Functional testing (Incremental)

# Default values

Set Resource.Restart = Defaults

Verify parameters as follows:

AI blocks - all:

              L\_Type should be Direct

             OutScale should be "psi"

AO block: IO\_Opts should be PV for BkCal out

              Shed Options: NormalShed\_NormalReturn

PID block:

              Bypas = 1 (Off)

              Gain = 1,0

              Shed Options: NormalShed\_NormalReturn

IS block:

              Sel Type = First Good

Output Splitter block

              LockVal = No Lock

             In Array = 0, 50, 50, 100

             Out Array = 0, 50, 50 100

Control Selector

             Sel Type = Middle

Arithmetic

            Arith Type = Average

# Response codes

Write TB block header with incorrect PD\_TAG

Verify that the write is rejected, and that the response code is well-defined and correct.

# TB

## Advanced

### Persistence

Verify that “Advanced” key is preserved in the following cases:

* Firmware upgrade R3🡪R3
* Writing RB.RESTART=42 (creating factory defaults)
* Writing RB.RESTART=Restore TB to factory (restoring factory defaults)
* Restart
* Power cycle

### Functionality

Verify that parameters prohibited by Advanced are not affected internally

This correct the following behavior:

*The "filter" zeros memory within live ptb\_1. This has the following effects:*

*1. All zeros may not be a conforming value, such as e.g. it is a status*

*2. Much worse, zeroing the memory changes the behavior of TB and FBAP. E.g. if ANALOG\_INPUT is hooked up to an AI FB, the AI input is erratic. IOW, the device works differently when reads are on vs. when no-one is reading.*

Attempt to reproduce as follows

Configure "Advanced" to Standard.

Configure NI Dialog to read ANALOG\_INPUT (or any other prohibited parameter) many (e.g. 200) times. Observe that occasionally, status and/or value are 0

## FAILED\_STATE

Set TB.FAILED\_STATE.PROPAGATE\_MODE=Report in TB and modify a few other entries from defaults.

Verify that after RB.RESTART=Defaults TB.FAILED\_STATE.PROPAGATE\_MODE=Propagate to RB, and the other entries restored to their defaults.

## UNITS

Change pressure units from FF (PRESSURE\_RANGE.UNITS\_INDEX) and separately, from local UI.

Verify that PRESSURE\_RANGE.EU\_100 and pressure values are adjusted accordingly.

## APP mode

Connect AO FB to TB and make both in AUTO, with good setpoint.

Put TB in LO mode (APP setup mode)

Verify that AO is in IMAN and TB mode is LO and FINAL\_VALUE is unchanged.

Put AO in OOS.

Verify TB FINAL\_VALUE.STATUS is BAD.

After time exceeding XD\_FSTATE.STATE\_TIME\_1 (which could be set to 1 to speed up the test) + 10 s, change APP mode to Normal.

Verify that the APP mode changed correctly

## Extremes

Verify that on reset,

* POSTION\_EXTREMES
* PRESSURE\_EXTREMES
* TEMPERATURE\_EXTREMES
* IP\_CURRENT\_EXTREMES

have large MIN and small MAX values immediately overwritten by real data.

Write absurdly large MAX and small MIN values.

Reset the device

Verify that the written values are gone and correct initial values are set.

## Static revision

### Factory restore

According to FF spec, if host writes a parameter and discovers ST\_REV is incremented by 1, it may choose to read only the affected parameter. If ST\_REV is incremented by more than 1, it must read all the parameters.

So, RB.RESTART=”Restore TB to factory” has to increment ST\_REV by more than 1.

Verify that.

### Alerts data

Verify that writing to TB.ALERT and TB.HISTORIC\_ALERT do not increment ST\_REV.

The following Rev.2 behavior is corrected:

Writing to HISTORIC\_ALERT:

Increment ST\_REV and stick: DEVIATION NEAR\_CLOSED SUPPLY\_PRESSURE (HI, LO, LOLO) WORKING\_TIME

Not increment ST\_REV and revert: CYCLE\_COUNTER

Writing to ALERT:

Increment ST\_REV and revert: WORKING\_TIME

Increment ST\_REV and stick: DEVIATION NEAR\_CLOSED SUPPLY\_PRESSURE (HI, LO, LOLO) – but only if writing the same value

Not increment ST\_REV and revert: CYCLE\_COUNTER

## APP mode transition

Verify that the following behavior doesn’t occur:

*On transition from LO/Setup to MAN/Normal, FINAL\_VALUE is correctly updated but it is not written back automatically. So, APP keeps using the manual setpoint set in Setup mode and can diverge wildly from FINAL\_VALUE*

Attempt to reproduce it as follows:

[For the most dramatic view, turn off the air or enable slow rate limit]

Set TB target mode to MAN.

In APP Setup mode, change setpoint to -5% with local UI. Wait for position to go to 0.

Now change setpoint to 160%.

Go to Normal mode. [If you turned off the air, turn it back on]

Observe that FINAL\_VALUE is near 0 and setpoint in LCD is 160% and the valve follows the local setpoint.

## XD\_FSTATE

Configure air action to ATC and XD\_FSTATE to fail open (another case - fail closed), and small timeout (like 10) to observe easily.

After running OK, set AO FB to OOS to elicit bad FINAL\_VALUE.

Observe that the valve fails in the correct direction as configured.

Test against requirements [Design\DO FB Channel FAULT.docx](Design/DO%20FB%20Channel%20FAULT.docx)

# DO FB

## Discrete control

Configure TB setpoint source to DO FB FINAL\_VALUE\_D, and in a DO FB, set CHANNEL to TB.FINAL\_VALUE\_D. Make sure DO and TB are in AUTO, and readback of DO configured to FINAL\_POSITION\_VALUE\_D

Make valve position limited (lo or high).

Verify that “limited” status of readback is

* The same as of FINAL\_POSITION\_VALUE\_D, if DO is set up for direct action
* The opposite of FINAL\_POSITION\_VALUE\_D, if DO is set up for reverse action

Put TB in (a) MAN and (b) LO.

Verify that DO bkcal\_out and rcas\_out have valid and sensible substatus code.

## Numeric control

Like [Discrete control](#_Discrete_control) except setpoint source and DO channel are set up for FINAL\_VALUE\_DINT.

# Local UI

## Find stops

Execute “Find stops” method from local UI.

Verify that raw position shown in the first line of the LCD is sensible and matches, more or less, the raw position read with SA:  


## Setpoint presentation

Write FINAL\_VALUE outside UI range [-5%, 175%], e.g. -50% or 180%

Verify that the cycling menu displays setpoint correctly in “SETPOINT” UI screen.

Put TB in LO (APP in Setup)

Navigate to “MAN POS”

Verify that setpoint is presented correctly.

Press \* to edit the setpoint.

Verify that the value is clamped to [-5%, 175%].

## Autotune

Run autotune from local UI. While it is in progress, turn off the air.

Verify that autotune fails and LCD displays “TUNE ERR”

# Physical analog input

Connect analog input to 1-5V source via 250 Ohm resistor.

Verify that

* The input is shown correctly in TB.ANALOG\_INPUT
* Status is GOOD when TB is not OOS, except when the value is outside signaling range [3.7-20.4] mA

Configure AI block to receive analog input as a channel.

Verify that the value and status propagate correctly.

# Find stops

Turn off the air and run “find stops” (from FF with TB in MAN and/or from local UI in Setup).

Verify that FAULT\_FIND\_STOPS\_FAILED is set

Verify that TB cannot be placed in AUTO.

Write a stop from TB.TRAVEL\_CALIBRATION.

Verify that FAULT\_FIND\_STOPS\_FAILED is cleared and TB can be placed in AUTO.

Execute RB.RESTART=”Restore TB to factory”

Verify restore is successful

Verify that

* Stops are unchanged in TB.TRAVEL\_CALIBRATION
* FAULT\_FIND\_STOPS\_FAILED is not set

Change position sensor type in TB.ACCESSORY.REMOTE\_SENSOR.

Verify that FAULT\_FIND\_STOPS\_FAILED is set

# DO FB Channel FAULT and TB

Verify that fault channel works according to design

# Parameters owned by APP

Verify that the following parameters are updated automatically, whether they are read periodically (with e.g. NI Configurator) or not: