

Run #5



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 specific sets of System Requirements (SYSRS) and their corresponding Software Requirements (SWRS) so that I can proceed with the detailed analysis and evaluation.



Meaningfulness of the link:

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 of the link:

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.



Thanks for sharing the evaluation criteria for rating the link between the system requirements (SYSRS) and their corresponding software requirements (SWRS). Please provide the specific requirements for evaluation, and I will proceed with the detailed analysis based on the criteria provided.



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 System Requirement SYSRS 013 and Linked Software Requirements SWRS 013, 014, and 015

Condition Check:

- **SYSRS 013** specifies that the sensor IC should amplify the transducer input signal with specific conditions: using multiple parameters, including *CALIB_analogGain*, *SPX_dynGainStartValue*, *CRM_CMD 0x4 Parameter CalibrationGain*, and *YPX_calibrationGainOffset*.

SWRS 013, 014, 015 all describe actions triggered by the reception of the *CRM_CMD 0x4: Start Calibration Measurement*. This condition is consistent across all software requirements, matching the conditional trigger in the system requirement.

Outcome: The condition specified in the system requirement (triggered by *CRM_CMD 0x4*) is consistently reflected in the software requirements.

Action and Response Check:

- **SYSRS 013** describes a series of actions regarding amplification using four parameters.

SWRS 013 specifies that *CALIB_analogGain* is applied to the register *ASP_CFG.g_ana*.

SWRS 014 specifies that *SPX_dynGainStartValue* is applied to the register *ENVP_AMPD_CFG.g_dig*.

SWRS 015 specifies that both *CalibrationGain* and *YPX_calibrationGainOffset* are applied together to the register *ENVP_GCAL.g_cal*.

Outcome: The actions and responses specified in the system requirement are clearly detailed in the software requirements, with each parameter being set to the appropriate register, covering all actions required by the system.

Parameter Sources Check:

- **SYSRS 013** mentions four parameters: *CALIB_analogGain*, *SPX_dynGainStartValue*, *CRM_CMD 0x4 Parameter CalibrationGain*, and *YPX_calibrationGainOffset*.

Each software requirement addresses these parameters:

- **SWRS 013** covers *CALIB_analogGain*.
- **SWRS 014** covers *SPX_dynGainStartValue*.
- **SWRS 015** covers both *CalibrationGain* and *YPX_calibrationGainOffset*.

Outcome: All parameters mentioned in the system requirement are addressed in the software requirements, with each parameter being assigned to its respective register.

Additional Conditions Check:

- There are no additional conditions in the software requirements that are not reflected in the system requirement. The condition of *CRM_CMD 0x4: Start Calibration Measurement* is consistently used across the software requirements, and no extraneous conditions are introduced.

Outcome: No irrelevant or additional conditions are found in the software requirements.

Rating

Meaningfulness Rating: High

- **Reason:** The software requirements clearly and directly support the system requirement. Each software requirement aligns logically with the specific actions and parameters described in the system requirement, ensuring that the amplification factors are applied appropriately.

Completeness Rating: High

- **Reason:** The software requirements cover all aspects of the system requirement in full detail. Each parameter and action is accounted for, and no significant information is missing.

Overall Quality of the Link: High

- **Reason:** Both the meaningfulness and completeness ratings are high. The software requirements are strongly aligned with the system requirement, providing a clear and complete implementation.

Suggested Changes or Revisions:

- No revisions are necessary as the software requirements fully cover the conditions, actions, responses, and parameters mentioned in the system requirement.



```
"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 System Requirement SYSRS 003 and Linked Software Requirement SWRS 003

Condition Check:

- **SYSRS 003** specifies the condition: **BEFORE sending CRM_RCC after reception of CRM_FCC**, the Sensor IC shall set the CRM_RSP parameter *CrmStatus* to a *Status Value*.

SWRS 003 mentions the condition: **BEFORE responding with the CRM_RSP**, the software shall include the *CrmStatus* parameter into the CRM_RSP.

Outcome:

- **SYSRS 003** specifically refers to an event sequence involving the CRM_RCC and CRM_FCC, while **SWRS 003** simplifies this to just **before responding with CRM_RSP**. Although the core idea (setting the *CrmStatus* before sending the response) is captured, the reference to **CRM_RCC** and **CRM_FCC** is not explicitly mentioned in the software requirement.

This may lead to a potential gap, as the system requirement's condition involves both the CRM_RCC and CRM_FCC, whereas the software requirement omits these details, making the condition partially fulfilled.

Action and Response Check:

- **SYSRS 003** specifies that the Sensor IC shall set the *CrmStatus* parameter to a status value in the CRM_RSP.

SWRS 003 states that the software shall include the *CrmStatus* parameter in the CRM_RSP, but it does not explicitly mention setting this parameter to a **specific status value** as per the system requirement.

Outcome:

- The action described in the software requirement partially fulfills the system requirement. The software requirement does not address setting *CrmStatus* to a **specific status value**, which is a critical part of the system action. Thus, the action is incomplete.

Parameter Sources Check:

- **SYSRS 003** mentions the parameter *CrmStatus* and specifies that it should be set to a **status value**.

SWRS 003 addresses the *CrmStatus* parameter but does not clarify how or from where this parameter is being sourced or whether it is being set to a **specific status value**.

Outcome:

- The software requirement does not clarify the source of the *CrmStatus* parameter or how its value is determined, leaving ambiguity about the handling of this parameter. This needs further detail to match the system requirement fully.

Additional Conditions Check:

- **SWRS 003** does not introduce any additional conditions that are irrelevant to the system requirement. It remains focused on the *CrmStatus* parameter and the CRM_RSP response.

Outcome:

- No extraneous or unrelated conditions are introduced in the software requirement.

Rating

Meaningfulness Rating: Medium

- **Reason:** While the software requirement generally supports the system requirement, there is a lack of detail in the specific conditions (CRM_RCC and CRM_FCC) and the setting of the *CrmStatus* to a **status value**. The overall link is somewhat logical but could be stronger with better alignment of these aspects.

Completeness Rating: Medium

- **Reason:** The software requirement addresses the core aspect of including the *CrmStatus* in the CRM_RSP, but it misses critical details from the system requirement, such as the setting of a **specific status value** and the exact event flow involving CRM_RCC and CRM_FCC. These gaps affect the completeness.

Overall Quality of the Link: Medium

- **Reason:** Both meaningfulness and completeness are rated as medium, as the software requirement partially aligns with the system requirement but misses important details related to the event sequence and the setting of a specific value for *CrmStatus*.

Suggested Changes or Revisions:

1. **Clarify the Event Sequence:** The software requirement should explicitly mention the full sequence **CRM_RCC after CRM_FCC** to align with the system requirement.
2. **Specify the Setting of the Status Value:** The software requirement should ensure that *CrmStatus* is set to a **specific status value**, as described in the system requirement, rather than simply including the parameter.
3. **Identify the Source of the Status Value:** Add details about how the *CrmStatus* value is sourced or determined in the software requirement to provide more clarity.



"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 System Requirement SYSRS 005 and Linked Software Requirements SWRS 005 and SWRS 006

Condition Check:

- **SYSRS 005** specifies a condition: **IF** the Sensor IC generates more information than possible to store in the Event Buffer or Time Series Data Buffer, **THEN** it shall forget the according information.

SWRS 005 and **SWRS 006** mention different conditions:

- **SWRS 005** states that the software shall store information for up to 5 Echo Events, but it does not explicitly address what happens if the buffer overflows or how the system should behave if the buffer cannot store more events.
- **SWRS 006** addresses holding at least 100 time series data samples and mentions a fallback to 50 if memory constraints occur. However, it does not directly mention what happens when these limits are exceeded or how the system forgets the excess information.

Outcome:

- The software requirements partially capture the memory constraints mentioned in the system requirement but do not fully address the specified **condition** about the IC "forgetting" information when the buffer limits are exceeded. The handling of overflow conditions is not explicitly stated in either software requirement, which leaves the condition in SYSRS 005 unmet.

Action and Response Check:

- **SYSRS 005** specifies that **IF** buffer limits are exceeded, **THEN** the Sensor IC shall forget the extra information.
 - **SWRS 005** focuses on storing information for 5 Echo Events, but it does not describe any action when this limit is exceeded or how the system handles overflow.
 - **SWRS 006** describes handling time series data samples, with a fallback to 50 samples if memory constraints arise. However, it also does not specify any action or response for situations when even this reduced limit is exceeded, leaving the behavior in such cases unclear.

Outcome:

- The system action of "forgetting" information when buffer limits are exceeded is not clearly addressed in either software requirement. Both requirements focus on storage limits but omit the system's response when those limits are surpassed.

Parameter Sources Check:

- **SYSRS 005** does not specify particular parameters but implicitly refers to the Event Buffer and Time Series Data Buffer as storage spaces that can become full.
 - **SWRS 005** references the Echo Events (up to 5), while **SWRS 006** references the time series data samples (100, with a fallback to 50). However, neither explicitly addresses the **buffers** as described in the system requirement, leaving the buffer aspect somewhat abstract in the software requirements.

Outcome:

- The software requirements refer to limits for Echo Events and time series data samples but do not clarify how these limits relate to the specific **Event Buffer** and **Time Series Data Buffer** mentioned in the system requirement. This leaves some ambiguity about the relationship between the buffers and the stored data.

Additional Conditions Check:

- Neither **SWRS 005** nor **SWRS 006** introduce additional conditions that are unrelated to the system requirement. Both software requirements focus on the number of events and data samples stored.

Outcome:

- No irrelevant or additional conditions are present in the software requirements, but there is a lack of detail concerning the core condition about buffer overflow and the "forgetting" behavior.

Rating

Meaningfulness Rating: Medium

- **Reason:** The software requirements somewhat support the system requirement by mentioning limits for event storage and data samples. However, the condition of "forgetting" information when the buffers are full is not addressed, and the specific buffer terms mentioned in SYRS 005 are not clearly referenced.

Completeness Rating: Medium

- **Reason:** The software requirements provide partial coverage of the system requirement by setting limits for Echo Events and time series data samples, but they fail to describe what happens when these limits are exceeded, a critical aspect of the system requirement. Additionally, the relationship to the Event Buffer and Time Series Data Buffer is not made explicit.

Overall Quality of the Link: Medium

- **Reason:** Both meaningfulness and completeness are rated as medium. The software requirements cover some aspects of the system requirement but omit important details, particularly around the behavior when buffer limits are exceeded.

Suggested Changes or Revisions:

1. **Clarify Overflow Handling:** Both software requirements should explicitly describe what happens when the event or data sample storage limits are exceeded, aligning with the system requirement's condition that the IC "forgets" the excess information.
2. **Specify Buffer Usage:** The software requirements should directly refer to the **Event Buffer** and **Time Series Data Buffer** mentioned in the system requirement, ensuring clarity about how these buffers are involved in storing Echo Events and time series data samples.
3. **Define a Fallback Action:** Include a fallback action for both Echo Events and time series data if storage capacity is exceeded, in line with the system requirement's instruction to "forget" the excess data.



"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 System Requirement SYSRS 004 and Linked Software Requirement SWRS 004

Condition Check:

- **SYSRS 004** specifies the condition: **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 replicates the same conditions: **If performing Erase Backup AND If Addressed with the Unicast Command**, then the software shall prepare to respond with *CRM_RSP 0x0: Report Acknowledgement*.

Outcome:

- Both the system and software requirements include the same conditions of performing an **Erase Backup** and being **addressed with the Unicast Command**. There is no discrepancy between the conditions.

Action and Response Check:

- **SYSRS 004** specifies that the Sensor IC shall **respond** with *CRM_RSP 0x0: Report Acknowledgement*.

SWRS 004 states that the software shall **prepare to respond** with *CRM_RSP 0x0: Report Acknowledgement*. This introduces a subtle difference: the software requirement refers to preparation to respond, rather than directly stating that the response will be sent.

Outcome:

- The software requirement does not explicitly state that the response will be **sent**, only that the software will prepare to respond. This is an incomplete description of the action required in the system requirement, which clearly states that the Sensor IC should **respond**. The act of preparing is not equivalent to the actual response.

Parameter Sources Check:

- **SYSRS 004** specifies the response should be **CRM_RSP 0x0: Report Acknowledgement**.

SWRS 004 correctly refers to the same response parameter (*CRM_RSP 0x0: Report Acknowledgement*), ensuring that the necessary parameter is addressed.

Outcome:

- The parameter *CRM_RSP 0x0: Report Acknowledgement* is consistent between the system and software requirements, and no issues arise regarding parameter sources.

Additional Conditions Check:

- **SWRS 004** does not introduce any additional conditions that are not specified in the system requirement. It remains focused on the conditions of performing an Erase Backup and being addressed with the Unicast Command.

Outcome:

- No irrelevant or extraneous conditions are introduced in the software requirement.

Rating

Meaningfulness Rating: Medium

- **Reason:** The software requirement aligns with the system requirement by replicating the conditions and specifying the correct response. However, it only states that the software **prepares to respond** rather than ensuring the actual **response is sent**, which weakens the link between the two requirements.

Completeness Rating: Medium

- **Reason:** The software requirement covers most aspects of the system requirement but lacks the critical detail that the response should actually be **sent**. The act of preparing to respond is not sufficient to meet the system requirement fully, leading to an incomplete action description.

Overall Quality of the Link: Medium

- **Reason:** Both meaningfulness and completeness are rated as medium. While the software requirement correctly covers the conditions and the parameter, it does not fully describe the action of responding, leading to a medium overall quality.

Suggested Changes or Revisions:

1. **Clarify the Action of Responding:** The software requirement should be revised to explicitly state that the software **will respond** with *CRM_RSP 0x0: Report Acknowledgement*, rather than just preparing to respond. This would fully align it with the system requirement.
2. **Ensure Action Completeness:** Ensure that the description includes the actual transmission of the response, as preparation alone is not sufficient to meet the system requirement's intent.



```
"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 System Requirement SYSRS 006 and Linked Software Requirement SWRS 007

Condition Check:

- **SYSRS 006** specifies the condition: **If performing Memory Access in Unicast AND IF the state of the Memory Area identified by CRM_CMD 0x8 Parameter MemoryArea is Locked**, then the Sensor IC shall respond with *CRM_RSP 0x8: Report Lock Status* and set the *CRM_RSP 0x8 Flag KeyRequest = 0b1 ("key requested")*.

SWRS 007 mirrors this condition: **If performing Memory Access in Unicast AND IF the state of the Memory Area identified by CRM_CMD 0x8 Parameter MemoryArea is Locked**, then the software shall prepare to respond with *CRM_RSP 0x8: Report Lock Status* and set *CRM_RSP 0x8 Flag KeyRequest = 0b1*.

Outcome:

- Both the system and software requirements share the same condition involving memory access, memory area identification via **CRM_CMD 0x8**, and the locked state. There is no discrepancy in the conditions, making this a good match.

Action and Response Check:

- **SYSRS 006** specifies that if the memory area is locked, the Sensor IC shall **respond** with *CRM_RSP 0x8: Report Lock Status* and set the *KeyRequest* flag to ****0b1 ("key requested")***.

SWRS 007, similar to previous cases, states that the software shall **prepare to respond** with *CRM_RSP 0x8: Report Lock Status* and set the *KeyRequest* flag to ****0b1 ("key requested")***. This wording introduces a difference: the system requirement explicitly calls for an immediate response, while the software requirement refers to **preparing** to respond.

Outcome:

- The software requirement does not fully capture the system requirement's action, which is to **respond** with the relevant CRM_RSP and set the *KeyRequest* flag. The action of merely **preparing** to respond is incomplete, as the system requirement specifies a definitive action (response) rather than a preparatory step.

Parameter Sources Check:

- **SYSRS 006** mentions the parameters *CRM_CMD 0x8 Parameter MemoryArea* and *CRM_CMD 0x8 Operation Flag* to determine the state of the memory area. It also specifies the response *CRM_RSP 0x8: Report Lock Status* and the flag *KeyRequest = 0b1*.

SWRS 007 includes all of these parameters: *CRM_CMD 0x8 Parameter MemoryArea*, *Operation Flag*, *CRM_RSP 0x8: Report Lock Status*, and *KeyRequest = 0b1*.

Outcome:

- The parameters mentioned in the system requirement are correctly referenced in the software requirement, and there are no missing or unclear parameter sources.

Additional Conditions Check:

- **SWRS 007** does not introduce any additional conditions that are not specified in **SYSRS 006**. It remains focused on memory access, the locked state of the memory area, and the response with the appropriate flags.

Outcome:

- There are no irrelevant or additional conditions in the software requirement.

Rating

Meaningfulness Rating: Medium

- **Reason:** The software requirement reflects the correct conditions and parameters but does not fully implement the system requirement. The system requirement specifies that the Sensor IC **shall respond**, while the software requirement only mentions preparing to respond, which reduces the clarity and direct alignment.

Completeness Rating: Medium

- **Reason:** The software requirement captures most aspects of the system requirement but omits the key action of actually sending the response. By focusing on **preparing to respond** rather than ensuring the response is sent, the software requirement leaves out a critical step, affecting completeness.

Overall Quality of the Link: Medium

- **Reason:** Both meaningfulness and completeness are rated as medium. The software requirement is closely aligned with the system requirement in terms of conditions and

parameters, but it does not fully describe the expected action, leading to a medium overall quality rating.

Suggested Changes or Revisions:

1. **Clarify the Action of Responding:** The software requirement should be revised to clearly state that the software will **respond** with *CRM_RSP 0x8: Report Lock Status* and set *KeyRequest = 0b1*, rather than just **preparing** to respond.
2. **Ensure Action Completeness:** Make sure the software requirement specifies the actual transmission of the response and the setting of the flag, as preparation alone does not fulfill the system requirement.



"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 System Requirement SYSRS 007 and Linked Software Requirement SWRS 008

Condition Check:

- **SYSRS 007** specifies a series of conditions: **If Performing Memory Access AND If the Key is Set AND If a Read Operation is Requested AND If the Memory Area is Set Read Accessible**, then the Sensor IC shall set the state of the memory area to "Locked."

SWRS 008 mirrors these conditions exactly: **If Performing Memory Access AND If the Key is Set AND If a Read Operation is Requested AND If the Memory Area is Set Read Accessible**, then the software shall set the state of the memory area to "Locked."

Outcome:

- Both the system and software requirements share identical conditions, with no discrepancies or omissions. The software requirement accurately reflects the conditional flow outlined in the system requirement.

Action and Response Check:

- **SYSRS 007** specifies that **if** all the conditions are met, the Sensor IC shall **set the state** of the Memory Area identified by *CRM_CMD 0x8 Parameter MemoryArea* to "Locked."

SWRS 008 reflects the same action: the software shall **set the state** of the memory area to "Locked" under the same conditions.

Outcome:

- The action described in the system requirement (setting the memory area to "Locked") is fully captured in the software requirement. Both requirements specify the same action with no ambiguities.

Parameter Sources Check:

- **SYSRS 007** refers to the parameter *CRM_CMD 0x8 Parameter MemoryArea* for identifying the memory area, which should be set to "Locked."

SWRS 008 also references the same parameter, *CRM_CMD 0x8 Parameter MemoryArea*, ensuring that the memory area is correctly identified in the software implementation.

Outcome:

- The parameters mentioned in the system requirement are correctly referenced in the software requirement. There are no ambiguities or missing parameter sources.

Additional Conditions Check:

- **SWRS 008** does not introduce any additional or irrelevant conditions. It remains strictly focused on the conditions and actions outlined in **SYSRS 007**.

Outcome:

- No additional or extraneous conditions are introduced in the software requirement, ensuring it remains relevant and aligned with the system requirement.

Rating

Meaningfulness Rating: High

- **Reason:** The software requirement is directly aligned with the system requirement, replicating the conditions, parameters, and actions without introducing any discrepancies. The link between them is strong and logical.

Completeness Rating: High

- **Reason:** The software requirement covers all necessary aspects of the system requirement in full detail, with no missing information or unclear areas. It fully describes the actions and conditions.

Overall Quality of the Link: High

- **Reason:** Both meaningfulness and completeness are rated as high. The software requirement is a perfect match for the system requirement, fully supporting its implementation.

Suggested Changes or Revisions:

- **No revisions needed.** The software requirement fully meets the expectations of the system requirement and implements it clearly and effectively.



"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 System Requirement SYSRS 008 and Linked Software Requirement SWRS 009

Condition Check:

- **SYSRS 008** specifies the following conditions: **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 to "Locked."

SWRS 009, however, specifies a different condition: **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 to "Locked."

Outcome:

- The conditions in **SWRS 009** differ from those in **SYSRS 008**:
 - **SYSRS 008** requires the **key to be set**, while **SWRS 009** explicitly states that the **key is NOT set**.
 - **SYSRS 008** references the memory area being **write accessible**, while **SWRS 009** refers to the memory area being **read accessible**.

- **SWRS 009** references *CRM_CMD 0x10* instead of *CRM_CMD 0x8*, which is specified in **SYSRS 008**.

These discrepancies indicate that the software requirement does not fully align with the system requirement in terms of conditions.

Action and Response Check:

- **SYSRS 008** specifies that if the conditions are met, the Sensor IC shall **set the state** of the memory area to "Locked."

SWRS 009 also specifies the same action: setting the memory area to "Locked." However, since the conditions differ significantly between the system and software requirements, this action is not aligned with the action expected by the system requirement.

Outcome:

- The action in the software requirement (setting the memory area to "Locked") aligns in terms of the outcome but is triggered under different conditions than those specified in the system requirement. Therefore, the action is not fully supported.

Parameter Sources Check:

- **SYSRS 008** refers to the parameter *CRM_CMD 0x8 Parameter MemoryArea* for identifying the memory area.

SWRS 009, however, refers to *CRM_CMD 0x10 Parameter MemoryArea*. This is a different parameter from what is specified in the system requirement.

Outcome:

- The parameter source mentioned in the software requirement (*CRM_CMD 0x10*) does not match the one specified in the system requirement (*CRM_CMD 0x8*), leading to a misalignment in parameter handling.

Additional Conditions Check:

- **SWRS 009** introduces the condition that the **key is NOT set** and that the memory area is **read accessible**, which are not mentioned in **SYSRS 008**. These additional conditions are not relevant to the system requirement.

Outcome:

- The software requirement introduces additional and unrelated conditions (key not set, read accessible memory), which makes it less relevant to the system requirement.

Rating

Meaningfulness Rating: Low

- **Reason:** The software requirement has significantly different conditions than the system requirement. It introduces new conditions (key not set, memory area read accessible) and refers to a different command parameter, making the link between them weak and unclear.

Completeness Rating: Low

- **Reason:** The software requirement does not fully address the conditions or actions specified in the system requirement. It refers to different conditions, parameters, and operations, making it incomplete in terms of fulfilling the system requirement.

Overall Quality of the Link: Low

- **Reason:** Both meaningfulness and completeness are rated low. The software requirement does not align with the system requirement, introducing different conditions and actions, leading to a poor overall link.

Suggested Changes or Revisions:

1. **Revise Conditions:** The software requirement should be revised to reflect the conditions in **SYSRS 008**:
 - The key should be **set** (not "NOT set").
 - The memory area should be **write accessible** (not "read accessible").
 - The appropriate command parameter (*CRM_CMD 0x8*) should be used instead of *CRM_CMD 0x10*.
2. **Remove Irrelevant Conditions:** Conditions such as the memory area being "read accessible" and the key being "not set" should be removed, as they do not align with the system requirement.
3. **Ensure Consistency in Parameters:** The parameter reference should match the system requirement (*CRM_CMD 0x8 Parameter MemoryArea*) to ensure consistency between the system and 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 System Requirement SYSRS 009 and Linked Software Requirement SWRS 010

Condition Check:

- **SYSRS 009** specifies the condition: **If Performing Memory Access in Broadcast AND IF the state of the Memory Area identified by CRM_CMD 0x8 Parameter MemoryArea 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 mirrors this condition: **If Performing Memory Access in Broadcast AND IF the state of the Memory Area identified by CRM_CMD 0x8 Parameter MemoryArea 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")*.

Outcome:

- The conditions in both the system and software requirements are identical, involving memory access in broadcast mode and the locked state of the memory area. There is no discrepancy in the conditions.

Action and Response Check:

- **SYSRS 009** specifies that the Sensor IC shall **respond** with *PDCM_RSP P0* containing the required information.

SWRS 010 states that the software shall **prepare to respond** with *PDCM_RSP P0* containing the same information.

Outcome:

- Similar to previous examples, the software requirement states that the software will "prepare to respond," rather than explicitly stating that it will **respond**. This introduces a slight difference, as "preparing to respond" is not equivalent to the actual response being sent, which the system requirement specifies.

Parameter Sources Check:

- **SYSRS 009** refers to several parameters: *CRM_CMD 0x8 Parameter MemoryArea*, *CRM_CMD 0x8 Operation Flag*, and *CRM_RSP 0x8: Report Lock Status* with *CRM_RSP 0x8 Flag KeyRequest = 0b1*.

SWRS 010 correctly references all these parameters, including *CRM_CMD 0x8 Parameter MemoryArea*, *CRM_CMD 0x8 Operation Flag*, and the corresponding response and flag values.

Outcome:

- The parameters referenced in the system requirement are accurately addressed in the software requirement. No parameter sources are missing or incorrectly referred to.

Additional Conditions Check:

- **SWRS 010** does not introduce any additional or irrelevant conditions. It stays focused on the broadcast memory access and the locked state of the memory area.

Outcome:

- No additional or extraneous conditions are introduced in the software requirement, making it fully relevant to the system requirement.

Rating

Meaningfulness Rating: Medium

- **Reason:** The software requirement largely supports the system requirement, but it does not fully capture the required action. The system requirement clearly states that the Sensor IC should **respond**, while the software requirement refers to merely **preparing to respond**. This weakens the clarity and meaningfulness of the link between the two.

Completeness Rating: Medium

- **Reason:** The software requirement covers most aspects of the system requirement, but it fails to include the critical action of actually sending the response. By only stating that the software will "prepare to respond," it leaves the required action incomplete.

Overall Quality of the Link: Medium

- **Reason:** Both meaningfulness and completeness are rated as medium. The software requirement addresses the conditions and parameters but does not fully align with the system requirement's expectations for action, leading to a medium overall quality rating.

Suggested Changes or Revisions:

1. **Clarify the Action of Responding:** The software requirement should explicitly state that the software will **respond** with *PDCM_RSP P0* containing *CRM_RSP 0x8: Report Lock Status* and *CRM_RSP 0x8 Flag KeyRequest = 0b1*, rather than just **preparing to respond**.
2. **Ensure Action Completeness:** The software requirement should make it clear that the response will be transmitted, as the system requirement expects an actual response rather than preparation.



"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 System Requirements SYSRS 010 and SYSRS 011 and Linked Software Requirement SWRS 011

Condition Check:

- **SYSRS 010** specifies that **if performing memory area modification AND after the reception of CRM_CMD 0x9: Take Data or Send Data**, the Sensor IC shall **cache the CRM_CMD 0x9 Parameter Data to the Data Cache**.

SYSRS 011 specifies the same condition (memory area modification after reception of CRM_CMD 0x9) but adds a different action: the Sensor IC shall **increment the CRM_CMD 0x8 Parameter StartAddress by 1 word address**.

SWRS 011 specifies a similar condition but adds a key difference: the software shall **store the CRM_CMD 0x9 Parameter Data to RAM** and wait for the **CRM_CMD 0xA: Perform Memory Operation with MemoryOperation == 0x2 "Write to NVM"** before proceeding.

Outcome:

- The condition in **SWRS 011** generally aligns with the system requirements (memory modification and reception of CRM_CMD 0x9). However, **SWRS 011** introduces the additional condition involving **CRM_CMD 0xA** and the operation to "write to NVM," which is not mentioned in either **SYSRS 010** or **SYSRS 011**. This introduces a divergence from the original conditions outlined in the system requirements.

Action and Response Check:

- **SYSRS 010** specifies that the Sensor IC should **cache the CRM_CMD 0x9 Parameter Data to the Data Cache**.

SYSRS 011 specifies that the Sensor IC should **increment the CRM_CMD 0x8 Parameter StartAddress by 1 word address**.

SWRS 011 describes a similar but not identical action: it states that the software should **store the CRM_CMD 0x9 Parameter Data to RAM** but adds that this data is stored **until** the reception of **CRM_CMD 0xA** with the specific **MemoryOperation == 0x2**. This action differs from the system requirement's instruction to **cache** the data immediately, as **SWRS 011** defers the actual memory operation until a further command is received.

Outcome:

- **SWRS 011** partially fulfills the action described in **SYSRS 010** by handling the data, but it adds a further delay by waiting for **CRM_CMD 0xA**, which is not mentioned in the system requirement. Additionally, **SWRS 011** does not address the action from **SYSRS 011** regarding the increment of **StartAddress**.

Parameter Sources Check:

- **SYSRS 010** and **SYSRS 011** mention the parameters **CRM_CMD 0x9 Parameter Data** and **CRM_CMD 0x8 Parameter StartAddress**, respectively.

SWRS 011 correctly addresses **CRM_CMD 0x9 Parameter Data** but omits any reference to **CRM_CMD 0x8 Parameter StartAddress**. Moreover, **SWRS 011** introduces a new parameter, **CRM_CMD 0xA**, which is not mentioned in the system requirements.

Outcome:

- **SWRS 011** correctly handles the **CRM_CMD 0x9 Parameter Data** but fails to address **CRM_CMD 0x8 Parameter StartAddress**. Additionally, the introduction of **CRM_CMD 0xA** and its associated parameters is not aligned with the system requirements.

Additional Conditions Check:

- **SWRS 011** introduces additional conditions involving **CRM_CMD 0xA** and waiting for a memory operation before proceeding with the storage of data. These conditions are not mentioned in either **SYSRS 010** or **SYSRS 011** and are not directly relevant to the system requirements as described.

Outcome:

- The additional condition involving **CRM_CMD 0xA** in **SWRS 011** is not relevant to the system requirements, and it adds complexity that is not mentioned or required by **SYSRS 010** or **SYSRS 011**.

Rating

Meaningfulness Rating: Low

- **Reason:** While **SWRS 011** handles the storage of **CRM_CMD 0x9 Parameter Data**, it introduces an unrelated condition involving **CRM_CMD 0xA** and omits important details such as the caching and the increment of **StartAddress**. This misalignment reduces the meaningfulness of the link between the system and software requirements.

Completeness Rating: Low

- **Reason:** **SWRS 011** covers only part of the system requirements, handling the data storage but failing to address the caching and increment actions. Additionally, the introduction of new conditions and parameters (involving **CRM_CMD 0xA**) detracts from the completeness of the software requirement in relation to the system requirements.

Overall Quality of the Link: Low

- **Reason:** Both meaningfulness and completeness are rated low due to the introduction of unrelated conditions, missing actions (such as caching and incrementing the address), and the overall misalignment between the system and software requirements.

Suggested Changes or Revisions:

1. **Remove the Reference to CRM_CMD 0xA:** The software requirement should be revised to focus solely on **CRM_CMD 0x9**, as the additional condition involving **CRM_CMD 0xA** is not relevant to the system requirements.
2. **Ensure Data Caching:** The software requirement should explicitly state that the **CRM_CMD 0x9 Parameter Data** should be **cached** to the Data Cache, as specified in **SYSRS 010**.
3. **Include the Address Increment:** The software requirement should incorporate the action of incrementing **CRM_CMD 0x8 Parameter StartAddress** by 1 word, as specified in **SYSRS 011**.
4. **Remove Additional Conditions:** Eliminate the conditional behavior related to waiting for **CRM_CMD 0xA** and focus on the immediate actions required by **SYSRS 010** and **SYSRS 011**.



```
"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 System Requirement SYSRS 012 and Linked Software Requirement SWRS 012

Condition Check:

- **SYSRS 012** specifies that **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**, the Sensor IC shall set *Error: err_discPhysAddrMismatch = 0x1* ("error detected").

SWRS 012 includes a similar condition: **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)**, the software shall set *Error: err_discPhysAddrMismatch = 0x1* ("error detected"). The software also introduces an additional condition: **OTHERWISE**, the software shall set *Error: err_discPhysAddrMismatch = 0x0* ("no error").

Outcome:

- The condition in **SWRS 012** generally aligns with **SYSRS 012**, but there is a slight distinction in how the two address the parameter comparison:
 - **SYSRS 012** compares the **effective value** of the parameter with its **default value**, while **SWRS 012** explicitly compares the values in **RAM** and **NVM**.
 - Additionally, **SWRS 012** introduces an **else** condition that is not mentioned in **SYSRS 012** (setting the error to **0** if no mismatch is detected).

Overall, the core condition of detecting a mismatch is aligned, but the detailed comparison between RAM and NVM and the added **else** clause in **SWRS 012** introduce some differences.

Action and Response Check:

- **SYSRS 012** specifies that if the condition is met (values are not equal), the Sensor IC shall **set Error: err_discPhysAddrMismatch = 0x1** ("error detected").

SWRS 012 mirrors this action but adds an **else** clause: if the values are equal, the software shall set *Error: err_discPhysAddrMismatch = 0x0* ("no error").

Outcome:

- The core action of setting the error flag to **0x1** when there is a mismatch is consistent between **SYSRS 012** and **SWRS 012**. However, **SWRS 012** includes the additional action of setting the error flag to **0x0** if no mismatch is detected, which is not specified in the system requirement.

Parameter Sources Check:

- **SYSRS 012** refers to the parameter **DSI3_physicalAddress** but does not specify the source (RAM or NVM), while **SWRS 012** explicitly compares the **RAM** and **NVM** values of the **DSI3_physicalAddress** parameter.

Outcome:

- **SWRS 012** adds more detail by specifying that the comparison is between the values in **RAM** and **NVM**, while **SYSRS 012** is less explicit about the source of the parameter values. The extra detail in **SWRS 012** is useful, but it does introduce a slight deviation from the abstraction level of the system requirement.

Additional Conditions Check:

- **SWRS 012** introduces an additional condition: if no mismatch is detected, the software shall set *Error: err_discPhysAddrMismatch* = 0x0 ("no error"). This condition is not mentioned in **SYSRS 012**, which only specifies what to do if a mismatch occurs.

Outcome:

- The introduction of the **else** condition (setting the error flag to **0x0**) is additional and not required by the system requirement. This could be seen as an enhancement, but it does add functionality not explicitly covered in **SYSRS 012**.

Rating

Meaningfulness Rating: Medium

- **Reason:** The software requirement generally supports the system requirement but introduces more detail (comparing values in RAM and NVM) and adds an **else** clause (setting the error flag to **0**), which is not present in the system requirement. This makes the link somewhat aligned but with additional complexity.

Completeness Rating: Medium

- **Reason:** While the software requirement addresses the core action of detecting a mismatch and setting the error flag, it adds extra conditions and details that are not in the system requirement, leaving some ambiguity about whether these additional actions are necessary. The comparison between RAM and NVM is not specified in **SYSRS 012**, and the **else** clause introduces functionality beyond what is required.

Overall Quality of the Link: Medium

- **Reason:** The software requirement supports the system requirement but adds additional conditions and actions that are not explicitly required. While the core functionality is preserved, the extra detail creates some divergence, making the overall quality medium.

Suggested Changes or Revisions:

1. **Clarify Parameter Comparison:** The software requirement should clarify whether comparing the **RAM** and **NVM** values of **DSI3_physicalAddress** is explicitly necessary based on system-level requirements. If not, this comparison could be simplified to match the system requirement.

2. **Remove the Else Condition (if unnecessary):** If setting *Error: err_discPhysAddrMismatch = 0x0* ("no error") is not explicitly required by the system design, it should be removed to align better with **SYSRS 012**. If it is required, it should be included in the system requirement for consistency.
3. **Ensure Consistency in Parameter Sources:** Ensure that the parameter sources (RAM, NVM) are aligned with the intended abstraction level of the system requirement. If **SYSRS 012** needs more explicit detail about where the values are sourced, this should be updated accordingly.



"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 System Requirement SYSRS 014 and Linked Software Requirement SWRS 016

Condition Check:

- **SYSRS 014** specifies that **IF the Event_MAX is generated**, the Sensor IC shall **measure the 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 DSI3 packet.

SWRS 016 mirrors this condition by stating that 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.

Outcome:

- The condition regarding the occurrence of **Event_MAX** and the relative measurement of time is consistently present in both **SYSRS 014** and **SWRS 016**. The only difference is the terminology for the transmission medium: **SYSRS 014** refers to a **DSI3 packet**, while **SWRS 016** refers to a **PDCM frame**. Depending on whether these terms are interchangeable or refer to different transmission protocols, this may or may not be an issue.

Action and Response Check:

- **SYSRS 014** specifies that the Sensor IC shall measure the event time and provide the *EventTimeStamp<i>* for the event, transmitted in a DSI3 packet.

SWRS 016 states that the software shall send the MAX events, measure the event time, and provide the *EventTimeStamp<i>* in a PDCM frame.

Outcome:

- Both the system and software requirements describe the same action: measuring the event time and transmitting the *EventTimeStamp<i>* with the event. The only discrepancy is the difference in transmission terminology (DSI3 packet vs. PDCM frame).

Parameter Sources Check:

- **SYSRS 014** specifies that the parameter *PDCM_RSP P12, P13 Parameter EventTimeStamp<i>* must be provided for the <i>th event in a DSI3 packet.

SWRS 016 correctly refers to the same parameter, *PDCM_RSP P12, P13 Parameter EventTimeStamp<i>*, but notes that it will be provided in a PDCM frame.

Outcome:

- The parameter *PDCM_RSP P12, P13 EventTimeStamp<i>* is referenced correctly in both the system and software requirements, but the medium (DSI3 packet vs. PDCM frame) differs. If these terms refer to different transmission mechanisms, this needs to be clarified.

Additional Conditions Check:

- **SWRS 016** does not introduce any additional conditions or actions beyond what is specified in **SYSRS 014**.

Outcome:

- No additional or extraneous conditions are introduced, ensuring that the software requirement remains relevant and focused on the task described in the system requirement.

Rating

Meaningfulness Rating: Medium

- **Reason:** While the software requirement supports the system requirement by ensuring the time measurement and transmission of the event timestamp, the difference in transmission terminology (DSI3 packet vs. PDCM frame) introduces a potential mismatch. If these terms are interchangeable, the link is strong; if not, this is a significant issue.

Completeness Rating: Medium

- **Reason:** The software requirement captures the core action and parameters of the system requirement but fails to clarify whether the transmission medium (PDCM frame vs. DSI3 packet) is consistent with the system requirement. This could lead to incomplete alignment if these terms are not interchangeable.

Overall Quality of the Link: Medium

- **Reason:** Both meaningfulness and completeness are rated medium. While the core functionality (event timing and transmission) is captured, the potential difference in transmission protocol terminology raises questions about the alignment of the requirements.

Suggested Changes or Revisions:

1. **Clarify Transmission Medium:** Ensure consistency between the system and software requirements regarding the transmission medium. If **DSI3 packet** and **PDCM frame** refer to the same thing, this should be made explicit to avoid confusion. If they are different, the software requirement should be revised to match the system requirement.
2. **Verify Terminology:** Double-check the use of terms like **DSI3 packet** and **PDCM frame** in both requirements to ensure that they refer to the correct technical specifications and align with the system's intended architecture.



"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 System Requirement SYSRS 015 and Linked Software Requirement SWRS 017

Condition Check:

- **SYSRS 015** specifies the condition: **AFTER Sending PDCM Packet After Reception of the BRC**, the Sensor IC shall increment the *PDCM_RSP Parameter KAC* by 1, with the added detail that the counter is reset to 0 at overflow.

SWRS 017 mirrors this condition: **AFTER Sending PDCM Packet After Reception of the BRC**, the software shall increment the *PDCM_RSP Parameter KAC* by 1.

Outcome:

- Both requirements specify the same condition—after sending the PDCM packet following the reception of the BRC, the **KAC** parameter is incremented. The condition is fully aligned between the system and software requirements.

Action and Response Check:

- **SYRSR 015** specifies that the Sensor IC shall increment the *PDCM_RSP Parameter KAC* by 1, and **reset the counter to 0 at overflow**.

SWRS 017 specifies that the software shall increment the *PDCM_RSP Parameter KAC* by 1, but it **omits the instruction to reset the counter to 0 at overflow**.

Outcome:

- The software requirement captures the primary action of incrementing the **KAC** parameter but **does not include the critical detail** about resetting the counter at overflow, as specified in the system requirement. This is an important omission because it impacts how the counter behaves when it reaches its maximum value.

Parameter Sources Check:

- Both **SYRSR 015** and **SWRS 017** refer to the **PDCM_RSP Parameter KAC**, which is correctly handled in both cases.

Outcome:

- The parameter source is correctly referenced and consistent between the system and software requirements.

Additional Conditions Check:

- **SWRS 017** does not introduce any additional conditions not mentioned in the system requirement. It remains focused on the task described in **SYRSR 015**.

Outcome:

- No additional conditions are introduced, and the software requirement remains aligned with the system requirement in terms of context.

Rating

Meaningfulness Rating: Medium

- **Reason:** While the software requirement supports the system requirement by handling the increment of the **KAC** parameter, it omits the critical detail about **resetting the counter at overflow**, which is an important aspect of the system requirement's functionality.

Completeness Rating: Medium

- **Reason:** The software requirement covers the increment of the **KAC** parameter but lacks the crucial instruction to reset the counter to 0 when it overflows, making it incomplete in terms of fully implementing the system requirement.

Overall Quality of the Link: Medium

- **Reason:** Both meaningfulness and completeness are rated as medium due to the omission of the overflow handling. While the core functionality (incrementing **KAC**) is addressed, the lack of handling for overflow leaves a gap in the overall implementation.

Suggested Changes or Revisions:

1. **Include Overflow Handling:** The software requirement should be revised to include the instruction to **reset the KAC counter to 0 at overflow**, as specified in the system requirement. This will ensure full alignment and proper behavior of the counter.
2. **Clarify Counter Behavior:** If the overflow condition is handled elsewhere in the system or software, this should be clarified in the software requirement to ensure that the behavior is fully documented and traceable.



"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 System Requirement SYSRS 016 and Linked Software Requirement SWRS 018

Condition Check:

- **SYSRS 016** specifies the condition: **After Reception of the Valid CRM_FCC_CMD**, the Sensor IC shall set the *PDCM_RSP Parameter KAC = 0x0*.

SWRS 018 introduces the same condition (after receiving a valid CRM_FCC_CMD) but specifies that the software shall set the *PDCM_RSP Parameter KAC = 0x3*.

Outcome:

- There is a clear discrepancy between **SYSRS 016** and **SWRS 018** regarding the value to which the **KAC** parameter should be set. **SYSRS 016** specifies that the **KAC** parameter should be set to **0x0**, while **SWRS 018** sets it to **0x3**. This constitutes a major misalignment, as the conditions are the same, but the actions diverge.

Action and Response Check:

- **SYSRS 016** specifies that the Sensor IC should set the *PDCM_RSP Parameter KAC* = 0x0 after receiving the valid **CRM_FCC_CMD**.

SWRS 018 specifies that the software should set the *PDCM_RSP Parameter KAC* = 0x3 after receiving the same command.

Outcome:

- The actions differ significantly between the system and software requirements. The system requirement calls for resetting **KAC** to **0x0**, while the software sets it to **0x3**, creating a functional mismatch.

Parameter Sources Check:

- Both **SYSRS 016** and **SWRS 018** refer to the **PDCM_RSP Parameter KAC**, which is correctly handled in terms of reference. However, the issue lies in the value that this parameter is being set to (0x0 vs. 0x3).

Outcome:

- The parameter source is referenced correctly, but the discrepancy in the values leads to a misalignment in functionality.

Additional Conditions Check:

- **SWRS 018** introduces another condition for incrementing **KAC** after sending a PDCM packet and receiving a **BRC**, which is not relevant to the specific behavior described in **SYSRS 016**.

Outcome:

- The additional condition in **SWRS 018** (incrementing **KAC** after BRC) is unrelated to the condition specified in **SYSRS 016**, which only focuses on the behavior after receiving **CRM_FCC_CMD**. This extra condition introduces unnecessary complexity to the specific action described in the system requirement.

Rating

Meaningfulness Rating: Low

- **Reason:** The software requirement diverges from the system requirement by setting the **KAC** parameter to **0x3** instead of **0x0**, which contradicts the system's intended functionality. Additionally, the introduction of unrelated conditions for incrementing **KAC** further reduces the meaningfulness of the link.

Completeness Rating: Low

- **Reason:** The software requirement does not fully implement the behavior described in the system requirement. The value set for **KAC** is incorrect, and additional unrelated conditions are introduced, leaving the software requirement incomplete in terms of fulfilling the system requirement.

Overall Quality of the Link: Low

- **Reason:** Both meaningfulness and completeness are rated as low due to the misalignment of values and the introduction of unrelated conditions. The software requirement does not correctly implement the system requirement, leading to a poor overall link.

Suggested Changes or Revisions:

1. **Correct the KAC Value:** The software requirement should be revised to set the **PDCM_RSP Parameter KAC = 0x0** after the reception of a valid **CRM_FCC_CMD**, aligning with **SYSRS 016**.
2. **Remove Unrelated Conditions:** The condition related to incrementing **KAC** after sending a PDCM packet and receiving a **BRC** should be removed if it is unrelated to the behavior specified in **SYSRS 016**. If this condition is necessary for a different system requirement, it should be documented separately.
3. **Ensure Consistency Across Requirements:** Any discrepancies in setting values for critical parameters like **KAC** should be resolved to ensure consistent functionality between the system and software requirements.



"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 System Requirement SYSRS 017 and Linked Software Requirement SWRS 019

Condition Check:

- **SYSRS 017** specifies a series of conditions: **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 simplifies the conditions to: **IF in Main-Measurement AND IF Break Burst Generation occurs**, then the software shall set *Error: err_burstGenerationStopped = 0x1 ("error")*.

Outcome:

- There is a substantial difference between the conditions in the system and software requirements:
 - **SYSRS 017** refers to **Standard Path Measurement** and specifies that the measurement is **stopped or blocked** before the **US-Transducer Control (Non-Modulated Sine Burst)** is fully performed.
 - **SWRS 019** only refers to **Main-Measurement** and checks for a **Break Burst Generation** condition. It omits the specifics of stopping an ongoing measurement or verifying the status of the US-Transducer Control functionality, which are crucial in **SYSRS 017**.

Action and Response Check:

- **SYSRS 017** specifies that, under the given conditions, the Sensor IC shall set *Error: err_burstGenerationStopped = 0x1* ("error").

SWRS 019 also specifies that the software shall set *Error: err_burstGenerationStopped = 0x1* under its conditions.

Outcome:

- While both the system and software requirements agree on the action (setting the error flag to **0x1**), the conditions under which this action occurs are different. **SWRS 019** does not fully reflect the specific conditions in **SYSRS 017**, meaning the action is taken under different circumstances in each case.

Parameter Sources Check:

- Both **SYSRS 017** and **SWRS 019** refer to the error parameter **err_burstGenerationStopped**, which is correctly referenced in both cases.

Outcome:

- The parameter source is correctly referenced in both requirements, and there is no issue with how the error flag is set.

Additional Conditions Check:

- **SWRS 019** introduces a new condition related to **Break Burst Generation** without referencing the more detailed checks for **Stopped Ongoing Measurement** and **US-Transducer Control** from **SYSRS 017**.

Outcome:

- The software requirement simplifies the conditions and omits critical details from the system requirement, which results in the introduction of unrelated or insufficient conditions that do not fully meet the system requirement.

Rating

Meaningfulness Rating: Low

- **Reason:** The software requirement omits several critical conditions mentioned in the system requirement, such as the **Stop Ongoing Measurement (Blocked)** and the status of the **US-**

Transducer Control. These conditions are vital to the correct functioning of the system, and their omission weakens the meaningfulness of the link between the system and software requirements.

Completeness Rating: Low

- **Reason:** The software requirement is incomplete because it does not account for the specific and detailed conditions mentioned in the system requirement. The simplification of the conditions means that the software may set the error under inappropriate circumstances or miss the correct situations where the error should be set.

Overall Quality of the Link: Low

- **Reason:** Both meaningfulness and completeness are rated as low. The software requirement fails to capture the critical aspects of the system requirement and introduces simplified conditions that do not align with the intended functionality.

Suggested Changes or Revisions:

1. **Align Conditions:** The software requirement should be revised to include the detailed conditions from **SYSRS 017**:
 - It should explicitly check if the measurement is **stopped or blocked** and verify whether the **US-Transducer Control (Non-Modulated Sine Burst)** has been fully performed.
 - The reference to **Break Burst Generation** should be either aligned with the system requirement or explained if it's an equivalent condition.
2. **Ensure Consistency:** The condition of **stopping an ongoing measurement** and checking the state of the **US-Transducer Control** should be directly linked to setting the error flag, as described in the system requirement. This ensures that the software reflects the system's functionality accurately.



"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 System Requirement SYSRS 018 and Linked Software Requirement SWRS 020

Condition Check:

- **SYSRS 018** specifies the following condition: **If Performing Memory Access AND If the Key is NOT set**, then the Sensor IC shall set the state of the memory area to **Unlocked**.

SWRS 020 describes a different condition: **If Performing Memory Access AND If the Key is set**, then the software shall set the state of the memory area to **Locked**.

Outcome:

- The conditions between **SYSRS 018** and **SWRS 020** are fundamentally different:
 - **SYSRS 018** specifies the condition for when **no key is set**, which results in the memory area being set to **Unlocked**.
 - **SWRS 020** describes the opposite scenario, where **the key is set**, leading to the memory area being **Locked**.

The two conditions are contradictory, as **SYSRS 018** deals with the scenario when no key is set, while **SWRS 020** describes what happens when the key is set. They do not align in terms of logic or operation.

Action and Response Check:

- **SYSRS 018** specifies that when the key is not set, the Sensor IC shall set the memory area state to **Unlocked**.

SWRS 020 specifies that when the key is set, the software shall set the memory area state to **Locked**.

Outcome:

- The actions differ significantly: **SYSRS 018** deals with setting the state to **Unlocked**, while **SWRS 020** deals with setting the state to **Locked**. These are opposite actions, with no overlap or alignment in their outcomes.

Parameter Sources Check:

- Both **SYSRS 018** and **SWRS 020** correctly reference the parameter **CRM_CMD 0x8 Parameter MemoryArea** for identifying the memory area.

Outcome:

- The parameter source is correctly referenced in both cases, but the actions and conditions tied to this parameter are contradictory.

Additional Conditions Check:

- **SWRS 020** introduces a condition about the key being set and the memory area being locked, which is not relevant to the condition outlined in **SYSRS 018** (where the key is not set and the memory area is unlocked).

Outcome:

- **SWRS 020** addresses a different condition and action from **SYSRS 018**, introducing an unrelated scenario (memory being locked when the key is set). This condition is unrelated to the system requirement and causes a significant misalignment.

Rating

Meaningfulness Rating: Low

- **Reason:** The software requirement describes a completely different scenario (setting the memory area to **Locked** when the key is set) from the system requirement (setting the memory area to **Unlocked** when the key is not set). The link between the two requirements is weak and illogical due to the opposing conditions and actions.

Completeness Rating: Low

- **Reason:** The software requirement does not cover the specific case outlined in **SYSRS 018**. Instead of handling the scenario where the key is not set and the memory area should be unlocked, it handles a different case. This makes the software requirement incomplete in terms of implementing the system requirement.

Overall Quality of the Link: Low

- **Reason:** Both meaningfulness and completeness are rated low due to the contradictory nature of the conditions and actions. The software requirement addresses a different scenario from the system requirement, resulting in a poor overall link.

Suggested Changes or Revisions:

1. **Align the Conditions:** The software requirement should be revised to handle the scenario outlined in **SYSRS 018**—when the key is **not set**, the memory area should be set to **Unlocked**. This would bring the software requirement in line with the system requirement.
2. **Create Separate Requirements if Necessary:** If the scenario where the key is set and the memory area should be locked is also required, it should be documented as a separate system and software requirement pair. This will avoid confusion and ensure each scenario is handled appropriately.
3. **Ensure Consistency:** The software requirement should be checked for consistency with the system requirement, ensuring that the actions (setting **Unlocked** vs. **Locked**) and conditions (key set vs. key not set) are correctly implemented.



"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 System Requirement SYSRS 019 and Linked Software Requirement SWRS 021

Condition Check:

- **SYSRS 019** specifies the condition: **If Performing Memory Access in Unicast AND AFTER reception of the CRM_FCC_CMD AND If Addressed with the Broadcast Command**, the Sensor IC shall set the parameter *Status Value* = 0x2 ("invalid").

SWRS 021 mirrors this condition exactly: **If Performing Memory Access in Unicast AND AFTER reception of the CRM_FCC_CMD AND If Addressed with the Broadcast Command**, the software shall set the parameter *Status Value* = 0x2 ("invalid").

Outcome:

- Both the system and software requirements share the same conditions: memory access in unicast, reception of the **CRM_FCC_CMD**, and being addressed by a broadcast command. There is no discrepancy in the conditions.

Action and Response Check:

- **SYSRS 019** specifies that under these conditions, the Sensor IC shall set the *Status Value* to **0x2** ("invalid").

SWRS 021 mirrors this action, specifying that the software shall also set the *Status Value* to **0x2** ("invalid").

Outcome:

- The actions are identical in both the system and software requirements, with the same response (setting the *Status Value* to **0x2**) being triggered by the same conditions.

Parameter Sources Check:

- Both **SYSRS 019** and **SWRS 021** refer to the parameter *Status Value* and specify that it should be set to **0x2** ("invalid").

Outcome:

- The parameter is correctly referenced in both the system and software requirements, and there are no issues with parameter handling.

Additional Conditions Check:

- **SWRS 021** does not introduce any additional or irrelevant conditions. It stays aligned with the condition specified in **SYSRS 019**.

Outcome:

- There are no additional conditions introduced in the software requirement, ensuring full relevance to the system requirement.

Rating

Meaningfulness Rating: High

- **Reason:** The software requirement directly supports the system requirement by mirroring the conditions and actions exactly. The link between the two is strong and logical, with no discrepancies.

Completeness Rating: High

- **Reason:** The software requirement fully covers all aspects of the system requirement, including the conditions and the specific action of setting the *Status Value* to **0x2**. No critical information is missing, and the requirement is fully implemented.

Overall Quality of the Link: High

- **Reason:** Both meaningfulness and completeness are rated high. The software requirement is perfectly aligned with the system requirement, making the overall quality of the link high.

Suggested Changes or Revisions:

- **No revisions needed:** The software requirement fully matches the system requirement, with all conditions and actions correctly implemented.



"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 System Requirement SYSRS 020 and Linked Software Requirement SWRS 022

Condition Check:

- **SYSRS 020** specifies the condition: **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 *Status Value* = 0x2 ("invalid").

SWRS 022, on the other hand, specifies the condition: **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 *Status Value* = 0x2 ("invalid").

Outcome:

- There is a significant difference between the conditions:
 - **SYSRS 020** deals with the scenario where the memory access is performed in **Broadcast** mode but the command is addressed with a **Unicast** command.
 - **SWRS 022** instead handles the situation where both the memory access and the command are in **Broadcast** mode.

These are opposite conditions. The system requirement addresses a mismatch between Broadcast and Unicast commands, while the software requirement handles Broadcast-to-Broadcast scenarios, creating a misalignment between the two.

Action and Response Check:

- **SYSRS 020** specifies that under the given conditions, the Sensor IC shall set the *Status Value* to **0x2** ("invalid").

SWRS 022 also specifies that the software shall set the *Status Value* to **0x2** ("invalid") under its conditions.

Outcome:

- The action of setting the *Status Value* to **0x2** is consistent in both the system and software requirements. However, since the conditions under which this action occurs differ significantly, the alignment between the action and its trigger is incorrect.

Parameter Sources Check:

- Both **SYSRS 020** and **SWRS 022** reference the parameter *Status Value* and specify that it should be set to **0x2** ("invalid").

Outcome:

- The parameter is correctly referenced in both the system and software requirements, with no discrepancies in handling the parameter itself.

Additional Conditions Check:

- **SWRS 022** introduces a different condition—addressing the command in Broadcast mode—whereas **SYSRS 020** specifically mentions addressing the command in **Unicast** mode. This difference leads to a mismatch in the conditions between the system and software requirements.

Outcome:

- The condition in **SWRS 022** is not aligned with **SYSRS 020**, creating a significant deviation from the intended system behavior.

Rating

Meaningfulness Rating: Low

- **Reason:** The software requirement addresses a different condition (Broadcast addressing Broadcast) from the system requirement (Broadcast addressing Unicast). This misalignment makes the link between the two requirements weak and illogical.

Completeness Rating: Low

- **Reason:** The software requirement does not cover the specific case outlined in **SYSRS 020**, where the system must handle the condition of Broadcast memory access being addressed with a Unicast command. This results in an incomplete implementation of the system requirement.

Overall Quality of the Link: Low

- **Reason:** Both meaningfulness and completeness are rated low because the software requirement deals with an entirely different scenario from the system requirement. The conditions under which the action occurs are inconsistent, resulting in a poor overall link.

Suggested Changes or Revisions:

1. **Align the Conditions:** The software requirement should be revised to reflect the condition outlined in **SYSRS 020**—where the system performs **Memory Access in Broadcast** mode but receives a **Unicast** command. This would align the software requirement with the system requirement.
2. **Remove or Separate the Broadcast Condition:** If handling Broadcast-to-Broadcast addressing is necessary, this should be documented as a separate system and software requirement pair to avoid confusion and ensure that each scenario is handled correctly.
3. **Ensure Consistency in Conditions:** The conditions (Broadcast addressing Unicast) should be implemented consistently across both the system and software requirements to ensure correct functionality.



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 System Requirement SYSRS 021 and Linked Software Requirements SWRS 023 and SWRS 024

Condition Check:

- **SYSRS 021** specifies that the firmware shall allow configurability to select retry for **open and short faults independently**.

SWRS 023 introduces the condition: **IF the software detects an open or short error AND retry is configured as enabled** in the **DEFAULT OPEN SHORT RETRY** value, the affected channel shall be re-enabled for one timeslice at least every 200 milliseconds.

SWRS 024 mentions the configuration item: **DEFAULT OPEN SHORT RETRY**.

Outcome:

- **SYSRS 021** focuses on independent configurability for retrying **open** and **short** faults, while **SWRS 023** and **SWRS 024** collectively focus on retrying both open and short errors based on a **default configuration** without specifically mentioning the ability to configure retries for open and short faults independently.
- **SWRS 023** includes a retry mechanism for open or short errors, but it does not explicitly describe whether open and short retries are configurable independently, as required by **SYSRS 021**.

Action and Response Check:

- **SYSRS 021** describes the ability to **configure retries independently** for open and short faults.

SWRS 023 specifies that if an open or short error is detected and retry is enabled, the software shall re-enable the affected channel for a timeslice every 200 milliseconds. However, there is no mention of independent configurability for open and short faults, which is a key action described in **SYSRS 021**.

Outcome:

- **SWRS 023** outlines the action for retrying open and short faults, but it does not address the configurability to handle open and short faults independently, which is a critical aspect of the system requirement.

Parameter Sources Check:

- **SYSRS 021** implies that there should be a configuration for independently selecting retry behavior for open and short faults.

SWRS 023 and **SWRS 024** mention **DEFAULT OPEN SHORT RETRY** as a configuration parameter, but there is no indication that this parameter supports independent retry configuration for open and short faults.

Outcome:

- While the software requirements reference a **DEFAULT OPEN SHORT RETRY** configuration, they do not specify if or how this parameter allows independent retry configuration for open and short faults, leaving an important detail unaddressed.

Additional Conditions Check:

- **SWRS 023** introduces the condition that retries occur **every 200 milliseconds**, but **SYSRS 021** does not specify a retry interval or timing, meaning this is an additional condition that could add complexity or deviate from the system requirement's flexibility.

Outcome:

- The addition of the 200-millisecond retry interval in **SWRS 023** is an extra condition not explicitly mentioned in **SYSRS 021**. While this might be acceptable, the system requirement does not specify any timing, leaving room for flexibility that is constrained by this extra condition.

Rating

Meaningfulness Rating: Medium

- **Reason:** The software requirements address the retry mechanism for open and short faults but do not explicitly cover the independent configurability of retries for each fault type, as required by **SYSRS 021**. The retry functionality is implemented but with missing detail about configurability.

Completeness Rating: Medium

- **Reason:** The software requirements capture the retry mechanism but lack the explicit mention of independent configurability for open and short faults. Additionally, the 200-millisecond retry interval adds a condition not specified in the system requirement, leaving the configurability incomplete.

Overall Quality of the Link: Medium

- **Reason:** Both meaningfulness and completeness are rated medium. While the retry functionality is addressed, the critical aspect of independent configurability for open and short faults is not clearly implemented in the software requirements. The additional retry interval adds a layer of detail not mentioned in the system requirement, creating a gap in alignment.

Suggested Changes or Revisions:

1. **Add Independent Configurability for Open and Short Faults:** The software requirements should be updated to explicitly mention that open and short faults can be configured for retry **independently**, as required by **SYSRS 021**.
2. **Clarify the Role of DEFAULT OPEN SHORT RETRY:** If the **DEFAULT OPEN SHORT RETRY** parameter supports independent configurability, this should be clearly stated in the software requirements. Otherwise, the configuration should be revised to allow for separate control of open and short retry behavior.
3. **Review the 200-Millisecond Retry Interval:** If the retry interval is a necessary system feature, it should be included in **SYSRS 021** for consistency. Alternatively, if flexibility is required, the software requirement should be revised to allow configurable retry timing.



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 System Requirement SYSRS 002 and Linked Software Requirement SWRS 002

Condition Check:

- **SYSRS 002** specifies that the **Sensor IC firmware shall have a watchdog** to detect a firmware or hardware malfunction by **resetting the watchdog timer** in an appropriate time within the Fault Tolerant Time Interval (FTTI).

SWRS 002 similarly specifies that the software shall **enable a watchdog** to detect software or hardware malfunction, and **if a malfunction is present**, perform a reset within the FTTI of **20ms**.

Outcome:

- The conditions in both the system and software requirements are generally aligned. Both requirements specify the use of a watchdog to detect malfunctions, but **SWRS 002** explicitly mentions a Fault Tolerant Time Interval (FTTI) of **20ms**, whereas **SYSRS 002** refers to FTTI without specifying the exact time.

The mention of "20ms" in **SWRS 002** adds more detail, which could be acceptable, but the system requirement leaves the exact FTTI open.

Action and Response Check:

- **SYSRS 002** requires the Sensor IC firmware to detect malfunctions and **reset the watchdog timer** in an appropriate time within the FTTI.

SWRS 002 similarly specifies that the software shall enable a watchdog to detect malfunctions and **perform a reset** within the FTTI. The software requirement also clarifies that this reset is triggered if a malfunction is detected.

Outcome:

- The software requirement aligns with the system requirement in terms of action: both involve detecting malfunctions and performing a reset within the FTTI. However, **SWRS 002** is more specific about resetting only when a malfunction is detected, which is implied but not explicitly mentioned in **SYSRS 002**.

Parameter Sources Check:

- **SYSRS 002** refers to the **Fault Tolerant Time Interval (FTTI)** without specifying an exact time, while **SWRS 002** defines the FTTI as **20ms**.

Outcome:

- The software requirement provides additional detail regarding the FTTI duration (20ms). While this level of specificity is not present in the system requirement, it is not contradictory unless the FTTI is expected to be configurable. If **20ms** is fixed, this addition is appropriate.

Additional Conditions Check:

- **SWRS 002** introduces a specific FTTI value of **20ms** but does not add any unrelated or extraneous conditions.

Outcome:

- There are no irrelevant conditions in the software requirement. The only difference is the explicit mention of the **20ms** FTTI, which may be acceptable if the system requirement allows for it.

Rating

Meaningfulness Rating: High

- **Reason:** The software requirement supports the system requirement closely by implementing the watchdog for detecting hardware and software malfunctions and performing a reset within the Fault Tolerant Time Interval (FTTI). The explicit mention of **20ms** provides additional detail without deviating from the system requirement.

Completeness Rating: High

- **Reason:** The software requirement fully implements the functionality described in the system requirement, adding specificity with the **20ms** FTTI but otherwise covering all necessary actions related to the watchdog reset mechanism.

Overall Quality of the Link: High

- **Reason:** Both meaningfulness and completeness are rated as high. The software requirement aligns with the system requirement, with the added detail of the **20ms** FTTI enhancing the implementation without introducing contradictions or unnecessary complexity.

Suggested Changes or Revisions:

- **No major revisions needed**, but if the **20ms** FTTI is intended to be configurable or if the system requirement leaves it intentionally flexible, this should be noted in both the system and software requirements to ensure consistency across all levels of design.



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 System Requirement SYSRS 001 and Linked Software Requirement SWRS 001

Condition Check:

- **SYSRS 001** specifies that the **Sensor IC shall digitize the Transducer Input Signal** with a sampling frequency of (**CALIB_samplingFreq plus YPX_samplingFreqOffset**).

SWRS 001 introduces the 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 plus YPX_samplingFreqOffset**).

Outcome:

- The conditions in **SWRS 001** add two specific events:
 1. **AFTER** the reception of the command **CRM_CMD 0x1**.
 2. **BEFORE** the start of the measurement.

These events are not mentioned in **SYSRS 001**, which only focuses on the task of digitizing the signal with the specified sampling frequency. While the additional conditions in **SWRS 001** provide more detail, they are not contradictory to **SYSRS 001** and can be considered a valid refinement of when the sampling frequency is set.

Action and Response Check:

- **SYSRS 001** specifies that the **Sensor IC shall digitize** the transducer input signal using the specified sampling frequency.

SWRS 001 describes the action of setting the register **BRG.FS.f_s** to the specified sampling frequency (**CALIB_samplingFreq plus YPX_samplingFreqOffset**) before measurement begins, ensuring the correct frequency is set for the signal digitization process.

Outcome:

- The software requirement describes the action of configuring the sampling frequency in the register before measurement, which aligns with the system requirement's goal of digitizing the signal using that frequency. The software requirement's action is a more detailed step in the overall process of achieving the system requirement.

Parameter Sources Check:

- **SYRS 001** specifies the parameters **CALIB_samplingFreq** and **YPX_samplingFreqOffset** for calculating the sampling frequency.

SWRS 001 correctly references the same parameters **CALIB_samplingFreq** and **YPX_samplingFreqOffset**, ensuring consistency in how the frequency is calculated.

Outcome:

- The parameters are correctly referenced and sourced in both the system and software requirements, ensuring that the software implements the same frequency calculation.

Additional Conditions Check:

- **SWRS 001** introduces additional conditions related to the command **CRM_CMD 0x1** and the timing of setting the frequency before measurement starts. These conditions refine when the sampling frequency is configured but do not introduce irrelevant or unrelated conditions.

Outcome:

- The additional conditions in **SWRS 001** (relating to the timing of setting the frequency) are logical refinements and do not conflict with the system requirement.

Rating

Meaningfulness Rating: High

- **Reason:** The software requirement supports the system requirement by providing more specific details on when and how the sampling frequency is set. The additional conditions of waiting for the command and setting the frequency before measurement enhance the implementation without deviating from the core requirement.

Completeness Rating: High

- **Reason:** The software requirement fully implements the task described in the system requirement, including the use of the correct sampling frequency. The additional timing conditions ensure the sampling frequency is set correctly before the signal is digitized.

Overall Quality of the Link: High

- **Reason:** Both meaningfulness and completeness are rated as high. The software requirement aligns with the system requirement, providing additional clarity and refinement without introducing contradictions or unnecessary complexity.

Suggested Changes or Revisions:

- **No revisions needed:** The software requirement fully implements the system requirement, with all conditions and parameters correctly aligned. The added timing details enhance the clarity of when the frequency setting occurs and ensure proper execution.

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

 Unarchive