("invalid").

Condition Check:

- **SYSRS 020 Condition:**
 - The system requirement explicitly mentions the condition where memory access is performed in broadcast mode, and after receiving the CRM_FCC_CMD, the IC is addressed with a **Unicast Command**.
- **SWRS 022 Condition:**
 - The software requirement also mentions the memory access in broadcast mode and after receiving the CRM_FCC_CMD, but it checks for a **Broadcast Command** instead of a **Unicast Command** as in SYSRS 020.

Assessment:

- There is a mismatch between the conditions in the system and software requirements. SYSRS 020 refers to addressing with a **Unicast Command**, whereas SWRS 022 refers to addressing with a **Broadcast Command**. This discrepancy in the condition is significant because it affects the triggering behavior of setting the status value to invalid.

Action and Response Check:

- **SYSRS 020 Action:**
 - The system requirement specifies that the Sensor IC should set the parameter ``Status Value = 0x2`` when the specified conditions are met.
- **SWRS 022 Action:**
 - The software requirement similarly instructs that the parameter ``Status Value = 0x2`` be set under its specified conditions.

Assessment:

- Both the system and software requirements correctly specify setting ``Status Value = 0x2`` as the action. However, due to the condition mismatch (Unicast vs. Broadcast command), the actions are triggered under different circumstances, which could lead to incorrect behavior.

Parameter Sources Check:

- **SYSRS 020 Parameters:**
 - The key parameter involved is ``Status Value``, which should be set to ``0x2`` when the IC is addressed with a Unicast Command.
- **SWRS 022 Parameters:**
 - SWRS 022 references the same parameter (``Status Value``), but it is set under different conditions (when addressed with a Broadcast Command).

Assessment:

- Both requirements correctly reference the same parameter (``Status Value``), but the condition mismatch (Unicast vs. Broadcast) creates a discrepancy in when this parameter is modified.

Additional Conditions Check:

- **SWRS 022 Additional Condition:**
 - SWRS 022 does not introduce any additional conditions but replaces the "Unicast Command" with a "Broadcast Command," which changes the meaning of the requirement.

Assessment:

- While there are no completely new conditions, the replacement of "Unicast" with "Broadcast" significantly alters the logic of the software requirement.

Ratings:

- **Meaningfulness Rating: Low**
 - **Reason:** The software requirement (SWRS 022) does not properly align with the system requirement (SYSRS 020). The system requirement expects the action to be taken when a **Unicast Command** is received, while the software requirement takes action on receiving a **Broadcast Command**. This mismatch in the condition makes the link weak and unclear.
- **Completeness Rating: Low**

- **Reason:** The software requirement fails to address the scenario described in the system requirement (Unicast Command). Therefore, it does not fully implement the system requirement's intent, making the link incomplete.

- **Overall Quality Rating: Low**

- **Reason:** The overall quality of the link is low due to the critical mismatch in the condition (Unicast vs. Broadcast). This discrepancy significantly affects the meaningfulness and completeness of the software requirement's alignment with the system requirement.

Suggested Revisions:

- **SWRS 022** should be revised to:
 - Align with the condition in **SYSRS 020** by checking for a **Unicast Command** instead of a Broadcast Command.
 - Ensure that the action of setting ``Status Value = 0x2`` occurs under the correct condition as specified in SYSRS 020.

This change would ensure that the software behavior correctly reflects the system requirement's intent and avoids potential implementation errors.



```
"SYSRS 019: If Performing Memory Access in Unicast
AND AFTER reception of the CRM_FCC_CMD,
AND If Addressed with the Broadcast Command
THEN
the Sensor IC shall
set the parameter Status Value = 0x2 ("invalid")."    "SWRS 021: If Performing Memory Access in
Unicast
AND AFTER reception of the CRM_FCC_CMD,
AND If Addressed with the Broadcast Command
THEN
the Software shall
set the parameter Status Value = 0x2 ("invalid")."
```



Evaluation of SYSRS 019 and Linked SWRS 021

SYSRS 019:

- **Condition:**
 - **IF performing memory access in Unicast,**
 - **AND AFTER reception of the CRM_FCC_CMD,**
 - **AND IF addressed with the Broadcast Command.**
- **Action:**
 - The Sensor IC shall set the parameter ``Status Value = 0x2`` ("invalid").

SWRS 021:

- **Condition:**
 - **IF performing memory access in Unicast,**
 - **AND AFTER reception of the CRM_FCC_CMD,**
 - **AND IF addressed with the Broadcast Command.**
- **Action:**
 - The Software shall set the parameter ``Status Value = 0x2`` ("invalid").

Condition Check:

- **SYSRS 019 Condition:**
 - The system requirement specifies that memory access is performed in Unicast mode, and after receiving the CRM_FCC_CMD, the Sensor IC is addressed with a **Broadcast Command**.
- **SWRS 021 Condition:**
 - The software requirement exactly mirrors the condition described in SYSRS 019, specifying that memory access is performed in Unicast mode, and after receiving the CRM_FCC_CMD, the system is addressed with a **Broadcast Command**.

Assessment:

- The conditions in SWRS 021 are a direct match with those in SYSRS 019, indicating complete alignment on the conditions that trigger the action.

Action and Response Check:

- **SYSRS 019 Action:**
 - The system requirement instructs the Sensor IC to set the ``Status Value`` to ``0x2`` ("invalid") when the specified conditions are met.
- **SWRS 021 Action:**
 - The software requirement specifies the same action—setting the ``Status Value`` to ``0x2`` ("invalid")—under the same conditions.

Assessment:

- The action in SWRS 021 exactly matches the action in SYSRS 019, ensuring that the software requirement properly implements the system requirement's directive.

Parameter Sources Check:

- **SYSRS 019 Parameters:**
 - The key parameter referenced is ``Status Value``, which should be set to ``0x2`` when the specified conditions are met.
- **SWRS 021 Parameters:**
 - The software requirement references the same parameter, ``Status Value``, and correctly specifies that it should be set to ``0x2`` under the same conditions.

Assessment:

- Both SYSRS 019 and SWRS 021 reference the same parameter, and its handling is consistent across both requirements.

Additional Conditions Check:

- **SWRS 021 Additional Condition:**

- There are no additional conditions introduced in SWRS 021. The software requirement is consistent with the system requirement.

Assessment:

- No additional conditions are introduced, meaning the software requirement is fully aligned with the system requirement.

Ratings:

- **Meaningfulness Rating: High**

- **Reason:** The software requirement directly supports the system requirement. The conditions and actions are a perfect match, ensuring that the software correctly implements the system behavior.

- **Completeness Rating: High**

- **Reason:** The software requirement fully addresses the system requirement, with no missing details or conditions. The implementation is complete and aligns perfectly with the system requirement.

- **Overall Quality Rating: High**

- **Reason:** The alignment between SYSRS 019 and SWRS 021 is strong and logical, with both the conditions and actions being identical. This ensures that the system behavior is accurately implemented in the software.

Suggested Revisions:

- No revisions are necessary, as SWRS 021 fully meets the expectations set by SYSRS 019.



"SYSRS 018: - Unlocked, If No Key Is Set
If Performing Memory Access
AND If the Key is NOT set
THEN
the Sensor IC shall
set the state of the Memory Area identified by CRM_CMD 0x8 Parameter MemoryArea to the state
Memory Area State: Unlocked." "SWRS 020: - locked, If Key Is Set
If Performing Memory Access
AND If the Key is set
THEN
the Software shall
set the state of the Memory Area identified by CRM_CMD 0x8 Parameter MemoryArea to the state
Memory Area State: locked."



Evaluation of SYSRS 018 and Linked SWRS 020

SYSRS 018:

- **Condition:**
 - **IF performing memory access,**
 - **AND IF the key is NOT set.**
- **Action:**
 - The Sensor IC shall set the state of the memory area identified by the CRM_CMD 0x8 Parameter MemoryArea to the state **Memory Area State: Unlocked.**

SWRS 020:

- **Condition:**
 - **IF performing memory access,**
 - **AND IF the key is set.**
- **Action:**
 - The Software shall set the state of the memory area identified by the CRM_CMD 0x8 Parameter MemoryArea to the state **Memory Area State: locked.**

Condition Check:

- **SYSRS 018 Condition:**
 - The system requirement specifies that memory access is performed **if the key is not set.**
- **SWRS 020 Condition:**
 - The software requirement, on the other hand, specifies that memory access is performed **if the key is set.**

Assessment:

- There is a clear difference between the conditions of the system and software requirements. SYSRS 018 checks the absence of the key (unlocked state), whereas SWRS 020 checks the presence of the key (locked state). The conditions are inverse to one another. While these may describe different scenarios, the software requirement does not address the condition described in SYSRS 018.

Action and Response Check:

- **SYSRS 018 Action:**
 - If the key is not set, the system requirement specifies that the state of the memory area should be set to **Unlocked.**
- **SWRS 020 Action:**
 - If the key is set, the software requirement specifies that the state of the memory area should be set to **Locked.**

Assessment:

- While the actions in both requirements are similar in that they modify the state of the memory area, they reflect opposite conditions (locked vs. unlocked). SWRS 020 does not implement the action required by SYSRS 018 (setting the memory area to "Unlocked" when no key is set).

Parameter Sources Check:

- **SYSRS 018 Parameters:**
 - The key parameter mentioned is the **CRM_CMD 0x8 Parameter MemoryArea**, and its state should be set to "Unlocked."
- **SWRS 020 Parameters:**
 - SWRS 020 references the same parameter, **CRM_CMD 0x8 Parameter MemoryArea**, but sets its state to "Locked."

Assessment:

- Both requirements reference the same parameter, but they modify the state in opposite ways (locked vs. unlocked), depending on the condition. However, SWRS 020 does not address the scenario where the key is not set, as required by SYSRS 018.

Additional Conditions Check:

- **SWRS 020 Additional Condition:**
 - SWRS 020 introduces no additional conditions. It merely handles the inverse scenario of SYSRS 018.

Assessment:

- No new conditions are introduced, but the software requirement does not address the specific scenario described in SYSRS 018.

Ratings:

- **Meaningfulness Rating: Low**
 - **Reason:** SWRS 020 addresses a different condition (key is set) than SYSRS 018 (key is not set). While the conditions are related, SWRS 020 does not support or implement the action described in SYSRS 018, making the link weak and misaligned.
- **Completeness Rating: Low**
 - **Reason:** The software requirement fails to address the scenario described in SYSRS 018, where the memory area should be unlocked if the key is not set. As a result, it does not fully implement the required behavior, leading to incomplete coverage of the system requirement.
- **Overall Quality Rating: Low**
 - **Reason:** The mismatch between SYSRS 018 and SWRS 020 in terms of condition (key not set vs. key set) and action (unlocked vs. locked) results in a poor link. The software requirement does not fulfill the intent of the system requirement, making the overall quality low.

Suggested Revisions:

- **SWRS 020** should be revised to handle the scenario where the key is not set, as required by **SYSRS 018**. A new software requirement should be added or an adjustment should be made to cover the case where the memory area is set to **Unlocked** when the key is not set.

- Alternatively, if the locking and unlocking states are handled separately (as in SYSRS 018 and SWRS 020), there should be a distinct software requirement addressing the **unlocked** state to fully align with SYSRS 018.



"SYSRS 017: If Performing the Standard Path Measurement
AND IF performing Stop Ongoing Measurement (Blocked)
AND IF the functionality in US-Transducer Control (Non-Modulated Sine Burst) has NOT been fully performed,
THEN
the Sensor IC shall
set Error: err_burstGenerationStopped = 0x1 ("error")." "SWRS 019: IF in Main-Measurement
AND IF Break Burst Generation occurs
THEN
the software shall set
Error: err_burstGenerationStopped = 0x1 ("error")."



Evaluation of SYSRS 017 and Linked SWRS 019

SYSRS 017:

- **Condition:**
 - **IF performing the Standard Path Measurement,**
 - **AND IF performing Stop Ongoing Measurement (Blocked),**
 - **AND IF the functionality in US-Transducer Control (Non-Modulated Sine Burst) has NOT been fully performed.**
- **Action:**
 - The Sensor IC shall set `Error: err_burstGenerationStopped = 0x1` ("error").

SWRS 019:

- **Condition:**
 - **IF in Main-Measurement,**
 - **AND IF Break Burst Generation occurs.**
- **Action:**
 - The software shall set `Error: err_burstGenerationStopped = 0x1` ("error").

Condition Check:

- **SYSRS 017 Condition:**
 - The system requirement specifies that an error is triggered if the Stop Ongoing Measurement command is executed **and** the US-Transducer Control (Non-Modulated Sine Burst) has **not** been fully performed.
- **SWRS 019 Condition:**
 - The software requirement states that the error is triggered **if Break Burst Generation occurs during Main-Measurement**. It does not reference Stop Ongoing Measurement or

the incomplete US-Transducer Control process, which is a key condition in SYSRS 017.

Assessment:

- There is a mismatch in the conditions. SWRS 019 refers to a different situation (Break Burst Generation) rather than the specific scenario described in SYSRS 017, which involves the Standard Path Measurement, stopping an ongoing measurement, and incomplete execution of US-Transducer Control. As a result, the conditions do not align properly.

Action and Response Check:

- **SYSRS 017 Action:**
 - If the specified conditions are met, the system requirement instructs the Sensor IC to set ``err_burstGenerationStopped = 0x1``.
- **SWRS 019 Action:**
 - SWRS 019 correctly specifies setting ``err_burstGenerationStopped = 0x1``, but this occurs under different conditions.

Assessment:

- While both the system and software requirements trigger the same error (``err_burstGenerationStopped = 0x1``), they do so under different conditions. The action itself is correct, but the situation in which it occurs differs between SYSRS 017 and SWRS 019.

Parameter Sources Check:

- **SYSRS 017 Parameters:**
 - The system requirement specifies ``err_burstGenerationStopped = 0x1``, which is set when the Stop Ongoing Measurement (Blocked) occurs and the US-Transducer Control has not been completed.
- **SWRS 019 Parameters:**
 - The software requirement also sets ``err_burstGenerationStopped = 0x1``, but the triggering event is related to Break Burst Generation during Main-Measurement, not the scenario described in SYSRS 017.

Assessment:

- Both requirements use the same parameter (``err_burstGenerationStopped``), but they reference different events that lead to setting this parameter.

Additional Conditions Check:

- **SWRS 019 Additional Condition:**
 - SWRS 019 introduces a different condition: Break Burst Generation during Main-Measurement, which is not mentioned in SYSRS 017.

Assessment:

- SWRS 019 adds a condition not mentioned in SYSRS 017. While both conditions are related to burst generation, they represent different scenarios, and SWRS 019 does not account for the specific situation described in SYSRS 017 (Stop Ongoing Measurement and incomplete US-Transducer Control).

Ratings:

- **Meaningfulness Rating: Medium**

- **Reason:** While SWRS 019 shares a similar goal (triggering an error when burst generation is interrupted), it refers to a different situation (Break Burst Generation) than SYSRS 017, which describes a more specific condition involving Stop Ongoing Measurement and incomplete US-Transducer Control. The software requirement partially supports the system requirement but fails to address the full scope.

- **Completeness Rating: Low**

- **Reason:** SWRS 019 does not fully implement the conditions described in SYSRS 017, omitting key details such as the incomplete execution of the US-Transducer Control and the Stop Ongoing Measurement condition. The error setting action is present, but the underlying circumstances differ, making the link incomplete.

- **Overall Quality Rating: Medium**

- **Reason:** The overall quality is medium because while SWRS 019 triggers the same error, it does so under different circumstances than SYSRS 017. The software requirement is related, but it fails to capture the full complexity of the system requirement.

Suggested Revisions:

- **SWRS 019** should be revised to incorporate the full scope of conditions from **SYSRS 017**.

Specifically:

- It should account for the **Stop Ongoing Measurement** condition and the incomplete execution of **US-Transducer Control (Non-Modulated Sine Burst)**.
- The current reference to **Break Burst Generation** during Main-Measurement could remain if it's a related scenario, but it should not replace the condition outlined in SYSRS 017. If these are separate conditions, they should both be included in the software requirements.



"SYSRS 016: After Reception of the Valid CRM_FCC_CMD,
THEN
the Sensor IC shall
set the PDCM_RSP Parameter KAC = 0x0." "SWRS 018: Increment KAC
AFTER Sending PDCM Packet After Reception of the BRC
THEN
the Software shall
increment the PDCM_RSP Parameter KAC by 1;

After Reception of the Valid CRM_FCC_CMD,
THEN
the Software shall
set the PDCM_RSP Parameter KAC = 0x3."



Evaluation of SYSRS 016 and Linked SWRS 018

SYSRS 016:

- **Condition:**
 - **AFTER reception of the valid CRM_FCC_CMD.**
- **Action:**
 - The Sensor IC shall set the `PDCM_RSP` Parameter `KAC = 0x0`.

SWRS 018:

- **Condition 1:**
 - **AFTER sending PDCM Packet AFTER reception of the BRC.**
- **Action 1:**
 - The Software shall increment the `PDCM_RSP` Parameter `KAC` by 1.
- **Condition 2:**
 - **AFTER reception of the valid CRM_FCC_CMD.**
- **Action 2:**
 - The Software shall set the `PDCM_RSP` Parameter `KAC = 0x3`.

Condition Check:

- **SYSRS 016 Condition:**
 - This system requirement specifies that the action should be performed **after the reception of a valid CRM_FCC_CMD.**
- **SWRS 018 Condition:**
 - SWRS 018 has two conditions:
 - **Condition 1:** After sending a PDCM Packet after receiving a BRC (Broadcast Command).
 - **Condition 2:** After receiving a valid CRM_FCC_CMD.

Assessment:

- **Condition 2** in SWRS 018 matches the condition in SYSRS 016, as both refer to the valid reception of the CRM_FCC_CMD. However, **Condition 1** introduces an additional condition not mentioned in SYSRS 016, which is the sending of the PDCM packet after receiving the BRC. This additional condition does not align with the system requirement.

Action and Response Check:

- **SYSRS 016 Action:**
 - The system requirement specifies that the `PDCM_RSP` Parameter `KAC` should be set to `0x0` after receiving the valid CRM_FCC_CMD.
- **SWRS 018 Action:**
 - SWRS 018 specifies two actions:

- **Action 1:** Increment the `PDCM_RSP` Parameter `KAC` by 1 after sending a PDCM packet following the reception of the BRC.
- **Action 2:** Set the `PDCM_RSP` Parameter `KAC = 0x3` after receiving the valid CRM_FCC_CMD.

Assessment:

- There is a significant discrepancy between the actions described in SYSRS 016 and SWRS 018. SYSRS 016 specifies setting `KAC = 0x0`, whereas SWRS 018 specifies setting `KAC = 0x3` after receiving the valid CRM_FCC_CMD. This is a major inconsistency in the action performed under the same condition. Additionally, SWRS 018 introduces an increment action for `KAC` in a scenario (after receiving the BRC) that is not mentioned in SYSRS 016.

Parameter Sources Check:

- **SYSRS 016 Parameters:**
 - The key parameter is the `PDCM_RSP` Parameter `KAC`, which should be set to `0x0`.
- **SWRS 018 Parameters:**
 - The same parameter, `PDCM_RSP` Parameter `KAC`, is mentioned, but SWRS 018 provides conflicting values for this parameter: `KAC = 0x3` instead of `KAC = 0x0`.

Assessment:

- Both SYSRS 016 and SWRS 018 reference the same parameter (`PDCM_RSP` Parameter `KAC`), but the values assigned are inconsistent. SWRS 018 increments the `KAC` parameter in an unrelated condition and sets it to `0x3` instead of the expected `0x0`.

Additional Conditions Check:

- **SWRS 018 Additional Condition:**
 - SWRS 018 introduces a condition (sending a PDCM packet after receiving a BRC) that is not mentioned in SYSRS 016. This adds complexity not required by the system requirement.

Assessment:

- The additional condition introduced in SWRS 018 does not appear in SYSRS 016 and represents a deviation from the expected behavior described in the system requirement.

Ratings:

- **Meaningfulness Rating: Low**
 - **Reason:** SWRS 018 introduces conflicting conditions and actions compared to SYSRS 016. The condition of setting `KAC = 0x3` after receiving the valid CRM_FCC_CMD directly contradicts the system requirement, which specifies setting `KAC = 0x0`. Additionally, SWRS 018 adds a condition and action not mentioned in SYSRS 016 (incrementing `KAC` after receiving the BRC). This makes the link between the system and software requirements weak and unclear.
- **Completeness Rating: Low**

- **Reason:** SWRS 018 does not fully implement the condition or action described in SYSRS 016. The key action (setting `KAC = 0x0`) is missing, and an additional condition (BRC reception and PDCM packet sending) is introduced, making the link incomplete.

- **Overall Quality Rating: Low**

- **Reason:** The overall quality of the link is low because the software requirement contradicts the system requirement in a critical area (the value assigned to `KAC` after receiving the valid CRM_FCC_CMD). Additionally, SWRS 018 introduces extra conditions and actions not required by SYSRS 016, resulting in a significant misalignment.

Suggested Revisions:

- **SWRS 018** should be revised to:
 - Set the `PDCM_RSP` Parameter `KAC = 0x0` after receiving the valid CRM_FCC_CMD, as specified in **SYSRS 016**.
 - If the additional condition (BRC reception and incrementing `KAC`) is necessary for the overall system, it should be documented separately to avoid confusion. However, this condition should not override the action required by SYSRS 016.
 - Remove or clarify the conflicting value of `KAC = 0x3`, as this directly contradicts the system requirement.



"SYSRS 015: Increment KAC
AFTER Sending PDCM Packet After Reception of the BRC
THEN
the Sensor IC shall
increment the PDCM_RSP Parameter KAC by 1; the counter is reset to 0 at overflow" "SWRS 017:
Increment KAC
AFTER Sending PDCM Packet After Reception of the BRC
THEN
the Software shall
increment the PDCM_RSP Parameter KAC by 1; "



Evaluation of SYSRS 015 and Linked SWRS 017

SYSRS 015:

- **Condition:**
 - **AFTER sending PDCM Packet after reception of the BRC.**
- **Action:**
 - The Sensor IC shall increment the `PDCM_RSP` Parameter `KAC` by 1, and the counter is reset to 0 at overflow.

SWRS 017:

- **Condition:**

- **AFTER sending PDCM Packet after reception of the BRC.**
 - **Action:**
 - The Software shall increment the `PDCM_RSP` Parameter `KAC` by 1.
-

Condition Check:

- **SYSRS 015 Condition:**
 - The system requirement specifies that the condition is met **after sending a PDCM packet** following the reception of the BRC (Broadcast Command).
- **SWRS 017 Condition:**
 - The software requirement matches the system requirement, specifying that the condition is triggered **after sending a PDCM packet** following the reception of the BRC.

Assessment:

- The conditions are fully aligned between SYSRS 015 and SWRS 017. Both specify that the action occurs after sending a PDCM packet following the reception of the BRC.

Action and Response Check:

- **SYSRS 015 Action:**
 - The system requirement specifies that the Sensor IC shall increment the `PDCM_RSP` Parameter `KAC` by 1, with an additional detail that the counter is **reset to 0 at overflow**.
- **SWRS 017 Action:**
 - The software requirement specifies that the software shall increment the `PDCM_RSP` Parameter `KAC` by 1, but it **does not mention** the behavior when the counter overflows (resetting to 0).

Assessment:

- The action described in SWRS 017 is partially aligned with SYSRS 015. While the action of incrementing the `KAC` parameter is correctly described, SWRS 017 omits the detail about resetting the counter to 0 at overflow, which is an important aspect of the system requirement.

Parameter Sources Check:

- **SYSRS 015 Parameters:**
 - The key parameter is the `PDCM_RSP` Parameter `KAC`, which should be incremented by 1, and the counter should reset to 0 at overflow.
- **SWRS 017 Parameters:**
 - SWRS 017 references the same parameter (`PDCM_RSP` Parameter `KAC`), but it does not mention the overflow reset.

Assessment:

- The same parameter is referenced in both requirements, but SWRS 017 lacks the overflow reset condition that is present in SYSRS 015. This omission could lead to inconsistent behavior in handling the `KAC` parameter when it overflows.

Additional Conditions Check:

- **SWRS 017 Additional Condition:**
 - SWRS 017 does not introduce any additional conditions, but it omits a key detail from SYSRS 015 (resetting to 0 at overflow).

Assessment:

- There are no additional conditions, but the omission of the overflow reset behavior represents an important missing detail.

Ratings:

- **Meaningfulness Rating: Medium**
 - **Reason:** The software requirement partially supports the system requirement by describing the increment of the `KAC` parameter. However, it omits the overflow reset condition, which is a critical part of the system behavior. This reduces the overall meaningfulness of the link, as it only implements part of the required behavior.
- **Completeness Rating: Medium**
 - **Reason:** The software requirement lacks the critical detail about resetting the `KAC` counter to 0 at overflow. While it correctly describes the increment action, the absence of the overflow behavior makes the link incomplete.
- **Overall Quality Rating: Medium**
 - **Reason:** The link between the system and software requirements is generally accurate but incomplete. The omission of the overflow reset condition reduces the quality of the implementation, though the basic behavior of incrementing `KAC` is correctly implemented.

Suggested Revisions:

- **SWRS 017** should be revised to include the detail that the `KAC` counter **resets to 0 at overflow**, as specified in **SYSRS 015**. This would ensure that the software behavior fully aligns with the system requirement and avoids potential inconsistencies when handling the overflow scenario.



"SYSRS 014: IF the Event_MAX is generated,
THEN
the Sensor IC shall
measure the time of the occurrence of the event condition relative to the start of the
measurement and provide PDCM_RSP P12, P13 Parameter EventTimeStamp<i> for the <i>th
Event transmitted in a DSI3 packet." SWRS 016: The software shall send the MAX events with
the measured time of the occurrence of the event, condition relative to the start of the
measurement, and provide the PDCM_RSP P12,P13 Parameter EventTimeStamp<i> for the <i>th
Event transmitted in a PDCM frame.



Evaluation of SYSRS 014 and Linked SWRS 016

SYSRS 014:

- **Condition:**
 - **IF the Event_MAX is generated.**
- **Action:**
 - The Sensor IC shall measure the time of the occurrence of the event relative to the start of the measurement and provide the ``PDCM_RSP P12, P13 Parameter EventTimeStamp<i>`` for the ``<i>th Event`` transmitted in a DSI3 packet.

SWRS 016:

- **Action:**
 - The software shall send the MAX events with the measured time of the occurrence of the event, condition relative to the start of the measurement, and provide the ``PDCM_RSP P12, P13 Parameter EventTimeStamp<i>`` for the ``<i>th Event`` transmitted in a PDCM frame.

Condition Check:

- **SYSRS 014 Condition:**
 - The system requirement specifies that the action is triggered when **Event_MAX is generated.**
- **SWRS 016 Condition:**
 - SWRS 016 does not explicitly state "Event_MAX is generated," but it refers to the **MAX events** without clearly defining the trigger condition. The implication seems to be that the software will handle the MAX event in the same way, though the condition is not explicitly stated as in SYSRS 014.

Assessment:

- The condition in SWRS 016 is implicit and does not explicitly refer to the event generation as described in SYSRS 014. While the intent appears to be the same, SWRS 016 would benefit from clarifying the specific event generation trigger to match the system requirement.

Action and Response Check:

- **SYSRS 014 Action:**
 - The system requirement specifies that the Sensor IC should measure the time of the event's occurrence and provide ``PDCM_RSP P12, P13 Parameter EventTimeStamp<i>`` for the ``<i>th Event`` in a DSI3 packet.
- **SWRS 016 Action:**
 - The software requirement mirrors this action closely, stating that the software will send the MAX events with the measured time of occurrence and provide the same parameters (``PDCM_RSP P12, P13 Parameter EventTimeStamp<i>``) but in a **PDCM frame** rather than a DSI3 packet.

Assessment:

- The actions are largely aligned. The main difference is that SYSRS 014 mentions transmission in a **DSI3 packet**, while SWRS 016 refers to transmission in a **PDCM frame**. If these two terms

represent different formats or protocols, this could be a significant discrepancy, but if they refer to the same mechanism under different naming conventions, this would be a minor issue.

Parameter Sources Check:

- **SYSRS 014 Parameters:**
 - The system requirement refers to the ``PDCM_RSP P12, P13 Parameter EventTimeStamp<i>``, which is provided for the ``<i>th Event``.
- **SWRS 016 Parameters:**
 - The software requirement uses the same parameters, ``PDCM_RSP P12, P13 Parameter EventTimeStamp<i>``, for the same purpose.

Assessment:

- Both requirements reference the same parameters (``PDCM_RSP P12, P13 Parameter EventTimeStamp<i>``), ensuring consistency in the information being provided.

Additional Conditions Check:

- **SWRS 016 Additional Condition:**
 - SWRS 016 introduces no additional conditions or actions that are not present in SYSRS 014.

Assessment:

- There are no additional conditions or deviations introduced in SWRS 016 that would complicate the link to SYSRS 014.

Ratings:

- **Meaningfulness Rating: Medium**
 - **Reason:** The software requirement generally supports the system requirement, but the lack of an explicit reference to "Event_MAX" in SWRS 016 weakens the clarity of the link. Additionally, the difference in terms (DSI3 packet vs. PDCM frame) could cause confusion if these represent different transmission mechanisms.
- **Completeness Rating: High**
 - **Reason:** SWRS 016 fully implements the key action described in SYSRS 014, providing the required parameters and measuring the event time relative to the start of the measurement. The only concern is the difference in terminology, which may need clarification.
- **Overall Quality Rating: Medium**
 - **Reason:** The overall quality is medium because while the core action is correctly implemented, the lack of an explicit reference to "Event_MAX" and the potential discrepancy in transmission terminology (DSI3 packet vs. PDCM frame) reduce the clarity and alignment between the two requirements.

Suggested Revisions:

- **SWRS 016** should:
 - Clarify that the event is triggered by the **Event_MAX** generation, as specified in **SYSRS 014**.
 - Confirm whether the transmission mechanism (DSI3 packet vs. PDCM frame) refers to the same process or different ones. If they are the same, a clarification should be added to avoid confusion. If they are different, this distinction should be explicitly noted to ensure consistency with the system requirement.



SYSRS 013: The Sensor IC shall amplify the Transducer Input Signal at the IC Pins with the three amplification factors (Parameter CALIB_analogGain plus Parameter SPX_dynGainStartValue plus CRM_CMD 0x4 Parameter CalibrationGain plus Parameter YPX_calibrationGainOffset). "SWRS 013: Analog Amplification: AFTER reception of the command CRM_CMD 0x4: Start Calibration Measurement the software shall set the register the register ASP_CFG.g_ana to Parameter CALIB_analogGain .

SWRS 014: Digital Amplification: AFTER reception of the command CRM_CMD 0x4: Start Calibration Measurement the software shall set the register ENVP_AMPD_CFG.g_dig to Parameter SPX_dynGainStartValue .

SWRS 015: Calibration Gain: AFTER reception of the command CRM_CMD 0x4: Start Calibration Measurement the software shall set the register ENVP_GCAL.g_cal to (CRM_CMD 0x4 Parameter CalibrationGain + Parameter YPX_calibrationGainOffset).

"



Evaluation of SYSRS 013 and Linked SWRS 013, SWRS 014, SWRS 015

SYSRS 013:

- **Condition:**
 - The Sensor IC shall amplify the Transducer Input Signal at the IC Pins with the following three amplification factors:
 - `Parameter CALIB_analogGain`
 - `Parameter SPX_dynGainStartValue`
 - `CRM_CMD 0x4 Parameter CalibrationGain` + `Parameter YPX_calibrationGainOffset`

SWRS 013:

- **Action:**
 - **Analog Amplification:** After reception of the command `CRM_CMD 0x4: Start Calibration Measurement`, the software shall set the register `ASP_CFG.g_ana` to `Parameter CALIB_analogGain`.

SWRS 014:

- **Action:**

- **Digital Amplification:** After reception of the command ``CRM_CMD 0x4: Start Calibration Measurement``, the software shall set the register ``ENVP_AMPD_CFG.g_dig`` to ``Parameter SPX_dynGainStartValue``.

SWRS 015:

- **Action:**
 - **Calibration Gain:** After reception of the command ``CRM_CMD 0x4: Start Calibration Measurement``, the software shall set the register ``ENVP_GCAL.g_cal`` to `(`CRM_CMD 0x4 Parameter CalibrationGain + Parameter YPX_calibrationGainOffset`)`.

Condition Check:

- **SYSRS 013 Condition:**
 - The system requirement does not specify any particular condition other than that the Sensor IC must perform amplification with the given parameters.
- **SWRS 013, SWRS 014, SWRS 015 Conditions:**
 - All three software requirements (SWRS 013, SWRS 014, SWRS 015) specify that the respective amplification actions are triggered **after reception of the command** ``CRM_CMD 0x4: Start Calibration Measurement``.

Assessment:

- The system requirement does not explicitly reference the ``CRM_CMD 0x4`` command as a trigger for amplification, whereas all the software requirements are conditioned on the reception of this command. The software requirements add this condition, which may reflect an implementation detail but is not stated in SYSRS 013. This added condition could be relevant for execution but requires clarification.

Action and Response Check:

- **SYSRS 013 Action:**
 - The system requirement specifies that the Sensor IC shall amplify the Transducer Input Signal using a combination of four factors:
 1. ``Parameter CALIB_analogGain``
 2. ``Parameter SPX_dynGainStartValue``
 3. ``CRM_CMD 0x4 Parameter CalibrationGain``
 4. ``Parameter YPX_calibrationGainOffset``
- **SWRS 013 Action:**
 - SWRS 013 handles the **Analog Amplification** by setting ``ASP_CFG.g_ana`` to ``Parameter CALIB_analogGain``.
- **SWRS 014 Action:**
 - SWRS 014 handles the **Digital Amplification** by setting ``ENVP_AMPD_CFG.g_dig`` to ``Parameter SPX_dynGainStartValue``.
- **SWRS 015 Action:**
 - SWRS 015 handles the **Calibration Gain** by setting ``ENVP_GCAL.g_cal`` to `(`CRM_CMD 0x4 Parameter CalibrationGain + Parameter YPX_calibrationGainOffset`)`.

Assessment:

- The three software requirements collectively implement the action described in SYSRS 013 by separately addressing the different amplification factors. However, while SYSRS 013 implies that these factors are combined for a single amplification process, the software requirements treat each factor as a separate process. This may reflect the actual software design but could benefit from clarification on how these amplifications are combined or processed together.

Parameter Sources Check:

- **SYSRS 013 Parameters:**
 - The system requirement specifies four parameters:
 1. ``Parameter CALIB_analogGain``
 2. ``Parameter SPX_dynGainStartValue``
 3. ``CRM_CMD 0x4 Parameter CalibrationGain``
 4. ``Parameter YPX_calibrationGainOffset``
- **SWRS 013, SWRS 014, SWRS 015 Parameters:**
 - The software requirements correctly reference all the necessary parameters:
 - SWRS 013 handles ``Parameter CALIB_analogGain``
 - SWRS 014 handles ``Parameter SPX_dynGainStartValue``
 - SWRS 015 handles both ``CRM_CMD 0x4 Parameter CalibrationGain`` and ``Parameter YPX_calibrationGainOffset``

Assessment:

- The parameters are correctly addressed in the software requirements, matching those specified in the system requirement. Each parameter is processed in its corresponding software requirement.

Additional Conditions Check:

- **SWRS 013, SWRS 014, SWRS 015 Additional Conditions:**
 - All three software requirements introduce the condition of processing amplification only **after the reception of the ``CRM_CMD 0x4: Start Calibration Measurement`` command**, which is not mentioned in SYSRS 013.

Assessment:

- The added condition in the software requirements suggests that the amplification processes are triggered by a specific command, which might be an important implementation detail but is not explicitly required by SYSRS 013. This additional condition does not seem to contradict the system requirement but may represent a design choice.

Ratings:

- **Meaningfulness Rating: High**
 - **Reason:** The software requirements directly implement the amplification process described in SYSRS 013. Although they divide the amplification factors into separate processes, they accurately handle all the parameters and achieve the same goal as the system requirement.
- **Completeness Rating: Medium**

- **Reason:** While the software requirements correctly implement the required amplification factors, they introduce an additional condition (reception of the `CRM_CMD 0x4` command) that is not mentioned in the system requirement. This could require clarification or adjustment to fully align with SYSRS 013. Additionally, the system requirement implies that the factors are combined, but the software requirements handle them separately.

- **Overall Quality Rating: Medium**

- **Reason:** The overall quality is medium because, while the software requirements fully implement the necessary amplification factors, the added condition (requiring the `CRM_CMD 0x4` command) and the separation of the amplification process into distinct actions could require clarification. The separation might be valid for software design but needs further explanation in the context of the system requirement.

Suggested Revisions:

- **SWRS 013, SWRS 014, SWRS 015** should:
 - Clarify whether the amplification factors are combined after being set individually or whether they remain distinct processes throughout.
 - If the condition of receiving the `CRM_CMD 0x4` command is an essential part of the design, it should be reflected in **SYSRS 013** for consistency, or it should be clarified as an implementation-specific detail in the software requirements.



"SYSRS 009: If Performing Memory Access in Broadcast
AND IF the state if the Memory Area identified by CRM_CMD 0x8 Parameter MemoryArea for the operation as determined by CRM_CMD 0x8 Operation Flag is in the state Locked,
THEN
the Sensor IC shall
respond with PDCM_RSP P0: Report CRM_RSP Information containing CRM_RSP 0x8: Report Lock Status and CRM_RSP 0x8 Flag KeyRequest = 0b1 ("key requested").

" "SWRS 010: If Performing Memory Access in Broadcast
AND IF the state if the Memory Area identified by CRM_CMD 0x8 Parameter MemoryArea for the operation as determined by CRM_CMD 0x8 Operation Flag is in the state Locked,
THEN
the Software shall prepare to
respond with PDCM_RSP P0: Report CRM_RSP Information containing CRM_RSP 0x8: Report Lock Status and CRM_RSP 0x8 Flag KeyRequest = 0b1 ("key requested").
"



Evaluation of SYSRS 009 and Linked SWRS 010

SYSRS 009:

- **Condition:**
 - **IF performing memory access in Broadcast,**

- **AND IF the state of the Memory Area identified by CRM_CMD 0x8 Parameter MemoryArea for the operation, as determined by CRM_CMD 0x8 Operation Flag, is in the state Locked.**
- **Action:**
 - The Sensor IC shall respond with `PDCM_RSP P0: Report CRM_RSP Information` containing `CRM_RSP 0x8: Report Lock Status` and `CRM_RSP 0x8 Flag KeyRequest = 0b1` ("key requested").

SWRS 010:

- **Condition:**
 - **IF performing memory access in Broadcast,**
 - **AND IF the state of the Memory Area identified by CRM_CMD 0x8 Parameter MemoryArea for the operation, as determined by CRM_CMD 0x8 Operation Flag, is in the state Locked.**
- **Action:**
 - The software shall prepare to respond with `PDCM_RSP P0: Report CRM_RSP Information` containing `CRM_RSP 0x8: Report Lock Status` and `CRM_RSP 0x8 Flag KeyRequest = 0b1` ("key requested").

Condition Check:

- **SYSRS 009 Condition:**
 - The system requirement specifies that memory access is being performed in Broadcast mode, and the Memory Area for the operation (determined by the `CRM_CMD 0x8 Operation Flag`) is in the Locked state.
- **SWRS 010 Condition:**
 - The software requirement mirrors the condition described in SYSRS 009, specifying the same conditions of memory access in Broadcast and the state of the Memory Area being Locked.

Assessment:

- The conditions in SWRS 010 are an exact match to those in SYSRS 009, indicating that the software requirement is properly aligned with the system requirement in terms of triggering conditions.

Action and Response Check:

- **SYSRS 009 Action:**
 - The system requirement specifies that the Sensor IC shall **respond** with `PDCM_RSP P0: Report CRM_RSP Information` containing `CRM_RSP 0x8: Report Lock Status` and `CRM_RSP 0x8 Flag KeyRequest = 0b1` ("key requested").
- **SWRS 010 Action:**
 - The software requirement specifies that the software shall **prepare to respond** with the same `PDCM_RSP P0: Report CRM_RSP Information` containing `CRM_RSP 0x8: Report Lock Status` and `CRM_RSP 0x8 Flag KeyRequest = 0b1` ("key requested").

Assessment:

- The actions in SYSRS 009 and SWRS 010 are largely similar, but the software requirement introduces the phrase "prepare to respond," whereas the system requirement directly states "respond." This distinction could reflect an implementation detail (e.g., preparing the response before sending it), but it introduces a slight ambiguity about when the actual response occurs.

Parameter Sources Check:

- **SYSRS 009 Parameters:**
 - The system requirement references ``CRM_CMD 0x8 Parameter MemoryArea``, ``CRM_CMD 0x8 Operation Flag``, and the response parameters ``PDCM_RSP P0: Report CRM_RSP Information``, ``CRM_RSP 0x8: Report Lock Status``, and ``CRM_RSP 0x8 Flag KeyRequest = 0b1``.
- **SWRS 010 Parameters:**
 - The software requirement uses the same parameters, referencing ``CRM_CMD 0x8 Parameter MemoryArea``, ``CRM_CMD 0x8 Operation Flag``, and the response parameters ``PDCM_RSP P0``, ``CRM_RSP 0x8``, and ``KeyRequest = 0b1``.

Assessment:

- Both the system and software requirements reference the same parameters, ensuring consistency in how the MemoryArea and Lock Status are handled.

Additional Conditions Check:

- **SWRS 010 Additional Condition:**
 - SWRS 010 introduces no additional conditions beyond what is specified in SYSRS 009.

Assessment:

- No additional conditions are introduced, indicating that the software requirement remains consistent with the system requirement.

Ratings:

- **Meaningfulness Rating: High**
 - **Reason:** SWRS 010 clearly and directly supports the system requirement. It specifies the same conditions and parameters, and the distinction between "prepare to respond" and "respond" likely reflects an implementation detail, not a major misalignment. The link between the system and software requirements is strong.
- **Completeness Rating: Medium**
 - **Reason:** While the conditions and actions are well aligned, the difference between "prepare to respond" in SWRS 010 and "respond" in SYSRS 009 introduces slight ambiguity. It is unclear whether the software requirement fully addresses the need to ensure that the response is completed or just prepared. This could be clarified to ensure full alignment.
- **Overall Quality Rating: High**
 - **Reason:** Despite the minor ambiguity in the wording ("prepare to respond" vs. "respond"), the overall quality is high because the software requirement properly implements the system requirement in terms of conditions, parameters, and actions.

Suggested Revisions:

- **SWRS 010** could clarify whether the software simply prepares the response or actually completes the response as required by **SYSRS 009**. This would eliminate any ambiguity and ensure that the response is properly sent when the conditions are met. If "prepare to respond" is an intentional implementation step, it should be documented to explain when the actual response is issued.



```
"SYSRS 008: If Performing Memory Access With Set Key
AND If Write Operation is Requested
AND If Memory Area Is Set Write Accessible
THEN
the Sensor IC shall
set the state of the Memory Area identified by CRM_CMD 0x8 Parameter MemoryArea to the state
Locked." "SWRS 009: If Performing Memory Access
AND If the Key is NOT Set
AND If Write Operation is Requested
AND If Memory Area Is Set Read Accessible
THEN
the Software shall
set the state of the Memory Area identified by CRM_CMD 0x10 Parameter MemoryArea to the
state Locked.
"
```



Evaluation of SYSRS 008 and Linked SWRS 009

SYSRS 008:

- **Condition:**
 - **IF performing memory access with Set Key,**
 - **AND IF a Write Operation is requested,**
 - **AND IF the Memory Area is Set to Write Accessible.**
- **Action:**
 - The Sensor IC shall set the state of the Memory Area identified by `CRM_CMD 0x8 Parameter MemoryArea` to the state Locked.

SWRS 009:

- **Condition:**
 - **IF performing memory access,**
 - **AND IF the Key is NOT Set,**
 - **AND IF a Write Operation is requested,**
 - **AND IF the Memory Area is Set to Read Accessible.**
- **Action:**
 - The software shall set the state of the Memory Area identified by `CRM_CMD 0x10 Parameter MemoryArea` to the state Locked.

Condition Check:

- **SYSRS 008 Condition:**
 - The system requirement specifies that the memory access occurs **with a Set Key**, a Write Operation is requested, and the Memory Area is **Write Accessible**.
- **SWRS 009 Condition:**
 - The software requirement specifies that the memory access occurs **without a Set Key**, a Write Operation is requested, and the Memory Area is **Read Accessible**.

Assessment:

- There is a major mismatch in conditions between SYSRS 008 and SWRS 009:
 - **Key Status:** SYSRS 008 requires the key to be set, while SWRS 009 specifies that the key is **not set**.
 - **Memory Area Access:** SYSRS 008 requires the Memory Area to be **Write Accessible**, while SWRS 009 requires it to be **Read Accessible**.

The conditions are largely opposite in the two requirements, indicating that SWRS 009 does not implement the condition described in SYSRS 008.

Action and Response Check:

- **SYSRS 008 Action:**
 - If the conditions are met, the Sensor IC shall set the state of the Memory Area to **Locked**.
- **SWRS 009 Action:**
 - SWRS 009 also specifies setting the state of the Memory Area to **Locked** under different conditions.

Assessment:

- Both requirements involve locking the Memory Area, but the conditions under which this occurs are different. Since the conditions in SWRS 009 are not aligned with those in SYSRS 008, the action is triggered in fundamentally different scenarios.

Parameter Sources Check:

- **SYSRS 008 Parameters:**
 - The system requirement references the ``CRM_CMD 0x8 Parameter MemoryArea`` as the source of the Memory Area being accessed.
- **SWRS 009 Parameters:**
 - The software requirement references the ``CRM_CMD 0x10 Parameter MemoryArea``, a different parameter than in SYSRS 008.

Assessment:

- The parameters referenced in the two requirements are different (``CRM_CMD 0x8`` in SYSRS 008 and ``CRM_CMD 0x10`` in SWRS 009), indicating a discrepancy in how the Memory Area is identified. This further reduces the alignment between the two requirements.

Additional Conditions Check:

- **SWRS 009 Additional Condition:**

- SWRS 009 introduces the condition of the Memory Area being **Read Accessible**, whereas SYSRS 008 specifies that it must be **Write Accessible**.

Assessment:

- SWRS 009 introduces additional conditions (the Memory Area being Read Accessible) that are not mentioned in SYSRS 008. These additional conditions shift the meaning of the requirement and lead to a different scenario being handled.

Ratings:

- **Meaningfulness Rating: Low**

- **Reason:** The software requirement (SWRS 009) does not implement the condition described in SYSRS 008. The conditions related to the Key being set and the Memory Area being Write Accessible are directly contradicted in SWRS 009. The software requirement addresses a different scenario altogether.

- **Completeness Rating: Low**

- **Reason:** SWRS 009 fails to address the conditions and parameters described in SYSRS 008. It introduces different conditions and references a different parameter (``CRM_CMD 0x10`` instead of ``CRM_CMD 0x8``), making the link incomplete.

- **Overall Quality Rating: Low**

- **Reason:** The overall quality is low due to the major misalignment in both conditions and parameters. The software requirement does not meaningfully or fully implement the system requirement, addressing a different situation than what is described in SYSRS 008.

Suggested Revisions:

- **SWRS 009** should be revised to:
 - Match the conditions described in **SYSRS 008**, specifically ensuring that the key is **set**, the Memory Area is **Write Accessible**, and the correct parameter (``CRM_CMD 0x8``) is referenced.
 - If the current conditions in SWRS 009 (key not set, Memory Area Read Accessible) are valid, they should be addressed in a separate software requirement rather than conflicting with SYSRS 008.



"SYSRS 007: If Performing Memory Access
AND If the Key is Set
AND If Read Operation is Requested
AND If Memory Area Is Set Read Accessible
THEN

the Sensor IC shall

set the state of the Memory Area identified by CRM_CMD 0x8 Parameter MemoryArea to the state Locked.

" SWRS 008: If Performing Memory Access AND If the Key is Set AND If Read Operation is Requested AND If Memory Area Is Set Read Accessible THEN the Software shall set

the state of the Memory Area identified by CRM_CMD 0x8 Parameter MemoryArea to the state Locked.



Evaluation of SYSRS 007 and Linked SWRS 008

SYSRS 007:

- **Condition:**
 - **IF performing memory access,**
 - **AND IF the Key is Set,**
 - **AND IF a Read Operation is requested,**
 - **AND IF the Memory Area is Set to Read Accessible.**
- **Action:**
 - The Sensor IC shall set the state of the Memory Area identified by `CRM_CMD 0x8 Parameter MemoryArea` to the state Locked.

SWRS 008:

- **Condition:**
 - **IF performing memory access,**
 - **AND IF the Key is Set,**
 - **AND IF a Read Operation is requested,**
 - **AND IF the Memory Area is Set to Read Accessible.**
- **Action:**
 - The Software shall set the state of the Memory Area identified by `CRM_CMD 0x8 Parameter MemoryArea` to the state Locked.

Condition Check:

- **SYSRS 007 Condition:**
 - The system requirement specifies that the action should occur when the memory access is being performed, the Key is set, a Read Operation is requested, and the Memory Area is set to Read Accessible.
- **SWRS 008 Condition:**
 - The software requirement exactly mirrors the same conditions described in SYSRS 007, using the same conditions: memory access, Key set, Read Operation, and Read Accessible Memory Area.

Assessment:

- The conditions in SWRS 008 perfectly align with those in SYSRS 007, indicating a strong match between the system and software requirements.

Action and Response Check:

- **SYSRS 007 Action:**
 - The system requirement specifies that the Sensor IC shall set the state of the Memory Area identified by `CRM_CMD 0x8 Parameter MemoryArea` to Locked.

- **SWRS 008 Action:**
 - The software requirement specifies that the software shall set the state of the Memory Area identified by ``CRM_CMD 0x8 Parameter MemoryArea`` to Locked.

Assessment:

- The actions described in both requirements are identical, ensuring that the software implementation directly supports the system requirement.

Parameter Sources Check:

- **SYSRS 007 Parameters:**
 - The system requirement references the ``CRM_CMD 0x8 Parameter MemoryArea`` as the source for identifying the Memory Area that will be locked.
- **SWRS 008 Parameters:**
 - The software requirement references the same parameter, ``CRM_CMD 0x8 Parameter MemoryArea``, ensuring consistency in how the Memory Area is identified.

Assessment:

- Both requirements reference the same parameter (``CRM_CMD 0x8 Parameter MemoryArea``), ensuring consistency and alignment in how the memory area is identified.

Additional Conditions Check:

- **SWRS 008 Additional Condition:**
 - SWRS 008 introduces no additional conditions beyond what is specified in SYSRS 007.

Assessment:

- There are no additional conditions introduced, indicating that the software requirement remains consistent with the system requirement.

Ratings:

- **Meaningfulness Rating: High**
 - **Reason:** SWRS 008 clearly and directly supports the system requirement. The conditions, actions, and parameters are perfectly aligned, ensuring that the software requirement meaningfully implements the system behavior.
- **Completeness Rating: High**
 - **Reason:** SWRS 008 fully addresses the conditions and actions described in SYSRS 007. No details are missing, and the parameters are consistently referenced.
- **Overall Quality Rating: High**
 - **Reason:** The overall quality is high because SWRS 008 accurately implements the system requirement in full. The conditions, parameters, and actions are all perfectly aligned, ensuring a strong link between the system and software requirements.

Suggested Revisions:

- No revisions are necessary, as **SWRS 008** fully implements the conditions and actions specified in **SYSRS 007**. The link between the system and software requirements is complete and clear.



```
"SYSRS 006: If Performing Memory Access in Unicast
AND IF the state of the Memory Area identified by CRM_CMD 0x8 Parameter MemoryArea for the
operation as determined by CRM_CMD 0x8 Operation Flag is in the state Locked,
THEN
the Sensor IC shall
respond with CRM_RSP 0x8: Report Lock Status and CRM_RSP 0x8 Flag KeyRequest = 0b1 ("key
requested").
"
"SWRS 007: If Performing Memory Access in Unicast
AND IF the state of the Memory Area identified by CRM_CMD 0x8 Parameter MemoryArea for the
operation as determined by CRM_CMD 0x8 Operation Flag is in the state Locked,
THEN
the Software shall prepare to
respond with CRM_RSP 0x8: Report Lock Status and CRM_RSP 0x8 Flag KeyRequest = 0b1 ("key
requested").
"
```



Evaluation of SYSRS 006 and Linked SWRS 007

SYSRS 006:

- **Condition:**
 - **IF performing memory access in Unicast,**
 - **AND IF the state of the Memory Area identified by `CRM_CMD 0x8 Parameter MemoryArea` for the operation, as determined by the `CRM_CMD 0x8 Operation Flag`, is in the state Locked.**
- **Action:**
 - The Sensor IC shall respond with `CRM_RSP 0x8: Report Lock Status` and `CRM_RSP 0x8 Flag KeyRequest = 0b1` ("key requested").

SWRS 007:

- **Condition:**
 - **IF performing memory access in Unicast,**
 - **AND IF the state of the Memory Area identified by `CRM_CMD 0x8 Parameter MemoryArea` for the operation, as determined by the `CRM_CMD 0x8 Operation Flag`, is in the state Locked.**
- **Action:**
 - The software shall prepare to respond with `CRM_RSP 0x8: Report Lock Status` and `CRM_RSP 0x8 Flag KeyRequest = 0b1` ("key requested").

Condition Check:

- **SYSRS 006 Condition:**
 - The system requirement specifies that the condition is triggered when the memory access is performed in Unicast mode, and the Memory Area is in the state Locked.
- **SWRS 007 Condition:**
 - The software requirement mirrors the exact conditions described in SYSRS 006, specifying memory access in Unicast mode and checking if the Memory Area is Locked.

Assessment:

- The conditions are fully aligned between SYSRS 006 and SWRS 007. Both requirements use the same criteria for triggering the action, ensuring consistency in when the response should occur.

Action and Response Check:

- **SYSRS 006 Action:**
 - The system requirement specifies that the Sensor IC should respond with ``CRM_RSP 0x8: Report Lock Status`` and ``CRM_RSP 0x8 Flag KeyRequest = 0b1`` (indicating "key requested").
- **SWRS 007 Action:**
 - The software requirement specifies that the software should **prepare to** respond with the same information: ``CRM_RSP 0x8: Report Lock Status`` and ``CRM_RSP 0x8 Flag KeyRequest = 0b1``.

Assessment:

- The actions are almost identical, but SWRS 007 introduces the phrase **"prepare to respond"** instead of directly stating that the software will respond. This distinction might reflect an implementation detail, such as a staging process before the actual response is sent, but it introduces slight ambiguity about when the response occurs. The intent and content of the response remain the same.

Parameter Sources Check:

- **SYSRS 006 Parameters:**
 - The system requirement references the ``CRM_CMD 0x8 Parameter MemoryArea`` and the ``CRM_CMD 0x8 Operation Flag``, as well as the response fields ``CRM_RSP 0x8: Report Lock Status`` and ``CRM_RSP 0x8 Flag KeyRequest = 0b1``.
- **SWRS 007 Parameters:**
 - The software requirement uses the same parameters (``CRM_CMD 0x8 Parameter MemoryArea``, ``CRM_CMD 0x8 Operation Flag``, ``CRM_RSP 0x8``, and ``Flag KeyRequest = 0b1``), ensuring consistency between the system and software requirements.

Assessment:

- Both requirements reference the same parameters, ensuring that the information needed for the memory area and response is handled consistently.

Additional Conditions Check:

- **SWRS 007 Additional Condition:**
 - SWRS 007 introduces no additional conditions beyond what is specified in SYSRS 006.

Assessment:

- No additional conditions are introduced, indicating that the software requirement remains fully aligned with the system requirement.

Ratings:

- **Meaningfulness Rating: High**
 - **Reason:** SWRS 007 directly supports the system requirement by implementing the same conditions and parameters. The slight difference in phrasing ("prepare to respond" vs. "respond") could reflect an implementation detail but does not significantly alter the meaning or intent.
- **Completeness Rating: Medium**
 - **Reason:** SWRS 007 introduces the phrase "prepare to respond," which could suggest that the response is not immediately sent. This creates some ambiguity regarding whether the software is fully addressing the need to respond as described in SYSRS 006. Clarification would be helpful to ensure that the response is not just prepared but actually sent.
- **Overall Quality Rating: High**
 - **Reason:** The overall quality is high because SWRS 007 properly implements the conditions and actions described in SYSRS 006. The only minor issue is the potential ambiguity in the phrasing "prepare to respond," which could benefit from clarification but does not detract significantly from the implementation.

Suggested Revisions:

- **SWRS 007** could clarify whether the software simply prepares the response or actually completes the response as required by **SYSRS 006**. If "prepare to respond" is an implementation step before sending the response, it should be documented, but if the intent is to match the system requirement's behavior, it should state that the software **will respond** with the necessary information.



"SYSRS 005: IF the Sensor IC generates more information than possible to store in Event Buffer or Time Series Data Buffer
THEN
the Sensor IC shall forget the according information." "SWRS 005: - The software shall store information for up to 5 Echo Events.

SWRS 006: - The software shall hold at least the information for up to 100 time series data samples (time stamp and value)
IF NOT possible regarding memory constraints could be reduced up to 50 time series data samples. "



Evaluation of SYSRS 005 and Linked SWRS 005, SWRS 006

SYSRS 005:

- **Condition:**
 - **IF the Sensor IC generates more information than possible to store in the Event Buffer or Time Series Data Buffer.**
- **Action:**
 - The Sensor IC shall forget the corresponding information.

SWRS 005:

- **Action:**
 - The software shall store information for up to **5 Echo Events**.

SWRS 006:

- **Action:**
 - The software shall hold at least the information for up to **100 time series data samples** (time stamp and value).
 - **IF NOT possible regarding memory constraints**, this can be reduced to **50 time series data samples**.

Condition Check:

- **SYSRS 005 Condition:**
 - The system requirement specifies that the action should occur if more information is generated than can be stored in the **Event Buffer** or **Time Series Data Buffer**.
- **SWRS 005 and SWRS 006 Condition:**
 - Neither SWRS 005 nor SWRS 006 explicitly checks for the situation where **more information than can be stored** is generated. SWRS 006 references memory constraints, but it doesn't explicitly mention handling scenarios where buffers are full or exceeding capacity.

Assessment:

- The condition described in SYSRS 005 (information exceeding storage capacity) is not explicitly mirrored in either SWRS 005 or SWRS 006. SWRS 006 mentions memory constraints, but it does not directly address the system requirement's condition of excess information in the Event Buffer or Time Series Data Buffer.

Action and Response Check:

- **SYSRS 005 Action:**
 - The system requirement specifies that if the buffers are full, the Sensor IC should **forget** the excess information.
- **SWRS 005 Action:**
 - SWRS 005 specifies that the software will store up to **5 Echo Events** but does not mention what happens if more Echo Events are generated than can be stored.
- **SWRS 006 Action:**

- SWRS 006 specifies that the software will store up to **100 time series data samples** and reduce this to **50** if memory constraints arise. However, there is no mention of what happens if the software generates more information than can be stored.

Assessment:

- The system requirement clearly specifies that excess information should be **forgotten** if the buffers are full, but neither SWRS 005 nor SWRS 006 directly addresses this action. SWRS 006 provides a strategy for handling memory constraints by reducing the number of stored time series data samples, but it does not specify the handling of buffer overflows or what happens when the buffer is exceeded.

Parameter Sources Check:

- **SYSRS 005 Parameters:**
 - The system requirement refers to the **Event Buffer** and **Time Series Data Buffer** as storage locations for information.
- **SWRS 005 Parameters:**
 - SWRS 005 refers to **Echo Events**, but it does not explicitly mention the Event Buffer.
- **SWRS 006 Parameters:**
 - SWRS 006 refers to **time series data samples** and addresses memory constraints but does not explicitly mention the Time Series Data Buffer.

Assessment:

- The system requirement's reference to the **Event Buffer** and **Time Series Data Buffer** is implied in SWRS 005 and SWRS 006 but is not explicitly mentioned. Furthermore, the handling of excess information beyond the capacity of these buffers is missing in the software requirements.

Additional Conditions Check:

- **SWRS 005 and SWRS 006 Additional Conditions:**
 - SWRS 006 introduces the condition that memory constraints could reduce the number of stored time series data samples from 100 to 50, but it does not fully address what happens if more information is generated than can be stored.

Assessment:

- SWRS 006 introduces a condition related to reducing data stored based on memory constraints but does not address the system requirement's key condition of handling excess data in the event that the buffers overflow.

Ratings:

- **Meaningfulness Rating: Medium**
 - **Reason:** SWRS 005 and SWRS 006 partially address the system requirement by specifying limits for storing Echo Events and time series data samples. However, they fail to address the specific scenario where more information is generated than can be stored. The handling of excess information (as required by SYSRS 005) is missing.
- **Completeness Rating: Low**

- **Reason:** SWRS 005 and SWRS 006 do not fully implement the required behavior described in SYSRS 005. While they set limits on the number of stored data points, they do not specify what happens when these limits are exceeded, nor do they implement the behavior of "forgetting" the excess information, as required by SYSRS 005.

- **Overall Quality Rating: Medium**

- **Reason:** The overall quality is medium because SWRS 005 and SWRS 006 implement partial aspects of the system requirement by defining storage limits. However, the failure to address the overflow scenario and the lack of explicit handling of buffer overflows reduce the quality of the link between the system and software requirements.

Suggested Revisions:

- **SWRS 005 and SWRS 006** should be revised to:
 - Explicitly mention how the software handles scenarios where more information is generated than can be stored in the Event Buffer or Time Series Data Buffer, as required by **SYSRS 005**.
 - Implement the "forgetting" behavior when the buffer capacity is exceeded, ensuring alignment with the system requirement.
 - Clarify if the Event Buffer and Time Series Data Buffer mentioned in **SYSRS 005** are directly addressed by the software limits on Echo Events and time series data samples, and provide explicit references to these buffers.



"SYSRS 004: If performing Erase Backup
AND If Addressed with the Unicast Command
THEN
the Sensor IC shall
respond with CRM_RSP 0x0: Report Acknowledgement." "SWRS 004: If performing Erase Backup
AND If Addressed with the Unicast Command
THEN
the Software shall prepare to
respond with CRM_RSP 0x0: Report Acknowledgement."



Evaluation of SYSRS 004 and Linked SWRS 004

SYSRS 004:

- **Condition:**
 - **IF performing Erase Backup,**
 - **AND IF addressed with the Unicast Command.**
- **Action:**
 - The Sensor IC shall respond with `CRM_RSP 0x0: Report Acknowledgement`.

SWRS 004:

- **Condition:**
 - **IF performing Erase Backup,**
 - **AND IF addressed with the Unicast Command.**
 - **Action:**
 - The software shall prepare to respond with ``CRM_RSP 0x0: Report Acknowledgement``.
-

Condition Check:

- **SYSRS 004 Condition:**
 - The system requirement specifies that the action should occur when performing the Erase Backup operation and being addressed with a Unicast Command.
- **SWRS 004 Condition:**
 - The software requirement exactly mirrors the same conditions described in SYSRS 004: performing Erase Backup and being addressed with a Unicast Command.

Assessment:

- The conditions in SWRS 004 are fully aligned with those in SYSRS 004, ensuring that the software requirement matches the system requirement in terms of when the action should occur.

Action and Response Check:

- **SYSRS 004 Action:**
 - The system requirement specifies that the Sensor IC should respond with ``CRM_RSP 0x0: Report Acknowledgement``.
- **SWRS 004 Action:**
 - The software requirement specifies that the software shall **prepare to** respond with ``CRM_RSP 0x0: Report Acknowledgement``.

Assessment:

- The actions are almost identical, but SWRS 004 introduces the phrase "**prepare to respond**" rather than directly stating that the software will respond. This may reflect an implementation detail, such as a preparation step before sending the actual response, but it introduces slight ambiguity about when the response occurs. The intent and content of the response remain consistent with SYSRS 004.

Parameter Sources Check:

- **SYSRS 004 Parameters:**
 - The system requirement references ``CRM_RSP 0x0: Report Acknowledgement`` as the response.
- **SWRS 004 Parameters:**
 - SWRS 004 references the same response (``CRM_RSP 0x0: Report Acknowledgement``), ensuring consistency between the system and software requirements.

Assessment:

- Both requirements reference the same response parameter (``CRM_RSP 0x0``), ensuring that the correct information is communicated in both the system and software actions.

Additional Conditions Check:

- **SWRS 004 Additional Condition:**
 - SWRS 004 does not introduce any additional conditions beyond those in SYSRS 004.

Assessment:

- There are no additional conditions introduced, meaning the software requirement is aligned with the system requirement.

Ratings:

- **Meaningfulness Rating: High**
 - **Reason:** SWRS 004 directly supports the system requirement by specifying the same conditions and actions. The phrase "prepare to respond" could reflect an implementation detail but does not significantly change the intent of the action.
- **Completeness Rating: Medium**
 - **Reason:** SWRS 004 introduces the phrase "prepare to respond," which might imply a preparation step before the actual response is sent. This introduces some ambiguity regarding whether the software is fully addressing the need to respond immediately as described in SYSRS 004. Clarification would help ensure full alignment.
- **Overall Quality Rating: High**
 - **Reason:** The overall quality is high because SWRS 004 fully implements the system requirement in terms of conditions and actions. The only minor issue is the potential ambiguity introduced by the "prepare to respond" phrasing, which could benefit from clarification.

Suggested Revisions:

- **SWRS 004** could clarify whether the software simply prepares the response or actually sends the response as required by **SYSRS 004**. If "prepare to respond" reflects an implementation step, it should be documented, but if the intent is to match the system requirement, the phrasing should indicate that the software will directly respond with `CRM_RSP 0x0`.



```
"SYSRS 003: BEFORE Sending CRM_RCC After Reception of the CRM_FCC,
THEN
the Sensor IC shall
set CRM_RSP Parameter CrmStatus = Status Value.
"  "SWRS 003: BEFORE responding with the CRM_RSP,
the Software shall include the CRM_RSP Parameter CrmStatus into the CRM_RSP.

"
```



Evaluation of SYSRS 003 and Linked SWRS 003

SYSRS 003:

- **Condition:**
 - **BEFORE** sending `CRM_RCC` after reception of the `CRM_FCC`.
- **Action:**
 - The Sensor IC shall set the `CRM_RSP Parameter CrmStatus = Status Value`.

SWRS 003:

- **Condition:**
 - **BEFORE** responding with the `CRM_RSP`.
- **Action:**
 - The software shall include the `CRM_RSP Parameter CrmStatus` into the `CRM_RSP`.

Condition Check:

- **SYSRS 003 Condition:**
 - The system requirement specifies that the action must occur **before sending the `CRM_RCC`** after the reception of the `CRM_FCC`.
- **SWRS 003 Condition:**
 - The software requirement specifies that the action must occur **before responding with the `CRM_RSP`**.

Assessment:

- There is a slight difference in the condition:
 - **SYSRS 003** refers to setting the `CRM_RSP Parameter CrmStatus` before sending `CRM_RCC` after receiving `CRM_FCC`.
 - **SWRS 003** mentions that this should happen before responding with the `CRM_RSP`, but it does not explicitly reference the relationship between `CRM_RCC` and `CRM_FCC`.

This difference suggests that the software requirement may not fully capture the specific timing relationship described in SYSRS 003.

Action and Response Check:

- **SYSRS 003 Action:**
 - The system requirement specifies that the `CRM_RSP Parameter CrmStatus` should be set to the **Status Value**.
- **SWRS 003 Action:**
 - The software requirement specifies that the `CRM_RSP Parameter CrmStatus` should be included in the `CRM_RSP`.

Assessment:

- Both SYSRS 003 and SWRS 003 refer to the handling of the `CrmStatus` parameter, but **SYSRS 003** specifies setting this parameter to a specific **Status Value**, while **SWRS 003** only mentions including the parameter in the response. The software requirement does not mention explicitly setting `CrmStatus` to a specific value, which is a critical part of the action in SYSRS 003.

Parameter Sources Check:

- **SYSRS 003 Parameters:**
 - The system requirement references the ``CRM_RSP Parameter CrmStatus`` and specifies setting it to the **Status Value**.
- **SWRS 003 Parameters:**
 - The software requirement also references the ``CRM_RSP Parameter CrmStatus``, but it does not mention the **Status Value** or specify what the ``CrmStatus`` should be set to.

Assessment:

- The software requirement references the correct parameter (``CRM_RSP Parameter CrmStatus``), but it does not address the need to set this parameter to a specific **Status Value**, as required by SYSRS 003.

Additional Conditions Check:

- **SWRS 003 Additional Condition:**
 - SWRS 003 does not introduce any additional conditions beyond what is specified in SYSRS 003.

Assessment:

- There are no additional conditions introduced, but the omission of setting the ``CrmStatus`` to a **Status Value** remains a gap in the software requirement.

Ratings:

- **Meaningfulness Rating: Medium**
 - **Reason:** SWRS 003 partially supports the system requirement by including the ``CrmStatus`` parameter in the response, but it does not address the key action of setting the parameter to a specific **Status Value**. Additionally, it does not fully capture the timing relationship between ``CRM_RCC`` and ``CRM_FCC`` described in SYSRS 003.
- **Completeness Rating: Low**
 - **Reason:** SWRS 003 fails to implement a critical aspect of SYSRS 003—setting the ``CrmStatus`` to a specific **Status Value**. The software requirement only mentions including the parameter in the response, but the system requirement specifically requires that this parameter be set before sending ``CRM_RCC`` after receiving ``CRM_FCC``.
- **Overall Quality Rating: Medium**
 - **Reason:** The overall quality is medium because the software requirement correctly references the ``CrmStatus`` parameter but lacks the key action of setting it to a **Status Value** and does not fully address the timing specified in SYSRS 003.

Suggested Revisions:

- **SWRS 003** should be revised to:

- Explicitly state that the software will set the `CRM_RSP Parameter CrmStatus` to the **Status Value**, as required by **SYSRS 003**.
- Include the condition of setting `CrmStatus` **before sending** `CRM_RCC` **after receiving** `CRM_FCC` to fully align with the system requirement's timing expectations.



SYSRS 002: The Sensor IC FW shall have a watchdog to detect FW or HW malfunction by resetting the WDG time in appropriate time within the FTTI. SWRS 002: The software shall enable a watchdog to detect software or hardware malfunction AND IF a malfunction is present, perform a reset in an appropriate time within the Fault Tolerant Time Interval of 20ms



Evaluation of SYSRS 002 and Linked SWRS 002

SYSRS 002:

- **Condition:**
 - The Sensor IC firmware shall have a **watchdog** to detect firmware (FW) or hardware (HW) malfunction by resetting the **WDG** (watchdog) time in an appropriate time within the **Fault Tolerant Time Interval (FTTI)**.

SWRS 002:

- **Condition:**
 - The software shall enable a **watchdog** to detect software or hardware malfunction, and **IF a malfunction is present**, perform a reset within the **Fault Tolerant Time Interval (FTTI)** of 20ms.

Condition Check:

- **SYSRS 002 Condition:**
 - The system requirement specifies that the **watchdog** should detect firmware or hardware malfunction and reset the watchdog time within the Fault Tolerant Time Interval (FTTI).
- **SWRS 002 Condition:**
 - The software requirement also specifies enabling a **watchdog** to detect software or hardware malfunction. However, SWRS 002 adds the condition "**IF a malfunction is present**", which is not explicitly mentioned in SYSRS 002.

Assessment:

- SWRS 002 introduces the additional condition "**IF a malfunction is present**", which suggests that the reset is conditional upon detecting a malfunction. This is implicit in SYSRS 002, but not explicitly stated. While this added condition may reflect the correct software behavior, it introduces a slight difference from the system requirement's broader phrasing.

Action and Response Check:

- **SYSRS 002 Action:**
 - The system requirement specifies that the Sensor IC firmware should reset the **watchdog time** within the **FTTI** to detect and handle malfunctions.
- **SWRS 002 Action:**
 - The software requirement specifies that the software will **perform a reset** in the event of a malfunction within the Fault Tolerant Time Interval (FTTI) of 20ms.

Assessment:

- SWRS 002 specifies performing a reset **if a malfunction is detected**, whereas SYSRS 002 focuses on resetting the watchdog time to handle the malfunction. The concept of performing a reset is aligned with the idea of resetting the watchdog, but SWRS 002's action is more explicitly tied to malfunction detection and does not emphasize resetting the watchdog timer as part of regular operation (to avoid a reset).

Parameter Sources Check:

- **SYSRS 002 Parameters:**
 - The key parameters in SYSRS 002 include the **watchdog time** and the **FTTI** within which the reset must occur.
- **SWRS 002 Parameters:**
 - SWRS 002 references the **watchdog** and explicitly states the **FTTI** as 20ms. This is an added level of specificity not mentioned in SYSRS 002, which does not define the specific value of the FTTI.

Assessment:

- Both requirements reference the **watchdog** and the **Fault Tolerant Time Interval**, but SWRS 002 adds the explicit value of 20ms for the FTTI, which is not mentioned in SYSRS 002. This addition could represent implementation-specific information but would need to be consistent with the system requirement.

Additional Conditions Check:

- **SWRS 002 Additional Condition:**
 - SWRS 002 introduces the condition of **"IF a malfunction is present"** for performing the reset, which is not explicitly required by SYSRS 002.

Assessment:

- This additional condition represents a more specific implementation scenario where the reset is triggered only upon detecting a malfunction, which is a logical refinement of the system requirement but not directly stated in SYSRS 002.

Ratings:

- **Meaningfulness Rating: Medium**
 - **Reason:** SWRS 002 generally supports SYSRS 002 by implementing the watchdog functionality and referencing the FTTI. However, the added condition **"IF a malfunction is present"** and the focus on performing a reset rather than resetting the watchdog time introduces slight differences in how the requirement is interpreted.

- **Completeness Rating: Medium**

- **Reason:** SWRS 002 does not fully implement the watchdog reset behavior described in SYSRS 002. It focuses on resetting the system in response to a malfunction rather than resetting the watchdog time proactively to prevent a malfunction. Additionally, the explicit reference to the FTTI as 20ms could be a useful detail but is not mentioned in SYSRS 002.

- **Overall Quality Rating: Medium**

- **Reason:** The overall quality is medium because SWRS 002 captures the general intent of SYSRS 002 but introduces additional conditions and focuses on handling malfunctions rather than preventing them by resetting the watchdog time. The explicit FTTI value also introduces some specificity that may not align with SYSRS 002.

Suggested Revisions:

- **SWRS 002** should:
 - Clarify that the software will **reset the watchdog timer** regularly within the Fault Tolerant Time Interval to prevent malfunctions, as implied by **SYSRS 002**, in addition to performing a reset if a malfunction is detected.
 - Ensure that the value of 20ms for the **Fault Tolerant Time Interval** is aligned with the system requirement or mention that it is an implementation-specific detail if not required by the system requirement.



SYSRS 001: The Sensor IC shall digitize the Transducer Input Signal at the IC Pins with sampling frequency of (CALIB_samplingFreq plus YPX_samplingFreqOffset). "SWRS 001: AFTER reception of the command CRM_CMD 0x1: Start Standard Measurement AND BEFORE the start of the Measurement the software shall set the register BRG.FS.f_s to (CALIB_samplingFreq plus YPX_samplingFreqOffset). "



Evaluation of SYSRS 001 and Linked SWRS 001

SYSRS 001:

- **Condition:**
 - The Sensor IC shall digitize the **Transducer Input Signal** at the IC pins with a sampling frequency of `(CALIB_samplingFreq + YPX_samplingFreqOffset)`.

SWRS 001:

- **Condition:**
 - **AFTER reception of the command `CRM_CMD 0x1: Start Standard Measurement`,**
 - **AND BEFORE the start of the Measurement,** the software shall set the register `BRG.FS.f_s` to `(CALIB_samplingFreq + YPX_samplingFreqOffset)`.

Condition Check:

- **SYSRS 001 Condition:**
 - The system requirement simply states that the Sensor IC shall digitize the transducer input signal with a specific sampling frequency.
- **SWRS 001 Condition:**
 - The software requirement adds a specific condition: this action occurs **after receiving the** ``CRM_CMD 0x1`` **command** (Start Standard Measurement) and **before the start of the measurement**.

Assessment:

- SWRS 001 introduces additional conditions that are not mentioned in SYSRS 001, including the need for a command (``CRM_CMD 0x1``) and the timing (before the start of the measurement). These additions may reflect practical implementation details but are not specified in SYSRS 001.

Action and Response Check:

- **SYSRS 001 Action:**
 - The system requirement specifies that the Sensor IC shall digitize the transducer input signal using a specific sampling frequency ``(CALIB_samplingFreq + YPX_samplingFreqOffset)``.
- **SWRS 001 Action:**
 - The software requirement specifies setting the register ``BRG.FS.f_s`` to ``(CALIB_samplingFreq + YPX_samplingFreqOffset)`` before starting the measurement.

Assessment:

- Both SYSRS 001 and SWRS 001 refer to the same action of setting the sampling frequency to ``(CALIB_samplingFreq + YPX_samplingFreqOffset)``. However, SWRS 001 focuses on setting the **register** before measurement, while SYSRS 001 is concerned with the **digitization** process.

Parameter Sources Check:

- **SYSRS 001 Parameters:**
 - The system requirement mentions the parameters ``CALIB_samplingFreq`` and ``YPX_samplingFreqOffset``.
- **SWRS 001 Parameters:**
 - The software requirement references the same parameters, ``CALIB_samplingFreq`` and ``YPX_samplingFreqOffset``, and sets them to the register ``BRG.FS.f_s``.

Assessment:

- Both requirements reference the same parameters, ensuring consistency in the values used for the sampling frequency.

Additional Conditions Check:

- **SWRS 001 Additional Condition:**
 - SWRS 001 adds the condition of receiving the ``CRM_CMD 0x1`` command and the timing constraint of performing the action before the start of the measurement.

Assessment:

- These additional conditions provide more context for the software's implementation but are not required by SYSRS 001. They do not contradict the system requirement but represent a more detailed execution plan.

Ratings:

- **Meaningfulness Rating: High**
 - **Reason:** SWRS 001 closely aligns with SYSRS 001 by implementing the action of setting the sampling frequency. While it introduces additional conditions regarding the timing and command reception, these seem to be practical implementation details and do not detract from the meaningfulness of the link.
- **Completeness Rating: Medium**
 - **Reason:** SWRS 001 does not fully implement the digitization process described in SYSRS 001. It focuses on setting the sampling frequency before starting the measurement but does not explicitly mention the **digitization** of the transducer input signal, which is central to SYSRS 001. The software requirement addresses the preparatory step but omits the actual digitization process.
- **Overall Quality Rating: Medium**
 - **Reason:** While SWRS 001 addresses the key parameters and timing for setting the sampling frequency, it does not fully align with SYSRS 001's emphasis on digitization. The additional conditions (command reception and timing) are logical but should not replace the core action of digitization. The overall quality is medium due to this gap in focus.

Suggested Revisions:

- **SWRS 001** should:
 - Clarify that the software not only sets the sampling frequency but also ensures that the **digitization** of the transducer input signal occurs with the specified frequency ``(CALIB_samplingFreq + YPX_samplingFreqOffset)``, as required by **SYSRS 001**.
 - If the additional condition regarding the command ``CRM_CMD 0x1`` and the timing are necessary, ensure they are framed as implementation details without losing focus on the core digitization requirement from the system.

This conversation is archived. To continue, please unarchive it first.

 Unarchive