

# Position Master EDP300

## Digital Positioner



Partial Stroke Valve Test (PST)  
Control via HART Commands

—  
EDP300

### Introduction

The PositionMaster EDP300 is an electronically configurable positioner with communication capabilities designed for mounting to linear or rotary actuators.

The Position Master EDP300 provides advanced performance 4 to 20mA with HART.

This document provides information for the Partial Stroke control via HART Commands

### For more information

Additional documentation on Position Master EDP300 is available for download free of charge at [www.abb.com/positioners](http://www.abb.com/positioners). Alternatively simply scan this code:



# Table of contents

- 1 Introduction ..... 3**
- 2 Description ..... 3**
  - HART commands for the PST control ..... 3
  - Description of HART commands ..... 4
    - Read Diagnostic Limits ..... 4
    - Write Diagnostic Limits..... 5
    - Read Partial Stroke Configuration ..... 7
    - Write Partial Stroke Configuration ..... 8
    - Get Write Handle ..... 9
    - Read Valve Diagosic State ..... 10
  - EDP300 PST Sequence and business logic ..... 13
    - PST Start and result sequence ..... 13
    - PST Abort / Stop Sequence ..... 14
- 3 Version History ..... 14**

## 1 Introduction

This document describes how to start the Partial stroke valve diagnostics test of the EDP300 device in using its built-in HART commands.

The EDP300 built-in Partial Stroke Test (PST) can be used to check the function of the safe position of attached ESD (emergency shutdown) valves. Furthermore, the successfully executed PST demonstrates that certain unresolved errors that would otherwise go undetected, such as stuck valve or spring fractures in the spring chamber of the pneumatic actuator are identified. The test can be started both locally by the device itself in a time-controlled manner, or by using HART command control (e.g. via DTM) from any HART capable remote system.

As test procedure the EDP300 evacuates the output for the actuator/valve movement. If this does not happen within the defined time (timeout value) an alarm flag is generated as failure indication. Additionally, a self-monitoring is performed to establish whether the valve has moved to its end position within a defined period of time (dead time). If this has not occurred, the test is cancelled as a "failed" test and an alarm flag is generated as failure indication. At the end of the test, the EDP300 moves the valve to the last valid position and returns to the most recently active control mode.

For documentation purposes, the test result is entered in the event history and saved in the non-volatile memory of the positioner.

### Note

For ESD valves whose safe position is "closed", it is recommended that the following parameter be set to "On": "Device Setup->END STOP BEHAV.->Control at 100 %".

This ensures that the depressurizing time and "dead time" are as short as possible during closure.

## 2 Description

There are three parameters that need to be configured as criteria for the valve movement verification:

**"PST Valve Stroke"** as a percentage of the (adjusted) total valve travel

Parameter:

Value of position change in direction exhaust [0...100%]

→ **ValveDiagnosis.partialStrokeMovement**

**"PST Dead time"** as the time duration for the verification if valve responds to the invoked signal change

Parameter:

Startup Time [0...Positioning timeout] msec

→ **ValveDiagnosis.partialStrokeDeadTime**

**"PST Timeout"** as the time duration to ensure that the valve can reach the predetermined percentage travel for the PST test cycle.

Parameter:

Positioning timeout [0...100] sec

→ **ValveDiagnosis.partialStrokeTimeOut**

These parameter settings needs to be determined by the end user based on the process and expected performance of the PST valve and actuator assembly.

If the target position is out of range the result of the test is "failed".

The Partial Stroke test can be configured to a user configurable repeat interval of [0...1000] days, the test is repeated after the interval has expired.

→ **ValveDiagnosis.partialStrokeInterval**

If 0 days is specified the test will not be executed periodically.

The result of the partial stroke test is signalled as diagnosis status [**ValveDiagnosis.partialStrokeState**].

This diagnosis status will remain unchanged until the next result is available.

(A bad result can be cleared without performing a new test by resetting the device.)

## HART commands for the PST control

Command No	
Read command No 182	read diagnosis limits
Write command No 183	write diagnosis limits
Read command No 214	ReadValveDiagnosisState
Read command No 225	read Partial Stroke Configuration
Write command No 226	write Partial Stroke Configuration
Write command No 147	get Write Handle

## ... 2 Description

### Description of HART commands

#### Read Diagnostic Limits

READ-command 182

reads temperature limits / timer limits

RESPONSE		byte(s)	type	object (default)
+----->	#0..#1	TD_INT16	ValveDiagnosis.positioningTimeLimit	
+----->	#2..#9	TD_FLOAT	Pressure.pressureLimits[0..1]	
+----->	#10..#11	TD_USIGN16	DiagnosisServices.diagnosisTimeLatch	
+----->	#12..#13	TD_INT16	ValveDiagnosis.partialStrokeDeadTime	
+--->	#14..#17	TD_FLOAT	PositionProcessing.zeroPositionLimit	
+-----+				
+-----+				
0	17			

#### BRIEF DESCRIPTIONS

##### ValveDiagnosis.positioningTimeLimit

Limit for positioning-time in seconds. When the positioning time exceeds the limit. 0 means that the function is disabled.

##### Pressure.pressureLimits

Limits for signalling the pressure Alarm bit (set if out of specified limits)

##### DiagnosisServices.diagnosisTimeLatch

Timer, how long the alarms are active after the diagnosis is active even through the diagnosis is not active. A value of "0" disable's the function. the customer can specify a time between 1s and 30min

##### ValveDiagnosis.partialStrokeDeadTime

Deadtime for startup surveillance in partial stroke [ms] (0 = inactive)

##### PositionProcessing.zeroPositionLimit

limit for zero position detection

**Write Diagnostic Limits**

WRITE-command 183

write temperature limits / timer limits

```

-----
| REQUEST |
-----

```

	byte(s)	type	object (default)
+----->	#0..#1	TD_INT16	ValveDiagnosis.positioningTimeLimit
+----->	#2..#9	TD_FLOAT	Pressure.pressureLimits[0..1]
+----->	#10..#11	TD_USIGN16	DiagnosisServices.diagnosisTimeLatch
+----->	#12..#13	TD_INT16	ValveDiagnosis.partialStrokeDeadTime
+--->	#14..#17	TD_FLOAT	PositionProcessing.zeroPositionLimit
+-----+			
+-----+			
0	17		

```

-----
| RESPONSE |
-----

```

	byte(s)	type	object (default)
+----->	#0..#1	TD_INT16	ValveDiagnosis.positioningTimeLimit
+----->	#2..#9	TD_FLOAT	Pressure.pressureLimits[0..1]
+----->	#10..#11	TD_USIGN16	DiagnosisServices.diagnosisTimeLatch
+----->	#12..#13	TD_INT16	ValveDiagnosis.partialStrokeDeadTime
+--->	#14..#17	TD_FLOAT	PositionProcessing.zeroPositionLimit
+-----+			
+-----+			
0	17		

## ... 2 Description

### ... Description of HART commands

---

| BRIEF DESCRIPTIONS |

ValveDiagnosis.positioningTimeLimit

Limit for positioning-time in seconds. When the positioning time exceeds the limit. 0 means that the function is disabled.

Pressure.pressureLimits

Limits for signalling the pressure Alarm bit (set if out of specified limits)

DiagnosisServices.diagnosisTimeLatch

Timer, how long the alarms are active after the diagnosis is active even through the diagnosis is not active. A value of "0" disable's the function. the customer can specify a time between 1s and 30min

ValveDiagnosis.partialStrokeDeadTime

Deadtime for startup surveillance in partial stroke [ms] (0 = inactive)

PositionProcessing.zeroPositionLimit

limit for zero position detection

---

**Read Partial Stroke Configuration**

READ-command 226

```

-----
| RESPONSE |
-----

```

	byte(s)	type	object (default)
+----->	#0..#1	TD_INT16	ValveDiagnosis.partialStrokeTimeOut
+----->	#2..#3	TD_INT16	ValveDiagnosis.partialStrokeInterval
+----->	#4..#7	TD_FLOAT	ValveDiagnosis.partialStrokeMovement
+----->	#8..#11	TD_USIGN32	ValveDiagnosis.timeToPartialStroke
+-->	#12	TD_USIGN8	ValveDiagnosis.partialStrokeState
+---+---+---+---+---+			
+---+---+---+---+---+			
0	12		

```

-----
| ENUMERATIONS |
-----

```

ValveDiagnosis.partialStrokeState

```

[0x10] NO_PARTIAL_STROKE
[0x20] PS_START_INTERVAL
[0x40] PS_START_HMI_DTM
[0x06] PARTIAL_STROKE_TEST_PASSED - last "PST passed"
[0x07] PARTIAL_STROKE_TEST_TIME_OUT - last "PST failed time out"
[0x08] PARTIAL_STROKE_TEST_OUT_OF_RANGE - last "PST failed out of range"
[0x09] PARTIAL_STROKE_TEST_USER_ABORT - last "PST failed user abort"
[0x0a] PARTIAL STROKE TEST FAILED - last "PST failed unknown"

```

## ... 2 Description

### ... Description of HART commands

#### Write Partial Stroke Configuration

WRITE-command 225

REQUEST			
	byte(s)	type	object (default)
+----->	#0..#1	TD_INT16	ValveDiagnosis.partialStrokeTimeOut
+----->	#2..#3	TD_INT16	ValveDiagnosis.partialStrokeInterval
+----->	#4..#7	TD_FLOAT	ValveDiagnosis.partialStrokeMovement
+-->	#8	TD_USIGN8	ValveDiagnosis.partialStrokeState
+---+---+---+---+			
+---+---+---+---+			
0	8		

RESPONSE			
	byte(s)	type	object (default)
+----->	#0..#1	TD_INT16	ValveDiagnosis.partialStrokeTimeOut
+----->	#2..#3	TD_INT16	ValveDiagnosis.partialStrokeInterval
+----->	#4..#7	TD_FLOAT	ValveDiagnosis.partialStrokeMovement
+-->	#8	TD_USIGN8	ValveDiagnosis.partialStrokeState
+---+---+---+---+			
+---+---+---+---+			
0	8		

#### ENUMERATIONS

ValveDiagnosis.partialStrokeState

- [0x10] NO\_PARTIAL\_STROKE
- [0x20] PS\_START\_INTERVAL
- [0x40] PS\_START\_HMI\_DTM
- [0x06] PARTIAL\_STROKE\_TEST\_PASSED - last "PST passed"
- [0x07] PARTIAL\_STROKE\_TEST\_TIME\_OUT - last "PST failed time out"
- [0x08] PARTIAL\_STROKE\_TEST\_OUT\_OF\_RANGE - last "PST failed out of range"
- [0x09] PARTIAL\_STROKE\_TEST\_USER\_ABORT - last "PST failed user abort"
- [0x0a] PARTIAL\_STROKE\_TEST\_FAILED - last "PST failed unknown"



**Get Write Handle**

WRITE-command 147

gets the Write handle to write the HART objects

---

```

-----
| REQUEST |
-----

```

	byte(s)	type	object (default)
+->	#0	TD_USIGN8	Coordinator.hartRequest (REQUEST_PASSIVE)
+++			
+++			
0			

---

```

-----
| RESPONSE |
-----

```

	byte(s)	type	object (default)
+->	#0	TD_USIGN8	Coordinator.hartRequest (REQUEST_PASSIVE)
+++			
+++			
0			

---

```

-----
| BRIEF DESCRIPTIONS |
-----

```

Coordinator.hartRequest  
 access rights management, hart requests access

---

## ... 2 Description

### ... Description of HART commands

#### Read Valve Diagnostics State

READ-command 214

Read Valve Diagnosis States Progressbar State ,.....

```

-----
| RESPONSE |
-----
    
```

	byte(s)	type	object (default)
+----->	#0	TD_USIGN8	ValveDiagnosis.offlineDiagnosisMode
+----->	#1	TD_USIGN8	ValveDiagnosis.testProfile
+----->	#2	TD_USIGN8	ValveDiagnosis.testProfileEndless
+----->	#3	TD_USIGN8	ValveDiagnosis.offlineState
+----->	#4..#7	TD_FLOAT	ValveDiagnosis.offlineProgress
+----->	#8	TD_USIGN8	ValveDiagnosis.offlineDiagnosis
+---->	#9..#12	TD_USIGN32	ValveDiagnosis.diagnosticsFlags
+-----+			
+-----+			
0	12		

---

```
-----  
| ENUMERATIONS |  
-----
```

ValveDiagnosis.offlineDiagnosisMode

- [0] CONTROL\_LOOP\_TEST
- [1] LEAKAGE\_TEST
- [3] STEP\_RESPONSE
- [4] VALVE\_SIGNATURE
- [5] SPEED\_OVER\_POSITION
- [6] ZERO\_POINT\_TEST
- [7] OFFLINE\_INIT
- [8] MOVE\_TO\_LAST\_SETPOINT - move to setpoint (last action in offline diag
- [9] WAIT\_UNTIL\_DONE - wait until setpoint reached
- [10] OFFLINE\_READY - offline diagnosis done
- [11] OFFLINE\_FAILED - offline diagnosis done, internal failure

ValveDiagnosis.testProfile

- [0] CTRL\_TEST\_PROF\_SERI - for seri chart test
  - [1] CTRL\_TEST\_PROF\_OMCT - air liquide one million cycle test
  - [2] CTRL\_TEST\_PROF\_ADJ - similar to auto adjustment
  - [3] CTRL\_TEST\_PROF\_IEC61514 - some test cases of iec 61514
  - [4] CTRL\_TEST\_PROF\_RECTANGLE\_F1
  - [5] CTRL\_TEST\_PROF\_RECTANGLE\_F2
  - [6] CTRL\_TEST\_PROF\_RECTANGLE\_F3
  - [7] CTRL\_TEST\_PROF\_TRIANGLE\_F1
  - [8] CTRL\_TEST\_PROF\_TRIANGLE\_F2
  - [9] CTRL\_TEST\_PROF\_TRIANGLE\_F3
  - [10] CTRL TEST PROF SINUS SWEEP
-

## ... 2 Description

### ... Description of HART commands

---

#### ValveDiagnosis.offlineState

- [0] CONTROL\_LOOP\_TEST
- [1] LEAKAGE\_TEST
- [3] STEP\_RESPONSE
- [4] VALVE\_SIGNATURE
- [5] SPEED\_OVER\_POSITION
- [6] ZERO\_POINT\_TEST
- [7] OFFLINE\_INIT
- [8] MOVE\_TO\_LAST\_SETPOINT - move to setpoint (last action in offline diagnosis)
- [9] WAIT\_UNTIL\_DONE - wait until setpoint reached
- [10] OFFLINE\_READY - offline diagnosis done
- [11] OFFLINE\_FAILED - offline diagnosis done, internal failure

#### ValveDiagnosis.offlineDiagnosis

- [0] NO\_OFFLINE\_DIAGNOSIS - no offline diagnosis available
  - [1] DIAGNOSIS\_FAILED - offline Diagnosis failed
  - [2] ZERO\_POSITION\_TEST\_PASSED - position and universal AI in specified limits
  - [3] ZERO\_POSITION\_TEST\_FAILED - position or universal AI outside specified limits
  - [4] LEAKAGE\_TEST\_PASSED - no leakage
  - [5] LEAKAGE TEST FAILED - leakage in ... see diagnoseFlags
- 

```

-----
| BRIEF DESCRIPTIONS |
-----

```

#### ValveDiagnosis.testProfile

test scenario for control loop test

#### ValveDiagnosis.testProfileEndless

test will be performed endless (until stopped by target mode change) if this parameter is !=0

#### ValveDiagnosis.offlineState

state of the offline diagnosis function

#### ValveDiagnosis.offlineDiagnosis

last offline diagnosis result

#### ValveDiagnosis.diagnosisFlags

flags for diagnosis-subsystem

---

## EDP300 PST Sequence and business logic

### Note

HART read access is required first, means the read commands 182, 226 needs to be performed before PST.

The following objects inside above listed HART commands are relevant for PST:

ValveDiagnosis.partialStrokeTimeOut	[TINT16]
ValveDiagnosis.partialStrokeInterval	[TINT16]
ValveDiagnosis.partialStrokeMovement	[TFLOAT]
ValveDiagnosis.timeToPartialStroke	[TUSIGN32]
ValveDiagnosis.partialStrokeState	[TUSIGN8]
ValveDiagnosis.partialStrokeDeadTime	[TINT16]

### PST Start and result sequence

WRITE command 147 with Coordinator.hartRequest = 1

WRITE command 183 with ValveDiagnosis.partialStrokeDeadTime [TINT16]

WRITE command 225 with

ValveDiagnosis.partialStrokeTimeOut = 10 (value as example)

ValveDiagnosis.partialStrokeInterval = 100 (value as example)

ValveDiagnosis.partialStrokeMovement = 10 (value as example)

If Start "direct mode" THAN

ValveDiagnosis.partialStrokeState = 64 // ATTENTION System moves immediately

ELSE If Start in "Interval mode"

ValveDiagnosis.partialStrokeState = 32 // ATTENTION System moves after reached interval

WRITE command 147 with Coordinator.hartRequest = 0

POLL command 214 // for the test state and progress

READ ValveDiagnosis.offlineProgress

// for progress,

IF ValveDiagnosis.offlineProgress == 100%

READ partialStrokeState

EXIT POLL

ENDIF

END POLL

## ... 2 Description

### ... EDP300 PST Sequence and business logic

READ command 226 / for the final results

ValveDiagnosis.partialStrokeState

```
[0x10] NO_PARTIAL_STROKE  
[0x20] PS_START_INTERVAL  
[0x40] PS_START_HMI_DTM  
[0x06] PARTIAL_STROKE_TEST_PASSED - last "PST passed"  
[0x07] PARTIAL_STROKE_TEST_TIME_OUT - last "PST failed time out"  
[0x08] PARTIAL_STROKE_TEST_OUT_OF_RANGE - last "PST failed out of range"  
[0x09] PARTIAL_STROKE_TEST_USER_ABORT - last "PST failed user abort"  
[0x0a] PARTIAL_STROKE_TEST_FAILED - last "PST failed unknown"
```

#### PST Abort / Stop Sequence

WRITE command 147 with Coordinator.hartRequest = 1

WRITE command 225 with ValveDiagnosis.partialStrokeState = 16

WRITE command 147 with Coordinator.hartRequest = 0

## 3 Version History

Revision version:

25 July 2014

IIM-TI-04-2014-P

## Notes

---

## ABB Measurement & Analytics

For your local ABB contact, visit:  
[www.abb.com/contacts](http://www.abb.com/contacts)

For more product information, visit:  
[www.abb.com/positioners](http://www.abb.com/positioners)

---

We reserve the right to make technical changes or modify the contents of this document without prior notice. With regard to purchase orders, the agreed particulars shall prevail. ABB does not accept any responsibility whatsoever for potential errors or possible lack of information in this document.

We reserve all rights in this document and in the subject matter and illustrations contained therein. Any reproduction, disclosure to third parties or utilization of its contents – in whole or in parts – is forbidden without prior written consent of ABB.