

6.49 CAL104 Smart Data Publishing

Verifies that the DUT responds properly to Command 103 and 104. Checks Command Number, Response Code and Byte Count for Command 103 and 104.

References:

Specification	Rev.	Sections
<i>Common Practice Command Specification</i>	9.0	7.76, 7.75, 7.71, 7.70

Test Case A: Verify Command 104

Verify the triggers and update periods settings.

```
CALL IdentifyDevice
IF UNIV_REVISION < 7 THEN
    PRINT "WARNING: Implementation of HART 7 is strongly recommended."
    Abort Test
END IF
```

 (1015)

Verify that Burst Mode is supported on wired connection of device under test.

```
CheckReadyForBurst()
```

Do basic sanity checks first - can we access all the burst messages?

```
SEND Command 105 with bmsg = 0 to read MAX_BURST_MSGS
CALL VerifyResponseAndByteCount ( 0, 31 )

SET bmsg = 0
SET classification = Not Classified (0)
SET units = Not Used (250)
SET value = NaN (0x7F, 0xA0, 0x00, 0x00)

DO
    SEND Command 108 with cmd=3, bmsg
    CALL VerifyResponseAndByteCount ( "Success", 5 )
    CALL IssueCommand104 (bmsg, "Continuous", classification,
        units, value)
    CALL VerifyResponseAndByteCount ( "Success", 10 )
    SET bmsg = bmsg + 1
WHILE (bmsg < MAX_BURST_MSGS)

CALL IssueCommand104 (MAX_BURST_MSGS+1, "Continuous", Not Classified ,
    Not Used , NaN, 1017)
CALL VerifyResponseAndByteCount ( "Invalid Burst Message", 2 )
```

 (1016)
(1017)

Check "Too Few Data Bytes" and forward compatibility

```
SEND Command 104 with bmsg = 0, and "Continuous" only
CALL VerifyResponseAndByteCount ( "Too Few Data Bytes Received", 2 )

DO
    SEND Command 104 with bmsg = 0, "Continuous", Not Classified ,
        Not Used , NaN and one extra byte
    CALL TestValidFrame()
WHILE ( (RESPONSE_CODE == "Busy")
    OR (RESPONSE_CODE == "Delayed Response Initiated")
    OR (RESPONSE_CODE == "Delayed Response Running") )

CALL VerifyResponseAndByteCount ( "Success", 10 )
```

When trigger is continuous the trigger value should be NaN. Device should correct if not

```
SEND Command 3 to read PV_UNITS, PV_VALUE
SEND Command 8 to read PV_CLASSIFICATION

CALL IssueCommand104 ( 0, "Continuous", PV_CLASSIFICATION,
  PV_UNITS, PV_VALUE, 1018)                                     (1018)
IF (RESPONSE_CODE != "Burst Condition Conflict")
  AND (RESPONSE_CODE != "SUCCESS")
  THEN test result is FAIL                                     (1012)
END IF
```

Now play with the classification and units - first read the range values

```
Send Command 15 to read URV, LRV
SET delta = ABS( URV - LRV ) * 20%

CALL IssueCommand104 ( 0, "Window", PV_CLASSIFICATION,
  PV_UNITS, delta, 1019)                                       (1019)
CALL VerifyResponseAndByteCount ( "Success", 10 )

CALL IssueCommand104 ( 0, "Window", PV_CLASSIFICATION+1,
  PV_UNITS, delta, 1020)                                       (1020)
CALL VerifyResponseAndByteCount ( "Invalid Device Variable
  Classification", 2 )
```

Confirm Response Code Invalid Units Code

```
SET units = 0
DO
  SET units = units + 1
  CALL IssueCommand104 ( 0, "Window", PV_CLASSIFICATION,
    units, PV_VALUE, 1021)                                       (1021)
WHILE ((units < 239) AND (RESPONSE_CODE != "Invalid Units Code"))

IF ( units >= 239 )
  THEN Test Result is FAIL                                     (1022)
END IF
```

Now play with the trigger level

```
CALL IssueCommand104 ( 0, "Falling", PV_CLASSIFICATION,
  PV_UNITS, -1E30, 1023)                                       (1023)
CALL VerifyResponseAndByteCount ( "Passed Parameter Too Small", 2 )

CALL IssueCommand104 ( 0, "Rising", PV_CLASSIFICATION,
  PV_UNITS, 1E30, 1024)                                       (1024)
CALL VerifyResponseAndByteCount ( "Passed Parameter Too Large", 2 )
```

Verify trigger modes - window mode tested already

```
FOR trigger = {Rising, Falling}
    CALL IssueCommand104 (0, trigger, PV_CLASSIFICATION,
        PV_UNITS, PV_VALUE, 1025) (1025)
    CALL VerifyResponseAndByteCount ( "Success", 10 )
END FOR
```

```
CALL IssueCommand104 (0, "On-Change", Not Classified ,
    Not Used , NaN, 1014) (1014)
CALL VerifyResponseAndByteCount ( "Success", 10 )
```

```
CALL IssueCommand104 (0, trigger = 239, PV_CLASSIFICATION,
    PV_UNITS, PV_VALUE, 1026) (1026)
CALL VerifyResponseAndByteCount ( "Invalid Burst Trigger Mode
    Selection ", 2 )
```

And we are done

```
CALL IssueCommand104 (0, "Continuous", Not Classified ,
    Not Used , NaN, 1027) (1027)
CALL VerifyResponseAndByteCount ( "Success", 10 )
```

END TEST CASE