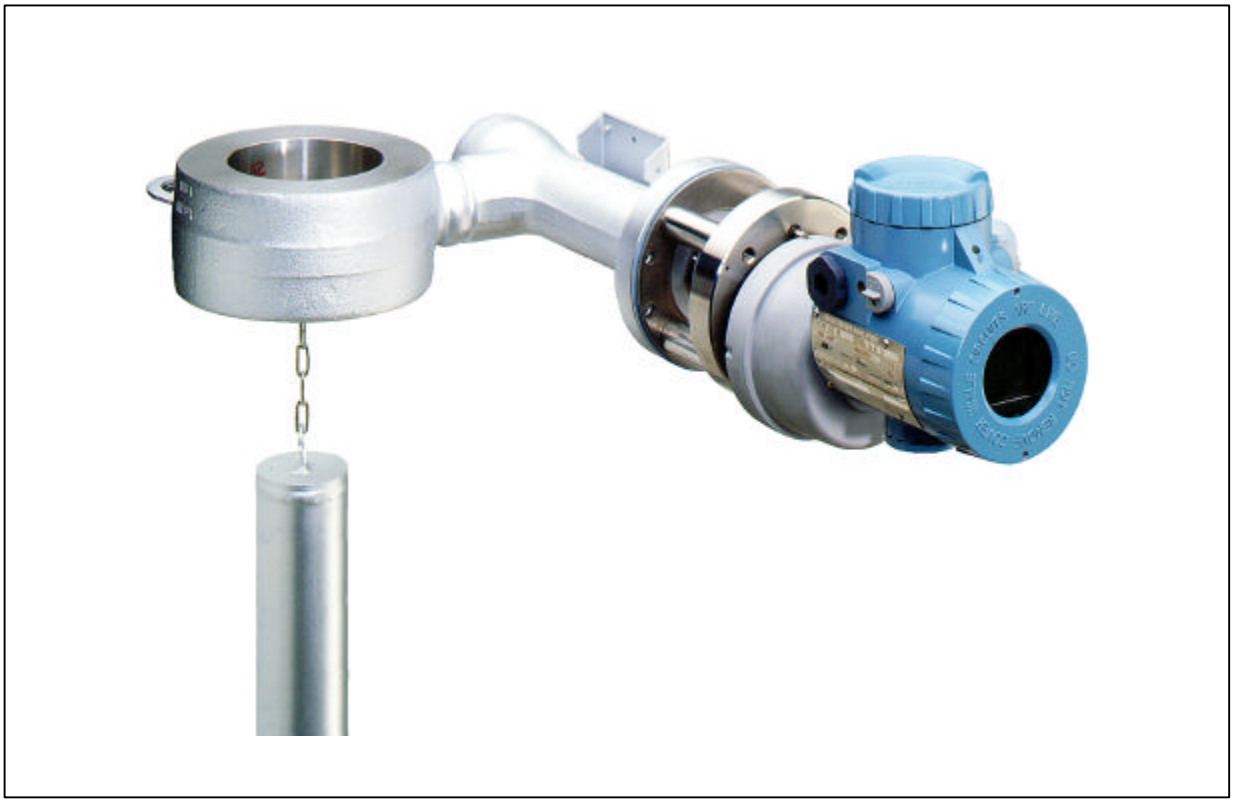
144LD, 244LD Intelligent Buoyancy Transmitters

Communication with FF-FIELDBUS



The intelligent buoyancy transmitters 144LD and 244LD are designed to perform measurements for liquid level, interface and density of liquids based on the Archimedes buoyancy principle and are accessible via fieldbus communication according Foundation Fieldbus (FF) specifications under the common device type 240FF.

FEATURES

- Communication FIELDBUS according FF Specifications
- Display in % or physical units
- Power supply DC 12...30V
- Current Imax 150mA
- Measure temperature –196°C to +400°C
- Connection according IEC 1158-2

FOXBORO ECKARDT

CONTENTS

1 CONFIGURATION VIA FIELDBUS

The intelligent transmitter 240FF performs function block (FB) application of measurements according the Foundation Fieldbus (FF) specifications. Physically the 240FF interfaces to external devices via one two-wire connection according the international standard IEC 1158-2. This two-wire connection is used for both power supply and digital communication of fieldbus. The general condition for the fieldbus agrees with the description of the communication protocol in [1] - [8].

The configuration of the transmitter 240FF consists of two major parts: management configuration and device configuration.

1.1 Management Configuration

The management configuration is divided into network management and system management. Network management provides for the configuration of the communication stack in the device while system management provides for the synchronization of the execution of the device function blocks and the communication of block parameters on the fieldbus. In the 240FF, the FF communication stack initializes implicitly most of the configuration parameters. Only some configuration parameters have to be customized for the FB application. Among these parameters, the following three are required in system management for identifying the device:

- Device identification (Device ID)
- Physical device tag
- Node address

For the 240FF, they are initialized with the following values:

Name	Value
Device identification	3858842506_240FF$<xx/yyyyyy>
Physical device tag	240FF$<xx/yyyyyy>
Node address	22 (0x16)

<xx/yyyyyy> = Device serial number (Fabrication number), e.g. 20/001000.

The 240FF may go through three major states before it can fully function on the network:

1. If the above values are assigned before the device is started, the 240FF system management starts in the state SM_OPERATIONAL. The network management agent of the device activates the application layer protocols, allowing applications to communicate across the network. To become fully operational, further network communication configurations may be necessary.

2. If the station cannot use the assigned node address for that this address is already used by another device, the device is assigned one of the default addresses between 0xF8 - 0xFF and the state is set to SM_INITIALIZED. In this case no other services are available except assigning a node address, clearing the physical device tag and identifying the device.

3. If Device Identification is set only, system management starts in state SM_UNINITIALIZED. In this state no other services but identifying the device and configuring the device with a physical device tag are available. For a more detailed description of system management services and procedures refer to [3].

The services listed below are supported by the 240FF:

Service	Type
Variable Access	Read, Write and Information Report
Event Management	Event Notification Event Notification with Type Acknowledge Event Notification Alter Event Condition Monitoring
Context Management	Initiate, Abort and Reject
Object Dictionary (OD) Management	Get OD
VFD (Virtual Filed Device) Status	Status, Unsolicited Status, Identify

1.2 Device Configuration

Device configuration is devided into function block connection and parametrization. For the former, the 240FF owns one function block whose output can be connected to the inputs of the function blocks of other devices, depending on the application interested.

Parametrization includes the configuration of the FB application objects like parameters, trend, alert and AP directory objects. The 240FF has an internal database containing all these objects that are a list of references to the parameters making up this application with data accessible via parameter indices, parameter names or DD-items (Device Description items). The device is configured by accessing the database through the communication of fieldbus. Together with the management configuration data, the database is stored in the device EEPROM with a static version number that increases after each of its updates. The parameters remain unchanged in case of power-off and keep so long until they are updated again.

The 240FF is set in plant with a factory setting. A user can do further configurations of his own based on this elementary setting. Inversely, the user can also restore this setting whenever required (see Section 2).

2 PARAMETER

The internal database of the 240FF is provided according the FF Standard Function Blocks (Resource Block and Analog Input (AI) Function Block) with additional manufacturer-specific extensions and FF Transducer Block called Standard Pressure with Calibration Basic Device Access with additional manufacturer-specific extensions. The data contained in the database are communicated over the fieldbus and the objects contain these data are described by object descriptions. A collection of all object descriptions is called Object Dictionary (OD).

2.1 Parameter Table

The following table shows all objects (parameters) of the object dictionary for the 240FF.

Table Legends:

Store

S: Static, the parameter must be stored non-volatile in EEPROM. Changing the parameter increases the static revision counter.

N: Non-volatile parameter stored in EEPROM. Changing the parameter does not increase the static revision counter.

D: Dynamic, the parameter is dynamic and is calculated or changed by the block. It is stored only in RAM.

Access

RO Read-only
WO Write-only, read always as a constant
RW Readable and writeable
ROW Read-only in general, writeable in Man-Mode
Mix Some elements in data structure are writeable while the others are read-only
DS Data structure
EI Element indices

Index	Parameter Name	Type	Size in Bytes	Store	Access	Valid Range	Default Value	Access Error Code
0	OD_OBJECT_DESCRIPTION	OD_HDR	44	N	RO	-	-	-
1-255	Reserved by FF	Data Type & DS	-	-	-	-	-	-
256 - 297	Reserved	Data Type & DS	-	-	-	-	-	-
298	AP_DIRECTORY	Array of USIGN16	60	N	RO	-	-	-
299	Unused/Reserved	-	-	-	-	-	-	-
Resource Block								
Standard Parameter								
300	BLK_DATA	DS-64	62	S	RW	-	-	-
301	ST_REV	USIGN16	2	S	RO	-	0	-
302	TAG_DESC	OSTRING	32	S	RW	-	Spaces	-
303	STRATEGY	USIGN16	2	S	RW	-	0	-
304	ALERT_KEY	USIGN8	1	S	RW	1 to 255	0	0x0005
305	MODE_BLK	DS-69	4	Mix	RW	O/S, Auto	O/S	0x0006
306	BLOCK_ERR	BSTRING	2	D	RO	-	-	-
307	RS_STATE	USIGN8	1	D	RO	0: Undefined 1: Start/Restart	Undefined	-

						2: Initialization 3: On-line Linking 4: On-line 5: Standby 6: Failure		
308	TEST_RW	DS-85	112	D	RW	-	-	-
309	DD_RESOURCE	VSTRING	32	S	RO	-	-	-
310	MANUFAC_ID	USIGN32	4	S	RO	Controlled by FF	0x385884	-
311	DEV_TYPE	USIGN16	2	S	RO	Set by manufacturer	0x2506	-
312	DEV_REV	USIGN8	1	S	RO	Set by manufacturer	1	-
313	DD_REV	USIGN8	1	S	RO	Set by manufacturer	1	-
314	GRANT_DENY	DS-70	2	D	RW	-	0,0	-
315	HARD_TYPES	BSTRING	2	S	RO	Set by manufacturer	0x8000 (Scalar Input)	-
316	RESTART	USIGN8	1	D	RW	0: Uninitialized 1: Run 2: Restart resource 3: Restart w. defaults 4: Restart processor	Run	0x0005
317	FEATURES	BSTRING	2	S	RO	Bit 0: Unicode strings (LSB) 1: Reports supported 2: Fault State supp. 3: Soft Write lock supported 4: Hard Write lock supported 5: Output readback supported 6: Direct write to output hardware 7: Change of BYPASS in an automatic mode	0x5800 (Reports, Soft Write lock, Hard Write lock)	-
318	FEATURE_SEL	BSTRING	2	S	RW	See FEATURES	0x5800	-
319	CYCLE_TYPE	BSTRING	2	S	RO	Bit 0: Scheduled (LSB) 1: Completion of block execution 2: Manufacturer specific	0xC000	-
320	CYCLE_SEL	BSTRING	2	S	RW	See CYCLE_TYPE	0xC000	-
321	MIN_CYCLE_T	USIGN32	4	S	RO	Set by manufacturer (Unit 1/32 ms)	960 (30 ms)	-
322	MEMORY_SIZE	USIGN16	2	S	RO	Set by manufacturer (Unit Kbytes)	0	-
323	NV_CYCLE_T	USIGN32	4	S	RO	(Unit 1/32 ms)	0 (It will be never automatically copied)	-
324	FREE_SPACE	FLOAT	4	D	RO	0 - 100%	0.0	-
325	FREE_TIME	FLOAT	4	D	RO	0 - 100%	0.0	-
326	SHED_RCAS	USIGN32	4	S	RW	(Unit 1/32 ms)	640000 (20 sec)	-
327	SHED_ROUT	USIGN32	4	S	RW	(Unit 1/32 ms)	640000 (20 sec)	-
328	FAULT_STATE	USIGN8	1	N	RO	1: Clear 2: Active	Clear	-
329	SET_FSTATE	USIGN8	1	D	RW	1: Off 2: Set	Off	-
330	CLR_FSTATE	USIGN8	1	D	RW	1: Off 2: Clear	Off	-
331	MAX_NOTIFY	USIGN8	1	S	RO	Set by manufacturer	8	-
332	LIM_NOTIFY	USIGN8	1	S	RW	0 to MAX_NOTIFY	MAX_NOTIFY	0x0005
333	CONFIRM_TIME	USIGN32	4	S	RW	(Unit 1/32 ms)	640000 (20 sec)	0x0005
334	WRITE_LOCK	USIGN8	1	S	RW	1: Unlocked 2: Locked	Unlocked	0x0005
335	UPDATE_EVT	DS-73	14	D	RO	-	0,0,0,0,0,0,9,0	-

336	BLOCK_ALM	DS-72	13	D	RW	-	0,0,0,0,0,0,0,8,0,0	0x0007
337	ALARM_SUM	DS-74	8	Mix	RW	-	0,0,0,0	-
338	ACK_OPTION	BSTRING	2	S	RW	0: Auto ack disabled 1: Auto ack enabled	Auto ack disabled	-
339	WRITE_PRI	USIGN8	1	S	RW	0 to 15	0	0x0005
340	WRITE_ALM	DS-72	13	D	RW	-	0,0,0,0,0,0,10,0,0	0x0007
341	ITK_VER	USIGN16	2	S	RO	Set by FF	4	-
Manufacturer-specific Parameter defined by Foxboro Eckardt								
342	TARGET_ERROR	Array of USIGN16	12	D	RO	-	0,0,0,0,0,0	-
343	DIAGNOSIS	Array of USIGN8	6	D	RO	-	0,0,0,0,0,0	-
344	RESET_HIST_STATUS	USIGN8	1	D	RW	0: Uninitialized 1: Clear history status 2: Clear all device status	Uninitialized	0x0005
345	FACTORY_RESET	USIGN8	1	D	RW	0: Uninitialized 1: Create fac setting 2: Restore fac setting	Uninitialized	0x0003, 0x0005
346	SOFTWARE_REVISION	VSTRING	16	N	RO	Format xx.yy	1.00	-
347	HARDWARE_REVISION	VSTRING	16	N	RO	Format xx	2	-
348	MODEL_CODE	VSTRING	32	S	RW	-	Spaces	0x0003
349	DEVICE_SER_NUM	VSTRING	16	S	RW	-	“00/000000”	0x0003
350	LOCAL_OP_ENA	USIGN8	1	S	RW	0: Uninitialized 1: Local op enable 2: Local op disable	Local op enable	0x0005
351	LOCAL_KEYS_CTRL	USIGN8	1	S	RW	0: Uninitialized 1: All keys enalbe 2: Span keys disable 3: Span & zeropoint keys diasble 4. All keys disable	All keys enable	0x0005
352	DISPLAY_MODE	USIGN8	1	S	RW	0: Uninitialized 1: NONE 2: PRV 3: OUT	OUT	0x0005
353	PASSWD_CTRL	USIGN8	1	S	RW	0: Uninitialized 1: Off 2: On	Off	0x0003, 0x0005
354	PASSWD	VSTRING	6	S	WO	-	“WKSHOP”	0x0009
355	MESSAGE_1	VSTRING	32	S	RW	-	Spaces	-
356	MESSAGE_2	VSTRING	32	S	RW	-	Spaces	-
357	MESSAGE_3	VSTRING	32	S	RW	-	Spaces	-
358	MESSAGE_4	VSTRING	32	S	RW	-	Spaces	-
359	MESSAGE_5	VSTRING	32	S	RW	-	Spaces	-
360 - 361	Unused/Reserved	-	-	-	-	-	-	-
Function Block								
Standard Parameter								
362	BLK_DATA	DS-64	62	S	RW	-	-	-
363	ST_REV	USIGN16	2	S	RO	-	0	-
364	TAG_DESC	OSTRING	32	S	RW	-	Spaces	-
365	STRATEGY	USIGN16	2	S	RW	-	0	-
366	ALERT_KEY	USIGN8	1	S	RW	1 to 255	0	0x0005
367	MODE_BLK	DS-69	4	Mix	RW	O/S, Auto, Man	O/S	0x0006
368	BLOCK_ERR	USIGN16	2	D	RO	-	-	-
369	PV	DS-65	5	D	RO	-	-	-
370	OUT	DS-65	5	N	ROW	-	-	0x0008
371	SIMULATE	DS-82	11	D	RW	-	Disable	0x2001
372	XD_SCALE	DS-68	11	S	RW	-	100.0, 0.0, 1342, 3	0x0008
373	OUT_SCALE	DS-68	11	S	RW	-	100.0, 0.0, 1342, 3	0x0008

374	GRANT_DENY	DS-70	2	D	RW	-	0, 0	-
375	IO_OPTS	USIGN16	2	S	RW	Bit 10: Low cutoff	0	0x0008
376	STATUS_OPTS	USIGN16	2	S	RW	Bit 3: Propagate fault forward 6: Uncertain if limited 7: BAD if limited 8: Uncertain in Man Mode	0	0x0008
377	CHANNEL	USIGN16	2	S	RW	1 to Mfgr limit	0	0x0005, 0x0008
378	L_TYPE	USIGN8	1	S	RW	1: Direct 2: Indirect 3: Ind sqr root	0	0x0005, 0x0008
379	LOW_CUT	FLOAT	4	S	RW	Non-negative	0	0x0005
380	PV_FTIME	FLOAT	4	S	RW	Non-negative	0	0x0005
381	FIELD_VAL	DS-65	5	D	RO	-	-	-
382	UPDATE_EVT	DS-73	14	D	RO	-	0,0,0,0,0,0,9,0	-
383	BLOCK_ALM	DS-72	18	D	RW	-	0,0,0,0,0,0,0,8,0,0	0x0007
384	ALARM_SUM	DS-74	8	Mix	RW	-	0,0,0,0	-
385	ACK_OPTION	USIGN16	2	S	RW	0: Auto ack disabled 1: Auto ack enabled	Auto ack disabled	-
386	ALARM_HYS	FLOAT	4	S	RW	0 to 50%	0.5	0x0005
387	HI_HI_PRI	USIGN8	1	S	RW	0 to 15	0	0x0005
388	HI_HI_LIM	FLOAT	4	S	RW	PV_SCALE, +INF	+INF	-
389	HI_PRI	USIGN8	1	S	RW	0 to 15	0	0x0005
390	HI_LIM	FLOAT	4	S	RW	PV_SCALE, +INF	+INF	-
391	LO_PRI	USIGN8	1	S	RW	0 to 15	0	0x0005
392	LO_LIM	FLOAT	4	S	RW	-INF, PV_SCALE	-INF	-
393	LO_LO_PRI	USIGN8	1	S	RW	0 to 15	0	0x0005
394	LO_LO_LIM	FLOAT	4	S	RW	-INF, PV_SCALE	-INF	-
395	HI_HI_ALM	DS-71	16	D	RW	-	0,0,0,0,0,0,0,4,0,0	0x0007
396	HI_ALM	DS-71	16	D	RW	-	0,0,0,0,0,0,0,2,0,0	0x0007
397	LO_ALM	DS-71	16	D	RW	-	0,0,0,0,0,0,0,1,0,0	0x0007
398	LO_LO_ALM	DS-71	16	D	RW	-	0,0,0,0,0,0,0,3,0,0	0x0007
Manufacturer-specific Parameter defined by Foxboro Eckardt								
399	Unused/Reserved	-	-	-	-	-	-	-
Transducer Block								
Standard Parameter								
400	BLK_DATA	DS-64	62	S	RW	-	-	-
401	ST_REV	USIGN16	2	S	RO	-	0	-
402	TAG_DESC	OSTRING	32	S	RW	-	Spaces	-
403	STRATEGY	USIGN16	2	S	RW	-	0	-
404	ALERT_KEY	USIGN8	1	S	RW	1 to 255	0	0x0005
405	MODE_BLK	DS-69	4	Mix	RW	O/S, Auto, Man	O/S	0x0006
406	BLOCK_ERR	BSTRING	2	D	RO	-	-	-
407	UPDATE_EVT	DS-73	14	D	RO	-	0,0,0,0,0,0,9,0	-
408	BLOCK_ALM	DS-72	13	D	RW	-	0,0,0,0,0,0,0,8,0,0	0x0007
409	TRANSDUCER_DIRECTORY	Array of USIGN16	4	N	RO	-	1,10	-
410	TRANSDUCER_TYPE	USIGN16	2	N	RO	-	100	-
411	XD_ERROR	USIGN8	1	D	RO	-	0	-
412	COLLECTION_DIRECTORY	Array of USIGN32	8	N	RO	-	1,13,0x80020380	-
413	PRIMARY_VALUE_TYPE	USIGN16	2	S	RW	-	110 (Level)	0x0003, 0x0008
414	PRIMARY_VALUE	DS-65	5	D	RO	-	-	-
415	PRIMARY_VALUE_RANGE	DS-68	11	S (N)	RW	-	100.0, 0.0, 1342, 3	0x0003, 0x0008
416	CAL_POINT_HI	FLOAT	4	S	RW	-	+INF	0x0003, 0x0008, 0x3001, 0x3002, 0x3003,

								0x3004, 0x3005
417	CAL_POINT_LO	FLOAT	4	S	RW	-	-INF	0x0003, 0x0008, 0x3001, 0x3002, 0x3003, 0x3004, 0x3005
418	CAL_MIN_SPAN	FLOAT	4	N	RO	-	0.0	-
419	CAL_UNIT	USIGN16	2	S	RW	-	1342 (%)	0x0003, 0x0008, 0x3000
420	SENSOR_TYPE	USIGN16	2	S	RW	-	121 (Pressure sensor unknown)	0x0003, 0x0008
421	SENSOR_RANGE	DS-68	11	S (N)	RW	-	19.613, 0.0, 1120, 3	0x0003, 0x0008
422	SENSOR_SN	VSTRING	32	N	RO	-	Spaces	-
423	SENSOR_CAL_METHOD	USIGN8	1	S	RW	-	103 (Factory trim standard calib.)	0x0003, 0x0005, 0x0008
424	SENSOR_CAL_LOC	VSTRING	32	S	RW	-	Spaces	0x0003, 0x0008
425	SENSOR_CAL_DATE	DS-11	7	S	RW	-	0,0,0,1,1,2	0x0003, 0x0008
426	SENSOR_CAL_WHO	VSTRING	32	S	RW	-	Spaces	0x0003, 0x0008
427	SENSOR_ISOLATOR_MTL	USIGN16	2	N	RO	To be defined by FF	0	-
428	SENSOR_FILL_FLUID	USIGN16	2	N	RO	To be defined by FF	0	-
429	SECONDARY_VALUE	FLOAT	4	D	RO	-	-	-
430	SECONDARY_VALUE_UNIT	USIGN16	2	S	RW	-	1001 (= °C)	-
Manufacturer-specific Parameter defined by Foxboro Eckardt								
431	TARGET_ERROR	Array of USIGN16	12	D	RO	-	0,0,0,0,0,0	-
432	LINE_FREQUENCY	USIGN8	1	S	RW	0: Uninitialized 1: 50Hz 2: 60Hz 3: Special	50Hz	0x0005
433	SENSOR_ID	Array of VSTRING	7	S	RO	-	“DEFAU ”	-
434	SENSOR_SUBTYPE	USIGN8	1	S	RO	-	6 (144LD)	-
435	SENSOR_VALUE	FLOAT	4	D	RO	-	-	-
436	REL_SENSOR_VALUE	FLOAT	4	D	RO	-	-	-
437	SMART_SMOOTH	Array of INT16 & FLOAT	8	S	RW	-	10,0,2.0	0x0003, 0x0005
438	SENSOR_ZERO_TRIM	USIGN8	1	D	RW	0: Uninitialized 1: Sen zero pt trim 2: Sen zero pt reset	Uninitialized	0x0003, 0x0005, 0x0008, 0x3004, 0x3005
439	USER_ZERO_TRIM	USIGN8	1	D	RW	0: Uninitialized 1: User zero pt trim 2: User zero pt reset	Uninitialized	0x0005, 0x0008, 0x3004, 0x3005
440	MAX_SENSOR_VALUE	FLOAT	4	D	RO	-	-INF	-
441	MIN_SENSOR_VALUE	FLOAT	4	D	RO	-	+INF	-
442	MAX_SENSOR_TEMPERATURE	FLOAT	4	D	RO	-	-INF	-
443	MIN_SENSOR_TEMPERATURE	FLOAT	4	D	RO	-	+INF	-
444	ELECTRONICS_TEMP	FLOAT	4	D	RO	-	-	-
445	ELECTRONICS_TEMP_UNIT	USIGN16	2	S	RW	-	1001 (= °C)	-
446	ELECTRONICS_PRODUCTION_NO	USIGN16	2	N	RO	-	0	-

447	ELECTRONICS_CAL_DATE	DS-11	7	N	RO	-	0,0,0,1,1,2	-
448	LIN_TYPE	USIGN8	1	S	RW	0: Uninitialized 1: Linear with input 2: Square root 3: Linearisation table	Linear with input	0x0005, 0x0008
449	TAB_ENTRY	USIGN8	1	D	RW	1 to 32	1	0x0005
450	TAB_X_Y_VALUE	Array of FLOAT	8	D	RW	-	0.0, 0.0	0x0008
451	TAB_MIN_NUMBER	USIGN8	1	N	RO	-	2	-
452	TAB_MAX_NUMBER	USIGN8	1	N	RO	-	32	-
453	TAB_OP_CODE	USIGN8	1	S	RW	0: Unintialized 1: Load new 2: End of transmission	Uninitialized	0x0005, 0x0008
454	TAB_STATUS	USIGN8	1	D	RO	0: Uninitialized 1: Good 2: Not monotonous inc 3: Not enough values 4: Table is currently loaded	Uninitialized	-
455	TAB_ACTUAL_NUMBER	USIGN8	1	N	RO	1 to 32	2	-
456	SENSOR_BASIC_DATA	Array of USIGN8 & FLOAT	39	N	RO	-	-	-
457	SPECIAL_UNIT_OUT	VSTRING	6	S	RW	-	Spaces	-
458 - 459	Unused/Reserved	-	-	-	-	-	-	-
Link Objects								
Standard Parameter								
460	FB_LINK01	DS-81	8	N	RW	-	-	-
461	FB_LINK02	DS-81	8	N	RW	-	-	-
462	FB_LINK03	DS-81	8	N	RW	-	-	-
463	FB_LINK04	DS-81	8	N	RW	-	-	-
464	FB_LINK05	DS-81	8	N	RW	-	-	-
465	FB_LINK06	DS-81	8	N	RW	-	-	-
466	FB_LINK07	DS-81	8	N	RW	-	-	-
467	FB_LINK08	DS-81	8	N	RW	-	-	-
468	FB_LINK09	DS-81	8	N	RW	-	-	-
469	FB_LINK010	DS-81	8	N	RW	-	-	-
470	FB_LINK011	DS-81	8	N	RW	-	-	-
471	FB_LINK012	DS-81	8	N	RW	-	-	-
472	FB_LINK013	DS-81	8	N	RW	-	-	-
473	FB_LINK014	DS-81	8	N	RW	-	-	-
474	FB_LINK015	DS-81	8	N	RW	-	-	-
475	FB_LINK016	DS-81	8	N	RW	-	-	-
476	FB_LINK017	DS-81	8	N	RW	-	-	-
477	FB_LINK018	DS-81	8	N	RW	-	-	-
478	FB_LINK019	DS-81	8	N	RW	-	-	-
479	FB_LINK020	DS-81	8	N	RW	-	-	-
480	FB_LINK021	DS-81	8	N	RW	-	-	-
481	FB_LINK022	DS-81	8	N	RW	-	-	-
482 - 489	Unused/Reserved	-	-	-	-	-	-	-
Alert Objects								
Standard Parameter								
490	ALERT_FLT01	DS-75	25	N	RW	-	-	-
491	ALERT_DIS01	DS-76	22	N	RW	-	-	-
492	ALERT_EVT01	DS-77	19	N	RW	-	-	-
493 – 499	Unused/Reserved	-	-	-	-	-	-	-
Trend Objects								
Standard Parameter								
500	TREND_FLT01	DS-78	97	N	RW	-	-	-

501 - 509	Unused/Reserved	-	-	-	-	-	-	-
View Objects								
Standard Parameter								
510	VIEW_1	List of EI	22	D	RO	-	-	-
511	VIEW_2	List of EI	30	D	RO	-	-	-
512	VIEW_3	List of EI	22	D	RO	-	-	-
513	VIEW_4	List of EI	31	D	RO	-	-	-
514 - 519	Unused/Reserved	-	-	-	-	-	-	-
520	VIEW_1	List of EI	31	D	RO	-	-	-
521	VIEW_2	List of EI	26	D	RO	-	-	-
522	VIEW_3	List of EI	31	D	RO	-	-	-
523	VIEW_4	List of EI	46	D	RO	-	-	-
524 - 529	Unused/Reserved	-	-	-	-	-	-	-
530	VIEW_1	List of EI	21	D	RO	-	-	-
531	VIEW_2	List of EI	16	D	RO	-	-	-
532	VIEW_3	List of EI	21	D	RO	-	-	-
533	VIEW_4	List of EI	117	D	RO	-	-	-
534	Unused/Reserved	-	-	-	-	-	-	-

2.2 Description of Parameters

A parameter is identified by its index in the OD. Index 0, called the object dictionary header, provides a description of the dictionary itself, and defines the first index for the object descriptions of the FB application. The object description of 240FF starts at the index 298.

2.2.1 OD Object Description

The OD object description is part of the device management, which provides a guide to all objects within the device [5]. An interface device such as a host desiring to access the objects described by the OD may read the information contained in this description. Besides, by reading the OD object description the structure description and the version number of the OD are made available. The OD object description has the following structure:

Name	Description	Size	Range	Value
Obj_code	Object code	USIGN16	1 = OD Object	1
Flag	Configure write protection	USIGN16	1 = no write protection 0 = write protection	0
Length	Size of names	USIGN16	0 to 32	32
Protection	Access protection	USIGN16	0 = no access protection 1 = access protection	0
Version	OD version	USIGN16		1
Int_addr	Internal address of OD	USIGN32		
Len_st_od	Length of S_OD	USIGN16		199
Int_addr_st_od	Internal address of S_OD	USIGN32		
First_index_s_od	Start index of ST_OD	USIGN16		298
Len_s_od	Length of ST_OD	USIGN16		212
Int_addr_s_od	Internal address of ST_OD	USIGN32		
First_index_dv_od	Start index of DV_OD	USIGN16		510
Len_dv_od	Length of DV_OD	USIGN16		25
Int_addr_dv_od	Internal address of DV_OD	USIGN32		
First_index_dp_od	Start index of DP_OD	USIGN16		0
Len_dp_od	Length of DP_OD	USIGN16		0
Int_addr_dp_od	Internal address of DP_OD	USIGN32		

Legends:

OD Object Dictionary

S_OD Static Object Dictionary which includes Action Object (not used in this application), Resource Block, Function Block(s), Transducer Block(s), Link Objects, Alert Objects, Trend Objects, Domain Object (not used in this application)

ST_OD Standard Object Dictionary

DV_OD Dynamic List of Variable Lists (View Object(s))

DP_OD Dynamic List of Program Invocations (Program Invocation Objects, not used in this application)

2.2.2 Data Types and Data Structures

Object indices 1 to 255 are reserved for standard Foundation data types and data structures. This portion of the OD is listed below. The C-type names of the various data types and structures are listed on the last column.

Index	Data Type	Name	C-Type
1	Data	Boolean	BOOL
2	Data	Integer 8	INT8
3	Data	Integer 16	INT16
4	Data	Integer 32	INT32
5	Data	Unsigned 8	USIGN8
6	Data	Unsigned 16	USIGN16
7	Data	Unsigned 32	USIGN32
8	Data	Floating Point	FLOAT
9	Data	Visible String	VSTRING
10	Data	Octet String	OSTRING
11	Data	Date	DATE_S
12	Data	Time of Day	TIME_OF_DAY_S
13	Data	Time Difference	TIME_DIFF_S
14	Data	Bit String	BSTRING
…			
21	Data	Time Value	TIME_VALUE_S
…			
64	Structure (DS-64)	Block	F_BLOCK
65	Structure (DS-65)	Value & Status - Float	FLOAT_S
66	Structure (DS-66)	Value & Status - Discrete	DISCRETE_S
67	Structure (DS-67)	Value & Status - Bitstring	BIT_STRING_S
68	Structure (DS-68)	Scaling	SCALE
69	Structure (DS-69)	Mode	MODE
70	Structure (DS-70)	Access Permissions	ACCESS_PERM
71	Structure (DS-71)	Alarm - Float	ALARM_FLOAT
72	Structure (DS-72)	Alarm - Discrete	ALARM_DISCRETE
73	Structure (DS-73)	Event - Update	EVENT
74	Structure (DS-74)	Alarm - Summary	ALARM_SUMMARY
75	Structure (DS-75)	Alert - Analog	ALERT_FLOAT
76	Structure (DS-76)	Alert - Discrete	ALERT_DISCRETE
77	Structure (DS-77)	Alert - Update	ALERT_EVENT
78	Structure (DS-78)	Trend - Float	TREND_FLOAT
79	Structure (DS-79)	Trend - Discrete	TREND_DISCRETE
80	Structure (DS-80)	Trend - Bitstring	TREND_BIT_STRING
81	Structure (DS-81)	FB Link	FB_LINK
82	Structure (DS-82)	Simulate - Float	SIMULATE_FLOAT
83	Structure (DS-83)	Simulate - Discrete	SIMULATE_DISCRETE
84	Structure (DS-84)	Simulate - Bitstring	SIMULATE_BIT_STRING
85	Structure (DS-85)	Test	TEST
86	Structure (DS-86)	Action - Instantiate/Delete	ACTION

Following the standard foundation data types and structures are the manufacturer-specific data types and structures, which reserve the indices 256 to 297. The 240FF has seven manufacturer-specific data structures defined:

Index	Data Type & Structure	Name	C-Type
256	Structure (fox_cmd)	Foxcom command	FOX_CMD
257	Structure (fox_rsp)	Foxcom response	FOX_RSP
258	Structure (diagnosis)	Diagnosis	DIAGNOSIS_FIELD
259	Structure (target_error)	Target error	TARGET_ERROR_FIELD
260	Structure (smart_smooth)	Smart smoothing	SMART_SMOOTHING
261	Structure (tab_x_y_value)	One couple table value	XY_VALUE
262	Structure (sen_basic_data)	Collection of sensor basic data	SEN_BASIC_DATA

2.2.3 AP Directory

The AP (Application Process) Directory contains the OD indices for the network visible objects of an AP (see [1], Section 4.2). The AP Directory is assigned with the index 298 which can be read from the “First_index_s_od“ entry of the OD Description Header. The AP Directory is defined as an array of constant Unsigned 16 values with its first six entries as the AP Directory Header. All entries that follow the header are references each of which is composed of two Unsigned 16 entries. For AP object references the first is the OD index of the AP object. For composite object references, it is the OD index of the first AP object in the composite. In both cases, the second entry is the number of consecutive OD entries for the object [1].

Name	Description	Value
Reserved	Reserved by FB Specification	0xFF
Rev_no	Directory revision number	2
No_of_dir_obj	Total number of directory objects	1
No_of_dir_entry	Total number of directory entries	9+1+1
First_comp_entry	Index of first composite	17
No_of_comp_entry	Number of composite entries	3
Action_start	Starting index of Action Objects	0
Action_count	Number of Action Objects	0
Link_start	Starting index of Link Objects	460
Link_count	Number of Link Objects	22
Alert_start	Starting index of Alert Objects	490
Alert_count	Number of Alert Objects	3
Trend_start	Starting index of Trend Objects	500
Trend_count	Number of Trend Objects	1
Domain_start	Starting index of Domain Objects	0
Domain_count	Number of Domain Objects	0
Rsc_ap_index	Resource Block AP directory index	23
Rsc_blk_count	Number of Resource Blocks	1
Tran_ap_index	Transducer Block AP directory index	25
Tran_blk_count	Number of Transducer Blocks	1
Func_ap_index	Function Block AP directory index	25+2*1
Func_blk_count	Number of Function Blocks	1
Rsc_od_index	Resource Block starting OD index	300
Rsc_blk_count	Number of objects in Resource Block	60
Trans_dir_blk_od_index1	Transducer Block 1 starting OD index	400
Trans_dir_blk_od_count1	Number of objects in Transducer Block 1	58
Fb_dir_blk_od_index1	Function Block 1starting OD index	362
Fb_dir_blk_od_count1	Number of objects in Function Block 1	37

2.2.4 Block Data Object (DS-64)

Block Data Object BLK_DATA is a universal parameter for all blocks. It is a data structure defined under index 64 and consists of the attributes of the related block. It has 13 elements defined:

Element Name	Data Type	Size	Values for Resource Block	Values for AI-Block	Values for Transducer Block
Block Tag	VSTRING	32	240FF_RES$<fab#>	240FF_AI$<fab#>	240FF_TD$<fab#>
DD Character ID	USIGN32	4	0x80020AF5	0x800201D7	0x80020639
DD Item ID	USIGN32	4	0x80020AF0	0x800201D0	0x80020630
DD Revision	USIGN16	2	1	1	1
Profile	USIGN16	2	0x133	0x101	0x115
Profile Revision	USIGN16	2	0x101	0x101	0x101
Execution Time	USIGN32	4	0	960 [1/32 ms]	0
Period of Execution	USIGN32	4	0	32000 [1/32 ms]	0
Number of Parameters	USIGN16	2	60	37	58
Next FB to Execute	USIGN16	2	0	0	0
Starting index of Views	USIGN16	2	510	520	530
Number of VIEW_3	USIGN8	1	1	1	1
Number of VIEW_4	USIGN8	1	1	1	2

<fab#> is the fabrication number of the device which gets its first value from the device serial number parameter (see the parameter with index 349 in Section 2.1)

2.2.5 Block Parameter Description

Parameter Name	Description
Universal Parameter for Resource, Function and Transducer Blocks	
ST_REV	The revision level of the static data associated with the block. To support tracking changes in static parameter attributes, the associated block's static revision parameter will be incremented each time a static parameter attribute value is changed. Also, the associated block's static revision parameter may be incremented if a static parameter attribute is written but the value is not changed.
TAG_DESC	The user description of the intended application of the block.
STRATEGY	The strategy field can be used to identify grouping of blocks. This data is not checked or processed by the block.
ALERT_KEY	The identification number of the plant unit. This information may be used in the host for sorting alarms, etc.
MODE_BLK	Data structure of type DS-69 (see [9], Part 1, Section 5.13.6) containing the actual, target, permitted, and normal modes of the block.
BLOCK_ERR	This parameter reflects the error status associated with the hardware or software components associated with a block. It is a bit string, so that multiple errors may be shown.
UPDATE_EVT	Data structure of type DS-73 (see [9], Part 1, Section 5.13.10). This alert is generated by any change to the static data.
BLOCK_ALM	Data structure of type DS-72 (see [9], Part 1, Section 5.13.9) holding the block alarm from all configuration, hardware, connection failure or system problems in the block. The cause of the alert is entered in the subcode field. The first alert to become active will set the Active status in the Status attribute. As soon as the Unreported status is cleared by the alert reporting task, another block alert may be reported without clearing the Active status, if the subcode has changed.
TARGET_ERROR	Manufacturer-specific parameter. It is a data structure of 6 read-only successive Unsigned 16 elements used to hold the access errors to block parameters. Each time a parameter is written, if error exists, the first element will be updated with the error reason, during which the old content of the element will not be discarded but shifted to the second element, and the second to the third and so on. If no error is detected, the first element will be updated to 0 after a new read access. So two error reports are normally separated by one non-error report, and totally 3 errors can be shown in this parameter. Note that the resource block and the function block use the same TARGET_ERROR parameter in the resource block.
Universal Parameter for Resource and Function Blocks	
GRANT_DENY	Data structure of type DS-70 (see [9], Part 1, Section 5.13.7) containing options for controlling access of host computer and local control panels to operating, tuning and alarm parameters of the block.
ACK_OPTION	Selection of whether alarms associated with the block will be automatically acknowledged.
ALARM_SUM	Data structure of type DS-74 (see [9], Part 1, Section 5.13.11) containing the current alert status, unacknowledged states, unreported states, and disabled states of the alarms associated with the block.
Parameter for Resource Block	
RS_STATE	State of the function block application state machine.

TEST_RW	This is a data structure of type DS-85 (see [9] Part 1, Section 5.13.22), a test parameter used only for read/write conformance testing.
DD_RESOURCE	String identifying the tag of the resource that contains the Device Description (DD) for this resource.
MANUFAC_ID	Manufacturer identification number - used by an interface device to locate the DD file for the resource.
DEV_TYPE	Manufacturer's model number associated with the resource - used by interface devices to locate the DD file for the resource.
DEV_REV	Manufacturer revision number associated with the resource - used by an interface device to locate the DD file for the resource.
DD_REV	Revision of the DD associated with the resource - used by an interface device to locate the DD file for the resource.
HARD_TYPES	The types of hardware available as channel numbers (see [9], Part 2, Section 5.4.1).
RESTART	Allows a manual restart to be initiated. Several degrees of restart are possible. They are 1: Run, 2: Restart resource, 3: Restart with defaults, and 4: Restart processor.
FEATURES	Used to show supported resource block options.
FEATURES_SEL	Used to select resource block options.
MIN_CYCLE_T	Time duration of the shortest cycle interval of which the resource is capable.
MEMORY_SIZE	Available configuration memory in the empty resource. To be checked before attempting a download.
NV_CYCLE_T	Minimum time interval specified by the manufacturer for writing copies of NV parameters to non-volatile memory. Zero means it will never be automatically copied. At the end of NV_CYCLE_TIME, only those parameters that have changed (as defined by the manufacturer) need to be updated in NVRAM.
FREE_SPACE	Percent of memory available for further configuration. Zero in a preconfigured resource.
FREE_TIME	Percent of the block processing time that is free to process additional blocks.
SHED_RCAS	Time duration at which to give up on computer writes to function block RCas locations. Shed from RCas shall never happen when SHED_RCAS = 0.
SHED_ROUT	Time duration at which to give up on computer writes to function block ROut locations. Shed from Rout shall never happen when SHED_ROUT = 0.
FAULT_STATE	Condition set by loss of communication to an output block, fault promoted to an output block or a physical contact. When Fault State condition is set, Then output function blocks will perform their FSTATE actions.
SET_FSTATE	Allows the Fault State condition to be manually initiated by selecting Set.
CLR_FSTATE	Writing a Clear to this parameter will clear the device fault state if the field condition, if any, has cleared.
MAX_NOTIFY	Maximum number of unconfirmed notify messages possible.
LIM_NOTIFY	Maximum number of unconfirmed alert notify messages allowed.
CONFIRM_TIME	The time the resource will wait for confirmation of receipt of a report before trying again. Retry shall not happen when CONFIRM_TIME = 0.
WRITE_LOCK	If set, no writes from anywhere are allowed, except to clear WRITE_LOCK. Block inputs will continue to be updated.
WRITE_PRI	Priority of the alarm generated by clearing the write lock.
WRITE_ALM	This alert is generated if the write lock parameter is cleared.
ITK_VER	Major revision number of the interoperability test case used in certifying this device as interoperable. The format and range of the version number is defined and controlled by the Fieldbus Foundation. Note: The value of this parameter will be zero (0) if the device has not been registered as interoperable by the FF.
DIAGNOSIS	Manufacturer-specific parameter containing detailed information of the device. It is a data structure of six successive Unsigned 8 values, bitwise coded. The first three values hold the current status of warnings, process and system, respectively. The next three values collect the history of the first three values.
RESET_HIST_STATUS	Manufacturer-specific parameter used to 1: Clear device history status, or 2: Clear all device status (i.e. both the actual and history status including all those stored in the EEPROM).
FACTORY_RESET	Manufacturer-specific parameter. It is used to 1: Create a factory setting, and 2: Restore the device to its factory setting state. The first option is a factory procedure protected by a password to unauthorized user.
SOFTWARE_REVISION	Manufacturer-specific parameter holding the revision number of the software of 240FF in format <xx.yy>.
HARDWARE_REVISION	Manufacturer-specific parameter holding the revision number for the hardware of 240FF.
MODEL_CODE	Manufacturer-specific parameter holding the model code of the device given by the manufacturer.
DEVICE_SER_NUM	Manufacturer-specific parameter containing the serial number (fabrication number) of the field device in format <xx/yyyyyy>.
LOCAL_OP_ENA	Manufacturer-specific parameter. It is used to enable (1) as well as to disable (2) the local operation in 240FF done by using the two menu buttons below the LCD-display or the two external keys.

LOCAL_KEYS_CTRL	Manufacturer-specific parameter used to lock or unlock the configurations selected by the two external keys of the device. To see how do the external keys work refer to [11].
DISPLAY_MODE	Manufacturer-specific parameter used to select the display content of the LCD-display, i.e. OUT, PRIMARY_VALUE (PRV) or nothing (NONE).
PASSWD_CTRL	Manufacturer-specific parameter used to work together with PASSWD to control the access of factory procedures.
PASSWD	Manufacturer-specific parameter holding a password used to get the access permission to all factory procedures.
MESSAGE_1	Manufacturer-specific parameters used to hold user-definable messages for the description of the device within the application or in the plant.
MESSAGE_2	
MESSAGE_3	
MESSAGE_4	
MESSAGE_5	
Parameter for Function Block	
PV	Either the primary analog value for use in executing the function, or a process value associated with it. It is a data structure of type DS-65 (see [9] Part 1, Section 5.13.2) containing a status byte followed by a floating value.
OUT	The primary analog value calculated as a result of executing the function. It is a data structure of type DS-65 (see [9] Part 1, Section 5.13.2).
SIMULATE	The analog input of the Function Block is supplied in general by the transducer output. This parameter allows the analog input of the Function Block to be manually supplied when simulate is enabled. When simulation is disabled, the simulate value and status track the actual value and status. This parameter is data structure of type DS-82 (see [9] Part 1, Section 5.13.19).
XD_SCALE	This is a data structure of type DS-68 (see [9] Part 1, Section 5.13.5) with four elements holding the high and low scale values, engineering units code, and number of digits to the right of the decimal point used with the value obtained from the transducer for a specified channel.
OUT_SCALE	This is a data structure of type DS-68 (see [9] Part 1, Section 5.13.5) holding the high and low scale values, engineering units code, and number of digits to the right of the decimal point to be used in displaying the OUT parameter and parameters which have the same scaling as OUT. Notice that the enginnering units code in OUT_SCALE is extended to include the code 1999 as a special code, i.e. besides all standard units provided by FF, the units of OUT can further be set to “special”, in which case the SPECIAL_UNIT_OUT parameter contains the corresponding units text. See also SPECIAL_UNIT_OUT.
IO_OPTS	Contains options that the user may select to alter input and output block processing (refer to [9] Part 2, Section 5.4.2.1). The options are bitwize coded. For AI Function Block, there is only one option “Low cutoff” defined by FF.
STATUS_OPTS	Contains options that the user may select in the block processing of status (refer to [9] Part 2, Section 5.4.2.3). For this parameter, the 240FF supports the options listed in the parameter table.
CHANNEL	The number of the logical hardware channel that is connected to this I/O block. This information defines the transducer to be used going to or from the physical world.
L_TYPE	Determines if the values passed by the transducer block to the AI block may be used directly (Direct) or if the value is in different units and must be converted linearly (Indirect), or with square root (Ind Sqr Root), using the input range defined by the transducer and the associated output range.
LOW_CUT	Limit used in square root processing if the “Low cutoff” option in IO_OPTS is selected. A value of zero percent of scale is used in block processing if the transducer value falls below this limit, in % of scale. This feature may be used to eliminate noise near zero for a flow sensor.
PV_FTIME	Time constant of a single exponential filter for the PV, in seconds.
FIELD_VAL	Raw value of the field device in percent of the PV range, with a status reflecting the Transducer condition, before signal characterization (L_TYPE) or filtering (PV_FTIME).
ALARM_HYS	Amount the PV must return within the alarm limits before the alarm condition clears. Alarm Hysteresis is expressed as a percent of the PV span.
HI_HI_PRI	Priority of the high high alarm.
HI_HI_LIM	The setting for high high alarm in engineering units.
HI_PRI	Priority of the high alarm.
HI_LIM	The setting for high alarm in engineering units.
LO_PRI	Priority of the low alarm.
LO_LIM	The setting for the low alarm in engineering units.
LO_LO_PRI	Priority of the low low alarm.
LO_LO_LIM	The setting of the low low alarm in engineering units.
HI_HI_ALM	The status for high high alarm and its associated time stamp.
HI_ALM	The status for high alarm and its associated time stamp.
LO_ALM	The status of the low alarm and its associated time stamp.
LO_LO_ALM	The status of the low low alarm and its associated time stamp.

Parameter for Transducer Block	
TRANSDUCER_DIRECTORY	In general it is an array of Unsigned 16 values with variable element number (see [10] Part 1, Section 3.4.7) where the first element contains the number of transducer definitions and each successive element contains the index of the beginning of the n-th transducer definition starting with the first definition. Each definition begins with the TRANSDUCER_TYPE parameter for that definition. Because 240FF has only one transducer block (definition), this array used for 240FF has two elements where the first element, i.e. the number of transducer definition is equal to 1 and the second element contains the (relative) index of the TRANSDUCER_TYPE parameter in this definition, here equal to 10.
TRANSDUCER_TYPE	TRANSDUCER_TYPE holds the code for this transducer definition defined by Fieldbus Foundation (see [10] Part 2, Section 4.2). The code for 240FF is 100, which means a transducer block of type “Standard Pressure with Calibration”.
XD_ERROR	The BLOCK_ERR bitstring parameter described in FF-890 [9] is not useful for the many errors possible in the transducer block. When a transducer error occurs, BLOCK_ALM should be used to send it out with a time stamp. An additional parameter XD_ERROR must be used to hold the single error subcode that the manufacturer considers most important when one or more errors occur. BLOCK_ERR will have bit 0 set whenever XD_ERROR is non-zero.
COLLECTION_DIRECTORY	Each transducer definition must have exactly one data collection directory that is represented by this parameter. The directory is an array of Unsigned 32 values with variable element number. The first element contains the number of data collections. After this element, the elements are paired off. The first entry of each pair contains the index of the beginning of the next data collection, and the second entry of each pair contains the DD item ID for the data collection referenced in the first entry of the element pair. The 240FF has only one data collection. Therefore, the first element is equal to 1, the first entry of the second element is equal to 13 (the relative index of the parameter PARIMARY_VALUE_TYPE), and the second entry of the second element is 0x80020380 (the DD item ID of PARIMARY_VALUE_TYPE).
PRIMARY_VALUE_TYPE	The type of measurement represented by the primary value. Some examples: Mass Flow, Volumetric Flow, Absolute Pressure and Differential Pressure etc. (see [10] Part 2, Section 4.1). The type for 240FF is Level.
PRIMARY_VALUE	This is a data structure of type DS-65 (see [9] Part 1, Section 5.13.2) and contains the measured value and status available to the Function Block.
PRIMARY_VALUE_RANGE	This is a data structure of type DS-68 (see [9] Part 1, Section 5.13.5) and contains the High and Low range limit values, the engineering units code and the number of digits to the right of the decimal point to be used to display the PRIMARY_VALUE.
CAL_POINT_HI	The highest calibrated value for sensor calibration. *In 240FF, writing this parameter via FF communication starts a high point calibration procedure.*
CAL_POINT_LO	The lowest calibrated value for sensor calibration. *In 240FF, writing this parameter via FF communication starts a low point calibration procedure.*
CAL_MIN_SPAN	The minimum calibration span value allowed. This minimum span information is necessary to ensure that when calibration is done, the two calibrated points (high and low) are not too close together.
CAL_UNIT	The Device Description engineering units code index for the calibration values.
SENSOR_RANGE	This is a data structure of type DS-68 (see [9] Part 1, Section 5.13.5) and contains the High and Low range limit values, the engineering units code and the number of digits to the right of the decimal point for the sensor.
SENSOR_SN	The sensor serial number (fabrication number). It consists of 6 parts of information related to sensor, i.e. sensor ID, sensor device type, sensor serial number, sensor manufacture date, sensor type and sensor sub-type. For example, 2267189/DEFAUB9000120212506 means, 2267189/DEFAU = sensor ID B9 = sensor device type (provided by the Foxboro company) 0001 = sensor serial number 2021 = sensor manufacture date (day/month/year coded in 2 bytes) 25 = sensor type (0x25 = Eckardt DMU) 06 = sensor sub-type (0x6 = 144 LD)
SENSOR_CAL_METHOD	The method of the last sensor calibration. ISO defines several standard methods of calibration (see [10] Part 2, Section 4.5). This parameter is intended to record that method, or if some other method was used.
SENSOR_CAL_LOC	The location of last sensor calibration. This describes the physical location at which the calibration was performed.
SENSOR_CAL_DATE	The date of the last sensor calibration. This is intended to reflect the calibration of that part of the sensor that is usually wetted by the process.
SENSOR_ISOLATOR_MTL	Defines the construction material of the isolating diaphragms. The content of this parameter is still to be defined by FF. When read, therefore, it is always 0. The necessary information however is provided in its copy redefined as one of the elements of SENSOR_BASIC_DATA.
SENSOR_FILL_FLUID	Defines the type of fill fluid used in the sensor. The content of this parameter is still to be defined by FF. When read, therefore, it is always 0. The necessary information however is provided in its copy redefined as one of the elements of SENSOR_BASIC_DATA.

SECONDARY_VALUE	The secondary value, related to the sensor. This is a data structure of type DS-65 (see [9] Part 1, Section 5.13.2). For 240FF it is used for the measurement of the sensor temperature.
SECONDARY_VALUE_UNIT	The engineering units to be used with SECONDARY_VALUE.
LINE_FREQUENCY	Manufacturer-specific parameter used to select the frequency used for the local power standard. NOTE: The new selected frequency is valid only after a new start of the device.
SENSOR_ID	Manufacturer-specific parameter used to hold the name for sensor ID. This information is also shown in SENSOR_SN.
SENSOR_SUBTYPE	Manufacturer-specific parameter used to hold the name for sensor sub-type. This information is also included in SENSOR_SN.
SENSOR_VALUE	Manufacturer-specific parameter to be used to hold the raw sensor value (the value before calibration).
REL_SENSOR_VALUE	Manufacturer-specific parameter to be used to hold the relative sensor value, i.e. sensor value (calibrated) plus an offset.
SMART_SMOOTH	Manufacturer-specific parameter used to provide time constant of a single exponential filter for the sensor value in the so-called deadband (see Section 2.1, the parameter with index 440, e.g., the default time constant is 10 s and the deadband is 2%).
SENSOR_ZERO_TRIM	Manufacturer-specific parameter used to set or reset the zero point in REL_SENSOR_VALUE. If set to 1, REL_SENSOR_VALUE will be zeroed with respect to the current sensor value. If set to 2, REL_SENSOR_VALUE will be restrored to its original form before the zero trim process.
USER_ZERO_TRIM	Manufacturer-specific parameter used to set or reset the zero point in REL_SENSOR_VALUE. The difference of USER_ZERO_TRIM to SENSOR_ZERO_TRIM is that the access of SENSOR_ZERO_TRIM is proteced with password while the access of USER_ZERO_TRIM is not. Furthermore the zeroing process using USER_ZERO_TRIM is based on the zeroed REL_SENSOR_VALUE whose zero point is set by SENSOR_ZERO_TRIM. Therefore, if 2 is written to USER_ZERO_TRIM, i.e., to reset the user zero point, only the zero point set by USER_ZERO_TRIM will be cleared; the zero point set by SENSOR_ZERO_TRIM remains unchanged. In contrast, if 2 is written to SENSOR_ZERO_TRIM, both settings done by SENSOR_ZERO_TRIM and USER_ZERO_TRIM will be cleared.
MAX_SENSOR_VALUE	Manufacturer-specific parameter used to hold the maximum sensor value in a power cycle.
MIN_SENSOR_VALUE	Manufacturer-specific parameter used to hold the minimum sensor value in a power cycle.
MAX_SENSOR_TEMPERATURE	Manufacturer-specific parameter used to hold the maximum sensor temperature in a power cycle.
MIN_SENSOR_TEMPERATURE	Manufacturer-specific parameter used to hold the minimum sensor temperature in a power cycle.
ELECTRONICS_TEMP	Manufacturer-specific parameter. This is a data structure of type DS-65 (see [9] Part 1, Section 5.13.2) and is used to hold the electronics temperature of the device.
ELECTRONICS_TEMP_UNIT	Manufacturer-specific parameter holding the engineering units to be used with ELECTRONICS_TEMP.
ELECTRONICS_PRODUCTION_NO	Manufacturer-specific parameter showing the production number of the device electronics.
ELECTRONICS_CAL_DATE	Manufacturer-specific parameter showing the date on which the electronics of the device is calibrated.
LIN_TYPE	Manufacturer-specific parameter. This parameter is used to extend the options of the standard parameter L_TYPE in Function Block, e.g. to provide a linearisation table which is not supported in L_TYPE.
TAB_ENTRY	Manufacturer-specific parameter to be used to identify which element of the linearisation table is in the TAB_X_Y_VALUE parameter currently.
TAB_X_Y_VALUE	Manufacturer-specific parameter containing one value couple of the linearisation table.
TAB_MIN_NUMBER	Manufacturer-specific parameter. For device internal reasons (e.g. calculation), sometimes it is necessary to use a certain number of table values in minimum. This number is provided in the TAB_MIN_NUMBER parameter and is 2 for 240FF.
TAB_MAX_NUMBER	Manufacturer-specific parameter. This is the maximum size of the linearisation table and is 32 for 240FF.
TAB_OP_CODE	Manufacturer-specific parameter. The modification of a table in a device influences the measurement or actuation algorithms of the device. Therefore an indication of a starting and an end point is necessary. The TAB_OP_CODE controls the transaction of the table. 240FF supports the following transactions: 0: not initialized 1: new operation characteristic, first value (TAB_ENTRY = 1) 2: last value, end of transmission, check table, swap the old curve with the new curve and update TAB_ACTUAL_NUMBER.
TAB_STATUS	Manufacturer-specific parameter. It is common to provide a plausibility check in the table. The result of this check is indicated in the TAB_STATUS parameter: 0: not initialized 1: good (new table is valid) 2: not monotonous increasing (old table is valid) 3: not enough values transmitted (old table is valid)

	4: table is currently loaded, set after TAB_OP_CODE = 1 (additional access to the table not valid, old values are valid)
TAB_ACTUAL_NUMBER	Manufacturer-specific parameter. This parameter contains the actual number of entries in the linearisation table. It should be calculated after the transmission of the table is finished.
SENSOR_BASIC_DATA	Manufacturer-specific parameter used to collect all basic properties of the sensor used for the application concernd. It is data structure of 24 elements, including the copies of the two standard parameters SENSOR_ISOLATOR_MTL and SENSOR_FILL_FLUID. The reason that the two parameters are redefined in this data structure is that the contents of these two standard parameters are still to be defined by FF in the future (see also SENSOR_ISOLATOR_MTL and SENSOR_FILL_FLUID).
SPECIAL_UNIT_OUT	Manufacturer-specific parameter used to hold the units text for OUT if the UNITS_INDEX element in OUT_SCALE is set to “special”.

2.2.6 View Objects

View objects allow groups of function block parameter values to be read with one read request. Such capability will be provided to enable group information to be efficiently communicated in a timely fashion. In agreement with the FF specifications the 240FF has four view objects for each of its 3 blocks.

Resource Block		
VIEW_1	ST_REV, MODE_BLK, BLOCK_ERR, RS_STATE, FREE_TIME, FAULT_STATE, ALARM_SUM	22 Bytes
VIEW_2	ST_REV, GRANT_DENY, FEATURE_SEL, CYCLE_SEL, NV_CYCLE_T, FREE_SPACE, SHED_RCAS, SHED_ROUT, LIM_NOTIFY, CONFIRM_TIME_WRITE_LOCK	30 Bytes
VIEW_3	ST_REV, MODE_BLK, BLOCK_ERR, RS_STATE, FREE_TIME, FAULT_STATE, ALARM_SUM	22 Bytes
VIEW_4	ST_REV, STRATEGY, ALERT_KEY, MANUFAC_ID, DEV_REV, DD_REV, RESTART, FEATURES, CYCLE_TYPE, MIN_CYCLE_T, MEMORY_SIZE, MAX_NOTIFY, ACK_OPTION, WRITE_PRI, ITK_VER	31 Bytes
Function Block		
VIEW_1	ST_REV, MODE_BLK, BLOCK_ERR, PV, OUT, FIELD_VAL, ALARM_SUM	31 Bytes
VIEW_2	ST_REV, XD_SCALE, OUT_SCALE, GRANT_DENY	26 Bytes
VIEW_3	ST_REV, MODE_BLK, BLOCK_ERR, PV, OUT, FIELD_VAL, ALARM_SUM	31 Bytes
VIEW_4	ST_REV, STRATEGY, ALERT_KEY, IO_OPTS, STATUS_OPTS, CHANNEL, L_TYPE, LOW_CUT, PV_TIME, ACK_OPTION, ALARM_HYS, HI_HI_PRI, HI_HI_LIM, HI_PRI, HI_LIM, LO_PRI, LO_LIM, LO_LO_PRI, LO_LO_LIM	46 Bytes
Transducer Block		
VIEW_1	ST_REV, MODE_BLK, BLOCK_ERR, TRANSDUCER_TYPE, XD_ERROR, PRIMARY_VALUE, SECONDARY_VALUE	21 Bytes
VIEW_2	ST_REV, TRANSDUCER_TYPE, PRIMARY_VALUE_TYPE, CAL_POINT_HI, CAL_POINT_LO, SECONDARY_VALUE_UNIT	12 Bytes
VIEW_3	ST_REV, MODE_BLK, BLOCK_ERR, TRANSDUCER_TYPE, XD_ERROR, PRIMARY_VALUE, SECONDARY_VALUE	21 Bytes
VIEW_4	1. ST_REV, STRATEGY, ALERT_KEY, TRANSDUCER_TYPE, PRIMARY_VALUE_RANGE, CAL_MIN_SPAN, CAL_UNIT, SENSOR_TYPE, SENSOR_RANGE, SENSOR_ISOLATOR_MTL, SENSOR_FILL_FLUID 2. ST_REV, SENSOR_SN, SENSOR_CAL_METHOD, SENSOR_CAL_LOC, SENSOR_CAL_DATE, SENSOR_CAL_WHO	41 Bytes 106 Bytes

The VIEW_4 for Transducer Block is split into multiple parts as described in [9], Part 1, Section 4.8.4 in order to represent all the necessary parameters.

VIEW_1 to VIEW_4 are categorized according to the information that the parameters may contaoin:

- VIEW_1 – Operration – Information required by a plant operator to run the process.
- VIEW_2 – Operation Static – Information which may need to be read once and then displayed along with the dynamic data.
- VIEW_3 – All Dynamic – Information which is changing and may need to be referenced in a detailed display.

- VIEW_4 – Other Static – Configuration and maintenance information.

2.3 Access Error Codes

In the last column of the parameter table, error codes for write access to corresponding parameters via FF communication are listed. These codes are shown in the TARGET_ERROR parameter of the related block whenever access error arises.

Access Error Code	Error Name	Description
0x0000	TARGET_NOT_INIT	Current block is not running.
0x0001	TARGET_NO_ERROR	No error for access.
0x0002	TARGET_WRITE_LOCKED	Parameters are write-protected.
0x0003	TARGET_FAC_PROCEDURE	This is a factory procedure. Writing to this parameter requires a password.
0x0004	TARGET_READ_ONLY	This parameter is read-only.
0x0005	TARGET_RANGE_ERROR	Value entered is beyond the value range of this parameter.
0x0006	TARGET_MODE_CHECK_ERROR	Mode entered is not allowed.
0x0007	TARGET_ALM_CHECK_ERROR	Alarm is acknowledged already.
0x0008	TARGET_WRONG_MODE2WRITE	The mode for writing this parameter is wrong.
0x0009	TARGET_INVALID_PASSWORD	The entered password is not correct.
0x1000	TARGET_RB_OOS	Target mode of the resource block is set to Out of Service.
0x2000	TARGET_AI_OOS	Target mode of the AI function block is set to Out of Service.
0x2001	TARGET_AI_NO_SIMU_JUMPER	Simulate-Jumper is required for simulation.
0x3000	TARGET_TB_CAL_UNIT_ERROR	Unit used for calibration must be % or in the same physical group of the sensor units.
0x3001	TARGET_TB_CAL_PAR_TOO_SMALL	The value entered for CAL_POINT_HI/LO is too small.
0x3002	TARGET_TB_CAL_PAR_TOO_LARGE	The value entered for CAL_POINT_HI/LO is too large.
0x3003	TARGET_TB_CAL_SPAN_TOO_SMALL	The applied calibration span is smaller than CAL_MIN_SPAN.
0x3004	TARGET_TB_APPL_PROC_TOO_HIGH	The applied process value for calibration is too high.
0x3005	TARGET_TB_APPL_PROC_TOO_LOW	The applied process value for calibration is too low.

3 OPERATION

3.1 Status

The AI function block of the 240FF receives its input from the transducer block and makes it available at its output hold by the OUT parameter. As has been seen from the parameter table in Section 2.1, OUT is a data structure of type DS-65 containing two elements, i.e., the value and the status to this value. The status is one byte long, bitwize coded, and provides information about the quality and state of the value. The coding of this byte is defined in FF specification, which is applied to the 240FF as below.

Status:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Quality		Quality Substatus				Limits	

3.1.1 Quality

7	6	5	4	3	2	1	0	
Quality		Substatus				Limits		
0	0							Bad
0	1							Uncertain
1	0							Good (Non Cascade)
1	1							Not used

3.1.2 Substatus

The coding of Substatus depends on the coding of Quality.

Quality = Bad (00):

7	6	5	4	3	2	1	0	
Quality		Substatus				Limits		
0	0	0	0	0	0			Non-specific
0	0	0	0	1	1			Device Failure
0	0	0	1	0	0			Sensor Failure
0	0	0	1	1	1			Out of Service

Quality = Uncertain (01):

7	6	5	4	3	2	1	0	
Quality		Substatus				Limits		
0	1	0	0	0	0			Non-specific
0	1	0	0	0	1			Last Usable Value
0	1	0	0	1	0			Substitute Value
0	1	0	1	0	0			Sensor Conversion not Accurate
0	1	0	1	0	1			Engineering Unit Range Violation
0	1	0	1	1	1			Configuration Error
0	1	1	0	0	0			Simulated Value

Quality = Good (Non Cascade) (10):

7	6	5	4	3	2	1	0	
Quality		Substatus				Limits		
1	0	0	0	0	0			Ok
1	0	0	0	0	1			Active Update Event
1	0	0	0	1	0			Active Advisory Alarm (Priority < 8)
1	0	0	0	1	1			Active Critical Alarm (Priority > 8)

3.1.3 Limits

The Limits bits indicate if the value from the sensor is in a valid conversion range or crosses already the sensor conversion boundaries.

Limits:

7	6	5	4	3	2	1	0	
Quality		Substatus				Limits		
						0	0	Ok
						0	1	Low Limit acceded
						1	0	High Limit acceded
						1	1	-

3.2 Operation Mode

The 240FF supports three operation modes, i.e., Out of Service (S/O), Automatic (AUTO) and Manual (MAN) mode. When applied to individual blocks, the allowable modes are

Block	Modes
RB	S/O, AUTO
FB	S/O, AUTO, MAN
TB	S/O, AUTO, MAN

An operator can set the desired mode for operation in the individual blocks via accessing the TARGET_MODE element in the MODE_BLK structure. Without a pre-setting, the default value for TARGET_MODE is O/S.

3.2.1 Automatic Mode (AUTO)

This mode is the normal operation mode of the related block. For function block, e.g., in this mode, the block gets input value from the transducer block, processes this value and outputs it in the OUT parameter. The function block changes to AUTO when TARGET_MODE is set to AUTO.

3.2.2 Out of Service (O/S)

This mode means that the block and hence the functionality of the block is out of service. This may take place, e.g. when the device is in offline state where configuration parameters are sending to the device or in a state where local operation such as menu configuration via the two menu buttons below the LCD-display of the device is just in progress. After the configuration, the operation mode will return to AUTO after the TARGET_MODE element in the MODE_BLK parameter is set to AUTO.

3.2.3 Manual Mode (MAN)

During device operation, this mode can be achieved after TARGET_MODE has been set to MAN. For function block e.g., this mode is required for doing override of the output, for which OUT is disconnected from the function block algorithm part and is written directly by operator. Using this mode, an operator can produce the output manually for other function blocks if the device is interfaced to other function block applications.

3.3 Diagnosis

During device operation, the 240FF sends error information to the BLOCK_ERROR parameter, the XD_ERROR parameter (only for transducer block) and the DIAGNOSIS parameter for diagnosis.

3.3.1 Block Error

BLOCK_ERR is a standard parameter in the resource block and is used to reflect the error status associated with the hardware or software components associated with a block. It is a bit string, so that multiple errors may be shown. For the 240FF, the following errors are coded:

BLOCK_ERR	Description
0x0001	Block is out of service.
0x0020	Lost static data.
0x0040	Memory failure (ROM, RAM or EEPROM).
0x0100	Input failure (available if the option “Propagate fault forward” in the STATUS_OPTS parameter is set).
0x0800	Local override resulted from local operation (e.g. configuration via two menu buttons below the LCD-display), used in resource block.
0x1000	Simulate active, used in the resource block to indicate that a hardware simulate jumper is present.
0x4000	Block configuration error (e.g. if L_TYPE = 0 or XD_SCALE and OUT_SCALE are different when L_TYPE = 1, etc.).

3.3.2 XD Error

XD_ERROR is a standard parameter used only in transducer block. Because the BLOCK_ERR bitstring parameter described in FF-890 [9] is not useful for the many errors possible in the transducer block, XD_ERROR is introduced to hold the single error subcode that the manufacturer considers most important when one or more errors occur. BLOCK_ERR will have bit 0 set whenever XD_ERROR is non-zero. For the 240FF, XD_ERROR may be set to the following status:

XD_ERROR	Description
18	Internal calibration error.
19	Zero point configuration error (relates to the action from the SENSOR_ZERO_TRIM parameter).
24	Software error. This error tells that the task stack used in the operating system reaches the warning limit (In the firmware of the 240FF, a supervision of the stack limit is set. If the device operates normally and nevertheless this error is reported, this stack size must be increased.).

3.3.3 Diagnosis

Besides BLOCK_ERR and XD_EEROR, the 240FF defines also a manufacturer-specific parameter DIAGNOSIS in the resource block to hold all device internal status that are collected and stored in the EEPROM of the device for off-line diagnosis. DIAGNOSIS is a data structure of six unsigned 8 bytes, bitwise coded. The first three bytes hold the current status grouped as warnings, process errors and system errors, respectively, while the next three bytes use the same coding but collect the history of the first three bytes. The coding of warnings, process errors and system errors are shown below.

Byte 1 (warnings):

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Power Fail	Stack Warn	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved

Bit 7 Power Fail: Power failed
Bit 6 Stack Warn: The task stack in the device operating system reaches the warning limit.
Bit 5 Reserved
Bit 4 Reserved
Bit 3 Reserved
Bit 2 Reserved
Bit 1 Reserved
Bit 0 Reserved

Byte 2 (process error):

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Sen Temp OOR	Brd Temp OOR	Sen Value OOR	Out Value OOR	Reserved	Reserved	Reserved	Reserved

Bit 7	Sen Temp OOR:	Sensor temperature out of limit.
Bit 6	Brd Temp OOR:	Electronics (Board) temperature out of limit.
Bit 5	Sen Value OOR:	Sensor value out of range
Bit 4	Out Value OOR:	OUT value out of range
Bit 3	Reserved	
Bit 2	Reserved	
Bit 1	Reserved	
Bit 0	Reserved	

Byte 3 (system error):

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Zero Pos Error	Range Cfg Error	EEPROM Wr Error	RAM Chk Error	ROM Chk Error	Reserved	Reserved	Reserved

Bit 7	Zero Pos Error:	In doing sensor zero trim using the SENSOR_ZERO_TRIM parameter the value entered is out of sensor range.
Bit 6	Range Cfg Error:	Configuration of measure range is not valid.
Bit 5	EEPROM Wr Error:	Error exists while writing EEPROM.
Bit 4	RAM Chk Error:	Error exists in checking RAM.
Bit 3	ROM Chk Error:	Error exists in calculating ROM checksum.
Bit 2	Reserved	
Bit 1	Reserved	
Bit 0	Reserved	

3.4 Reset

In operating the 240FF a user is provided with various alternatives to set or reset the device to some known states via accessing the FACTORY_RESET, RESTART and RESET_HIST_STATUS parameters.

3.4.1 Factory Reset

To ensure a reasonable start, the 240FF is shipped with a factory setting satisfying some pre-defined application requirements. Based on the factory setting, a user can make further configurations of his own. While doing that, it is often desirable to reset the device back to its factory state if something is wrong during the configuration. The manufacturer-specific parameter FACTORY_RESET can be used for this purpose and the factory setting can be restored by writing 2 (“Restore factory setting”) to this parameter.

FACTORY_RESET can also be used to create a new factory setting when 1 (“Create factory setting”) is writen to it. This option is used by manufacturer during production phase.

Note that a factory setting does not include the setting of the network parameters.

3.4.2 Restart

RESTART is a standard parameter used to re-start the device. When 3 (“Defaults”) is written to it, the device will be re-started with defaults. In the 240FF, the default state is different from the factory setting with which the device is shipped. Defaults here refer to the “raw” database values that are valid for the first device start. Factory setting however is not the same as defaults, but created by the manufacturer after some elementary configurations based on the defaults. Re-starting the device with value 3, therefore, will restore the device only to its raw state, but not to its factory setting state.

3.4.3 Clear Status

The manufacturer-specific parameter RESET_HIST_STATUS in the 240FF is used to handle device status. By writing 1 (“Clear history status”) to this parameter, the history status collected in the second three bytes of DIAGNOSIS (see Section 3.3) will be cleared. If 2 (“Clear all device status”) is written to this parameter, both the current status and the history status, i.e. all 6 elements in DIAGNOSIS will be set to their clear state.

3.5 Linearisation

The 240FF is provided with the possibility to convert measurements with a user-specific characteristic other than just with linear scaling or square root conversion provided by the standard parameter L_TYPE. With the help of the parameters TAB_ENTRY, TAB_X_Y_VALUE, TAB_OP_CODE and TAB_STATUS, one can create a linearisation table and store it in the EEPROM. The manufacturer-specific parameter LIN_TYPE then can be used to select the table for measurement conversion. The following steps show how a linearisation table can be created:

1. Set the transducer block to O/S mode.

2. Set TAB_OP_CODE = 1 (“Load new”), which means that the table is ready to accept new characteristic.

3. Use TAB_ENTRY to enter table entry index (1 to 32) and TAB_X_Y_VALUE to enter the values to this entry. Repeat this step until all desired entries are entered[1].

4. Set TAB_OP_CODE = 2 (“End of tranmission”) to check and create the table. The inputted new characteristic will be checked for validity. If valid, a new table will be created, and the TAB_ACTUAL_NUMBER parameter will be updated. During the characteristic checking, the following status may appear in the TAB_STATUS parameter:

   - 1 = good (new table is accepted).

   - 2 = not monotonous increasing (new table is not accepted, old table remains valid).

   - 3 = not enough values transmitted (new table is not accepted, old table remains valid).

   - 4 = table is currently loaded. This status is set after TAB_OP_CODE = 1 (additional access to the table not valid, old values are valid).

[1] For a given table length, e.g. 10, each table entry with index from 1 to 10 must be new entered no matter if the value of this entry is identical or not to its old one. Otherwise the created table will not be accepted for reason “not enough” because totally 10 inputs are expected.

4 MAINTENANCES

4.1 Installation

NI (National Instrument) configurator can be used as a host to operate the 240FF. Before operation, the Device Description (DD) files need to be installed. The files installed should be present in the appropriate directories given below:

```
ManufacturerID
        |
        DeviceType
                |
                DeviceRevDDRev.ffo
                DeviceRevDDRev.sym
                DeviceRevDDRevCFFRev.cff
```

Here *.ffo is the DD tokenizer output file, *.sym is the symbol file and *.cff is the common file format file.

For the transmitter 240FF, ManufactuerID = 0x385884 (The Foxboro Company), Device Type = 0x2506 (sensor type = 0x25 (Eckardt DMU) and sensor sub-type = 0x06), DeviceRev (device revision) = 1.0, DDRev (DD revision) = 1.0 and CFFRev (common file format file revision) = 1.0. Assume that the NI configurator is installed under C:\Nifbus. Then, the DD files of the 240FF should be put into the directory:

```
C:\Nifbus
    |
    data
      |
      385884
          |
          2506
              |
              0101.ffo
              0101.sym
              010101.cff
```

In the subdirectory ”data”, a current file of the FF dictionary file ”standard.dct” should be present also.

4.2 Access Right

4.2.1 Write Lock

The 240FF supports both hardware and software write protections. The hardware write protection is supplied with a hard jumper (no real hard jumper is used; instead the jumper is simulated using an internal parameter that can be “switched” by operating the two menu buttons) while the software write protection is implemented by the standard parameter WRITE_LOCK in the resource block. Write protection is valid when either hardware or software write protection is active. When protected, all writes to static and non-volatile parameters are rejected by the device, not only for those writes via the fieldbus but also for those writes via the menu bottons and external keys.

Note that write lock has higher priority than that of the password protection (to be described in Section 4.2.3).

4.2.2 Local Operation

Local operation is provided in the 240FF via the two menu buttons next to the LCD-display and the two external keys controlled by the LOCAL_KEYS_CTRL parameter. The host can enable and disable local operation by writing the manufacturer-specific parameter LOCAL_OP_ENA. If this parameter is set to 1 (“Local operation enable”), local operation is enabled. If 0 (“Local operation disable”), local operation is disabled.

In case of enabled, device parameters can be changed by local operation. The host is informed about the changes from local operation by checking if the standard parameter BLOCK_ERR is set with the code 0x0800.

It should be noticed that one configuration point in menu may change simultaneously two or more parameters, e.g. the configuration of the “units” element in the PRIMARY_VALUE_RANGE parameter sets automatically the same “units” to XD_SCALE for that the units code of XD_SCALE must match the channel units code (see [10], Part 2 Section 4.1.3). For more details of menu operation refer to [11].

4.2.3 Password Protected Access

The 4th entry of the OD Object Description in Section 2.2.1 shows that the device communication stack itself provides no protection for object accesses. However, protection of those objects related to important internal settings of the device is in general desired. For this intention, the 240FF has been designed to have protection for such parameters whose (write) accesses are only allowed during the production phase. Some so-called factory procedures are integrated into the accesses of these parameters, i.e. the procedures are activated by writing the corresponding parameters. Thus, factory procedures such as sensor calibrations etc. are protected by password from unauthorized users who may unintentionally change these settings. For authorized users, it is their duty to run the factory procedures properly to ensure the functionality of the device.

Two manufacturer-specific parameters, PASSWD_CTRL and PASSWD are provided for the control of password protected accesses. If PASSWD_CTRL is “Off”, no protection is given for the write access of the protected parameters including PASSWD (the protected parameters will be listed later). In this case one can write the parameters, run all factory procedures and enter new password to PASSWD. It should be noticed that the 2400FF is shipped with PASSWD_CTRL being set to “On”. In order to change PASSWD_CTRL to “Off”, the default password is required. Without the default password, PASSWD_CTRL cannot go to “Off” and hence the password protection cannot be removed.

Having entered the default password successfully, PASSWD_CTRL changes automatically to “Off”, and the access restriction on those protected parameters is removed. PASSWD_CTRL should be set back to “On” after the all accesses are finished. Next time another access is hoped, the new password must be entered to get a new access permission. If the password was not changed before, the default password remains valid further.

In summary,

1. For doing protected settings:
    a. Eenter valid password to change PASSWD_CTRL to “Off”.
    b. Do configurations of corresponding parameters.
    c. Set PASSWD_CTRL back to “On”.
2. For changing password:
    a. Eenter valid password to change PASSWD_CTRL to “Off”.
    b. Enter new password to PASSWD.

c. Set PASSWD_CTRL back to “On”.

The following parameters are protected by password.

Parameter Name	Access Values	Factory Procedure Activated
PASSWD_CTRL	1 = “Off”	-
PASSWD	Password text	-
FACTORY_RESET	1 = “Create factory setting”	Writing 1 to this parameter creates a factory setting based on the up-to-date configurations (Note that the setting includes only the part for FB application, but not the part for management).
MODEL_CODE	Model code text	-
PRIMARY_VALUE_TYPE	Unsigned 16	-
CAL_POINT_HI	Float value	Writing a float value to this parameter activates a high point calibration procedure for sensor, which uses the float value and the present sensor value as the two calibration point values.
CAL_POINT_LO	Float value	Writing a float value to this parameter activates a low point calibration procedure for sensor, which uses the float value and the present sensor value as the two calibration point values.
CAL_UNIT	Unit code	Writing a unit code to this parameter updates the CAL_MIN_SPAN parameter simultaneously.
SENSOR_TYPE	Unsigned 16	-
SENSOR_RANGE	DS-68	-
SENSOR_CAL_METHOD	Unsigned 8	-
SENSOR_CAL_LOC	Location text	-
SENSOR_CAL_DATE	DS-11	-
SENSOR_CAL_WHO	Name text	-
SMART_SMOOTH	SMART_SMOOTHING	-
SENSOR_ZERO_TRIM	1 = “Sensor Zero Point Trim” 2 = “Sensor Zero Point Reset”	Writing 1 to this parameter activates a zero point setting based on the present sensor value. Writing 2 to this parameter restores the zero point to the state before zero point trim.

4.3 Calibration

The 240FF supports two-point calibration and sensor zero-point calibration by applying the CAL_POINT_LO, the CAL_POINT_HI and the SENSOR_ZERO_TRIM parameters. These calibrations are factory procedures that are protected by password.

Before a calibration is started it is in general necessary to set PV_FTIME and the time constant in SMART_SMOOTH, i.e. the first element of SMART_SMOOTH to 0 so that the process value used for the calibration can trace in time the changes from the sensor and thus time can be saved for the calibration. After the calibration PV_FTIME and SMART_SMPPTH should be set back to their old values.

4.3.1 Two-Point Calibration

Writing the CAL_POINT_LO/HI parameters activates a two-point calibration for the sensor. This goes ahead along the steps:

1. In the resource block, if PASSWD_CTRL is “On”, enter password to change PASSWD_CTRL to “Off”.

2. Set the transducer block in O/S mode.

3. Hang a weight on the sensor of the device to get the desirable process value for calibration (several seconds waiting time may be required to get a stable process value).

4. Start calibration by writing the calibration point value to CAL_POINT_LO/HI.

5. After the calibration, set PASSWD_CTRL back to “On”.

Error information may be returned to the TARGET_ERROR parameter of the Transducer Block if the calibration process is not successful (for error information refer to Section 2.3).

4.3.2 Sensor Zero-Point Calibration

Sensor zero-point calibration is provided to set another reference point for the process sensor value, which is done internally by subtracting an offset linearly from the process sensor value. Sensor zero-point calibration is a factory procedure and is used for the correction of the zero-point of the process sensor value during the production phase. This calibration follows the steps:

1. In the resource block, if PASSWD_CTRL is “On”, enter password to change PASSWD_CTRL to “Off”.

2. Set the transducer block to O/S mode.

3. Hang a weight on the sensor to get the desirable process value used as offset to create the zero-point (several secodes waitng time may be required to get a stable value).

4. Start calibration by writing 1 (“Sensor Zero Point Trim”) to SENSOR_ZERO_TRIM.

5. After the calibration, set PASSWD_CTRL back to “On”.

Writing the value 2 (“Sensor Zero Point Reset”) to SENSOR_ZERO_TRIM clear the calibrated sensor zero-point.

Besides the sensor zero-point calibration that is protected by password, the 240FF provides further user zero-point calibration using the USER_ZERO_TRIM parameter, which requires no password. Such a calibration corrects the zero-point based on correction by SENSOR_ZERO_TRIM and can be cleared if USER_ZERO_TRIM is written with 2, and the zero-point set by SENSOR_ZERO_TRIM remains unchanged. In contrast, if SENSOR_ZERO_TRIM is written with 2, i.e., to reset the sensor zero-point, both settings done by SENSOR_ZERO_TRIM and USER_ZERO_TRIM will be cleared[2].

[2] A user zero-point can also be set via the two external keys of the device (for the operation of external keys see [11]). Writing 2 to USER_ZERO_TRIM will clear this setting.

5 REFERENCES

[1] Foundation Specifications: System Architecture (FF-800 -1.4)

[2] Foundation Specifications: Communication Profile (FF-940 -1.4)

[3] Foundation Specifications: System Management (FF-880 -1.4)

[4] Foundation Specifications: Network Management (FF-801-1.4)

[5] Foundation Specifications: Fieldbus Message Specification (FF-870 -1.4)

[6] Foundation Specifications: Fieldbus Access Sublayer (FF-875 -1.4)

[7] Foundation Specifications: Data Link Services Subset (FF-821-1.4)

[8] Foundation Specifications: Data Link Protocol Specification (FF-822 -1.4)

[9] Foundation Specifications: Function Block Application Process Part 1 + 2 (FF-890-1.3 + FF-891-1.4)

[10] Foundation Specifications: Transducer Block Application Process Part 1 + 2 (FF-902 + FF-903 - PS 3.0)

[11] Master Instruction for 240FF, Foxboro Eckardt GmbH, 2002