|  |  |  |
| --- | --- | --- |
|  | **Document Name** | **SRS for Mold Remote Condition Monitoring System** |
| **Based on Template** | **ZiMetrics** |
| **Created By** | **ZiMetrics** |

**Software Requirement Specification for Mold Remote Condition Monitoring System**

**Version – 1.0**

**Copyright Notice**

|  |
| --- |
| This document contains proprietary information of ZiMetrics. No part of this document may be reproduced, stored, copied, or transmitted in any form or by means of electronic, mechanical, photocopying or otherwise, without the express consent of Zimetrics. This document is intended for internal circulation only and not meant for external distribution. |

**Reviewers**

|  |  |
| --- | --- |
| **Designation** | **Name/Title** |
| Practice Head - Embedded Engineering | Anurag Srivastava |
| Principal Practice Architect - Embedded Engineering | Parag Gupta |
| Senior Technical Lead | Manoj Sharma |
| Solution architect | Jeganathan Palanisamy |

**Modification History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Originator** | **Comments (List impact by section)** |
| Version - 0.1 | 18/04/2022 | Nitesh Srivastava | Initial Draft Version |
| Version -1.0 | 30/09/2022 | Priyanka Yadav | Phase 1 requirement incorporated |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

[**1.** **Summary** 4](#_Toc104218060)

[**2.** **Scope** 4](#_Toc104218061)

[**2.1** **Compliance** 4](#_Toc104218062)

[**2.1.1** **Quality Compliance** 4](#_Toc104218063)

[**2.2** **System Requirements** 4](#_Toc104218064)

[**2.2.1** **Hardware Platform Support** 4](#_Toc104218065)

[**2.2.2** **Operating System Support** 4](#_Toc104218066)

[**2.3** **System Overview & Context Diagram** 5](#_Toc104218067)

[**2.4** **Software Requirements** 6](#_Toc104218068)

[**2.5** **Assumptions** 14](#_Toc104218069)

[**2.6** **Dependencies** 14](#_Toc104218070)

[**3.** **Acceptance Criteria** 15](#_Toc104218071)

[**4.** **Validation** 15](#_Toc104218072)

[**4.1** **Validation Strategy** 15](#_Toc104218073)

[**4.2** **Validation criteria** 15](#_Toc104218074)

[**4.3** **Validation Constraints** 15](#_Toc104218075)

[**5.** **References** 15](#_Toc104218076)

[**6.** **Glossary** 16](#_Toc104218077)

# **Summary**

The purpose of this document is to collect, analyse and define high-level requirements and features of the MOLD Remote Condition Monitoring System This document provides the focus for the project team to understand the project’s need, importance and provide guidance for decision-making.

# **Scope**

This document identifies the functionalities of MOLD Remote Condition Monitoring System. The document also highlights the interdependencies between various high-level requirements. This document will be an input for software design and software implementation of MOLD Remote Condition Monitoring System.

# **Compliance**

# **Quality Compliance**

[ZMX-MOLD-NFUNC-0001] MOLD RCM MUST be MISRA C: 2012 compliance.

# **System Requirements**

# **Hardware Platform Support**

[ZMX-MOLD-NFUNC-0002] MOLD RCMS SHALL support modular PLC having CPU module power supply module, analog input module, digital input module and RTD module.

MOLD RCMS SHALL collect mold machine data using the connected temperature sensors, flow meter, contact sensor,vibration sensors.

# **Operating System Support**

[ZMX-MOLD-NFUNC-0003] MOLD RCMS MUST be developed to support RTOS Linux with pfc-firmware-sdk-G2 as supported firmware,must support MQTT library, OPEN SSL library, AWS library

# **System Overview & Context Diagram**

The broad level architecture of the MOLD is shown in the diagram below:

|  |
| --- |
| MOLD Machine  sensors  **Wago PFC**  **Input**  **Module**  +69+69  Battery backup |

AWS Server

Figure 1: Mold RCMS Architecture

The brief description about each of the components is highlighted below:

* Mold Machine – Machine where actually molding process happens.
* Wago PFC – It is the hardware PLC controller used for moldRCM.
* Device – Device refers to the wago PFC controller which hosts client firmware.
* Server – Server refers to amazon web server to which device communicates with.
* Input module – Hardware interface through which sensor is connected to controller.
* Battery backup – Power backup system in case there is main power failure.
* Sensors – hardware device that are installed on mold machine which detects its various parameters.

# **Description of terms used in document**

* Molding Cycle – Process from close to open to close is defined as molding cycle.
* Cycle time – Time taken for mold to complete one molding cycle.
* Face Plate – Two faces of molding tool in which cavities are made.
* Face plate alignment – Offset measurement between edges of two face plate of mold tool, this offset must be as low as possible and ideally zero.
* Cavities – Groove on mold tool face plate which is shaped as per object to be produced.
* Blocked cavities – Condition where the cavities are not able to prepare the end product.
* Historical counter – defines the lifetime of wago PFC.
* Cumulative counter – Defines the lifetime of mold.
* Dry cycle event - Cycle time is observed as ½ Standard Cycle time or less.
* RCR – Refers to Remote condition record.
* Reset – Reset functionality which occurs only when reset command is sent by the server to the device, following functionality should be performed when reset command is received by the device
* All saved certificates must be deleted.
* All stored data and logs must be deleted.
* Cumulative counters must be set to 0, historical counter must not be reset.

# **Software Requirements**

**Provisioning and certificates**

1. There **SHALL** be unique Device ID assigned for each device.
2. Device unique ID **SHALL** be sent to AWS server during provisioning.
3. Device **SHALL** be able to get provisioned only once with AWS provided certificates.
4. After successful provisioning, Device **SHALL** retain the provision status.
5. Device **SHALL** be able to connect to the AWS server with the configured certificates.
6. Device **SHALL** persist AWS certificates in the persistent memory.

**Condition record**

1. Device **SHALL** be able to count the molding cycles performed on the mold tool.
2. Device **SHALL** be able to detect the cycle time defined in milliseconds.
3. Device **SHALL** be able to send the warning alerts set to -+2% of the deviation in cycle time.
4. Device **SHALL** be able to alarm alert triggers set to +-5% of the deviation in cycle time.
5. Device **SHALL** be able to capture and count the downtime events if an interval of 2x Standard Cycle time is observed which will be used in other analytics formulation.
6. Device **SHALL** be able to ignore the downtime event if Dry cycle is observed as no product will be generated. In this case Dry cycle events will also be captured and used in other analytics.
7. Device **SHALL** persist the cumulative count of molding cycles in persistent memory.
8. Device **SHALL** be able to send the cumulative count to the AWS server as part of periodic telemetry message (PTM) or real-time telemetry message on the pre-configured topic.
9. Device **SHALL** calculate the average cycle time of up to last 100 molding cycles.
10. Device **SHALL** persist the average cycle time of up to last 100 molding cycles in persistent memory until it is successfully sent to the server.
11. Device **SHALL** be able to send the average cycle time of up to last 100 molding cycles on AWS server as part of PTM on the pre-configured topic.
12. Device **SHALL** calculate the average cycle time of up to last 1000 molding cycles.
13. Device **SHALL** persist the average cycle time of up to last 1000 molding cycles in persistent memory until it is successfully sent to the server.
14. Device **SHALL** be able to send the average cycle time of up to last 1000 molding cycles on AWS server as part of PTM on the pre-configured topic.
15. Device **SHALL** calculate the average cycle time of up to last 10000 molding cycles.
16. Device **SHALL** persist the average cycle time of up to last 10000 molding cycles in persistent memory until it is successfully sent to the server.
17. Device **SHALL** be able to send the average cycle time of up to last 10000 molding cycles on AWS server as part of PTM on the pre-configured topic.
18. Device **SHALL** support up to 4 lines for monitoring the line temperature of cooling manifold on the mold tool.
19. Device **SHALL** be able to detect the temperature at the entry point of the configured line of the cooling manifold on the mold tool.
20. Device **SHALL** persist the temperature at the entry point of the configured line of the cooling manifold until it is successfully sent to the server.
21. Device **SHALL** be able to send the temperature at the entry point of the configured line of the cooling manifold to the AWS server as part of PTM on the pre-configured topic.
22. Device **SHALL** be able to detect the temperature at the exit point of the configured line of the cooling manifold on the mold tool.
23. Device **SHALL** persist the temperature at the exit point of the configured line of the cooling manifold until it is successfully sent to the server.
24. Device **SHALL** be able to send the temperature at the exit point of the configured line of the cooling manifold to the AWS server as part of PTM on the pre-configured topic.
25. Device **SHALL** support up-to 4 lines for monitoring the lines flow rate of cooling manifold on mold tool.
26. Device **SHALL** be able to detect the flow rate at the entry point of the configured line of the cooling manifold on the mold tool.
27. Device **SHALL** persist the flow rate at the entry point of the configured line of cooling manifold until it is successfully sent to the server.
28. Device **SHALL** be able to send the flow rate at the entry point of the configured line of the cooling manifold to the AWS server as part of PTM on the pre-configured topic.
29. Device **SHALL** be able to detect the flow rate at the exit point of the configured line of the cooling manifold on the mold tool.
30. Device **SHALL** persist the flow rate at the exit point of the configured line of cooling manifold until it is successfully sent to the server.
31. Device **SHALL** be able to send the flowrate at exit point of the configured line of the cooling manifold to the AWS server as part of PTM on the pre-configured topic.
32. Device **SHALL** use PTM for sending RCR to the AWS server.
33. Device **SHALL** be configurable by AWS server for PTM message frequency.
34. Device **SHALL** persist the latest periodic telemetry message frequency.

**Periodic inspection message**

1. Device **SHALL** be able to execute mold's face-plate alignment precision test periodically.
2. Device **SHALL** persist the face-plate alignment precision test result until it is successfully sent to the server.
3. Device **SHALL** be able to send the face-plate alignment precision test result to the AWS server as periodic inspection message (PIM) with timestamp on the pre-configured topic.
4. Device **SHALL** be able to execute blocked Cavity test periodically.
5. Device **SHALL** persist the blocked Cavity test result until it is successfully sent to the server.
6. Device **SHALL** be able to send the blocked Cavity test result to the AWS server as PIM with timestamp on the pre-configured topic.
7. Device **SHALL** be able to execute periodic Rig Vibration Test.
8. Device **SHALL** persist the Rig Vibration Test result until it is sent to the server.
9. Device **SHALL** be able to send the Rig Vibration Test result to the AWS server as PIM with timestamp on the pre-configured topic.
10. Device **SHALL** be configurable by AWS server for PIM message frequency.

**System event message**

1. Device **SHALL** consider main power OFF as a system event ID 3.
2. Device **SHALL** persist the main power OFF event until it is successfully sent to server.
3. Device **SHALL** be able to send main 41\*8885 event as SEM to AWS server with timestamp at real time on pre-configured topic.
4. Device **SHALL** consider battery charge level below 30% capacity as a system event ID 4.
5. Device **SHALL** be able to send battery low event as SEM to AWS server with time stamp at real time on pre-configured topic.
6. Device **SHALL** be able to send battery low event at every 2% level drop below a configured percentage level as SEM to AWS server with timestamp at real time on pre-configured topic.
7. Device **SHALL** persist the updated battery low event until it is successfully sent to server.
8. Device **SHALL** consider any sensor error as a system event ID 6.
9. Device **SHALL** persist the sensor error event until it is successfully sent to server.
10. Device **SHALL** be able to send any Sensor Error event as SEM to AWS server with time stamp at real time on pre-configured topic.
11. Device **SHALL** consider network connection lost as a system event ID 7.
12. Device **SHALL** persist the network connection lost event until it is successfully sent to the server.
13. Device **SHALL** be able to send network connection lost event as SEM to AWS server with timestamp on the pre-configured topic as it reconnects.
14. Device **SHALL** consider mold’s system power ON as system event ID 20.
15. Device **SHALL** persist mold’s system power ON event until it is successfully sent to the server.
16. Device **SHALL** be able to send mold’s system power ON event as SEM to AWS server with the timestamp at real time on the pre-configured topic.
17. Device **SHALL** consider successful System Provisioning as a system event ID 21.
18. Device **SHALL** persist the System Provisioned event until it is successfully sent to server.
19. Device **SHALL** be able to send successfully System Provisioned event as SEM to AWS server with timestamp on pre-configured topic at real time.
20. Device **SHALL** consider mold machine power shutdown as system event ID 22.
21. Device **SHALL** persist the mold machine power shutdown event until it is successfully sent to server.
22. Device **SHALL** be able to send mold machine power shutdown event as SEM to AWS server with timestamp on the pre-configured topic at real-time.
23. Device **SHALL** be able to sync RTC clock with the server.
24. Device **SHALL** consider Real Time Clock (RTC) showing the wrong time as system event ID 24.
25. Device **SHALL** persist RTC error event until it is successfully sent to the server.
26. Device **SHALL** be able to send the RTC error as SEM to the AWS server with the timestamp in Realtime on the pre-configured topic.
27. Device **SHALL** consider the opening of control box cabinet as system tempering event ID 40.
28. Device **SHALL** record the time of occurrence of system tampering.
29. Device **SHALL** persist control box open error event until it is successfully sent to the server.
30. Device **SHALL** be able to send the control box open error to the AWS server as part of system event message at real-time on the pre-configured topic.
31. Device **SHALL** run BIST each time it is powered up or when the command to run the test is sent by the server.
32. Device **SHALL** test if the device can read data from the temperature sensor as a part of BIST.
33. Device **SHALL** test if the device can read data from the contact sensor as a part of BIST.
34. Device **SHALL** test if the device can read data from flow meter as a part of BIST.
35. Device **SHALL** test if the battery is not below critical level as a part of BIST.
36. Device **SHALL** send BIST report to the server as SEM with the timestamp.
37. Device **SHALL** keep a record of the last status of all the sensors and system events that occurred as On Demand Diagnostic message.
38. Device **SHALL** persist the ODM until it is successfully sent to the server.
39. Device **SHALL** send ODM when the server sends the command to the device for the ODM message.
40. Device **SHALL** send ODM as message type 2 for PTM when the server sends the command to the device for ODM message.
41. Device Timestamp sent in PTM record **SHALL** match with the server Time stamp as requested by AWS server.
42. Device **SHALL** be able to make the updates in the configuration files for the time of sending the PTM messages.
43. Device **SHALL** be able to make updates in the frequency for saving the PTM messages.
44. Device **SHALL** be able to update the frequency for publishing the PTM messages.
45. Device **SHALL** be able to set the Time zone in GMT for frequency and publishing the PTM messages.
46. Device **SHALL** be able to save PTM at every 5th min with configured PTM message format.
47. Device **SHALL** be able to publish the RCR at every 1 hour to the cloud or AWS Server.

**Network connection**

1. Device **SHALL** keep retrying to connect with the network after every 2 min of time duration if it is not connected to the network.
2. Device **SHALL** send the data to the persistent memory when network disconnection happens.
3. Device **SHALL** save and publish the data when the network reconnects.

# **Hardware Requirements**

1. Device **SHALL** work with the temperature Sensor with the range of -10 to 85 ˚C.
2. Device **SHALL** have the temperature Sensor with min wire length of 3 meter approx. as distance between sensor and controller is 5 feet.
3. Device **SHALL** have the temperature Sensor with the Head length of 3 Inch.
4. Device **SHALL** have the contact sensor (MICRO SWITCHES BZC) with dependable performance up to 20 million mechanical cycles.
5. Device **SHALL** have the contact sensor (MICRO SWITCHES BZC) with Ambient Humidity of -20～+80°C (-4～176oF).
6. Device **SHALL** have the power supply module supporting single phase 110-220V input and 22-28V DC as output
7. Device **SHALL** have battery with Power (Watt) backup for 1 Hour
8. Device **SHALL** have battery with continuous discharge rate (C rating) with 8 lithium ions.
9. Device **SHALL** have battery with nominal voltage of 25.9V.
10. Device **SHALL** have battery with max continuous output current of 8 Ampere.
11. Device **SHALL** have PLC with operating system Real-time Linux (with RT-Preempt patch).
12. Device **SHALL** have PLC with CPU Cortex A8, 1 GHz.
13. Device **SHALL** have PLC with Main memory (RAM) 512 MB.
14. Device **SHALL** have pfc-firmware-sdk-s2 installed on OS (RTOS Linux).
15. Device **SHALL** have the digital input module to sense digital signal.
16. Device **SHALL** have Each input module with a noise-rejection filter with different time constant on each input module
17. Device **SHALL** have the analog input module consisting of 8 channels.
18. Device **SHALL** have RTD module on which 2 wire RTD sensor can be connected.
19. Device **SHALL** have the RTD module which automatically linearizes the entire temperature range.
20. Device **SHALL** have the analog input module in which a sensor error (short circuit, wire break or out-of-measurement range) is indicated by a red LED.
21. Device SHALL have the analog input module with a response time 3s at max.
22. Device SHALL have the lifecycle of 4 million cycles.

# **Server Requirements**

1. Server **SHALL** have API to provision the tenant, each tenant will be associated with CA.
2. Server **SHALL** have device provision with SSL certificates.
3. Server **SHALL** have associate certificate with IOT core policy for authorization.
4. Server **SHALL** have system command (SEM) to send configs to device.
5. Server **SHALL** have endpoints to do bulk operation (SEM).
6. Server **SHALL** receive Periodic Telemetry Message (PTM) having a Remote Condition Record, send at user defined frequency, stores in S3 and latest will be reflected in device virtual machine.
7. Server **SHALL** receive Realtime Telemetry Message () having a Remote Condition Record, sends as it happens, stores in S3 and latest will be reflected in device virtual machine.
8. Server **SHALL** receive Periodic Inspection Message (PIM) having automated Inspection Report Record, send at a frequency.
9. Server **SHALL** receive Periodic Inspection Message (PIM) having automated Inspection Report Record, send at a frequency.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Timestamp(ts)** | **TestID** | **Description** | **Test Status** | **TestResult (8Byte Txt)** | **TestResult (4Byte Float)** |
| Ts | 1 | FacePlateAlignment Precision | 1: Test was successful, and result is reliable. | Free text value field with value of test in text up to 8 bytes | Float value field with value of test in text up to 4 bytes |
| 0: Test failed, and outcome is not reliable | Error Message1 | Error Message2 |
| Ts | 2 | Open Cavities test | 1: Test was successful, and result is reliable. | Free text value field with value of test in text up to 8 bytes | Float value field with value of test in text up to 4 bytes |
| 0: Test failed, and outcome is not reliable | Error Message1 | Error Message2 |
| Ts | 3 | Remote Module test | 1: Test was successful, and result is reliable. | Free text value field with value of test in text up to 8 bytes | Float value field with value of test in text up to 4 bytes |
| 0: Test failed, and outcome is not reliable | Error Message1 | Error Message2 |
| Ts | 4 | RigVibrationTest | 1: Test was successful, and result is reliable. | Free text value field with value of test in text up to 8 bytes | Float value field with value of test in text up to 4 bytes |
| 0: Test failed, and outcome is not reliable | Error Message1 | Error Message2 |
|  |  |  |  |  |  |

1. Server **SHALL** report Device last known status.
2. Server **SHALL** receive Sys Event Message (SEM) with event type “SensorDisconnected” as soon as any of the sensors disconnects; and the Sensor ID is sent in Payload of the message. Alert will be generated and notifies respective stakeholders.
3. Server **SHALL** receive Sys Event Message (SEM) with event type “MainsOff” as soon as main power cut off happens. Alert will be generated and notifies respective stakeholders.
4. Server **SHALL** receive Sys Event Message (SEM) to the server with event type “BatteryLow” as soon as battery level is found below a configured percentage level, Alert will be generated and notifies respective stakeholders.
5. Server **SHALL** receive “BatteryLow” at every 5% level drop found below a configured percentage level. Alert will be generated and notifies respective stakeholders.
6. Server **SHALL** receive Sys Event Message (SEM) with event type “SensorError” as soon as the PLC gives the error for connected sensor. Alert will be generated and notifies respective stakeholders.
7. **Server SHALL** receive Sys Event Message (SEM) with event type “ConnLost” as soon as connectivity with the network is lost. Then send the same when reconnects to the server, Alert will be generated and notifies respective stakeholders.
8. Server **SHALL** receive a message when the connection with the network is lost and keep in persistent storage. This message will be cleaned from persistent storage once message is successfully sent to Server, Alert will be generated and notifies respective stakeholders.
9. Server **SHALL** receive Sys Event Message (SEM) with event type “SysStart” as soon as the system starts, Alert will be generated and notifies respective stakeholders.
10. Server **SHALL** receive Sys Event Message (SEM) with event type “SysProvision” as soon as the system gets provisioned, Alert will be generated and notifies respective stakeholders.
11. **Server SHALL** receive Sys Event Message (SEM) with event type “SysShutDown” as soon as the system is going to shut down, Alert will be generated and notifies respective stakeholders.
12. Server **SHALL** receive Sys Event Message (SEM) with event type “SystemCriticalError” as soon as any critical error occurs on the device., Alert will be generated and notifies respective stakeholders.
13. Server **SHALL** able to send the lock command
14. Server **SHALL** provide reset command
15. Server **SHALL** provide command to map sensor.
16. Server **SHALL** provide command to test stack pointer
17. Server **SHALL** provide command to test clock frequency
18. Server **SHALL** provide command to test program counter.
19. Server **SHALL** provide command to test CPU registers
20. Server **SHALL** provide test invariable memory
21. Server **SHALL** provide test variable memory
22. Server **SHALL** provide configure maximum number of flash locations to be tested during
23. Server **SHALL** provide CRC initialization value.
24. Server **SHALL** provide to initialize Power ON Self-Test parameters.
25. Server **SHALL** provide end point to trigger on demand health diagnosis (ODM)
26. Server **SHALL** provide endpoint to provide Health summary of the Device.
27. Server **SHALL** have endpoint to setup threshold for devices.
28. Server **SHALL** have backup of all , PTM and ODM, health records in S3.

# **Dependencies**

* App development and testing requires physical hardware devices.
* Code Versioning: GIT would be used for code versioning.

# **Acceptance Criteria**

Sprint Realization with review and functionally working.

* + Acceptance strategy should be defined as part of the Sprint planning meeting.
  + Review criteria should be set for every delivery.

# **Validation**

# **Validation Strategy**

* + Test plan and procedures will be written which will capture all the requirements mentioned in this SRS document.
  + A traceability matrix will be created which will match the requirements in the SRS document with the test case ID in the test procedure document.

# **Validation criteria**

* + The Validation of the requirements will be done based on traceability matrix created for this application.
  + All the requirements should be matched to corresponding test cases.

# **Validation Constraints**

* + Availability of all standard services, setup, and tools.
  + Certification testing based on MISRA compliance.
  + Memory usage check will be performed based on Memory Optimization and bug fixing and Memory Testing & Analysis.

# **References**

|  |  |  |
| --- | --- | --- |
| **Sr. No** | **Document Name** | **Remarks** |
| 1 |  |  |
| 2 |  |  |

Table 1: References

# **Glossary**

|  |  |
| --- | --- |
| **Abbreviation** | **Description** |
| RCMS | Remote condition monitoring system |
| RCR | Remote condition record |
| AWS | Amazon web server |
| PTM | Periodic telemetry message |
|  | Realtime telemetry message |
| PIM | Periodic inspection message |
| SEM | System event message |
| ODM | On demand diagnostic message |
| BIST | Built in self-test |

Table 2: Glossary