

# iC-PZ Series

HIGH-RESOLUTION REFLECTIVE ABSOLUTE ENCODER



Rev F1, Page 1/85

## FEATURES

- ◆ Reflective, compact, high-resolution, absolute encoder iCs
- ◆ Series of 3 basic encoder iCs for matched code discs ( $\varnothing$  9 mm, 26 mm) and linear scales (max. 6.71 m)
- ◆ Unique *FlexCode*<sup>®</sup> circuitry for arbitrary code disc diameters
- ◆ Monolithic HD Phased Array with excellent signal matching
- ◆ *EncoderBlue*<sup>®</sup>: System-on-chip design with embedded blue LED for excellent signal quality
- ◆ LED power control
- ◆ Absolute singleturn resolution of 22 bit ( $\varnothing$  26 mm) with on-chip interpolation
- ◆ Automatic adjustment features on command
- ◆ Calibration and configuration storage via external I2C EEPROM
- ◆ Digital BiSS, SSI, and SPI interfaces with CMOS I/O
- ◆ Operational and temperature monitoring with alarm messaging
- ◆ Operation at 4.5 V to 5.5 V within  $-40^{\circ}\text{C}$  to  $125^{\circ}\text{C}$
- ◆ 2.5 V, 3.3 V, and 5.0 V compatible I/O ports
- ◆ Unique *FlexCount*<sup>®</sup> interpolation for arbitrary ABZ resolution
- ◆ UVW commutation signals for motors with 1 to 32 pole pairs
- ◆ Configurable analog outputs
- ◆ Absolute Data Interface (ADI) to multiturn sensors
- ◆ Wide assembly tolerances ensure easy installation

## APPLICATIONS

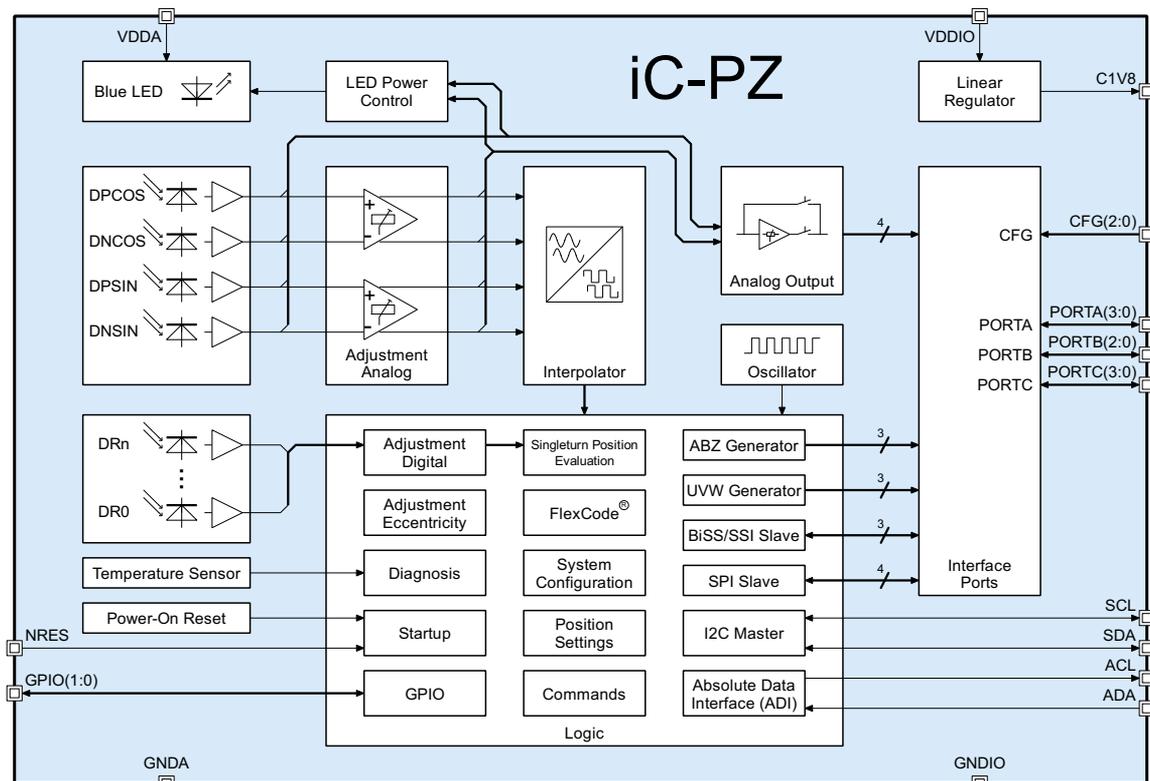
- ◆ Low-height, absolute optical position encoders
- ◆ Factory automation and robotics
- ◆ Servo motors
- ◆ Linear actuators

## PACKAGES



optoQFN32-5x5  
5 mm x 5 mm x 0.9 mm  
RoHS compliant

## BLOCK DIAGRAM



### DESCRIPTION

The iC-PZ Series represents advanced optical-reflective absolute encoder ICs featuring integrated photosensors. Utilizing an *HD Phased Array* and an *Embedded Blue LED Chip*, an outstanding signal fidelity is obtained at relaxed alignment tolerances.

Typical applications are high-resolution rotary and linear position encoders.

Blue-enhanced photosensors are adapted to the short wavelength of the embedded blue LED, improving the signal contrast for an outstanding jitter performance. The optical crosstalk is minimized due to the unique assembly technology of the emitter and sensor.

The internally powered blue LED ensures constant illumination without any external wiring. A sine square or sum control mode is selectable for LED power control.

Automatic adjustment features provide fast and reliable signal calibration on command.

Unique features like FlexCount® and FlexCode® guarantee highest flexibility regarding encoder resolution and diameter.

The configuration and calibration data is stored in an external I2C EEPROM. Additional user data can be stored and accessed via iC-PZ.

The analog circuitry and the integrated blue LED are operated at  $5\text{ V} \pm 10\%$ . For the digital supply includ-

ing the I/O ports, a voltage range of 2.25 V to 5.5 V is possible. Both supply inputs can be shorted and operated at 5 V.

BiSS, SSI, and SPI are supported as digital interfaces to ensure easy system-integration.

#### General notice on materials under excessive conditions

Epoxy resins (such as solder resists, IC package and injection molding materials, as well as adhesives) may show discoloration, yellowing, and surface changes in general when exposed long-term to high temperatures, humidity, irradiation, or due to thermal treatments for soldering and other manufacturing processes.

Equally, standard molding materials used for IC packages can show visible changes induced by irradiation, among others when exposed to light of shorter wavelengths, blue light for instance. Such surface effects caused by visible or IR LED light are rated to be of cosmetic nature, without influence to the chip's function, its specifications and reliability.

Note that any other material used in the system (e.g. varnish, glue, code disc) should also be verified for irradiation effects.

#### General notice on application-specific programming

Parameters defined in the datasheet represent supplier's attentive tests and validations, but - by principle - do not imply any warranty or guarantee as to their accuracy, completeness or correctness under all application conditions. In particular, setup conditions, register settings and power-up have to be thoroughly validated by the user within his specific application environment and requirements (system responsibility).

The chip's performance in application is impacted by system conditions like the quality of the optical target, the illumination, temperature and mechanical stress, sensor alignment and initial calibration.

### CONTENTS

<b>PACKAGING INFORMATION</b>	<b>5</b>	<b>SINGLETURN POSITION EVALUATION</b>	<b>31</b>
SENSOR LAYOUT . . . . .	5	<b>ANALOG OUTPUT</b>	<b>32</b>
PIN CONFIGURATION		Signal Routing . . . . .	32
oQFN32-5x5 (5 mm x 5 mm) . . . . .	6	Output Driver . . . . .	32
PACKAGE DIMENSIONS . . . . .	7	<b>ABZ GENERATOR</b>	<b>33</b>
<b>ABSOLUTE MAXIMUM RATINGS</b>	<b>8</b>	AB Periods . . . . .	33
<b>THERMAL DATA</b>	<b>8</b>	AB Direction . . . . .	33
<b>ELECTRICAL CHARACTERISTICS</b>	<b>9</b>	Z Gating . . . . .	34
ELECTRICAL CHARACTERISTICS: Diagrams	12	Z Polarity . . . . .	34
<b>OPERATING REQUIREMENTS</b>	<b>13</b>	Hysteresis . . . . .	34
Supply Voltages . . . . .	13	Minimum Transition Distance . . . . .	35
BiSS Slave . . . . .	14	ABZ Offset . . . . .	36
SSI Slave . . . . .	15	ABZ Preset . . . . .	36
SPI Slave . . . . .	16	Startup . . . . .	37
Absolute Data Interface (ADI) . . . . .	17	<b>UVW GENERATOR</b>	<b>38</b>
<b>SIGNAL DEFINITIONS</b>	<b>18</b>	Pole Pairs, Direction and Polarity . . . . .	38
<b>LINEAR REGULATOR</b>	<b>19</b>	UVW Offset . . . . .	39
<b>EEPROM SELECTION</b>	<b>19</b>	UVW Preset . . . . .	39
<b>POWER-ON RESET</b>	<b>19</b>	Startup . . . . .	40
<b>STARTUP</b>	<b>19</b>	<b>POSITION SETTINGS</b>	<b>41</b>
<b>INTERFACE PORTS</b>	<b>20</b>	Position Data Length . . . . .	41
<b>CIRCUIT DESIGN PROPOSALS</b>	<b>21</b>	Position Offset . . . . .	41
<b>MEMORY ORGANIZATION</b>	<b>22</b>	Position Preset . . . . .	41
EEPROM Access . . . . .	22	<b>BiSS SLAVE</b>	<b>43</b>
Bank Selection . . . . .	22	Single Cycle Data . . . . .	43
Register Protection Level (RPL) . . . . .	22	Sign-of-Life Counter . . . . .	44
<b>REGISTER MAP</b>	<b>23</b>	Cyclic Redundancy Check . . . . .	44
<b>INTERFACE PORTS CONFIGURATION</b>	<b>29</b>	Fixed & Adaptive Timeout . . . . .	44
Slew Rate I/O-Pads . . . . .	29	BiSS protocol commands . . . . .	44
Port Disabling . . . . .	29	<b>SSI SLAVE</b>	<b>46</b>
VDDIO Monitoring . . . . .	29	<b>SPI SLAVE</b>	<b>48</b>
<b>LED POWER CONTROL</b>	<b>30</b>	Read Registers . . . . .	48
<b>INTERPOLATOR</b>	<b>31</b>	Write Registers . . . . .	48
		Read Position . . . . .	49
		Write Command . . . . .	50
		Read Diagnosis . . . . .	50
		Request Data From I2C Slave . . . . .	51
		Transmit Data To I2C Slave . . . . .	51
		Get Transaction Info . . . . .	51
		Multi-Slave Configurations with iC-PZ . . . . .	52
		Activate Slave In Chain . . . . .	52

<b>COMMANDS</b>	<b>53</b>	<b>FLEXCODE®</b>	<b>69</b>
<b>ABSOLUTE DATA INTERFACE (ADI)</b>	<b>55</b>	<b>ADJUSTMENT ANALOG</b>	<b>70</b>
Interface Enable . . . . .	55	Offset . . . . .	70
Single- and Multi-Slave Operation . . . . .	55	Amplitude . . . . .	71
Synchronization . . . . .	56	Phase . . . . .	72
Error and Warning Bits . . . . .	56	Adjustment Analog, Static . . . . .	73
Interface Settings . . . . .	57	Adjustment Analog, Dynamic . . . . .	74
Startup . . . . .	58	Adjustment Analog, Applied Correction . . . . .	74
Diagnosis . . . . .	58		
<b>I2C MASTER</b>	<b>60</b>	<b>ADJUSTMENT DIGITAL</b>	<b>75</b>
Communication with 8-bit devices . . . . .	60	Adjustment Digital, Static . . . . .	75
Communication with 16-bit devices . . . . .	60	Adjustment Digital, Dynamic . . . . .	76
		Adjustment Digital, Applied Correction . . . . .	76
<b>GPIO</b>	<b>62</b>	<b>ADJUSTMENT ECCENTRICITY</b>	<b>77</b>
GPIO as Input . . . . .	62		
GPIO as Output . . . . .	62	<b>SAFETY ADVICE</b>	<b>78</b>
<b>TEMPERATURE SENSOR</b>	<b>63</b>	<b>HANDLING ADVICE</b>	<b>78</b>
		<b>DEVICE OVERVIEW</b>	<b>79</b>
<b>DIAGNOSIS</b>	<b>64</b>	<b>DESIGN REVIEW: Notes On Chip Functions</b>	<b>80</b>
Errors & Warnings . . . . .	66	<b>REVISION HISTORY</b>	<b>81</b>
CRC Status . . . . .	66		
<b>SYSTEM CONFIGURATION</b>	<b>67</b>		
Revision & Identification . . . . .	67		
System Definition . . . . .	67		
BiSS Profile & Identifier . . . . .	68		

# iC-PZ Series

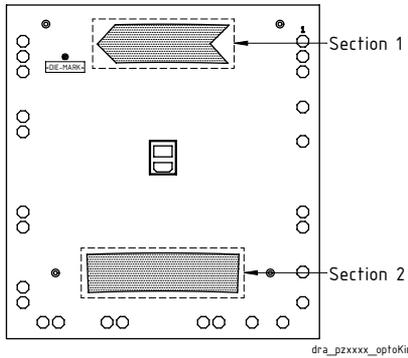
HIGH-RESOLUTION REFLECTIVE ABSOLUTE ENCODER



Rev F1, Page 5/85

## PACKAGING INFORMATION

### SENSOR LAYOUT



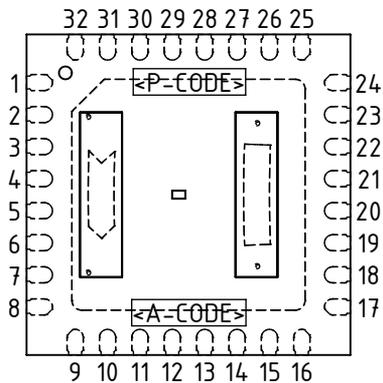
### AOI CRITERIA

<Die Mark>	<Section>	<Area Class> <sup>1</sup>
iC-PZxxxx	1	A16
	2	A25

<sup>1</sup> Inspection class for the optical inspection of detector areas. Refer to [Optical Selection Criteria](#) for further description.

### PIN CONFIGURATION

oQFN32-5x5 (5 mm x 5 mm)



### PIN FUNCTIONS

No.	Name	Function
1	CFG(0)	Port Configuration Input
8	PORTA(3)	Interface Port A
9	PORTA(2)	Interface Port A
10	PORTA(1)	Interface Port A
11	PORTA(0)	Interface Port A
12	C1V8	Core Voltage Buffer Capacitance (see Elec. Char. R01)
13	GNDIO <sup>3</sup>	Digital I/O Ground
14	VDDIO	+2.25 V... +5.5 V Digital I/O Supply Voltage Input
15	PORTB(2)	Interface Port B
16	PORTB(1)	Interface Port B
17	PORTB(0)	Interface Port B
18	ACL	Absolute Data Interface, clock line
19	ADA	Absolute Data Interface, data line
20	GPIO(1)	General Purpose I/O
21	GPIO(0)	General Purpose I/O
22	NRES	Reset Input
23	PORTC(3)	Interface Port C
24	PORTC(2)	Interface Port C
25	PORTC(1)	Interface Port C
26	PORTC(0)	Interface Port C
27	SCL	I2C Master, clock line
28	SDA	I2C Master, data line
29	GNDA <sup>3</sup>	Analog Ground
30	VDDA	+4.5 V... +5.5 V Analog Supply Voltage Input
31	CFG(2)	Port Configuration Input
32	CFG(1)	Port Configuration Input
2..7	n.c. <sup>1</sup>	
	BP <sup>2</sup>	Backside Pad / Exposed Pad

IC top marking: <P-CODE> = product code, <A-CODE> = assembly code (subject to changes);

<sup>1</sup> Pin numbers marked with n.c. are not connected.

<sup>2</sup> The backside pad has to be connected by a single link to GNDA. A current flow across the pad is not permissible.

<sup>3</sup> GNDA and GNDIO must be at the same potential but should be connected with separate lines from a star point on the PCB.

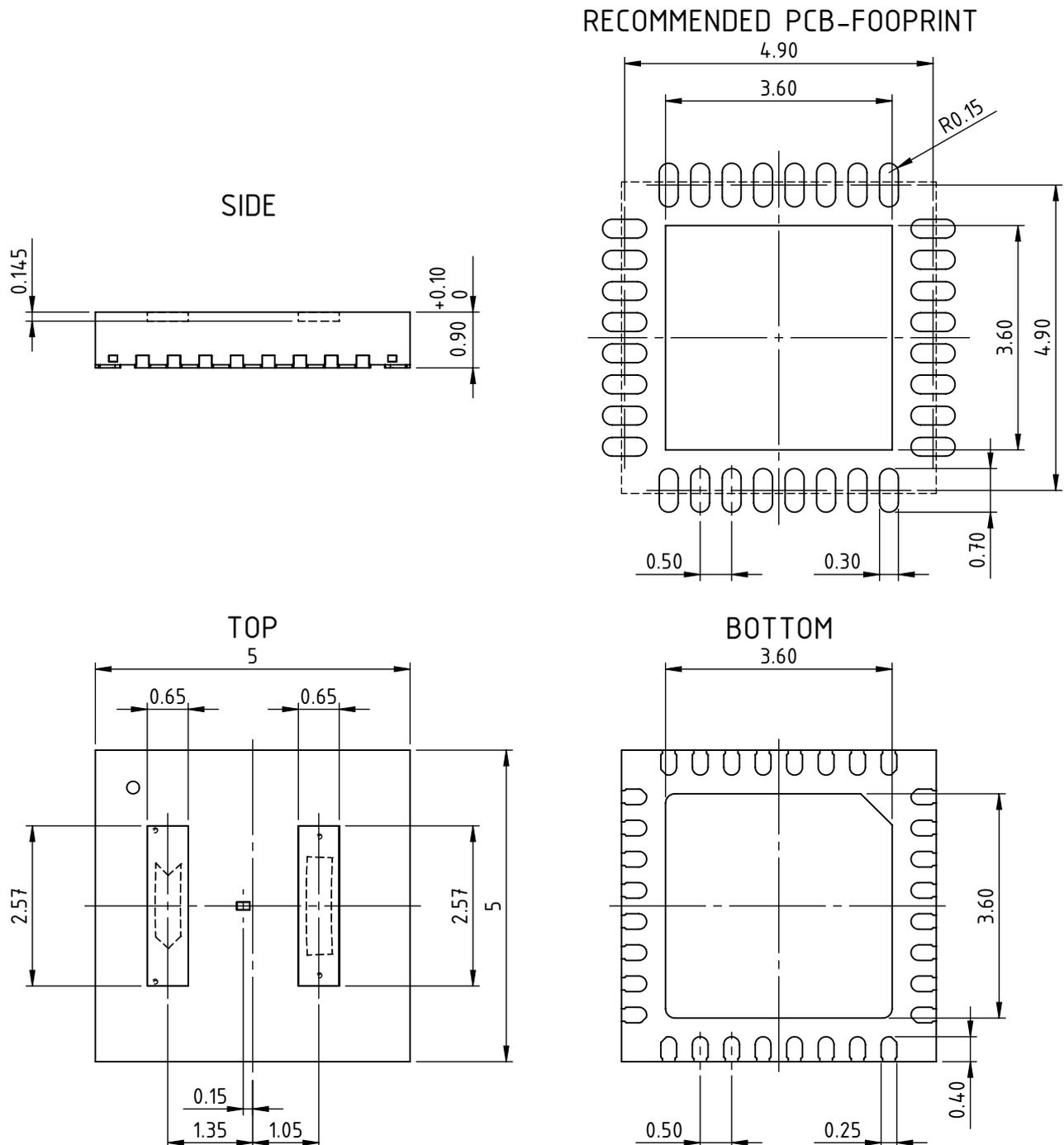
# iC-PZ Series

HIGH-RESOLUTION REFLECTIVE ABSOLUTE ENCODER



Rev F1, Page 7/85

## PACKAGE DIMENSIONS



All dimensions given in mm. General Tolerances of form and position according to JEDEC MO-220. Positional tolerance of sensor pattern:  $\pm 70\mu\text{m}$  /  $\pm 1^\circ$  (with respect to center of backside pad). Maximum molding excess  $+20\mu\text{m}$  /  $-75\mu\text{m}$  versus surface of glass. Small pits in the mold surface, which may occasionally appear due to the manufacturing process, are cosmetic in nature and do not affect reliability.

### ABSOLUTE MAXIMUM RATINGS

Beyond these values damage may occur; device operation is not guaranteed.

Item No.	Symbol	Parameter	Conditions	Limits		Unit
				Min.	Max.	
G001	VDDA	Voltage at VDDA	Referenced to GNDA	-0.3	6	V
G002	I(VDDA)	Current in VDDA		-20	100	mA
G003	VDDIO	Voltage at VDDIO	Referenced to GNDA	-0.3	6	V
G004	I(VDDIO)	Current in VDDIO		-20	100	mA
G005	GNDIO	Voltage at GNDIO	Referenced to GNDA Supply voltage at VDDA or VDDIO applied	-0.3	0.3	V
G006	V(C1V8)	Voltage at C1V8	Referenced to GNDIO	-0.3	2.0	V
G007	I(C1V8)	Current in C1V8		-20	20	mA
G008	V(PORTC)	Voltage at PORTC(3:0)	Referenced to GNDA	-0.3	VDDA + 0.3	V
G009	I(PORTC)	Current in PORTC(3:0)		-20	20	mA
G010	V()	Pin Voltage, all remaining pins	Referenced to GNDA	-0.3	VDDIO + 0.3	V
G011	I()	Pin Current, all remaining pins		-20	20	mA
G012	Vd()	ESD Susceptibility	HBM, 100 pF discharged through 1.5 kΩ		2	kV
G013	Tj	Junction Temperature		-40	140	°C

<sup>1</sup> JEDEC document JEP 155: 500V HBM allows safe manufacturing with a standard ESD control process

### THERMAL DATA

Item No.	Symbol	Parameter	Conditions	Limits			Unit
				Min.	Typ.	Max.	
T01	Ta	Operating Ambient Temperature Range		-40		125	°C
T02	Ts	Permissible Storage Temperature Range		-40		125	°C
T03	Tpk	Soldering Peak Temperature	tpk < 20 s, convection reflow  MSL 3 (max. floor life 168 h at 30 °C and 60 % RH); Refer to <a href="#">Handling and Soldering Conditions</a> for details.			245	°C
T04	Rthja	Thermal Resistance Chip to Ambient	Package mounted on PCB according to JEDEC standard		50		K/W

All voltages are referenced to ground unless otherwise stated.

All currents flowing into the device pins are positive; all currents flowing out of the device pins are negative.

### ELECTRICAL CHARACTERISTICS

Operating conditions: VDDA = 4.5...5.5 V, VDDIO = 2.25...VDDA V, GNDA = GNDIO = 0 V, Tj = -40...140 °C, unless otherwise noted

Item No.	Symbol	Parameter	Conditions				Unit
				Min.	Typ.	Max.	
<b>General</b>							
001	VDDA	Analog Supply Voltage	referenced to GNDA	4.5	5.0	5.5	V
002	I(VDDA)	Analog Supply Current	no external load, LED off refer to Table 145 for details without analog output buffer with analog output buffer		18.4 22.2		mA mA
003	VDDIO	I/O Supply Voltage	referenced to GNDIO	2.25		VDDA	V
004	I(VDDIO)	I/O Supply Current	ports configured to BiSS/ABZ/Analog, default register values, 600 RPM (iC-PZ2656) VDDIO = 5.5 V, no external load, NRES = low VDDIO = 2.25 V, no external load, NRES = high VDDIO = 5.5 V, no external load, NRES = high		3.8 14.5 15.0		mA mA mA
005	SR(VDDIO)	Slewrate of VDDIO	in range 1.0 V..VDDIOon	25			V/s
006	V(C1V8)	Digital Core Voltage	generated internally	1.62	1.8	1.98	V
007	$\alpha()$ e,mx	Maximum Permissible Acceleration	$\alpha()$ e = $\frac{d\omega}{dt}$ for sine/cosine  for linear systems: $a()$ mx = $\frac{\alpha()$ e,mx}{2\pi} \cdot r_{nat} with $r_{nat}$ native resolution, i.e. line distance of the incremental track on the scale PZ01L: $r_{nat} = 204.8 \mu\text{m} \rightarrow a()$ mx = $6\,500 \frac{\text{m}}{\text{s}^2}$			200	$10^6$ rade/s <sup>2</sup>
<b>Digital I/O Pads</b>							
201	V()	Pin-Open Voltage CFG(2:0)		40	50	60	%VDDIO
202	Vt()hi	Threshold high CFG(2:0)				90	%VDDIO
203	Vt()med	Threshold medium CFG(2:0)		40		60	%VDDIO
204	Vt()lo	Threshold low CFG(2:0)		10			%VDDIO
205	Vt()hys	Threshold Hysteresis medium/high or low/medium CFG(2:0)			10		%VDDIO
211	Rpu()	Pull-Up Resistor CFG(2:0)	V(CFG) = 0 V	130	200	280	k $\Omega$
212	Rpd()	Pull-Down Resistor CFG(2:0)	V(CFG) = VDDIO	130	200	280	k $\Omega$
213	Vt()hi	Threshold high NRES, GPIO(1:0), SCL, SDA, ADA, PORTA(3:0), PORTB(2:0), PORTC(3:0)	pins of PORTA/B/C only if configured as inputs			70	%VDDIO
215	Vt()lo	Threshold low NRES, GPIO(1:0), SCL, SDA, ADA, PORTA(3:0), PORTB(2:0), PORTC(3:0)	pins of PORTA/B/C only if configured as inputs	30			%VDDIO
217	Vt()hys	Threshold Hysteresis NRES, GPIO(1:0), SCL, SDA, ADA, PORTA(3:0), PORTB(2:0), PORTC(3:0)	pins of PORTA/B/C only if configured as inputs		15		%VDDIO
219	lpu()	Pull-Up Current NRES, GPIO(1:0), ADA, PORTA(3:0), PORTB(2:0), PORTC(3:0)	V() = GNDIO pins of PORTA/B/C only if configured as inputs pin GPIO(0) only for GPIO0_CFG = 00/01 pin GPIO(1) only for GPIO1_CFG = 00/01 and IC not in reset	-50		-8	$\mu\text{A}$
220	lpd()	Pull-Down Current GPIO(1:0)	V() = VDDIO pin GPIO(0) only for GPIO0_CFG = 10 pin GPIO(1) only for GPIO1_CFG = 10 or IC in reset	8		50	$\mu\text{A}$
222	isc()hi	Short-Circuit Current high GPIO(1:0), ACL, PORTA(3:0), PORTB(2:0), PORTC(3:0)	V() = GNDIO, pins of PORTA/B/C only if configured as outputs VDDIO = 2.5 V $\pm$ 10 % VDDIO = 3.3 V $\pm$ 10 % VDDIO = 5 V $\pm$ 10 %	-55 -75 -120			mA mA mA

### ELECTRICAL CHARACTERISTICS

Operating conditions: VDDA = 4.5...5.5 V, VDDIO = 2.25...VDDA V, GNDA = GNDIO = 0 V, Tj = -40...140 °C, unless otherwise noted

Item No.	Symbol	Parameter	Conditions				Unit
				Min.	Typ.	Max.	
224	Isc()lo	Short-Circuit Current lo GPIO(1:0), SCL, SDA, ACL, PORTA(3:0), PORTB(2:0), PORTC(3:0)	V() = VDDIO, pins of PORTA/B/C only if configured as outputs VDDIO = 2.5 V ± 10 % VDDIO = 3.3 V ± 10 % VDDIO = 5 V ± 10 %			55 75 120	mA mA mA
226	Vs()hi	Saturation Voltage high GPIO(1:0), ACL, PORTA(3:0), PORTB(2:0), PORTC(3:0)	Vs()hi = VDDIO - V() I() = -4 mA			0.4	V
227	Vs()lo	Saturation Voltage low GPIO(1:0), ACL, PORTA(3:0), PORTB(2:0), PORTC(3:0)	I() = 4 mA			0.4	V
228	Is()lo	Saturation Current low SCL, SDA	V() = 400 mV V() = 600 mV	3 6			mA mA
229	tr()	Rise Time GPIO(1:0), ACL, PORTA(3:0), PORTB(2:0), PORTC(3:0)	CL = 20 pF, rise 30..70 % VDDIO PADx_CFG = 00 and GPIO(1:0) PADx_CFG = 01 PADx_CFG = 10 PADx_CFG = 11			220 75 15 3.5	ns ns ns ns
230	tf()	Fall Time GPIO(1:0), ACL, PORTA(3:0), PORTB(2:0), PORTC(3:0)	CL = 20 pF, fall 70..30 % VDDIO PADx_CFG = 00 and GPIO(1:0) PADx_CFG = 01 PADx_CFG = 10 PADx_CFG = 11			220 75 15 3.5	ns ns ns ns
231	tf()	Fall Time SCL, SDA	CL = 10..100 pF, fall 70..30 % VDDIO VDDIO = 2.25..3.0 V VDDIO = 3.0..5.5 V	15 20		150 250	ns ns
<b>PRC Photocurrent Amplifiers and Comparators</b>							
306	fc,r()hi	Cut-Off Frequency PRC (-3 dB)		1000			kHz
<b>Quad Photocurrent Amplifier</b>							
501	fc,a()hi	Cut-Off Frequency Analog (-3 dB)		240			kHz
505	Vv()dc,eff	Effective DC Signal Level	LED_CTRL = 1 and LED_CUR = 0 (sum control) Vv()dc,eff = Vv()dc - Vv()d with Vv()d dark signal level	280		480	mV
<b>Analog Output</b>							
A01	Vout()dc	Output Signal DC Level	ANA_SEL = 00	48	50	52	%VDDA
A02	Vout()jac	Output Signal AC Amplitude	ANA_SEL = 00 ANA_OS = 0 ANA_OS = 1		1000 250		mV mV
A03	I()mx	Permissible Load Current	ANA_SEL = 00 ANA_OS = 0 ANA_OS = 1	-1 -6		1 6	mA mA
A04	C()mx	Permissible Capacitive Load	ANA_SEL = 00			50	pF
<b>ABZ Generator</b>							
C01	AAabs (INL)	Absolute Angle Accuracy (Integral Nonlinearity)	Ideal waveform, f()sin ≤ 60 kHz IPO_FILTER1 = 0xEA, IPO_FILTER2 = 0x4		0.5		°e
C02	AArel (DNL)	Relative Angle Accuracy (Differential Nonlinearity)	Ideal waveform IPO_FILTER1 = 0xEA, IPO_FILTER2 = 0x4 ABZ_PER ≤ 2 <sup>SYS_eff</sup> x 256, f()sin = 128 Hz ABZ_PER ≤ 2 <sup>SYS_eff</sup> x 256, f()sin = 2.56 kHz ABZ_PER ≤ 2 <sup>SYS_eff</sup> x 1024, f()sin = 128 Hz ABZ_PER ≤ 2 <sup>SYS_eff</sup> x 1024, f()sin = 2.56 kHz see also Figure 1		15 2.5 35 7		% % % %
C03	fout()	Maximum Frequency per Output	ABZ_MTD = 0x0 (default), f()sin ≤ 240 kHz	6			MHz
<b>LED Power Control</b>							
L01	Iop()	Permissible LED Current	except startup	0.5		30	mA
L02	Ictrl()	Controlled LED Output Current	refer to Table 145 for details		5..10		mA
L03	Iop()min	Minimum LED Current			50		%Iop()

### ELECTRICAL CHARACTERISTICS

Operating conditions: VDDA = 4.5...5.5 V, VDDIO = 2.25...VDDA V, GNDA = GNDIO = 0 V, Tj = -40...140 °C, unless otherwise noted

Item No.	Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
L07	Ic,led	LED Constant Current	LED_CUR = 0x0	30	40	70	mA
			LED_CUR = 0x1	0	0	0.1	mA
			LED_CUR = 0x2	0.75	1	2.3	mA
			LED_CUR = 0x3	1.5	2	4.3	mA
			LED_CUR = 0x4	3	4	8	mA
			LED_CUR = 0x5	6	8	16	mA
			LED_CUR = 0x6	12	16	29	mA
			LED_CUR = 0x7	24	32	56	mA
<b>Oscillator</b>							
O01	fosc	Oscillator Frequency		72	80	88	MHz
<b>Power-On Reset &amp; Voltage Monitoring</b>							
P01	VDDAon	Turn-On Threshold VDDA (power-on release)	increasing voltage at VDDA			3.95	V
P02	VDDAoff	Turn-Off Threshold VDDA (power-down reset)	decreasing voltage at VDDA	3.00			V
P03	VDDAhys	Threshold Hysteresis VDDA	VDDAhys = VDDAon - VDDAoff	200	300	600	mV
P04	VDDIOpor	Threshold VDDIO for Voltage Regulator (power-on)	increasing voltage at VDDIO			2.20	V
P05	VDDIOon	Turn-On Threshold VDDIO (diagnosis-ok)	VDDIOSEL = 00 (2.5 V)			2.20	V
			VDDIOSEL = 01 (3.3 V)			2.95	V
			VDDIOSEL = 10 (5.0 V)			3.95	V
P06	VDDIOoff	Turn-Off Threshold VDDIO (diagnosis-error)	VDDIOSEL = 00 (2.5 V)	1.3			V
			VDDIOSEL = 01 (3.3 V)	2.2			V
			VDDIOSEL = 10 (5.0 V)	3.0			V
P07	VDDIOhys	Threshold Hysteresis VDDIO	VDDIOhys = VDDIOon - VDDIOoff				mV
			VDDIOSEL = 00 (2.5 V)		250		mV
			VDDIOSEL = 01 (3.3 V)		200		mV
VDDIOSEL = 10 (5.0 V)		250		mV			
<b>Linear Regulator</b>							
R01	C(C1V8)	Recommended Capacity at C1V8			100		nF
<b>Adjustment Analog</b>							
S01	COROS()min	Minimum Offset Correction	COS_OFF(9:0) = 0x3FF and/or SIN_OFF(9:0) = 0x3FF			-100	mV
S02	COROS()mx	Maximum Offset Correction	COS_OFF(9:0) = 0x1FF and/or SIN_OFF(9:0) = 0x1FF	100			mV
S03	CFA()min	Minimum Amplitude Correction Factor	SC_GAIN = 0x200			0.825	
S04	CFA()mx	Maximum Amplitude Correction Factor	SC_GAIN = 0x1FF	1.20			
S05	CORPH()min	Minimum Shift of Phase	SC_PHASE(9:0) = 0x1FF			-10	°e
S06	CORPH()mx	Maximum Shift of Phase	SC_PHASE(9:0) = 0x3FF	10			°e
<b>Temperature Sensor</b>							
T01	Trng	Temperature Sensor Range		-50		175	°C
T02	Tres	Temperature Sensor Resolution			0.1		°C
T03	Tacc	Temperature Sensor Accuracy	Tj = -40..140 °C and after adjustment at iC-Haus	-3		3	°C
<b>Startup</b>							
W01	tstart()	Startup Time	VDDA > VDDAon VDDIO > VDDIOpor EEPROM with valid configuration attached to SCL/SDA		14		ms

### ELECTRICAL CHARACTERISTICS

Operating conditions: VDDA = 4.5...5.5 V, VDDIO = 2.25...VDDA V, GNDA = GNDIO = 0 V, Tj = -40...140 °C, unless otherwise noted

Item No.	Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
				Serial Interfaces			
Z01	fBISS	Permissible BiSS-Slave Frequency	point-to-point configuration bus configuration (chaining multiple chips)			20 10	MHz MHz
Z02	fSSI	Permissible SSI-Slave Frequency				10	MHz
Z03	fSPI	Permissible SPI-Slave Frequency	point-to-point configuration bus configuration (chaining multiple chips), as of $t_{p1}$ and $t_{p3}$ 2 chips 3 chips 4 chips $n$ chips, $n \geq 2$ : $f = \frac{100}{7n - 4}$			12 10 5.88 4.17 f	MHz MHz MHz MHz
Z04	fI2C	I2C-Master Frequency	EEPROM attached to SCL/SDA		220	400	kHz
Z05	tsample	Period of Adaptive Timeout Sampling Clock (for EDS)	refer to characteristics in BiSS SLAVE on page 43	50	67	75	ns
Z06	tout(adapt)	Adaptive Slave Timeout at SLO	BISS_NTOA = 0 refer to timing Figure 4 $t_{init}$ measured as first 1.5 · T(MA) each frame	0.075	$t_{init} + 0.2$	24	µs
Z07	tout(fixed)	Fixed Slave Timeout at SLO	BISS_NTOA = 1 refer to timing Figure 4	16	20	24	µs

### ELECTRICAL CHARACTERISTICS: Diagrams

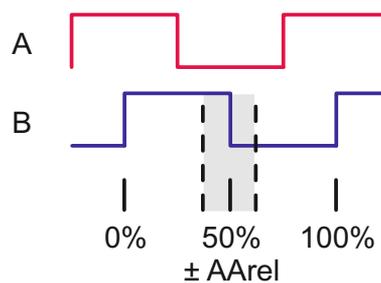


Figure 1: Definition of AB duty cycle variation.

### OPERATING REQUIREMENTS: Supply Voltages

Item No.	Symbol	Parameter	Conditions	Min. / Max.		Unit
				Min.	Max.	
<b>Supply Voltages</b>						
I001	$t_{r,VDDIO}$	VDDIO Rise Time		refer to Elec. Char. 005		
I002	VDDIO(t)	Voltage VDDIO at time t			VDDA(t)	V

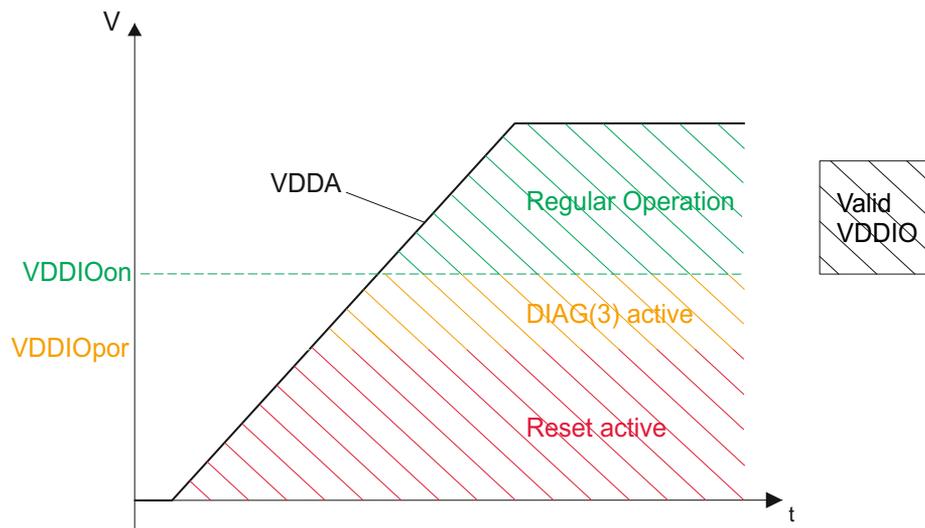


Figure 2: Supply voltages at startup

### OPERATING REQUIREMENTS: BiSS Slave

Item No.	Symbol	Parameter	Conditions	Timing		Unit
				Min.	Max.	
<b>BiSS protocol</b>						
I101	$t_{frame}$	Permissible Frame Repetition		(*)	indefinite	
I102	$t_C$	Permissible Clock Period		50		ns
I103	$t_{L1}$	Clock Signal hi Level Duration		20	$t_{out}$	ns
I104	$t_{L2}$	Clock Signal lo Level Duration		20	$t_{out}$	ns
I105	$t_{busy}$	Processing Time with Start Bit Delay			$5 \cdot t_C$	
I106	$t_{P3}$	Propagation Delay: SLO stable after MA lo $\rightarrow$ hi	CL = 20 pF, rise to 70 % VDDIO or fall to 30 % VDDIO PADx_CFG = 00 PADx_CFG = 01 PADx_CFG = 10 PADx_CFG = 11		400 150 60 35	ns ns ns ns
I107	$t_{out}$	Adaptive Slave Timeout		refer to Elec. Char. Z06		
I108	$t_{S1}$	Setup Time: SLI stable before MA hi $\rightarrow$ lo		5		ns
I109	$t_{H1}$	Hold Time: SLI stable after MA hi $\rightarrow$ lo		10		ns

(\*) Allow  $t_{out}$  to elapse.

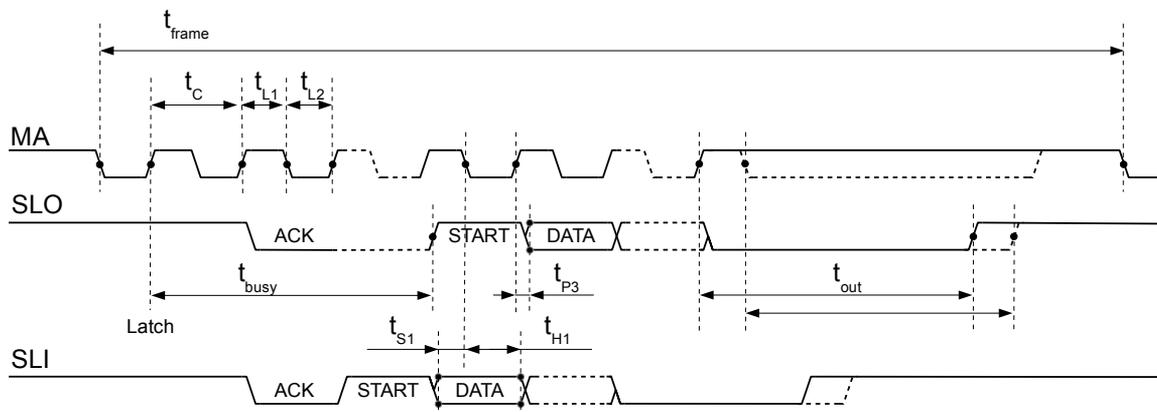


Figure 3: BiSS timing

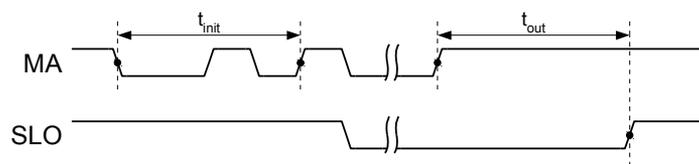


Figure 4: BiSS slave timeout

### OPERATING REQUIREMENTS: SSI Slave

Item No.	Symbol	Parameter	Conditions	Min.	Max.	Unit
<b>SSI protocol</b>						
I201	$t_{frame}$	Permissible Frame Repetition		(*)	indefinite	
I202	$t_C$	Permissible Clock Period		100		ns
I203	$t_{L1}$	Clock Signal hi Level Duration		25	$t_{out}$	ns
I204	$t_{L2}$	Clock Signal lo Level Duration		25	$t_{out}$	ns
I205	$t_{RQ}$	REQ Signal lo Level Duration		50		ns
I206	$t_{P3}$	Propagation Delay: SLO stable after MA lo $\rightarrow$ hi	CL = 20 pF, rise to 70 % VDDIO or fall to 30 % VDDIO PADx_CFG = 00 PADx_CFG = 01 PADx_CFG = 10 PADx_CFG = 11		400 150 60 35	ns ns ns ns
I207	$t_{out}$	Adaptive Slave Timeout		refer to Elec. Char. Z06		

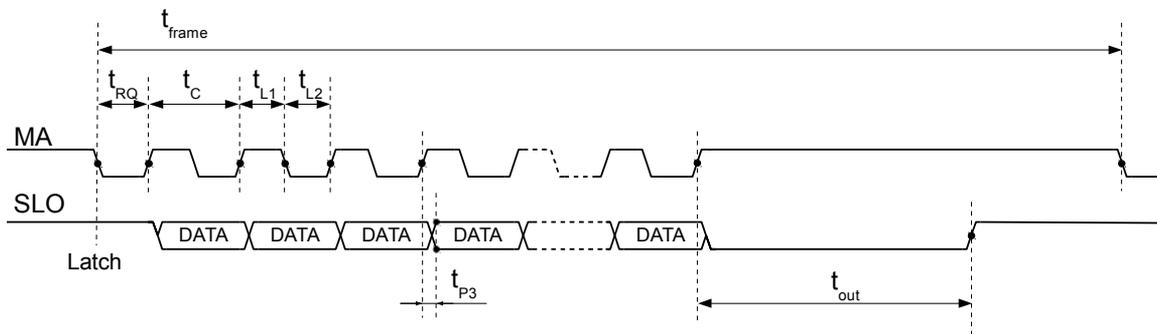


Figure 5: SSI timing

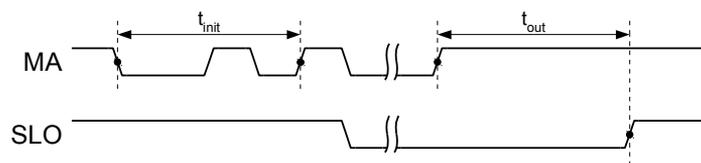


Figure 6: SSI slave timeout

**OPERATING REQUIREMENTS: SPI Slave**

Item No.	Symbol	Parameter	Conditions			Unit
				Min.	Max.	
<b>SPI protocol</b>						
I301	$t_{C1}$	Permissible Clock Cycle Time		83.33		ns
I302	$t_{L1}$	Clock Signal lo Level Duration		30		ns
I303	$t_{L2}$	Clock Signal hi Level Duration		30		ns
I304	$t_{S1}$	Setup Time: NCS lo before SCLK lo $\rightarrow$ hi		41.67		ns
I305	$t_{H1}$	Hold Time: NCS lo after SCLK lo $\rightarrow$ hi	valid for SPI mode 3	41.67		ns
I306	$t_{H3}$	Hold Time: NCS lo after SCLK hi $\rightarrow$ lo	valid for SPI mode 0	41.67		ns
I307	$t_{W2}$	Wait Time: NCS hi before SCLK change		200		ns
I308	$t_{W1}$	Wait Time: between NCS lo $\rightarrow$ hi and NCS hi $\rightarrow$ lo		200		ns
I309	$t_{H2}$	Hold Time: MOSI stable after SCLK lo $\rightarrow$ hi		5		ns
I310	$t_{S2}$	Setup Time: MOSI stable before SCLK lo $\rightarrow$ hi		15		ns
I311	$t_{P1}$	Propagation Delay: MISO stable after SCLK hi $\rightarrow$ lo	CL = 20 pF, rise to 70 % VDDIO or fall to 30 % VDDIO PADA_CFG = 00 PADA_CFG = 01 PADA_CFG = 10 PADA_CFG = 11		400 150 60 35	ns ns ns ns
I312	$t_{P3}$	Propagation Delay: MISO stable after MOSI change	CL = 20 pF, rise to 70 % VDDIO or fall to 30 % VDDIO PADA_CFG = 00 PADA_CFG = 01 PADA_CFG = 10 PADA_CFG = 11		400 150 60 35	ns ns ns ns
I313	$t_{P4}$	Propagation Delay: MISO stable after NCS hi $\rightarrow$ lo	CL = 20 pF, rise to 70 % VDDIO or fall to 30 % VDDIO PADA_CFG = 00 PADA_CFG = 01 PADA_CFG = 10 PADA_CFG = 11		400 150 60 35	ns ns ns ns
I314	$t_{P2}$	Propagation Delay: MISO hi impedance after NCS lo $\rightarrow$ hi			35	ns

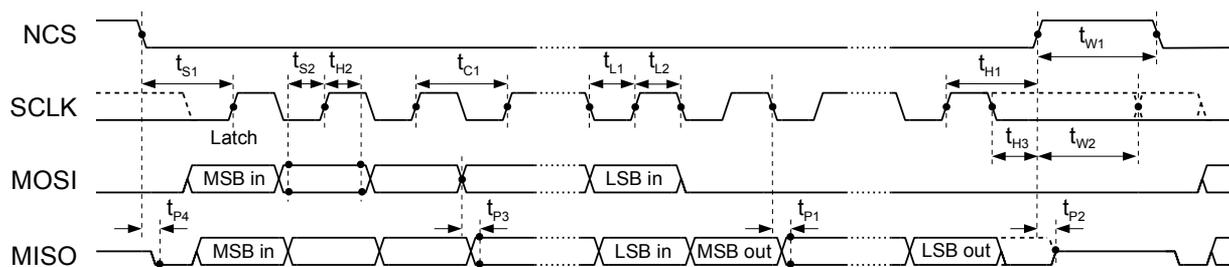


Figure 7: SPI timing

### OPERATING REQUIREMENTS: Absolute Data Interface (ADI)

Item No.	Symbol	Parameter	Conditions	Min.   Max.		Unit
				Min.	Max.	
<b>ADI protocol (SSI)</b>						
I401	$f_{ACL}$	ADI clock frequency	ADI_CFG(6) = 0 (slow mode) ADI_CFG(6) = 1 (fast mode)	150 1500		kHz kHz
I402	$T_{ACL}$	ADI clock period		1/ $f_{ACL}$		
I403	$t_{hi}, t_{lo}$	ADI clock hi/lo level duration		50		% $T_{ACL}$
I404	$T_{frame}$	Frame repetition period	Normal operation Startup, if ADI_CFG(1) = 0 Startup, if ADI_CFG(1) = 1	1500 1500 200		$\mu$ s $\mu$ s $\mu$ s
I405	$t_s$	Setup time: ADA stable before falling ACL edge		100		ns
I406	$t_h$	Hold time: ADA stable after falling ACL edge		100		ns
I407	$t_{out}$	Permissible slave timeout	Rising-edge-triggered monoflop Falling-edge-triggered monoflop Dual-edge-triggered monoflop	11.5 11.5 5.75	40 40 40	$\mu$ s $\mu$ s $\mu$ s

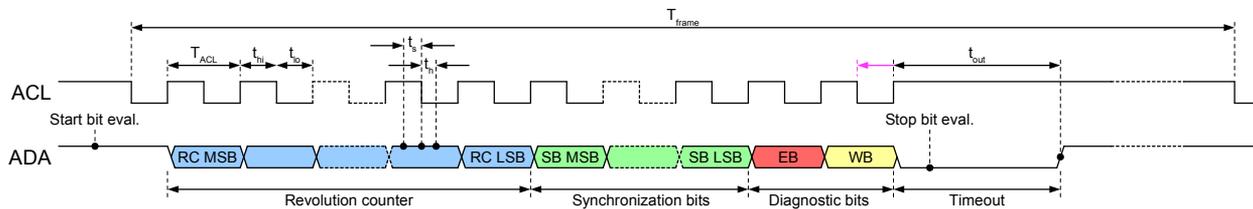


Figure 8: ADI timing with SSI protocol

#### ADA sampling and setup/hold times

ADA is sampled with the double frequency of ACL. With each rising ACL edge iC-PZ processes the previously sampled ADA, i.e., the ADA that has been sampled with the previous falling ACL edge. As a consequence, ADA has to be stable after  $t = (T_{ACL}/2 - t_s)$  after a rising ACL edge.

### SIGNAL DEFINITIONS

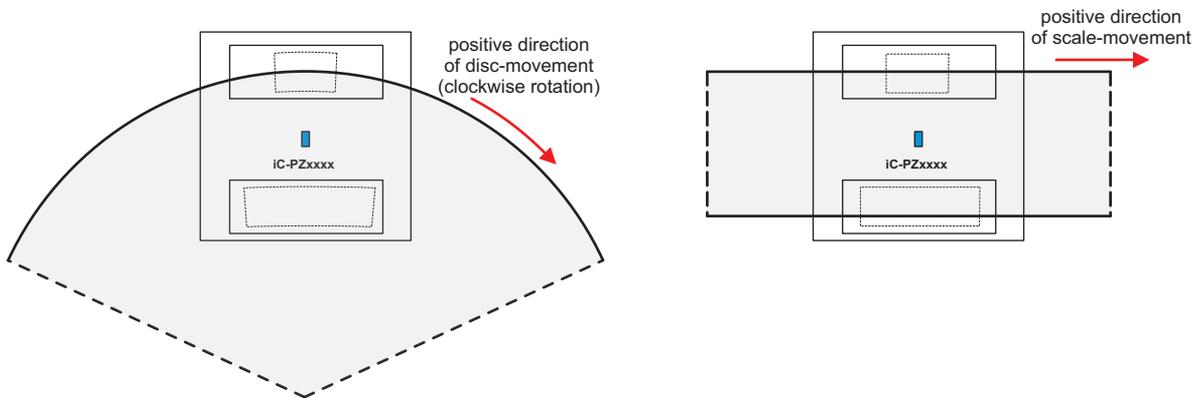


Figure 9: Definition of positive direction of movement

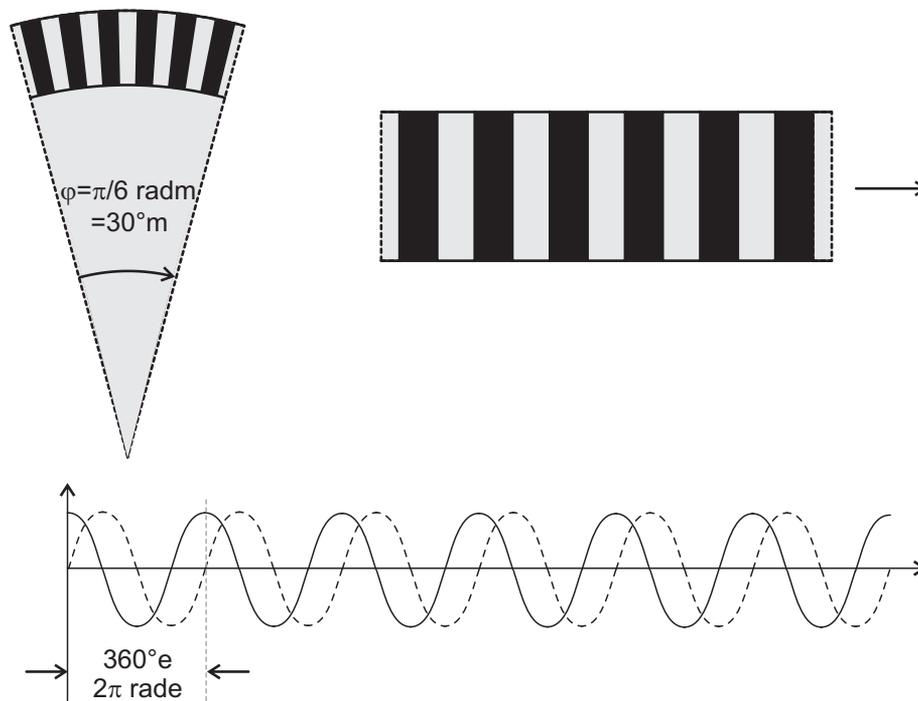


Figure 10: Definition of mechanical degrees / radians ( $^{\circ}\text{m}$  / radm) and electrical degrees / radians ( $^{\circ}\text{e}$  / rade)

A section of a code disc and a section of a linear scale are shown in Figure 10. Six sine/cosine periods are generated by either of them as illustrated. In this example, the code disc has to be moved by 30 mechanical degrees ( $^{\circ}\text{m}$ ) to generate those six signal periods. Each period represents 360 electrical degrees ( $^{\circ}\text{e}$ ).

If radians are used instead of degrees, the code disc has to be moved by  $\frac{\pi}{6}$  mechanical radians (radm). Respectively, each period represents  $2\pi$  electrical radians (rade).

In this document, the frequency of sine/cosine signals is denoted as  $f(\sin)$ .

### LINEAR REGULATOR

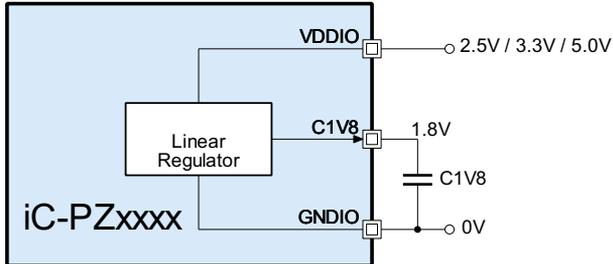


Figure 11: External capacitance blocking pin C1V8

An integrated linear regulator generates the digital core voltage of 1.8 V from the digital supply VDDIO/GNDIO. To ensure a stable regulated voltage, a blocking capacitor at pin C1V8 is required. The recommended capacitance is given in R01 on page 11.

**Note:** The regulated voltage is for internal use only and must not be used to supply additional circuitry. Sensing the regulated voltage with a high impedance, e. g. for safety reasons, is possible.

### EEPROM SELECTION

For proper usage of iC-PZ, an external I2C-EEPROM with the characteristics defined below is required:

- Size of 2 kbit up to 16 kbit (larger sizes are recommended to provide storage for BiSS EDS and user data).
- Supply voltage from 1.8 V up to desired VDDIO. If VDDIO is shorted to VDDA, the supply voltage may begin at 3.0 V (see P02 on page 9).
- Compatible with 400 kHz I2C bus mode.
- Page size  $\geq$  8 byte.
- 7-bit I2C device address is set to 0x50.
- Protocol with one address byte only, as shown in Figure 48

**ATTENTION:** EEPROMs which consider block selection bits as don't care should not be used. This can be the case with 8-pin devices, as well as with 5-pin devices not featuring A2, A1, A0 pins.

Be aware of potential conflicts:

If a user addresses memory beyond the 2 kbit range, iC-PZ configuration data will be overwritten.

If further I2C slave devices are operated on the same bus, higher device addresses may be occupied.

RPL (register protection level) may be passed over.

### POWER-ON RESET

To ensure correct startup, the system is reset until all power-on thresholds defined in ELECTRICAL CHARACTERISTICS section Power-On Reset are exceeded:

- VDDA must exceed threshold VDDAon
- VDDIO must exceed threshold VDDIOpor

### STARTUP

A system-restart is triggered by one of the following events:

- Power-on reset (VDDA, VDDIO)
- Pin NRES (0 = reset, 1 = active)
- Command **REBOOT** via serial interface

While iC-PZ is in reset state or during power-up phase, pin GPIO(0) is low, indicating that the system is not yet ready. Communication using one of the serial interfaces is not possible at this time. After leaving the

reset state, iC-PZ performs its internal startup routines, including reading the configuration stored in the external EEPROM and optionally the multiturn position from an external multiturn-device. During startup, communication using one of the serial interfaces is not allowed. Afterwards, pin GPIO(0) goes high and iC-PZ is ready.



In case the communication with the EEPROM fails, iC-PZ will load its default configuration.

### INTERFACE PORTS

Five interfaces, of which three are multiplexed to ports A, B, and C, are provided by iC-PZ. The individual port configuration is made via pins CFG(2:0). By connecting those pins to the appropriate levels GNDIO (L), VDDIO/2 (M), or VDDIO (H), one of the configurations shown in Table 1 is selected. The pin assignment to the corresponding configuration is defined in Table 2. For further configurations refer to INTERFACE PORTS CONFIGURATION on page 29.

**Note:** As BiSS and SSI share its physical ports, the interface of choice has to be enabled additionally via [SSI\\_EN](#). Refer to BiSS SLAVE on page 43.

Pin Level			Port Function		
CFG(2)	CFG(1)	CFG(0)	Port A	Port B	Port C
L	L	L	SPI	BiSS/SSI	ABZ
L	L	M	SPI	BiSS/SSI	Analog
L	L	H	SPI	BiSS/SSI	UVW
H	L	L	BiSS/SSI	ABZ	UVW
H	L	M	BiSS/SSI	ABZ	Analog
H	L	H	BiSS/SSI	UVW	Analog
H	H	L	SPI	ABZ	UVW
H	H	M	SPI	ABZ	Analog
H	H	H	SPI	UVW	Analog
M	L	H	ABZ	UVW	Analog
Others			Reserved (do not use)		

Table 1: Pin configured port function

	Pin Assignment			
Port A	PORTA(3)	PORTA(2)	PORTA(1)	PORTA(0)
Port B	-	PORTB(2)	PORTB(1)	PORTB(0)
Port C	PORTC(3)	PORTC(2)	PORTC(1)	PORTC(0)
Interface	Interface Signal			
BiSS/SSI	-	MA	SLI	SLO
SPI	NCS	SCLK	MOSI	MISO
ABZ	-	A	B	Z
UVW	-	U	V	W
Analog	NSIN	PSIN	NCOS	PCOS

Table 2: Pin assignment to interface signal

### CIRCUIT DESIGN PROPOSALS

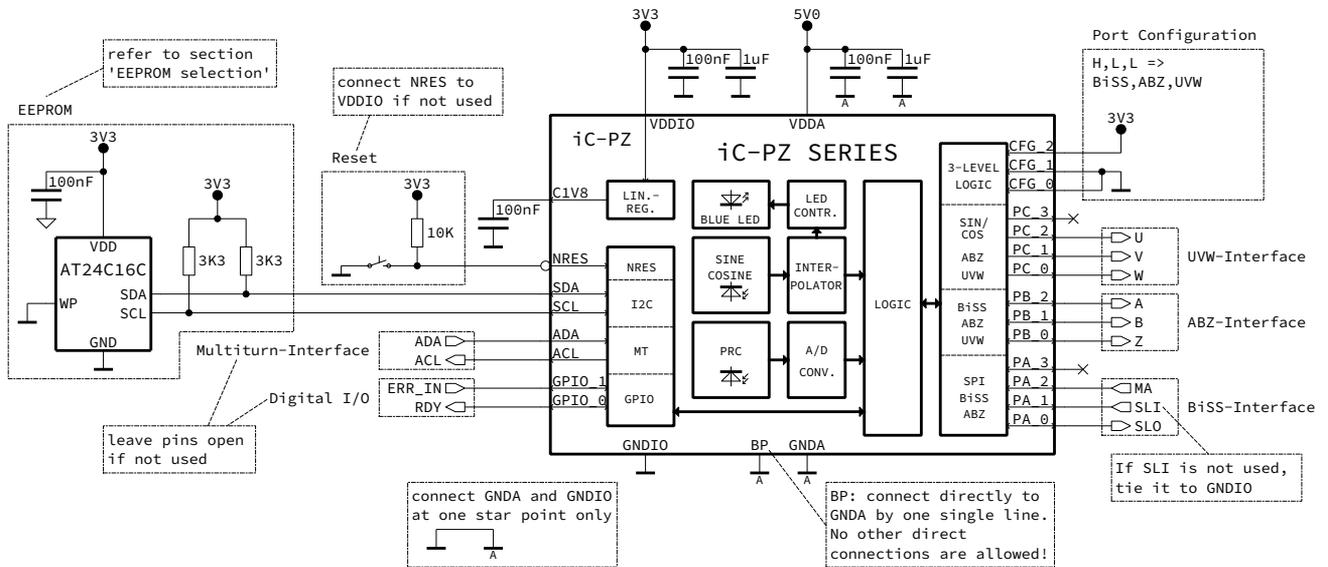


Figure 12: Circuit schematic with BiSS, ABZ and UVW port configuration

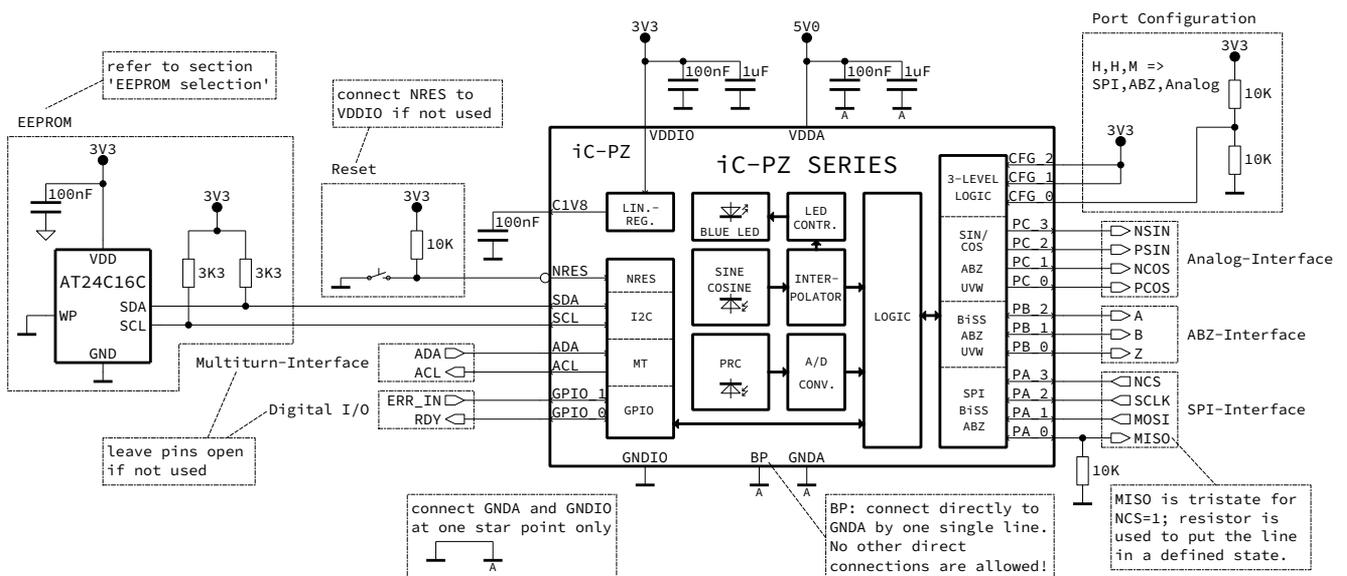


Figure 13: Circuit schematic with SPI, ABZ and analog port configuration

### MEMORY ORGANIZATION

Besides the on-chip RAM, data from an external EEPROM or other I2C devices can be accessed via iC-PZ. An overview of the memory organization is given in Table 3.

Registers are organized in banks. Banks 0x00..0x0E contain device configuration registers, which are located in the on-chip RAM. Registers 0x40..0x7F are not

assigned to banks and are accessed directly disregarding the active bank. Banks 0x20..0x3F can be used to access registers in an external EEPROM. Additional MT devices can be configured via bank 0x20. EDS and user data is stored in banks 0x24..0x3F. Additionally, data from up to four external I2C devices can be accessed using banks 0x40..0x4F.

Bank	Address	Content	Location	RPL
0x00..0x0E	0x00..0x3F	iC-PZ Device Configuration (volatile)	On-chip RAM	r/w
all	0x40..0x7F	iC-PZ Direct Access	On-chip RAM	n. a.
0x20	0x00..0x3F	MT Device Configuration (e. g. iC-PV)	EEPROM: 0x000 - 0x03F	r/w
X	X	iC-PZ Device Configuration (non-volatile)	EEPROM: 0x040 - 0x0FF	n. a.
0x24..0x3F	0x00..0x3F	EDS, User Data	EEPROM: 0x100 - 0x7FF	r/w
0x40..0x43	0x00..0x3F	I2C device 0 data	I2C device memory	n. a.
0x44..0x47	0x00..0x3F	I2C device 1 data	I2C device memory	n. a.
0x48..0x4B	0x00..0x3F	I2C device 2 data	I2C device memory	n. a.
0x4C..0x4F	0x00..0x3F	I2C device 3 data	I2C device memory	n. a.

Table 3: Memory organization overview

#### EEPROM Access

The external EEPROM is used to store the iC-PZ device configuration non-volatile, so that the data will be remaining after power-down. The memory addressing and data transmission to exchange the device configuration with the EEPROM is fully handled by iC-PZ, when receiving the appropriate command from any serial interface. Either a single bank or the complete device configuration can be read or written. When writing a bank to the EEPROM, the CRC value is calculated automatically by iC-PZ. Refer to COMMANDS on page 53 for details.

The device configuration data is secured by an 8-bit CRC value for every bank. Each CRC value has a Hamming Distance of 3 bits. After power-on, the data is read in bank by bank. In case a CRC value is incorrect, the bank is read again up to 3 times in total. If the CRC value of a bank is correct, the data is used. Otherwise, the bank uses its default data values. Invalid value are marked individually for each bank in [CRC\\_STAT](#) and as an error bit in [DIAG](#).

The EDS and user data is read and written immediately to/from the EEPROM, when accessing an address in the appropriate bank. As I2C is used for the communication between iC-PZ and EEPROM, data transmission will take a certain amount of time until completed. The corresponding address accessed in the EEPROM can be calculated according to the formula below:

$$EEPROM\_ADR = (BSEL - 0x20) * 0x40 + ADR$$

#### Bank Selection

The active bank is selected via [BSEL](#). Registers 0x40..0x7F are not affected and can always be accessed disregarding the active bank.

<b>BSEL(7:0)</b>		Addr. 0x40; bit 7:0	default: 0x00
Code	Value		
0x00..0x4F	Active bank		

Table 4: Bank Selection

#### Register Protection Level (RPL)

The banks containing device configuration, EDS and user data, can be individually protected from write and/or read access. Therefore, a Register Protection Level (RPL) can be set to the active bank by either executing the command [RPL\\_SET\\_RO](#) (read only) or [RPL\\_SET\\_NA](#) (no access, neither read nor write). To become persistent, the RPL settings have to be written to the EEPROM. When writing a complete bank from the on-chip RAM to the EEPROM, the RPL is stored automatically. For any other bank located in external memory, the RPL settings are stored in bank 0xF. By writing all banks to the EEPROM, all RPL settings become persistent. To check the RPL that is set for the active bank, the command [RPL\\_GET](#) can be executed.



Once the RPL is stored in the EEPROM, it can not be removed anymore. Nevertheless, setting the RPL from read only (RO) to no access (NA) is possible.

### REGISTER MAP

Register map								
Bank,Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Interface Ports Configuration</b>								
0x0, 0x00	PADM_CFG(1:0)		PADC_CFG(1:0)		PADB_CFG(1:0)		PADA_CFG(1:0)	
0x0, 0x01	-	PORTC_DIS	PORTB_DIS	PORTA_DIS	-		VDDIOSEL(1:0)	
<b>Interpolator</b>								
0x0, 0x02	<i>reserved [0x21]</i> <sup>1</sup>							
0x0, 0x03	IPO_FILT1(7:0)							
0x0, 0x04	<i>reserved [00000]</i> <sup>1</sup>					IPO_FILT2(2:0)		
0x0, 0x05	<i>reserved [0xE6]</i> <sup>1</sup>							
0x0, 0x06	<i>reserved [0x10]</i> <sup>1</sup>							
<b>System Definition</b>								
0x0, 0x07	SYS_OVR(3:0)				-		CD_FLIP	CD_INV
<b>Position Data Length</b>								
0x0, 0x08	-		ST_PDL(5:0)					
0x0, 0x09	-		MT_PDL(5:0)					
<b>Absolute Data Interface (ADI)</b>								
0x0, 0x0A	ADI_OS(4:0)				ADI_SBL(2:0)			
0x0, 0x0B	ADI_MSO(2:0)			ADI_EBP	ADI_EBL(3:0)			
0x0, 0x0C	ADI_CFG(7:1)							<i>reserved [0]</i> <sup>1</sup>
0x0, 0x0D	-							ADI_CFG(8)
<b>Singleturn Position Evaluation</b>								
0x0, 0x0E	-				<i>reserved [0x0]</i> <sup>1</sup>			
0x0, 0x0F	RAN_FLD	<i>reserved [001]</i> <sup>1</sup>			RAN_TOL(3:0)			
0x0, 0x10	CFG_CRC_0(7:0)							
<b>Adjustment Analog (static)</b>								
0x1, 0x00	COS_OFF(1:0)		-					
0x1, 0x01	COS_OFF(9:2)							
0x1, 0x02	SIN_OFF(1:0)		-					
0x1, 0x03	SIN_OFF(9:2)							
0x1, 0x04	SC_GAIN(1:0)		-					
0x1, 0x05	SC_GAIN(9:2)							
0x1, 0x06	SC_PHASE(1:0)		-					
0x1, 0x07	SC_PHASE(9:2)							
<b>Adjustment Digital (static)</b>								
0x1, 0x08	AI_PHASE(1:0)		-					
0x1, 0x09	AI_PHASE(9:2)							
0x1, 0x0A	AI_SCALE(0)	-						
0x1, 0x0B	AI_SCALE(8:1)							
0x1, 0x10	CFG_CRC_1(7:0)							
<b>Adjustment Analog (static + dynamic) – read-only, not stored in EEPROM</b>								
0x1, 0x20	COS_OFFS(1:0)		-					
0x1, 0x21	COS_OFFS(9:2)							
0x1, 0x22	SIN_OFFS(1:0)		-					
0x1, 0x23	SIN_OFFS(9:2)							

<sup>1</sup> Reserved registers must not be changed (default value in square brackets).

Register map									
Bank,Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
0x1, 0x24	SC_GAINS(1:0)					-			
0x1, 0x25	SC_GAINS(9:2)								
0x1, 0x26	SC_PHASES(1:0)					-			
0x1, 0x27	SC_PHASES(9:2)								
Adjustment Digital (static + dynamic) – read-only, not stored in EEPROM									
0x1, 0x28	AI_PHASES(1:0)					-			
0x1, 0x29	AI_PHASES(9:2)								
0x1, 0x2A	AI_SCALES(0)							-	
0x1, 0x2B	AI_SCALES(8:1)								
Adjustment Analog Configuration									
0x2, 0x00	SC_GAIN_SEL(3:0)				SC_OFF_SEL(3:0)				
0x2, 0x01	-				SC_PHASE_SEL(3:0)				
0x2, 0x02	<i>reserved [0x00]</i> <sup>1</sup>								
Adjustment Digital Configuration									
0x2, 0x03	AI_S_SEL(3:0)				AI_P_SEL(3:0)				
Adjustment Eccentricity (static)									
0x2, 0x04	ECC_AMP(7:0)								
0x2, 0x05	ECC_AMP(15:8)								
0x2, 0x06	ECC_AMP(23:16)								
0x2, 0x07	ECC_AMP(31:24)								
0x2, 0x08	ECC_PHASE(5:0)								-
0x2, 0x09	ECC_PHASE(13:6)								
0x2, 0x0A	-							ECC_EN	
0x2, 0x10	CFG_CRC_2(7:0)								
Analog Output									
0x3, 0x00	-			LED_CONST	LED_CUR(2:0)		LED_CTRL		
0x3, 0x01	-				ANA_OS	ANA_SEL(1:0)			
0x3, 0x10	CFG_CRC_3(7:0)								
ABZ Generator									
0x4, 0x00	ABZ_PER(7:0)								
0x4, 0x01	ABZ_PER(15:8)								
0x4, 0x02	ABZ_PER(23:16)								
0x4, 0x03	<i>reserved [00000]</i> <sup>1</sup>					ABZ_PER(26:24)			
0x4, 0x04	ABZ_HYS(7:0)								
0x4, 0x05	-				ABZ_MTD(3:0)				
0x4, 0x06	<i>reserved [0]</i> <sup>1</sup>	ABZ_CFG(2:0)			ABZ_ZGATE(3:0)				
0x4, 0x08	ABZ_OFF(7:0)								
0x4, 0x09	ABZ_OFF(15:8)								
0x4, 0x10	CFG_CRC_4(7:0)								
UVW Generator									
0x5, 0x00	-			UVW_PP(4:0)					
0x5, 0x01	-					UVW_CFG(1:0)			
0x5, 0x02	UVW_OFF(7:0)								
0x5, 0x03	UVW_OFF(15:8)								
0x5, 0x10	CFG_CRC_5(7:0)								

<sup>1</sup> Reserved registers must not be changed (default value in square brackets).

# iC-PZ Series

HIGH-RESOLUTION REFLECTIVE ABSOLUTE ENCODER



Rev F1, Page 25/85

Register map								
Bank,Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>BiSS/SSI Slave</b>								
0x6, 0x00	BISS_EM(7:0)							
0x6, 0x01	BISS_EM(15:8)							
0x6, 0x02	BISS_EM(23:16)							
0x6, 0x03	BISS_EM(31:24)							
0x6, 0x04	BISS_WM(7:0)							
0x6, 0x05	BISS_WM(15:8)							
0x6, 0x06	BISS_WM(23:16)							
0x6, 0x07	BISS_WM(31:24)							
0x6, 0x08	-	BISS_ST_DL(5:0)						
0x6, 0x09	-	BISS_MT_DL(5:0)						
0x6, 0x0A	-	SSI_EN	SSI_EXT	SSI_GRAY	-	BISS_NTOA	BISS_CRC16	BISS_ENSOL
0x6, 0x0B	-	BISS_CRCS(5:0)						
0x6, 0x10	CFG_CRC_6(7:0)							
<b>SPI Slave</b>								
0x7, 0x00	SPI_EM(7:0)							
0x7, 0x01	SPI_EM(15:8)							
0x7, 0x02	SPI_EM(23:16)							
0x7, 0x03	SPI_EM(31:24)							
0x7, 0x04	SPI_WM(7:0)							
0x7, 0x05	SPI_WM(15:8)							
0x7, 0x06	SPI_WM(23:16)							
0x7, 0x07	SPI_WM(31:24)							
0x7, 0x08	-	SPI_ST_DL(5:0)						
0x7, 0x09	-	SPI_MT_DL(5:0)						
0x7, 0x0A	-							SPI_EXT
0x7, 0x0B	-	SPI_CRCS(5:0)						
0x7, 0x10	CFG_CRC_7(7:0)							
<b>FlexCode®</b>								
0x8, 0x00	FCL(7:0)							
0x8, 0x01	-	FCL(14:8)						
0x8, 0x02	FCS(7:0)							
0x8, 0x03	-	FCS(14:8)						
0x8, 0x10	CFG_CRC_8(7:0)							
<b>GPIO</b>								
0x9, 0x00	GPIO0_M(7:0)							
0x9, 0x01	GPIO0_M(15:8)							
0x9, 0x02	GPIO0_M(23:16)							
0x9, 0x03	GPIO0_M(31:24)							
0x9, 0x04	GPIO1_M(7:0)							
0x9, 0x05	GPIO1_M(15:8)							
0x9, 0x06	GPIO1_M(23:16)							
0x9, 0x07	GPIO1_M(31:24)							
0x9, 0x08	GPIO1_SEL	GPIO1_DIAG	GPIO1_CFG(1:0)		GPIO0_SEL	GPIO0_DIAG	GPIO0_CFG(1:0)	
0x9, 0x10	CFG_CRC_9(7:0)							
<b>I2C Master</b>								
0xA, 0x00	I2C_DEV_ID_0(7:0)							

# iC-PZ Series

HIGH-RESOLUTION REFLECTIVE ABSOLUTE ENCODER



Rev F1, Page 26/85

Register map								
Bank,Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0xA, 0x01	I2C_DEV_ID_1(7:0)							
0xA, 0x02	I2C_DEV_ID_2(7:0)							
0xA, 0x03	I2C_DEV_ID_3(7:0)							
0xA, 0x04	I2C_T_0(6:0)							I2C_F_0
0xA, 0x05	I2C_T_1(6:0)							I2C_F_1
0xA, 0x06	I2C_T_2(6:0)							I2C_F_2
0xA, 0x07	I2C_T_3(6:0)							I2C_F_3
0xA, 0x10	CFG_CRC_A(7:0)							
Position Offset Singleturn								
0xB, 0x00	ST_OFF(7:0)							
0xB, 0x01	ST_OFF(15:8)							
0xB, 0x02	ST_OFF(23:16)							
0xB, 0x03	ST_OFF(31:24)							
0xB, 0x10	CFG_CRC_B(7:0)							
Position Offset Multiturn								
0xC, 0x00	MT_OFF(7:0)							
0xC, 0x01	MT_OFF(15:8)							
0xC, 0x02	MT_OFF(23:16)							
0xC, 0x03	MT_OFF(31:24)							
0xC, 0x10	CFG_CRC_C(7:0)							
Temperature Monitoring								
0xD, 0x00	TEMP_L_1(7:0)							
0xD, 0x01	-	TEMP_L_1(11:8)						
0xD, 0x02	TEMP_L_2(7:0)							
0xD, 0x03	-	TEMP_L_2(11:8)						
0xD, 0x04		-					TEMP_LT_2	TEMP_LT_1
0xD, 0x10	CFG_CRC_D(7:0)							
Profile (to be modified)								
0xE, 0x01	EDS_BANK_X(7:0)							
0xE, 0x02	BISS_PROFILE_ID_1_X(7:0)							
0xE, 0x03	BISS_PROFILE_ID_0_X(7:0)							
0xE, 0x04	SERIAL_3_X(7:0)							
0xE, 0x05	SERIAL_2_X(7:0)							
0xE, 0x06	SERIAL_1_X(7:0)							
0xE, 0x07	SERIAL_0_X(7:0)							
BiSS Identifier (to be modified)								
0xE, 0x08	DEV_ID_5_X(7:0)							
0xE, 0x09	DEV_ID_4_X(7:0)							
0xE, 0x0A	DEV_ID_3_X(7:0)							
0xE, 0x0B	DEV_ID_2_X(7:0)							
0xE, 0x0C	DEV_ID_1_X(7:0)							
0xE, 0x0D	DEV_ID_0_X(7:0)							
0xE, 0x0E	MFG_ID_1_X(7:0)							
0xE, 0x0F	MFG_ID_0_X(7:0)							
0xE, 0x10	CFG_CRC_E(7:0)							

Register map								
Bank,Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>Bank Selection</b>								
0x40	BSEL(7:0)							
<b>Profile</b>								
0x41	EDS_BANK(7:0)							
0x42	BISS_PROFILE_ID_1(7:0)							
0x43	BISS_PROFILE_ID_0(7:0)							
0x44	SERIAL_3(7:0)							
0x45	SERIAL_2(7:0)							
0x46	SERIAL_1(7:0)							
0x47	SERIAL_0(7:0)							
<b>Revision &amp; Identification</b>								
0x48	ID(7:0)							
0x49	ID(15:8)							
0x4A	ID(23:16)							
0x4B	ID(31:24)							
0x4C	SYS(3:0)				REV(3:0)			
<b>GPIO</b>								
0x4D	-				GPIO_IN(1:0)		GPIO_OUT(1:0)	
<b>Temperature Sensor</b>								
0x4E	TEMP(7:0)							
0x4F	TEMP(15:8)							
<b>Presets</b>								
0x50	ST_PRE(7:0)							
0x51	ST_PRE(15:8)							
0x52	ST_PRE(23:16)							
0x53	ST_PRE(31:24)							
0x54	MT_PRE(7:0)							
0x55	MT_PRE(15:8)							
0x56	MT_PRE(23:16)							
0x57	MT_PRE(31:24)							
0x58	ABZ_PRE(7:0)							
0x59	ABZ_PRE(15:8)							
0x5A	UVW_PRE(7:0)							
0x5B	UVW_PRE(15:8)							
<b>Autocalibration Configuration</b>								
0x5C	AC_SEL2(3:0)				AC_SEL1(3:0)			
0x5D	AC_ETO	reserved [000] <sup>1</sup>				AC_COUNT(3:0)		
<b>Absolute Data Interface (ADI) – received synchronization bits</b>								
0x5E	-				ADI_SB(3:0)			
<b>I2C Device Data</b>								
0x60	I2C_DATA_0(7:0)							
0x61	I2C_DATA_0(15:8)							
0x62	I2C_DATA_1(7:0)							
0x63	I2C_DATA_1(15:8)							
0x64	I2C_DATA_2(7:0)							

<sup>1</sup> Reserved registers must not be changed (default value in square brackets).

Register map								
Bank,Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x65					I2C_DATA_2(15:8)			
0x66					I2C_DATA_3(7:0)			
0x67					I2C_DATA_3(15:8)			
Diagnosis								
0x68					DIAG(7:0)			
0x69					DIAG(15:8)			
0x6A					DIAG(23:16)			
0x6B					DIAG(31:24)			
0x6C					ERR(7:0)			
0x6D					ERR(15:8)			
0x6E					ERR(23:16)			
0x6F					ERR(31:24)			
0x70					WARN(7:0)			
0x71					WARN(15:8)			
0x72					WARN(23:16)			
0x73					WARN(31:24)			
Bank CRC Status								
0x74					CRC_STAT(7:0)			
0x75					CRC_STAT(15:8)			
Commands								
0x76					CMD_STAT(7:0)			
0x77					CMD(7:0)			
BiSS Identifier								
0x78					DEV_ID_5(7:0)			
0x79					DEV_ID_4(7:0)			
0x7A					DEV_ID_3(7:0)			
0x7B					DEV_ID_2(7:0)			
0x7C					DEV_ID_1(7:0)			
0x7D					DEV_ID_0(7:0)			
0x7E					MFG_ID_1(7:0)			
0x7F					MFG_ID_0(7:0)			

Table 5: Register map

Please note: Register marked with '-' are not implemented. They cannot be written to and are always read as '0'.

### INTERFACE PORTS CONFIGURATION

#### Slew Rate I/O-Pads

The output slew rates for the digital I/O pads of PORTA, PORTB, PORTC, and the Absolute Data Interface are configured via [PADx\\_CFG\(1:0\)](#). Refer to ELECTRICAL CHARACTERISTICS 229 and 230 for details. Faster drivers support higher output frequencies but may increase interference.

Code	Description
00	Slow output driver (e.g. I2C, ADI-slow)
01	Medium output driver (e.g. ADI-fast)
10	Fast output driver (e.g. BiSS, ABZ)
11	Ultra fast output driver

Table 6: I/O Pads Configuration

#### Port Disabling

If ports are not used, disabling them via [PORTx\\_DIS](#) is recommended to prevent noise caused by potential crosstalk.

Bit	Description
0	Port is enabled
1	Port is disabled

Table 7: Port Disabling

#### VDDIO Monitoring

The VDDIO supply voltage can be monitored for undervoltage. Therefore, [VDDIOSEL](#) has to be set accordingly. In table 8 the nominal supply voltages for VDDIO are given. For details on the monitored undervoltage-levels, refer to section Power-On Reset & Voltage Monitoring in the ELECTRICAL CHARACTERISTICS.

Code	Description
00	VDDIO = 2.5 V
01	VDDIO = 3.3 V
10	VDDIO = 5.0 V
11	Illegal

Table 8: VDDIO Selection

### LED POWER CONTROL

The embedded blue LED is controlled by an on-chip regulator. One of two control modes can be selected via **LED\_CTRL**. The signal path is shown in Figure 14.

In Square Control Mode, the amplitudes of the analog signals are kept at a constant level. The LED power control uses the signal at the input of the interpolator as process value. Signal amplitude is typically 1000 mV.

In Sum Control Mode, the DC values of the analog signals are kept constant. In that case, the LED power control uses the signals at the outputs of the photocurrent amplifiers as process value. Signal amplitudes vary depending on the contrast of the system.

The control current of the LED can be limited to a maximum level via **LED\_CUR**. This is useful to avoid peak LED currents, especially during startup.

The LED power control can be switched off via **LED\_CONST**. If so, the LED is running with a constant current. In that case the current is set according to the value for **LED\_CUR**. Signal DC levels and amplitudes depend on the contrast of the system and the distance of the chip to the code disc. As the LED current is not controlled in a closed loop, this mode is suitable for signal adjustment and alignment. By setting **LED\_CUR** to 0x1 (0 mA), the LED can be switched off.

<b>LED_CTRL</b>		Addr. 0x3, 0x00; bit 0	default: 0
Code	Description		
0	Square Control Mode		
1	Sum Control Mode		

Table 9: LED Control Mode

<b>LED_CUR(2:0)</b>		Addr. 0x3, 0x00; bit 3:1	default: 0x0
Code	Description		
0x0	40 mA		
0x1	0 mA		
0x2	1 mA		
0x3	2 mA		
0x4	4 mA		
0x5	8 mA		
0x6	16 mA		
0x7	32 mA		

Table 10: LED Current Limit

<b>LED_CONST</b>		Addr. 0x3, 0x00; bit 4	default: 0
Code	Description		
0	Controlled current with limit		
1	Constant LED current		

Table 11: LED Constant Current

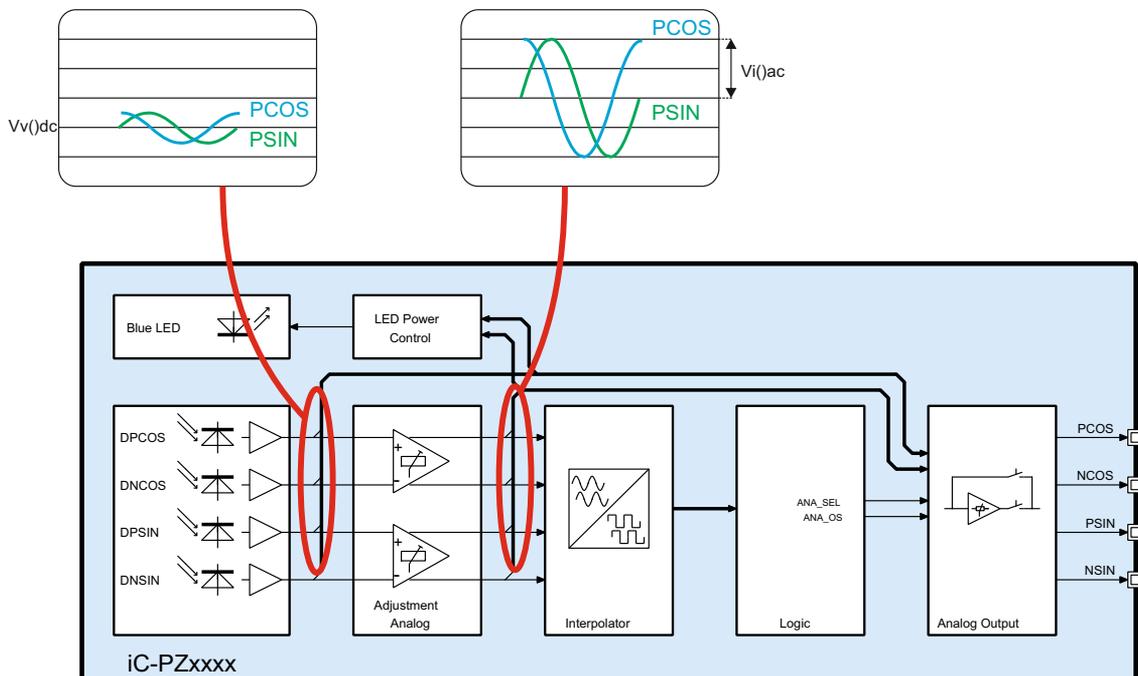


Figure 14: Analog signal paths and LED control in square and sum control mode

### INTERPOLATOR

A digital control loop provides a filtered interpolator position. For setting up the filter, two use cases are distinguished:

Use case	Description
Before analog adjustment	First system startup, analog autocalibration not yet executed
After analog adjustment	Analog autocalibration completed

Parameters `IPO_FILT1` and `IPO_FILT2` should be set according to the use-cases above. The default values `IPO_FILT1 = 0x6E` and `IPO_FILT2 = 0x4` ensure stable filter operation before the first analog autocalibration after startup is executed. After the first analog autocalibration has been completed, the parameters should be changed to `IPO_FILT1 = 0xEA` and `IPO_FILT2 = 0x4`.

<code>IPO_FILT1(7:0)</code>		Addr. 0x0, 0x03; bit 7:0	default: 0x6E
Code	Use case		
0x6E	Before analog adjustment		
0xEA	After analog adjustment		

Table 12: Filter Parameter 1

<code>IPO_FILT2(2:0)</code>		Addr. 0x0, 0x04; bit 2:0	default: 0x4
Code	Use case		
0x4	Suitable for any use case		

Table 13: Filter Parameter 2

### SINGLETURN POSITION EVALUATION

The absolute position information is provided by the sampled Pseudo Random Code (PRC) track. The sampled incremental track is interpolated to increase the resolution of the singleturn position. An internal counter is implemented that is incremented with each new sample of the incremental track. That counter is initialized during startup with the first sampled absolute position.

During operation, each sampled absolute position is compared to the internally counted position. In case both values do not match, the counted position is replacing the sampled position. This way the system provides a mechanism to mask single misreadings during PRC sampling.

An error is reported in `DIAG(11)` as soon as the tolerance for mismatches of the sampled and counted position set via `RAN_TOL` is exceeded. In case of an error, the counter value is either kept (`RAN_FLD = 0`) or reloaded with the sampled absolute position (`RAN_FLD = 1`). This feature is meant to provide increased availability of a valid position information in test environments, e. g. using a dirty code disc in the lab.

<code>RAN_TOL(3:0)</code>		Addr. 0x0, 0x0F; bit 3:0	default: 0x4
Code	Value		
0x0	No mismatches tolerated		
0x1	Low mismatching tolerance		
...	...		
0x4	Medium mismatching tolerance		
...	...		
0xF	High mismatching tolerance		

Table 14: PRC Mismatching Tolerance

<code>RAN_FLD</code>		Addr. 0x0, 0x0F; bit 7	default: 1
Code	Description		
0	Internal counter will never be reloaded after startup		
1	Internal counter is reloaded in case of error <code>DIAG(11)</code>		

Table 15: PRC Forced Loading



Setting `RAN_TOL = 0x0` and not receiving an error in `DIAG(11)` during operation indicates that zero misreadings have occurred.



If the internal counter is initialized with a wrong absolute position during startup (e. g. due to dirt on the PRC track of the disc), the error in `DIAG(11)` will not be set until `RAN_TOL` is exceeded.

### ANALOG OUTPUT

#### Signal Routing

The signals routed to the analog output port can be selected via **ANA\_SEL**. As shown in Figure 15, three routings along the analog signal path are selectable.

<b>ANA_SEL(1:0)</b>		Addr. 0x3, 0x01; bit 1:0	default: 00
Code	Description		
00	Buffered signals, amplitude and driver set by <b>ANA_OS</b>		
10	Pre Signal Conditioning signals (photocurrent amplifiers, weak output)		
11	Post Signal Conditioning signals (weak output)		

Table 16: Analog Signal Selection

By setting **ANA\_SEL** = 00, the analog signals are adjusted and amplified before being routed to the analog output port. This setting is recommended to be used during operation. Driver strength is selected via **ANA\_OS**.

By setting **ANA\_SEL** = 10, the raw signals of the photocurrent amplifiers are routed to the analog output port. Those signals can be used for testing and alignment purposes only and may not see any load, as they do only have weak drivers.

By setting **ANA\_SEL** = 11, the signals being affected by the analog adjustment are routed to the analog output port. Those signals can also be used for testing and alignment purposes only and may not see any load, as they do only have weak drivers.

#### Output Driver

The driver strength and amplitude of the analog output port is set via **ANA\_OS**. The driver is only applied to the routing **ANA\_SEL** = 00.

<b>ANA_OS</b>		Addr. 0x3, 0x01; bit 2	default: 1
Code	Description		
0	1 mA, 1000 mV amplitude (amplification x1), 2.5 V dc		
1	6 mA, 250 mV amplitude (amplification x0.25), 2.5 V dc		

Table 17: Analog Output Setting

**Note:** The amplitudes described for **ANA\_OS** are only valid for **LED\_CTRL** = 0 and **LED\_CONST** = 0. For other settings, the amplitudes are not controlled and depend on the system properties like the chosen LED power control (sum or constant LED current) or the contrast of the system.

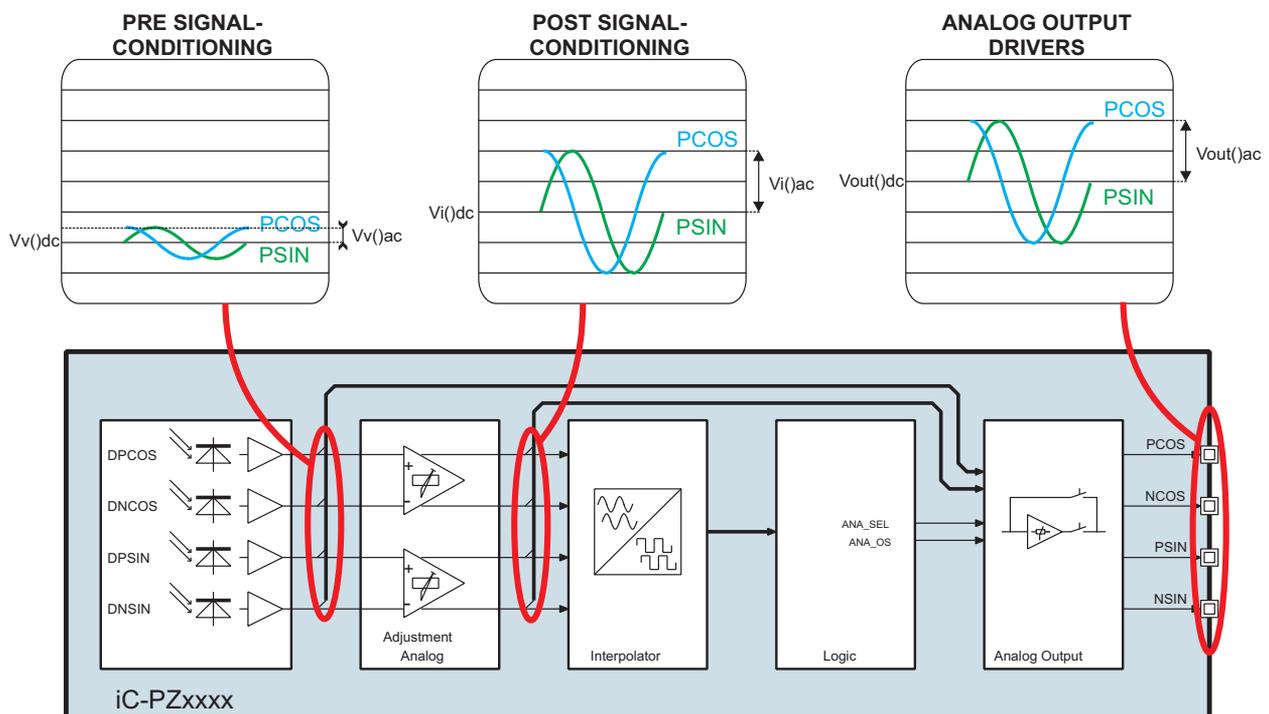


Figure 15: Analog signal path with single-ended signals referenced to GNDA (NCOS and NSIN not shown)

### ABZ GENERATOR

#### AB Periods

Making use of the FlexCount® feature, the resolution of the incremental (quadrature) signals can be arbitrarily adjusted. The AB periods per mechanical revolution (rotary) or per maximum scale length (linear) can be set via **ABZ\_PER**.

Code	AB periods
0x0000000	$2^{27}$
0x0000001	1
0x0000002	2
...	...
0x0002000	8192
...	...
0x7FFFFFFD	$2^{27} - 3$
0x7FFFFFFE	$2^{27} - 2$
0x7FFFFFFF	$2^{27} - 1$

Note: The maximum value allowed for the AB periods depends on the resolution of the system:  $ABZ\_PER \leq 2^{SYS\_eff+12}$ .

Example: For the rotary system with  $\varnothing$  26 mm ( $SYS\_eff = 8$ ), the maximum value allowed is  $= 2^{8+12} = 2^{20}$ .

Table 18: AB Periods

For rotary systems, **ABZ\_PER** defines the number of AB periods per mechanical revolution.

For linear systems with a native resolution of  $r_{nat}$  (line distance of the incremental track on the scale), one AB period corresponds to the length  $l_{AB}$  of:

$$l_{AB} = \frac{2^{SYS\_eff}}{ABZ\_PER} \cdot r_{nat}$$

Example:

For iC-PZ205,  $r_{nat} = 204.8 \mu m$  and  $SYS\_eff = 15$   
 To set one AB period corresponding to the length  $l_{AB} = 1.6 \mu m$ , **ABZ\_PER** has to be set to:

$$ABZ\_PER = \frac{r_{nat}}{l_{AB}} \cdot 2^{SYS\_eff} = \frac{204.8 \mu m}{1.6 \mu m} \cdot 2^{15} = 2^{22}$$

The speed of a linear system is limited to

$$v_{max} = \frac{l_{AB}}{4 \cdot MTD}$$

Example:

In the above example with  $l_{AB} = 1.6 \mu m$  and the maximum  $MTD = 37.5 ns$ , the maximum possible speed is

$$v_{max} = \frac{1.6 \mu m}{4 \cdot 37.5 ns} = 10.7 \frac{m}{s}$$



Independent of the above calculation, the speed must not exceed the limit given in Table 145.

#### AB Direction

The direction of the incremental signals can be switched via **ABZ\_CFG(0)**. The signals for all possible use cases are illustrated in Figure 16.

Code	AB direction
0	A leading B for pos. mech. direction of movement
1	B leading A for pos. mech. direction of movement

Table 19: AB Direction

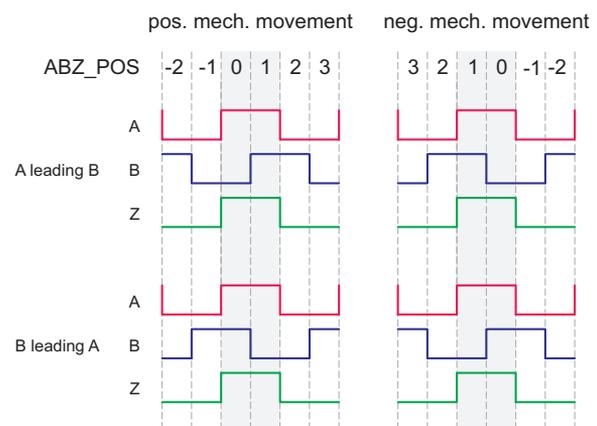


Figure 16: ABZ signals for different mechanical and electrical directions of movement (here: **ABZ\_ZGATE = 0x4**)

### Z Gating

Several gating options for the index signal Z can be configured via **ABZ\_ZGATE**.

ABZ_ZGATE(3:0)		Addr. 0x4, 0x06; bit 3:0	default: 0x0
Code	Function		
0x0	Z at AB = 10	90° Z gating options	
0x1	Z at AB = 11		
0x2	Z at AB = 01		
0x3	Z at AB = 00		
0x4	Z at A = 1	180° Z gating options	
0x5	Z at A = 0		
0x6	Z at B = 1		
0x7	Z at B = 0		
0x8	Z at A	360° Z gating options (ungated)	
0x9	Z at B		
Others	Not allowed		
Note: Z is always located at the internal zero position, AB is adapted with respect to Z.			

Table 20: Z Gating

ABZ signals for mechanical movement in positive direction (see SIGNAL DEFINITIONS on page 18) and **ABZ\_CFG(0) = 0** (A leading B) are shown in Figure 17.

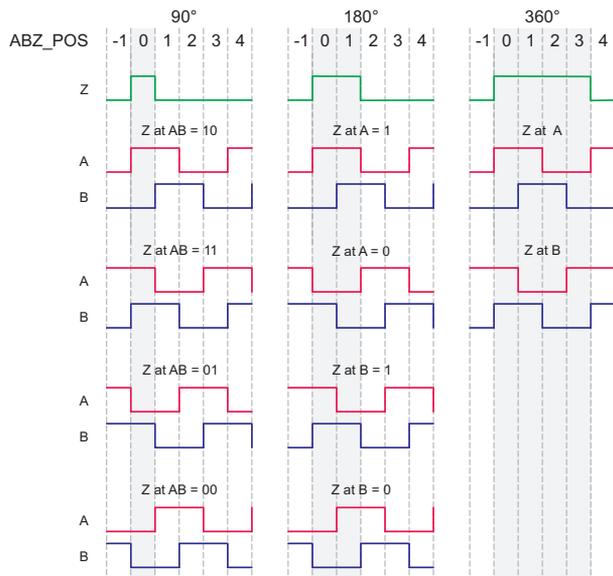


Figure 17: Gating and position of the Z signal

### Z Polarity

The polarity of Z can be switched via **ABZ\_CFG(1)**.

ABZ_CFG(1)		Addr. 0x4, 0x06; bit 5	default: 0
Code	Z polarity		
0	Standard output (Z active high)		
1	Inverted output (Z active low)		

Table 21: Z Polarity

### Hysteresis

As illustrated in Figure 18, the configurable hysteresis **ABZ\_HYS** corresponds to a slip existing between the two rotating directions. In this way multiple switching of the ABZ signals at the reversing point of a changing direction of movement can be prevented.

ABZ_HYS(7:0)		Addr. 0x4, 0x04; bit 7:0	default: 0x20
Code	ABZ hysteresis	Value	
unsigned	$2 \cdot \text{ABZ\_HYS} \cdot \text{LSB}$	$\frac{2 \cdot \text{ABZ\_HYS}}{2^{14}} \cdot 360^\circ e$	
0x00	0 LSB	0.000°e	
0x01	2 LSB	0.044°e	
...	...	...	
0x20	64 LSB	1.406°e	
...	...	...	
0xFE	508 LSB	11.162°e	
0xFF	510 LSB	11.206°e	
Note: In FlexCode®-systems, "Value" changes as described in FLEXCODE® on page 69.			

Table 22: ABZ Hysteresis

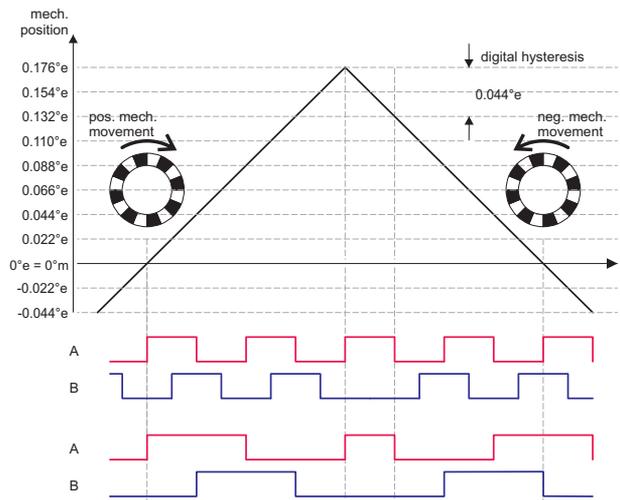


Figure 18: AB signals with 2 LSB hysteresis. Signals for max. AB resolution (top,  $2^{12}$  AB-Periods per sine-period) and for half of the max. resolution (bottom,  $2^{11}$  AB-Periods per sine-period)

### Minimum Transition Distance

The Minimum Transition Distance (MTD) sets the minimum time between two successive edges of the A and B signals. Setting `ABZ_MTD` is useful to avoid errors in the motor controller. Some controllers need a Minimum Transition Distance, and at speeds close to the maximum controllable speed AB-jitter may cause some faster edges.

Code	ABZ minimum transition distance
0x0	37.5 ns
0x1	50.0 ns
0x2	62.5 ns
0x3	75.0 ns
0x4	87.5 ns
0x5	100.0 ns
0x6	112.5 ns
0x7	125.0 ns
0x8	250.0 ns
0x9	500.0 ns
0xA	750.0 ns
0xB	1.0 $\mu$ s
0xC	2.5 $\mu$ s
0xD	5.0 $\mu$ s
0xE	7.5 $\mu$ s
0xF	10.0 $\mu$ s

Note: The given times are typical values, i.e., they hold for  $f_{osc} = 80$  MHz.

Table 23: ABZ Minimum Transition Distance

If the movement of the motor causes faster AB signals than the MTD (fine-dashed signals in Figure 19), the AB-edges will be output **incorrectly** with the fixed MTD between two successive edges. However, if AB-signals are continuously output with MTD, the internal AB-position increasingly differs from the actual absolute position. If the difference is too large, AB-calculation will no longer work correctly, and incorrect AB-signals will be output, e.g. with incorrect direction.

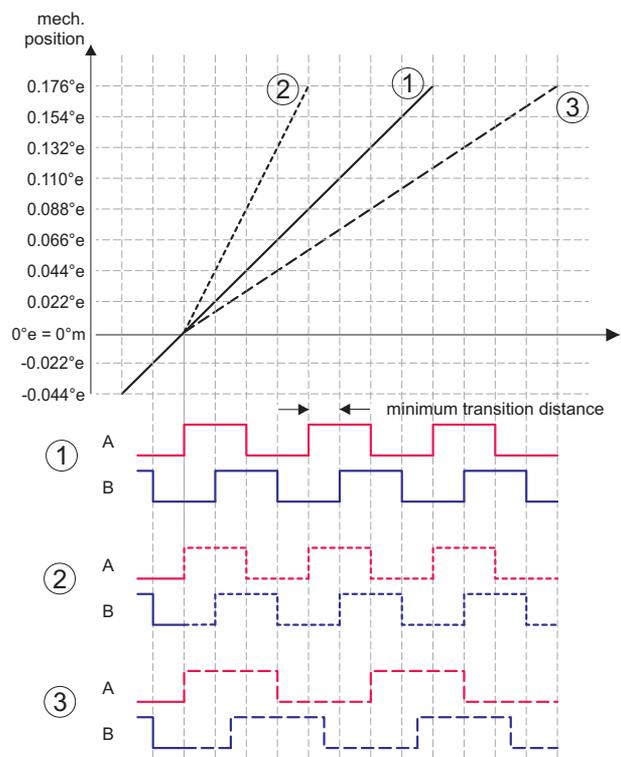


Figure 19: Minimum Transition Distance (MTD)

- ① equal to MTD
- ② faster than MTD
- ③ slower than MTD

### ABZ Offset

An individual offset can be applied to the ABZ signals via **ABZ\_OFF**. The offset is added before the ABZ signals are transferred, as shown in Figure 21.

<b>ABZ_OFF(7:0)</b>		Addr. 0x4, 0x08; bit 7:0	default: 0x0000
<b>ABZ_OFF(15:8)</b>		Addr. 0x4, 0x09; bit 7:0	
Code	ABZ offset	Value in °m	
Unsigned		$\frac{ABZ\_OFF}{2^{16}} \cdot 360^\circ\text{m}$	
0x0000	0	0.0000	
0x0001	1	0.0055	
0x0002	2	0.0110	
...	...	...	
0xFFFFD	$2^{16} - 3$	359.9835	
0xFFFFE	$2^{16} - 2$	359.9890	
0xFFFFF	$2^{16} - 1$	359.9945	

Table 24: ABZ Offset

### ABZ Preset

To automatically calculate the offset that is related to a certain position, a preset can be applied. First, the desired preset has to be set via **ABZ\_PRE**. Then, by executing the command **ABZ\_PRESET**, the offset is calculated so that the current position is set according to **ABZ\_PRE**. Refer to COMMANDS on page 53 for details.

For linear systems, **ABZ\_OFF** is calculated similarly,  $360^\circ\text{m}$  has to be replaced with the max. length of the scale. For iC-PZ205, e.g., the max. scale length is  $204.8 \mu\text{m} \cdot 2^{15} \approx 6.71 \text{m}$ .

<b>ABZ_PRE(7:0)</b>		Addr. 0x58; bit 7:0	default: 0x0000
<b>ABZ_PRE(15:8)</b>		Addr. 0x59; bit 7:0	
Value	Preset value for ABZ interface		

Table 25: ABZ Preset

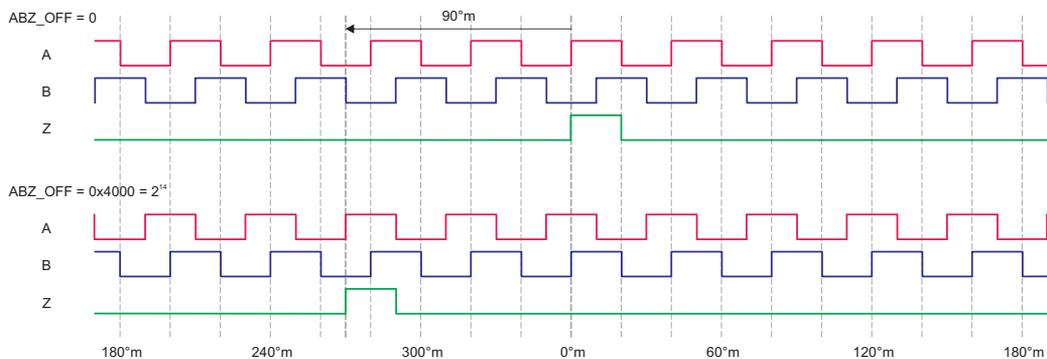


Figure 20: ABZ offset for a system with 9 periods

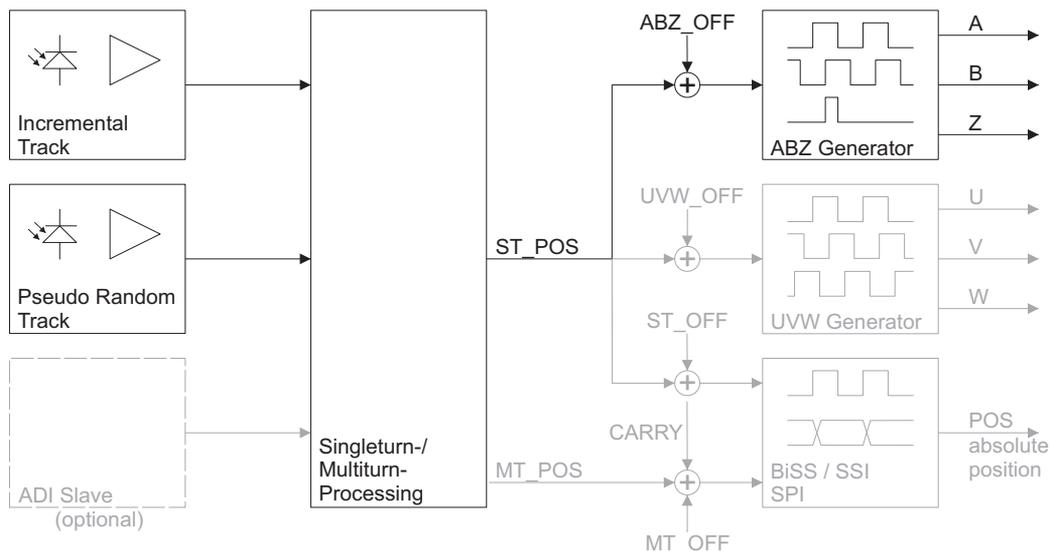


Figure 21: Programming of position offsets

### Startup

The startup behavior of the ABZ generator is configured via [ABZ\\_CFG\(2\)](#).

ABZ_CFG(2)		Addr. 0x4, 0x06; bit 6	default: 0
Code	ABZ Startup Behavior		
0	AB = 11, Z = 0 during startup, until position found		
1	ABZ counting from 0 position to actual position with programmed MTD		

Table 26: ABZ Startup Behavior

Startup behavior of the ABZ generator for both settings of [ABZ\\_CFG\(2\)](#) is illustrated in Figure 22 and Figure 23. For better orientation, different phases of the system startup are distinguished:

### RESET

iC-PZ is in the reset state, either due to power-down, external pin NRES or command [REBOOT](#).

### READ CFG

Configuration is read from an external EEPROM.

### STUP POS

Startup phase of the position processing (MT + ST).

### WAIT

Waiting time >50 μs, before the ABZ generator is enabled.

### STUP ABZ

Startup phase of the ABZ generator.

### NORMAL OP

Normal operation (output of valid ABZ signals, as configured).

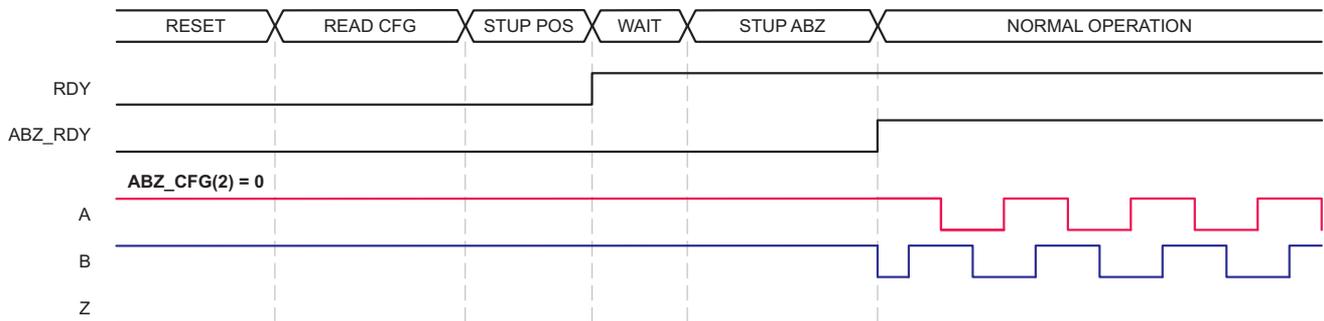


Figure 22: ABZ startup with ABZ\_CFG(2) = 0

For [ABZ\\_CFG\(2\)](#) = 0 the fastest MTD of 37.5 ns is used internally to minimize the duration of the startup phase. Once the position is found, ABZ is then immediately output with the programmed MTD. At this point the invalid AB-transition 11 → 00 is possible. As

illustrated in Figure 22, users can evaluate status bit [DIAG\(6\)](#) = not(ABZ\_RDY) (e. g. via GPIO). Once this bit becomes 1, the ABZ startup is finished and valid signals are output henceforth.

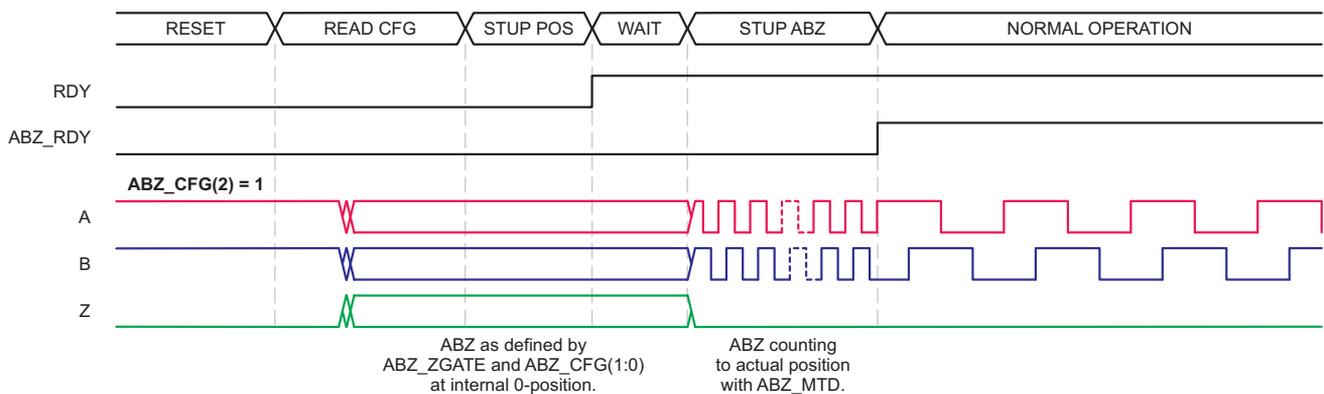


Figure 23: ABZ startup with ABZ\_CFG(2) = 1

For [ABZ\\_CFG\(2\)](#) = 1 it is possible that ABZ changes during EEPROM readout, as the levels of ABZ at the 0 position depend on the configuration. As illustrated in Figure 23, users can evaluate status bit

[DIAG\(24\)](#) = not(RDY) (e. g. via GPIO). Once this bit becomes 1, a valid position (ST + MT) is available via the serial interfaces and ABZ is enabled (begins to count to the position) at least 50 μs later.

### UVW GENERATOR

#### Pole Pairs, Direction and Polarity

The UVW generator of iC-PZ provides motor commutation signals for up to 32 pole pairs.

UVW_PP(4:0)	Addr. 0x5, 0x00; bit 4:0	default: 0x04
Code	UVW Pole Pairs	
0x00	32	
0x01	1	
...	...	
0x04	4	
...	...	
0x1F	31	

Table 27: UVW Pole Pairs per Mechanical Revolution

For rotary systems, **UVW\_PP** defines the number of UVW pole pairs per mechanical revolution.

For linear systems with a native resolution of  $r_{nat}$  (line distance of the incremental track on the scale), one UVW period corresponds to the length  $l_{UVW}$  of:

$$l_{UVW} = \frac{2^{SYS\_eff}}{UVW\_PP} \cdot r_{nat}$$

Example:

For iC-PZ205,  $r_{nat} = 204.8 \mu m$  and  $SYS\_eff = 15$ .

For **UVW\_PP** = 0x8, the length of one UVW period is:

$$l_{UVW} = \frac{2^{15}}{8} \cdot 204.8 \mu m \approx 0.84 m$$

Via **UVW\_CFG(0)** the direction and via **UVW\_CFG(1)** the polarity of the UVW signals can be adjusted, as shown in Figure 24.

UVW_CFG(0)	Addr. 0x5, 0x01; bit 0	default: 0
Code	UVW direction	
0	Standard direction	
1	Reversed direction	

Table 28: UVW Direction

UVW_CFG(1)	Addr. 0x5, 0x01; bit 1	default: 0
Code	UVW polarity	
0	Standard polarity	
1	Inverted polarity (180° phase shift)	

Table 29: UVW Polarity

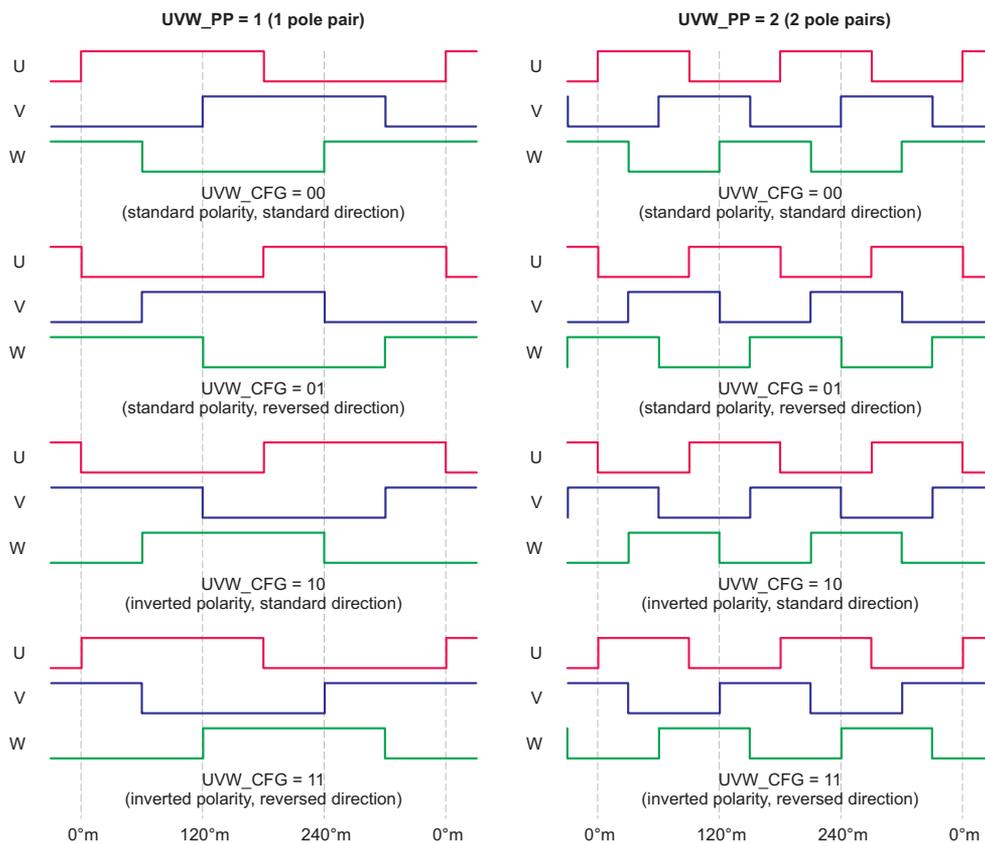


Figure 24: UVW pole pairs, direction and polarity for pos. mech. movement

### UVW Offset

The offset **UVW\_OFF** is added to the internal position before being used by the UVW generator (Figure 26). Figure 25 illustrates the effect of **UVW\_OFF**.

For linear systems, **UVW\_OFF** is calculated similarly, 360°m has to be replaced with the max. length of the scale. For iC-PZ205, e.g., the max. scale length is  $204.8\mu\text{m} \cdot 2^{15} \approx 6.71\text{ m}$

<b>UVW_OFF(7:0)</b>		Addr. 0x5, 0x02; bit 7:0	default: 0x0000
<b>UVW_OFF(15:8)</b>		Addr. 0x5, 0x03; bit 7:0	
Code	UVW position offset		
Unsigned		$\frac{\text{UVW\_OFF}}{2^{16}} \cdot 360^\circ\text{m}$	
0x0000	0	0.0000	
0x0001	1	0.0055	
0x0002	2	0.0110	
...	...	...	
0xFFFFD	$2^{16} - 3$	359.9835	
0xFFFFE	$2^{16} - 2$	359.9890	
0xFFFFF	$2^{16} - 1$	359.9945	

Table 30: UVW Offset

### UVW Preset

To automatically calculate the offset that is related to a certain position, a preset can be applied. First, the desired preset has to be set via **UVW\_PRE**. Then, by executing the command **UVW\_PRESET**, the offset is calculated so that the current position is set according to **UVW\_PRE**. Refer to COMMANDS on page 53 for details.

<b>UVW_PRE(7:0)</b>		Addr. 0x5A; bit 7:0	default: 0x0000
<b>UVW_PRE(15:8)</b>		Addr. 0x5B; bit 7:0	
Value	Preset value for UVW interface		

Table 31: UVW Preset

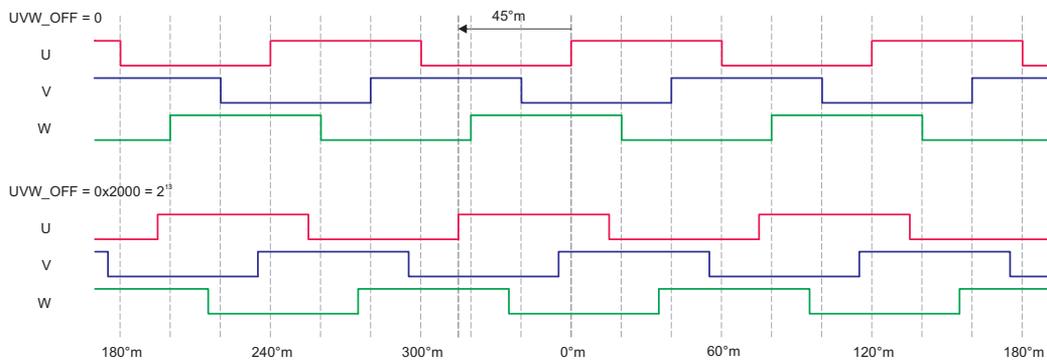


Figure 25: UVW offset for a system with 3 pole pairs

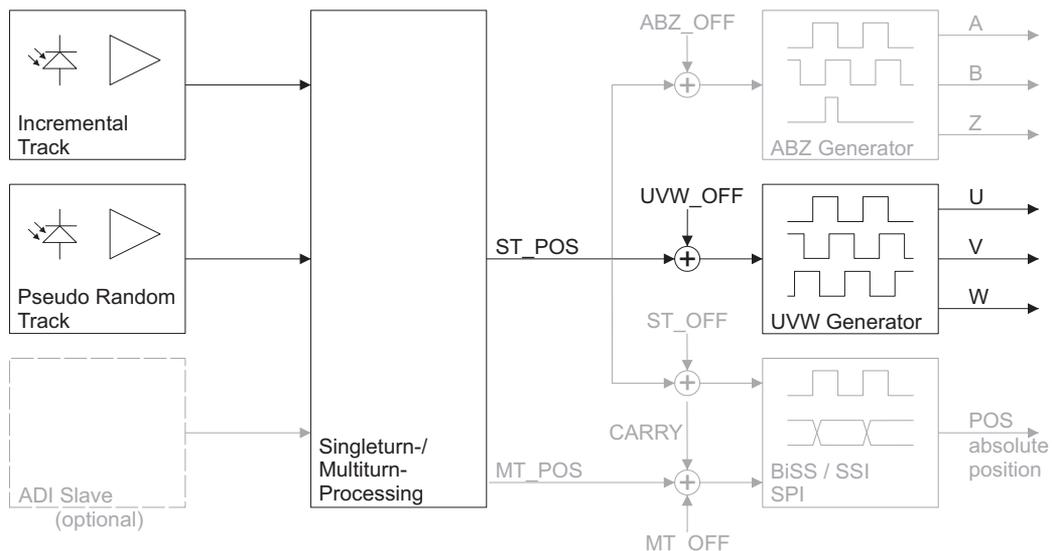


Figure 26: Programming of position offsets

### Startup

During startup, the UVW generator outputs constant signal levels, i.e., UVW = 000, until valid signals can be output.

Figure 27 shows the UVW startup behavior in the context of the overall system startup, where different phases can be distinguished:

- RESET** iC-PZ is in the reset state, either due to power-down, external pin NRES or command **REBOOT**.
- READ CFG** The configuration is read from an external EEPROM.

- STUP POS** Startup phase of the position processing (MT + ST).
- WAIT** Waiting time >50  $\mu$ s, before the UVW generator is enabled.
- STUP UVW** Startup phase of the UVW generator.
- NORMAL OP** Normal operation (output of valid UVW signals, as configured).

As illustrated in Figure 27, users can evaluate status bit **DIAG(7) = not(UVW\_RDY)** (via GPIO). Once this bit becomes 1, the UVW startup is finished and valid signals are output henceforth.

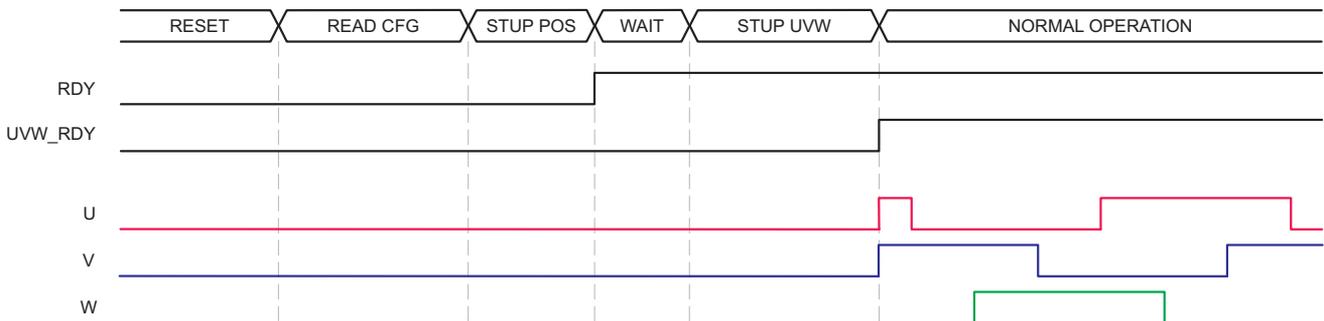


Figure 27: UVW startup

### POSITION SETTINGS

#### Position Data Length

The number of bits used for the singleturn (ST) and multiturn (MT) position data can be set via [ST\\_PDL](#) and [MT\\_PDL](#). Those settings affect the position data sent by any serial interface (BiSS, SSI and SPI). However, the position data format can be adjusted individually for any of those interfaces. Refer to BiSS SLAVE, SSI SLAVE and SPI SLAVE for details.

<b>ST_PDL(5:0)</b>		Addr. 0x0, 0x08; bit 5:0	default: 0x20
Code	Value		
0..32	Maximum number of singleturn position data bits. Unused bits are set to zero from right to left.		

Table 32: Singleturn Position Data Length

<b>MT_PDL(5:0)</b>		Addr. 0x0, 0x09; bit 5:0	default: 0x20
Code	Value		
0..32	Maximum number of multiturn position data bits. Unused bits are set to zero from left to right.		

Table 33: Multiturn Position Data Length

**Note:** Further hints on properly setting [MT\\_PDL](#) when using external MT slaves are given on page 55.

#### Position Offset

An individual offset can be applied to the ST and MT position via [ST\\_OFF](#) and [MT\\_OFF](#). The offset is added before the position data is transferred by one of the serial interfaces as shown in Figure 28.

<b>ST_OFF(7:0)</b>		Addr. 0xB, 0x00; bit 7:0	default:
<b>ST_OFF(15:8)</b>		Addr. 0xB, 0x01; bit 7:0	0x00000000
<b>ST_OFF(23:16)</b>		Addr. 0xB, 0x02; bit 7:0	
<b>ST_OFF(31:24)</b>		Addr. 0xB, 0x03; bit 7:0	
Code	Singleturn position offset		
0x00000000	0		
0x00000001	1		
0x00000002	2		
...	...		
0xFFFFFFFFD	$2^{32} - 3$		
0xFFFFFFFFE	$2^{32} - 2$		
0xFFFFFFFFF	$2^{32} - 1$		

Table 34: Singleturn Position Offset

<b>MT_OFF(7:0)</b>		Addr. 0xC, 0x00; bit 7:0	default:
<b>MT_OFF(15:8)</b>		Addr. 0xC, 0x01; bit 7:0	0x00000000
<b>MT_OFF(23:16)</b>		Addr. 0xC, 0x02; bit 7:0	
<b>MT_OFF(31:24)</b>		Addr. 0xC, 0x03; bit 7:0	
Code	Singleturn position offset		
0x00000000	0		
0x00000001	1		
0x00000002	2		
...	...		
0xFFFFFFFFD	$2^{32} - 3$		
0xFFFFFFFFE	$2^{32} - 2$		
0xFFFFFFFFF	$2^{32} - 1$		

Table 35: Multiturn Position Offset

#### Position Preset

To automatically calculate the offset that is related to a certain position, a preset position can be applied. First, the desired preset position has to be set via [ST\\_PRE](#) and [MT\\_PRE](#). Then, by executing either the command [MT\\_PRESET](#) or [MTST\\_PRESET](#), the position offsets are calculated so that the current position is set according to [ST\\_PRE](#) and [MT\\_PRE](#). Refer to COMMANDS on page 53 for details.

<b>ST_PRE(7:0)</b>		Addr. 0x50; bit 7:0	default:
<b>ST_PRE(15:8)</b>		Addr. 0x51; bit 7:0	0x00000000
<b>ST_PRE(23:16)</b>		Addr. 0x52; bit 7:0	
<b>ST_PRE(31:24)</b>		Addr. 0x53; bit 7:0	
Code	Singleturn position preset		
0x00000000	0		
0x00000001	1		
0x00000002	2		
...	...		
0xFFFFFFFFD	$2^{32} - 3$		
0xFFFFFFFFE	$2^{32} - 2$		
0xFFFFFFFFF	$2^{32} - 1$		

Table 36: Singleturn Position Preset

<b>MT_PRE(7:0)</b>		Addr. 0x54; bit 7:0	default:
<b>MT_PRE(15:8)</b>		Addr. 0x55; bit 7:0	0x00000000
<b>MT_PRE(23:16)</b>		Addr. 0x56; bit 7:0	
<b>MT_PRE(31:24)</b>		Addr. 0x57; bit 7:0	
Code	Multiturn position preset		
0x00000000	0		
0x00000001	1		
0x00000002	2		
...	...		
0xFFFFFFFFD	$2^{32} - 3$		
0xFFFFFFFFE	$2^{32} - 2$		
0xFFFFFFFFF	$2^{32} - 1$		

Table 37: Multiturn Position Preset

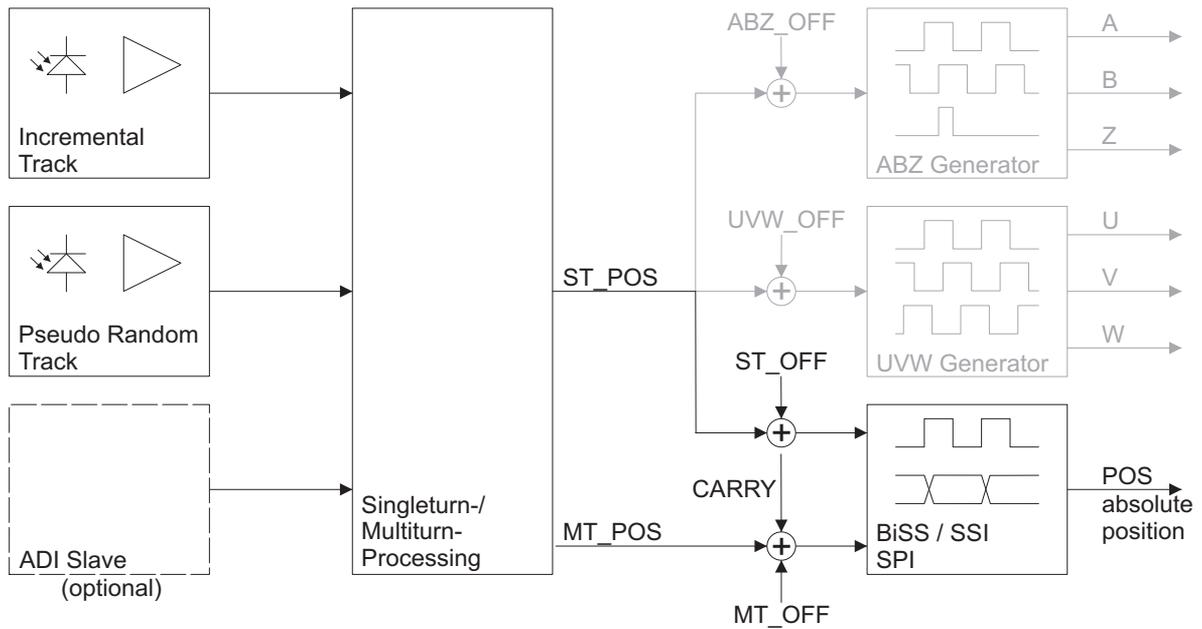


Figure 28: Programming of position offsets

### BISS SLAVE

The implemented BiSS slave interface can be selected via port configuration pins as described in INTERFACE PORTS on page 20. As BiSS and SSI share its physical ports, the interface of choice has to be enabled additionally via **SSI\_EN**. A BiSS transmission is illustrated in Figure 29. Data is sampled on the first rising edge of MA and transferred with MSB first.

<b>SSI_EN</b>		Addr. 0x6, 0x0A; bit 6	default: 0
Code	Value		
0	BiSS enabled		
1	SSI enabled		

Table 38: SSI Enable

**Note:** For in-depth information about BiSS visit [www.biSS-interface.com](http://www.biSS-interface.com).

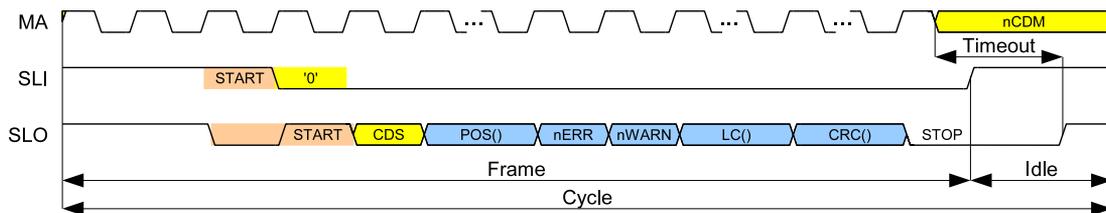


Figure 29: BiSS transmission sequence

<b>BiSS Slave Performance</b>		
Parameter	Symbol	Description
Clock Rate	$1/t_c$	Refer to Item No. Z01 on page 12
Process.T.	$t_{busy}$	Refer to Item No. I105 on page 14
Timeout	$t_{out}$	Refer to Item No. Z06, Z07 on page 12
<b>SCD Channel 1: Position Data</b>		
Bits/cycle	ID	Description
0..32	MT	Multiturn position - <b>MT_OFF</b> (right-aligned)
0..32	ST	Singleturn position - <b>ST_OFF</b> (left-aligned)
1	nE <sup>1</sup>	Error bit ERR
1	nW <sup>1</sup>	Warning bit WARN
(6)	LC	Sign-of-Life Counter
6 (16)	CRC <sup>2</sup>	Polynomial 0x43 (0x190D9), adjustable start value.
<b>CD Channel: Control Data</b>		
Bits/cycle	ID	Description
1	nCDM <sup>1</sup> , CDS	Support of bidirectional register access
	Slave IDs	1
	Commands	Support of selected BiSS Commands according to Table 49.
Notes	<sup>1</sup> Low active. <sup>2</sup> Bit inverted transmission.	

Table 39: BiSS slave performance

### Single Cycle Data

The Single Cycle Data (SCD) is transmitted in the format highlighted blue in Figure 29. According to Table 39, the format includes the multiturn position followed by the singleturn position, one low-active error bit, one low-active warning bit, an optional 6-bit sign-of-life counter (LC) and a 6-bit or 16-bit CRC value. The position data is affected by the offsets **MT\_OFF** and **ST\_OFF**.

The data length of the ST and MT values can be individually set in bits via **BISS\_ST\_DL** and **BISS\_MT\_DL**. Bit count not filling up full bytes is supported. If the values set for **BISS\_ST\_DL** and **BISS\_MT\_DL** exceed the resolution provided by the system, the surplus bits are padded with zeros.

<b>BISS_ST_DL(5:0)</b>		Addr. 0x6, 0x08; bit 5:0	default: 0x18
Code	Value		
0..32	Number of singleturn bits that are transmitted. Padded right with zeros if more than available. 0 is only allowed for <b>BISS_MT_DL</b> ≠ 0.		

Table 40: BiSS Singleturn Data Length

<b>BISS_MT_DL(5:0)</b>		Addr. 0x6, 0x09; bit 5:0	default: 0x18
Code	Value		
0..32	Number of multiturn bits that are transmitted. Padded left with zeros if more than available. 0 is only allowed for <b>BISS_ST_DL</b> ≠ 0..		

Table 41: BiSS Multiturn Data length



The total data length (ST, MT, nERR, nWARN, LC) is limited to 57 bits for the 6-bit CRC polynomial and 64 bits for the 16-bit CRC polynomial excluding the bits needed for the CRC value. Refer to [BISS\\_CRC16](#) for details.

**Note:** Common used profiles, like the BiSS Encoder Profile BP1 and BP3, do have certain restrictions regarding data length to be taken into account.

The sources triggering the active-low error bit nERR and the active-low warning bit nWARN can be selected individually. The diagnosis information logged in [DIAG](#) are only transferred via BiSS, if the according bit in the masks [BISS\\_WM](#) and [BISS\\_EM](#) is set to 1. Otherwise an error or warning will not be forwarded for corresponding events masked with 0. Refer to [DIAGNOSIS](#) on page 64 for further information.

Code	Value
<b>BISS_EM(7:0)</b>	Addr. 0x6, 0x00; bit 7:0 default: 0x0
<b>BISS_EM(15:8)</b>	Addr. 0x6, 0x01; bit 7:0 0x11000F13
<b>BISS_EM(23:16)</b>	Addr. 0x6, 0x02; bit 7:0
<b>BISS_EM(31:24)</b>	Addr. 0x6, 0x03; bit 7:0

Bit	Description
31:0	Activate error for corresponding <a href="#">DIAG</a> bit

Table 42: BiSS Error Bit Mask

Code	Value
<b>BISS_WM(7:0)</b>	Addr. 0x6, 0x04; bit 7:0 default: 0x0
<b>BISS_WM(15:8)</b>	Addr. 0x6, 0x05; bit 7:0 0x0200C00C
<b>BISS_WM(23:16)</b>	Addr. 0x6, 0x06; bit 7:0
<b>BISS_WM(31:24)</b>	Addr. 0x6, 0x07; bit 7:0

Bit	Description
31:0	Activate warning for corresponding <a href="#">DIAG</a> bit

Table 43: BiSS Warning Bit Mask

### Sign-of-Life Counter

The transmission of a 6-bit sign-of-life counter (LC) can be enabled via [BISS\\_ENSOL](#). The LC is initialized with 0, but counts from 1 to 63 skipping the 0 when wrapping. The counter is incremented with each BiSS frame.

Code	Value
<b>BISS_ENSOL</b>	Addr. 0x6, 0x0A; bit 0 default: 0
0	No sign-of-life counter is transmitted
1	A 6-bit sign-of-life counter is transmitted

Table 44: BiSS Enable Sign-of-life Counter

### Cyclic Redundancy Check

The Cyclic Redundancy Check (CRC) value is transmitted in its inverted state at the end of each BiSS frame. The CRC start value is defined by [BISS\\_CRCS](#). Via [BISS\\_CRC16](#) the usage of either a 6-bit or a 16-bit CRC polynomial is selected.

Code	Value
<b>BISS_CRCS(5:0)</b>	Addr. 0x6, 0x0B; bit 5:0 default: 0x00
0x00..0x3F	Start value for BiSS CRC calculation (both 6 and 16 bit), used by all slaves on the BiSS channel

Table 45: BiSS CRC Start Value

Code	CRC HEX Code	Description
<b>BISS_CRC16</b>	Addr. 0x6, 0x0A; bit 1 default: 0	
0	0x43	6-bit CRC polynomial: $X^6 + X^1 + X^0$ Hamming Distance: 3 max data length: 57 bits
1	0x190D9	16-bit CRC polynomial: $X^{16} + X^{15} + X^{12} + X^7 + X^6 + X^4 + X^3 + X^0$ Hamming Distance: 6 max data length: 64 bits

Table 46: BiSS CRC Polynomial Selection

### Fixed & Adaptive Timeout

Either a fixed or an adaptive timeout for BiSS communication can be selected via [BISS\\_NTOA](#). The adaptive BiSS timeout is recommended for fastest communication speed.

Code	Value
<b>BISS_NTOA</b>	Addr. 0x6, 0x0A; bit 2 default: 0
0	Adaptive BiSS timeout used
1	Fixed BiSS timeout used

Table 47: BiSS Not Timeout Adaptive

The adaptive BiSS timeout is set according to the period of the BiSS MA clock  $T_{MA}$  and the internal sampling frequency  $1/T_{SAMPLE}$  (see Elec. Char. item no. Z06). First, one and a half periods of the MA clock from first falling to second rising edge within each BiSS frame are measured during operation. Then, the timeout is calculated according to the equation below:

$$T_{SAMPLE} = \frac{16}{3 * f_{osc}}$$

Where  $f_{osc}$  is the system clock frequency (see Elec. Char. item no. O01 and Z06).

Timeout	Min.	Max.
$t_{tout}$	$1.5 * T_{BISS}$	$1.5 * T_{BISS} + 3.0 * T_{SAMPLE}$

Table 48: Adaptive Timeout Calculations

**Note:** More information on the adaptive timeout can be found in BiSS application note AN23 at [www.biss-interface.com](http://www.biss-interface.com).

### BiSS protocol commands

The following BiSS interface protocol commands are implemented.

CD Channel: BiSS Protocol Commands		
CMD	Availability	Function
<b>Addressed</b>		
00	Yes	Activate Single-Cycle Data channels
01	—*	Deactivate control communication
10	—	Reserved
11	—	Reserved
<b>Broadcast (all slaves)</b>		
00	Yes	Deactivate Single-Cycle Data channels
01	—*	Activate control communication
10	—	Reserved
11	—	Reserved
Notes	* Command w/o function, but will be acknowledged.	



Short BiSS-frames with less than 6 MA pulses cause just the one directly following BiSS-frame to contain not refreshed but outdated data. Such reduced BiSS-frames (e. g. BiSS-Init) should be avoided. Alternatively, an intermediate frame with at least 6 MA pulses can be inserted without using its position data.

Table 49: BiSS Protocol Commands

### SSI SLAVE

The implemented SSI slave interface can be selected via port configuration pins as described in INTERFACE PORTS on page 20. As BiSS and SSI share its physical ports, the interface of choice has to be additionally enabled via **SSI\_EN**. Refer to chapter BiSS SLAVE on page 43.

Two data formats are supported, of which one has to be selected via **SSI\_EXT**. Regardless of the format, position data is latched on the first falling edge of MA and transferred with MSB first. The position data coding can be switched between Natural Binary Code and Gray Code via **SSI\_GRAY**. A fixed or adaptive timeout can be set via **BISS\_NTOA**. However, the use of an adaptive timeout is not recommended for SSI.

<b>SSI_EXT</b>		Addr. 0x6, 0x0A; bit 5	default: 0
Code	Value		
0	Standard SSI		
1	Extended SSI		

Table 50: Enable Extended SSI

<b>SSI_GRAY</b>		Addr. 0x6, 0x0A; bit 4	default: 0
Code	Value		
0	Natural binary		
1	Gray code		

Table 51: Activate SSI Gray Coding

According to Table 52, the standard SSI protocol format includes the multiturn (MT) and singleturn (ST) position. The ST data length has to be set to exactly 13 bit via **BISS\_ST\_DL**. The MT data length has to be set to either 0 or 12 bit via **BISS\_MT\_DL**. A transmission using the standard SSI protocol is shown in Figure 30.

<b>Standard SSI Protocol Frame</b>	
Bit Length	Description
0 or 12	Multiturn position - <b>MT_OFF</b>
13	Singleturn position - <b>ST_OFF</b>

Table 52: Standard SSI Protocol Frame

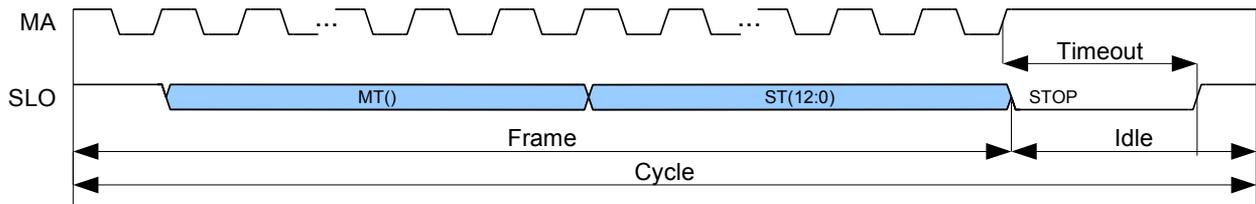


Figure 30: Standard SSI protocol

According to Table 53, the extended SSI protocol format includes the multiturn position followed by the singleturn position, one low-active error bit, one low-active warning bit, an optional 6-bit sign-of-life counter (LC) and a 6-bit or 16-bit CRC value. The format can be adjusted according to section Single Cycle Data in BiSS SLAVE on page 43.

Extended SSI Protocol Frame	
Bit Length	Description
0 - 32	Multiturn position - <b>MT_OFF</b>
0 - 32	Singleturn position - <b>ST_OFF</b>
1	Error bit nERR (active low)
1	Warning bit nWARN (active low)
0 or 6	Optional sign-of-life counter (LC)
6 or 16	CRC (inverted)

Table 53: Extended SSI Protocol Frame

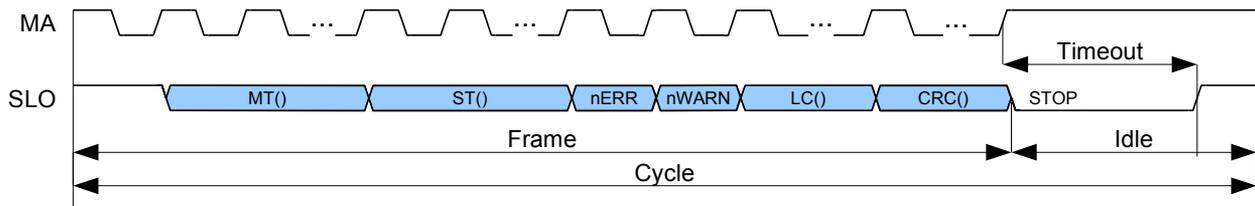


Figure 31: Extended SSI protocol



Short SSI-frames with less than 6 MA pulses cause just the one directly following SSI-frame to contain not refreshed but outdated data. Such short SSI-frames should be avoided.

### SPI SLAVE

The implemented SPI slave can be selected via port configuration pins as described in INTERFACE PORTS on page 20. SPI modes 0 and 3 are supported. Idle state of SCLK can be either low or high. Data is sampled on the rising edge of SCLK. Communication is initiated with a falling edge on NCS. While NCS is low, iC-PZ is set active. Each SPI transaction starts with one of the opcodes listed in Table 54. Data is sent byte by byte with MSB first. An SPI transmission including SCLK lines for modes 0 and 3 is illustrated in Figure 32.

OPCODE	
Code	Description
0x81	Read Registers
0xCF	Write Registers
0xA6	Read Position
0xD9	Write Command
0x9C	Read Diagnosis
0x97	Request Data From I2C Slave
0xD2	Transmit Data To I2C Slave
0xAD	Get Transaction Info
0xB0	Activate Slave In Chain

Table 54: SPI Operation Codes

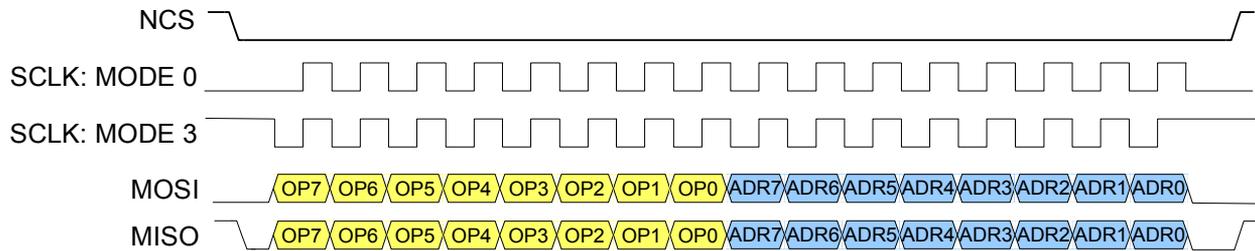


Figure 32: SPI transmission mode 0 and 3

**Note:** The output line MISO should have an external pull-up or pull-down resistor. Otherwise, this line will float when tristate (NCS high) and may produce crosscurrent in the following input stage.

### Read Registers

Opcode [Read Registers](#) (0x81) is used to read data from any number of consecutive registers in the on-chip RAM. As shown in Figure 33, the data stream to be sent on MOSI consists of the opcode 0x81, followed by the address of the first register to be read and a delay byte 0x00. Those first three bytes are also transmitted by iC-PZ on MISO, before sending the requested data (DATA1) from the register at address (ADR). As long as clock is sent and the slave stays active, data (DATA2) from the next register at the incremented address (ADR + 1) is transmitted. This procedure may be continued for any number of consecutive registers.

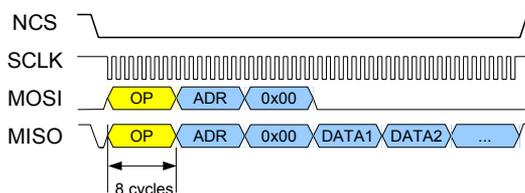


Figure 33: Read Registers

### Write Registers

Opcode [Write Registers](#) (0xCF) is used to write data to any number of consecutive registers in the on-chip RAM. As shown in Figure 34, the data stream to be sent on MOSI consists of the opcode 0xCF, followed by the address of the first register to be written and the data. With each data byte the address of the register to be written is incremented by one (ADR + 1). If successfully received, the same data stream is transmitted on MISO by iC-PZ.

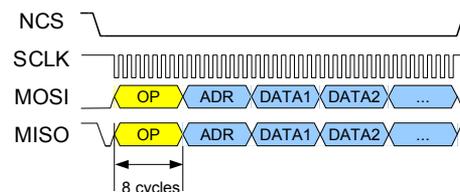


Figure 34: Write Registers

### Read Position

Opcode [Read Position](#) (0xA6) is used to read the absolute position data from iC-PZ. As shown in Figure 35, the position data is latched on the first rising edge of SCLK (REQ).

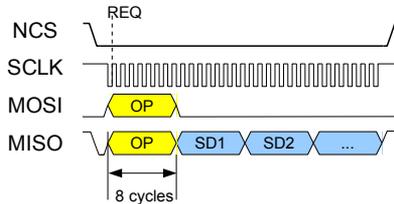


Figure 35: Read Position

As shown in Table 55, the position data format consists of the multiturn position followed by the singleturn position, an error bit, a warning bit, an optional sign-of-life counter (LC) and a CRC value. Singleturn and multiturn position data length can be adjusted individually. The position data includes the offset adjusted via [MT\\_OFF](#) and [ST\\_OFF](#).

Position Data Format	
Byte Length	Description
<b>for SPI_EXT = 0</b>	
0 - 4	Multiturn Position - <a href="#">MT_OFF</a>
0 - 4	Singleturn Position - <a href="#">ST_OFF</a>
1	nERR, nWARN, 6-bit CRC
<b>for SPI_EXT = 1</b>	
0 - 4	Multiturn Position - <a href="#">MT_OFF</a>
0 - 4	Singleturn Position - <a href="#">ST_OFF</a>
1	nERR, nWARN, 6-bit LC
2	16-bit CRC

Table 55: SPI Position Data Format

The position information is transmitted in one of two data formats selected via [SPI\\_EXT](#) as shown in Table 56. The first format ([SPI\\_EXT](#) = 0) includes one low-active error bit, one low-active warning bit and a 6-bit CRC. The second format ([SPI\\_EXT](#) = 1) includes one low-active error bit, one low-active warning bit, a 6-bit sign-of-life counter (LC) and a 16-bit CRC. The LC is initialized with 0, but counts from 1 to 63 skipping the 0 when wrapping. The total data length (ST, MT, nERR, nWARN, LC) is limited to 57 bits ([SPI\\_EXT](#) = 0) or 64 bits ([SPI\\_EXT](#) = 1) without the bits needed for the CRC value.

SPI_EXT		Addr. 0x7, 0x0A; bit 0	default: 0
Code	CRC HEX Code	Description	
0	0x43	6-bit CRC polynomial: $X^6 + X^1 + X^0$ Hamming Distance: 3 max data length: 57 bits  No sign-of-life counter is transmitted	
1	0x190D9	16-bit CRC polynomial: $X^{16} + X^{15} + X^{12} + X^7 + X^6 + X^4 + X^3 + X^0$ Hamming Distance: 6 max data length: 64 bits  A 6-bit sign-of-life counter is transmitted. The counter starts at 0 counting from 1 to 63 and omitting 0 when wrapping.	

Table 56: SPI Extended Position Data Format

The CRC value is transmitted in its inverted state at the end of each SPI frame. The CRC start value is defined by [SPI\\_CRCS](#).

SPI_CRCS(5:0)		Addr. 0x7, 0x0B; bit 5:0	default: 0x00
Code	Value		
0x00..0x3F	Start-value for SPI CRC calculation (both 6 and 16 bit)		

Table 57: SPI CRC Start Value

The data length of the ST and MT value can be individually set in bits via [SPI\\_ST\\_DL](#) and [SPI\\_MT\\_DL](#). Bit count not filling up full bytes is supported. Breaking up the position data at any bit during transmission is possible, if further parts of the data stream (nERR/nWARN, LC or CRC) are not needed. If the values set for [SPI\\_ST\\_DL](#) and [SPI\\_MT\\_DL](#) do not match the resolution provided by the system, the surplus bits are padded with zeros. In conjunction with some systems (e.g. microcontrollers) using full bytes for the position data is advisable, as it may help making data handling easier.

SPI_ST_DL(5:0)		Addr. 0x7, 0x08; bit 5:0	default: 0x18
Code	Value		
0..32	Number of singleturn bits that are transmitted. padded right with zeros if more than available. 0 is only allowed when <a href="#">SPI_MT_DL</a> ≠ 0.		

Table 58: SPI Singleturn Data Length

SPI_MT_DL(5:0)		Addr. 0x7, 0x09; bit 5:0	default: 0x18
Code	Value		
0..32	Number of multiturn bits that are transmitted. padded left with zeros if more than available. 0 is only allowed when SPI_ST_DL ≠ 0.		

Table 59: SPI Multiturn Data Length

The sources triggering the active-low error bit nERR and the active-low warning bit nWARN can be selected individually. The diagnosis information logged in DIAG is only transferred via SPI, if the according bit in the masks SPI\_WM and SPI\_EM is set to 1. Otherwise an error or warning will not be forwarded for corresponding events masked with 0. Refer to DIAGNOSIS on page 64 for further information.

SPI_EM(7:0)		Addr. 0x7, 0x00; bit 7:0	default:
SPI_EM(15:8)		Addr. 0x7, 0x01; bit 7:0	0x11000F13
SPI_EM(23:16)		Addr. 0x7, 0x02; bit 7:0	
SPI_EM(31:24)		Addr. 0x7, 0x03; bit 7:0	
Bit	Description		
31:0	Activate error for corresponding DIAG-bit		

Table 60: SPI Error Bit Mask

SPI_WM(7:0)		Addr. 0x7, 0x04; bit 7:0	default:
SPI_WM(15:8)		Addr. 0x7, 0x05; bit 7:0	0x0200C00C
SPI_WM(23:16)		Addr. 0x7, 0x06; bit 7:0	
SPI_WM(31:24)		Addr. 0x7, 0x07; bit 7:0	
Bit	Description		
31:0	Activate warning for corresponding DIAG-bit		

Table 61: SPI Warning Bit Mask

### Write Command

One of the commands specified in COMMANDS on page 53 can be executed via opcode **Write Command** (0xD9). The command is automatically written to the CMD register at address 0x77 before execution. As shown in Figure 36, the data stream to be sent on MOSI consists of the opcode 0xD9 followed by the command to be executed. Those two bytes are also transmitted by iC-PZ on MISO.

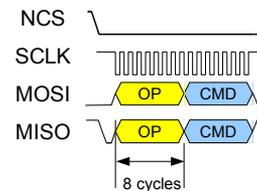


Figure 36: Write Command



Commands do require processing time until completed. Successful completion can be detected by polling the CMD register. Refer to COMMANDS on page 53 for details.

### Read Diagnosis

Opcode **Read Diagnosis** (0x9C) is used to read the registers at address 0x6C to 0x73 containing error ERR and warning WARN information. As shown in Figure 37, only the opcode 0x9C has to be sent on MOSI. The opcode, followed by a delay byte 0x00 and 4 bytes each for ERR and WARN are transmitted by iC-PZ on MISO. Refer to DIAGNOSIS on page 64 for further information.

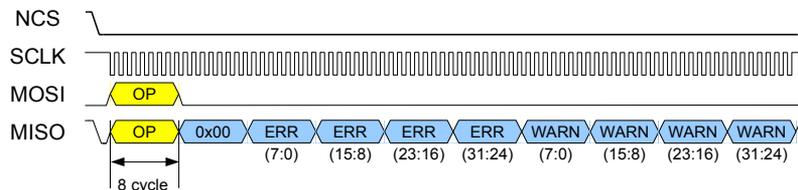


Figure 37: Read Diagnosis

### Request Data From I2C Slave

Data from external devices connected to the I2C master of iC-PZ (e. g. EEPROM) can be requested via opcode [Request Data From I2C Slave](#) (0x97). As shown in Figure 38, the opcode followed by the register address of the I2C slave has to be sent on MOSI. The same content is transmitted by iC-PZ on MISO. After the opcode has been received, the initiated I2C communication will take additional time until completed. Opcode [Get Transaction Info](#) is used to poll for the current I2C communication status and new data.

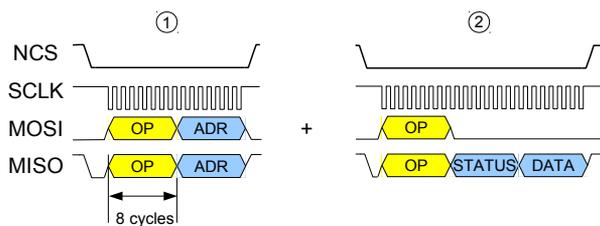


Figure 38: Request Data From I2C Slave

### Transmit Data To I2C Slave

Data to external devices connected to the I2C master of iC-PZ (e. g. EEPROM) can be transmitted via opcode [Transmit Data To I2C Slave](#) (0xD2). As shown in Figure 39, the opcode followed by the register address of the I2C slave and the data byte to be transmitted has to be sent on MOSI. The same content is transmitted by iC-PZ on MISO. After the opcode has been received, the initiated I2C communication will take additional time until completed. Opcode [Get Transaction Info](#) is used to poll for the current I2C communication status.

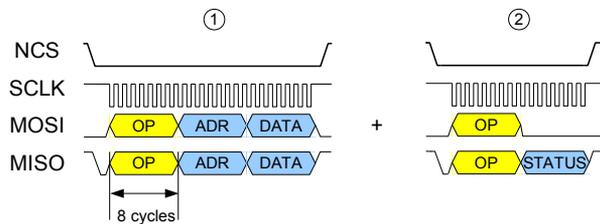


Figure 39: Transmit Data To I2C Slave

**Note:** For details about the address space of external I2C devices refer to MEMORY ORGANIZATION on page 22.

**Note:** For details on how to configure the I2C master refer to I2C MASTER on page 60.

### Get Transaction Info

Via opcode [Get Transaction Info](#) (0xAD) the status of the last initiated SPI transaction is returned. Opcodes [Read Position](#) (0xA6) and [Get Transaction Info](#) (0xAD) itself are not updating the status byte. As shown in Figure 40, only the opcode has to be sent on MOSI. The opcode followed by the status byte defined in Table 62 is transmitted by iC-PZ on MISO. The data byte is only defined, if opcode [Request Data From I2C Slave](#) has been sent before.

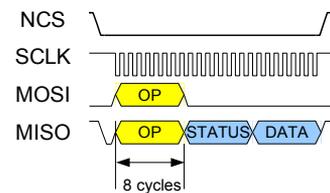


Figure 40: Get Transaction Info

SPI STATUS	
Bit	Description
7	Invalid Opcode
6:4	-
3	Illegal Address
2	Data Request Failed
1	Slave Busy
0	Data Valid

Table 62: SPI Status Byte

### Multi-Slave Configurations with iC-PZ

A common SPI bus configuration for two iC-PZ is illustrated in Figure 41. Each slave is selected individually by the SPI master via a dedicated NCS line. Only one slave may communicate at the same time.

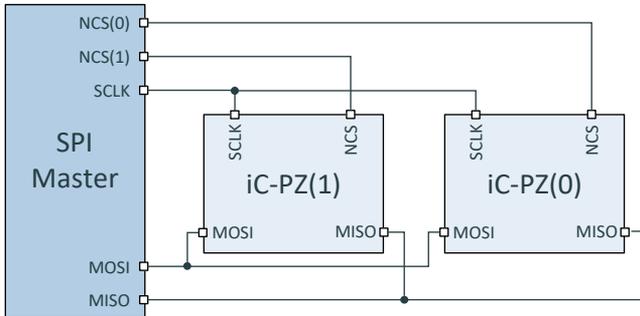


Figure 41: SPI bus configuration

Another possibility to connect multiple iC-PZ is by setting up an SPI daisy chain as shown in Figure 42. The MISO of each iC-PZ is connected to the MOSI of the next device. As only a single NCS line is used, individual slaves are selected via opcode [Activate Slave in Chain](#).

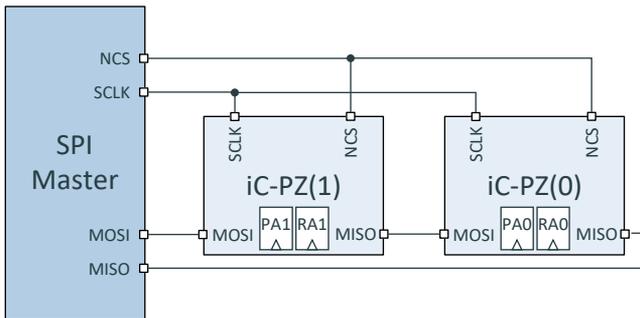


Figure 42: SPI daisy chain configuration

**Note:** Reading the current state of [RACTIVE](#) and [PACTIVE](#) is not possible.

### Activate Slave In Chain

Each iC-PZ provides two separate channels for register and position data transfer that can be switched on and off individually. Register communication with iC-PZ is only possible if [RACTIVE](#) = 1. Otherwise register communication attempts are ignored. Position data can only be acquired from iC-PZ if [PACTIVE](#) = 1. Otherwise opcode [Read Position](#) (0xA6) is ignored.

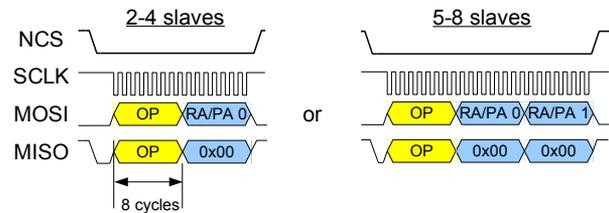


Figure 43: Activate Slave In Chain

By sending opcode [Activate Slave in Chain](#) (0xB0) each iC-PZ acts as a 2-bit shift register containing one [RACTIVE](#) and one [PACTIVE](#) configuration bit. [RACTIVE](#) and [PACTIVE](#) bits are initialized as '0', turning both data channels off. The SPI Status Byte is reset. Following the opcode, the desired [RACTIVE](#)/[PACTIVE](#) channel configuration is transmitted. Data bytes corresponding to all possible configurations are shown in Table 65. Slave number 0 is considered to be the last slave in chain directly connected to MISO of the SPI master.

RACTIVE		default: 1
Code	Description	
0	Register communication deactivated	
1	Register communication activated	

Table 63: RACTIVE (RA)

PACTIVE		default: 1
Code	Description	
0	Position data channel deactivated	
1	Position data channel activated	

Table 64: PACTIVE (PA)

Slaves	RA/PA configuration byte 0								RA/PA configuration byte 1							
2	0	0	0	0	RA0	PA0	RA1	PA1	Not used							
3	0	0	RA0	PA	RA1	PA1	RA2	PA2	Not used							
4	RA0	PA0	RA1	PA1	RA2	PA2	RA3	PA3	Not used							
5	0	0	0	0	0	0	RA0	PA0	RA1	PA1	RA2	PA2	RA3	PA3	RA4	PA4
6	0	0	0	0	RA0	PA0	RA1	PA1	RA2	PA2	RA3	PA3	RA4	PA4	RA5	PA5
7	0	0	RA0	PA0	RA1	PA1	RA2	PA2	RA3	PA3	RA4	PA4	RA5	PA5	RA6	PA6
8	RA0	PA0	RA1	PA1	RA2	PA2	RA3	PA3	RA4	PA4	RA5	PA5	RA6	PA6	RA7	PA7

Table 65: RA/PA channel configuration bits for 2-8 slaves

COMMANDS			
CMD(7:0)		Addr. 0x77; bit 7:0	default: 0x00
Code	Name	Description	
0x00	<NOP_OK>	<Return-code: last operation succeeded>	
0x10	REBOOT	Equivalent to power-on	
0x18	ADI_RESET	Reset the absolute data interface and re-synchronize with the external multiturn device	
0x20	SCLEAR	Clear the complete diagnosis register	
0x28	DIAG_USER0_RESET	Reset DIAG_USER(0) to 0	
0x29	DIAG_USER0_SET	Set DIAG_USER(0) to 1	
0x2A	DIAG_USER1_RESET	Reset DIAG_USER(1) to 0	
0x2B	DIAG_USER1_SET	Set DIAG_USER(1) to 1	
0x2C	DIAG_USER2_RESET	Reset DIAG_USER(2) to 0	
0x2D	DIAG_USER2_SET	Set DIAG_USER(2) to 1	
0x30	CRC_CALC	Calculate and apply valid CRC checksums for all configuration banks	
0x31	CRC_CHECK	Check for invalid CRC checksums in all configuration banks	
0x40	CONF_READ_ALL	Read configuration for all banks from EEPROM. This includes the RPL information for all banks and the RPL information for the EEPROM.	
0x41	CONF_WRITE_ALL	Write current configuration of all banks to EEPROM. This includes the RPL information for all banks and the RPL information for the EEPROM. Valid CRC checksums are always calculated automatically beforehand for all banks.	
0x42	CONF_READ	Read configuration for the active bank (*) from EEPROM. This includes the RPL information of the active bank.	
0x43	CONF_WRITE	Write current configuration of the active bank (*) to EEPROM. This includes the RPL information of the active bank. Valid CRC checksums are always calculated automatically beforehand for the active bank.	
0x80	MTST_PRESET	Calculate and apply <a href="#">MT_OFF</a> and <a href="#">ST_OFF</a> so that the current MT:ST-position will be equal to <a href="#">MT_PRE:ST_PRE</a>	
0x81	MT_PRESET	Calculate and apply <a href="#">MT_OFF</a> so that the current MT-position will be equal to <a href="#">MT_PRE</a>	
0x82	ABZ_PRESET	Calculate and apply <a href="#">ABZ_OFF</a> so that the current ABZ-position will be equal to <a href="#">ABZ_PRE</a>	
0x83	UVW_PRESET	Calculate and apply <a href="#">UVW_OFF</a> so that the current UVW-position will be equal to <a href="#">UVW_PRE</a>	
0x88	MTST_PRESET_STORE	Identical to MTST_PRESET, additionally store the containing bank 0xB..C to EEPROM.	
0x89	MT_PRESET_STORE	Identical to MT_PRESET, additionally store the containing bank 0xC to EEPROM.	
0x8A	ABZ_PRESET_STORE	Identical to ABZ_PRESET, additionally store the containing bank 0x4 to EEPROM.	
0x8B	UVW_PRESET_STORE	Identical to UVW_PRESET, additionally store the containing bank 0x5 to EEPROM.	
0x90	GPIO_OUT0_SET0	Set GPIO_OUT(0) to 0	
0x91	GPIO_OUT0_SET1	Set GPIO_OUT(0) to 1	
0x92	GPIO_OUT1_SET0	Set GPIO_OUT(1) to 0	
0x93	GPIO_OUT1_SET1	Set GPIO_OUT(1) to 1	
0xA0	RPL_SET_NA	Set the register-protection-level of the active bank (*) to no-access. Call command CONF_WRITE to store the restriction in EEPROM	
0xA1	RPL_SET_RO	Set the register-protection-level of the active bank (*) to read-only. Call command CONF_WRITE to store the restriction in EEPROM	
0xA2	RPL_GET	Get the register-protection-level status of the active bank (*). The result is stored in <a href="#">CMD_STAT</a>	
0xB0	AUTO_ADJ_ANA	Automatic analog adjustment. Refer to chapter ADJUSTMENT ANALOG for details.	
0xB1	AUTO_ADJ_DIG	Automatic digital adjustment (initial). Refer to chapter ADJUSTMENT DIGITAL for details.	
0xB2	AUTO_READJ_DIG	Automatic digital re-adjustment (in-field). Refer to chapter ADJUSTMENT DIGITAL for details.	
0xB3	AUTO_ADJ_ECC	Automatic eccentricity adjustment. Refer to chapter ADJUSTMENT ECCENTRICITY for details.	
0xC1	FORCE_BISS	Switch BiSS/SSI-Interface to BiSS (sets <a href="#">SSI_EN</a> = 0)	
0xC2	FORCE_SSI	Switch BiSS/SSI-Interface to SSI (sets <a href="#">SSI_EN</a> = 1)	
0xE0	CHIP_ID	Return CHIP_ID: for iC-PZ: 0x01	
0xE1	CHIP_REV	Return CHIP_REV: content identical to register <a href="#">REV</a>	
0xFF	<NOP_FAIL>	<Return-code: last operation failed>	
(*) The active bank is defined by <a href="#">BSEL</a> ; <a href="#">BSEL</a> must not be changed before the command finishes.			

Table 66: Commands

The **CMD** register is used to execute defined commands received via serial interface. As long as a command is queued or executed, **CMD** keeps the received command. Once the command has been completed successfully, **CMD** is set to 0x00. If an error occurs during command execution, **CMD** is set to 0xFF.

A separate **CMD** register is provided for each serial interface individually at address 0x77. It is forbidden to send a new command via the same interface while another command is still in queue or executed. Therefore, polling **CMD** after a command has been sent is essential. However, sending a single command via multiple interfaces is possible. If more than one command is received across different interfaces, execution is applied via first come first served scheduling so that delays may have to be taken into account.

For some commands useful information after execution, or in case an error occurred, are written to register **CMD\_STAT**. A separate **CMD\_STAT** register is provided for each serial interface individually at address 0x76.

<b>CMD_STAT(7:0)</b>		Addr. 0x76; bit 7:0	default: 0x00, read-only
Code	Value		
0x00	Last command succeeded		
others	Last command error-code. See Table 68 for details		

Table 67: Command Status

<b>CMD_STAT error-code</b>	
<b>CRC_CHECK</b>	
n	n = Amount of wrong bank CRCs. Refer to <b>CRC_STAT</b> for a detailed bitmask.
<b>CONF_WRITE, CONF_WRITE_ALL, PRESET_STORE (all)</b>	
0x10	Slave acknowledge missing, resulting in timeout. Slave may not be available or misconfigured.
0x11	Slave acknowledge missing after restart. Indicates a problem with the slave.
0x12	SCL stuck at zero
0x13	SDA stuck at zero
0x14	Arbitration lost
0x20	Unexpected event
<b>CONF_READ, CONF_READ_ALL</b>	
0x10..0x20	In case of communication issues with the EEPROM, the error code is identical to CONF_WRITE.
0x40	EEPROM communication ok, but at least one CRC is wrong. Refer to <b>CRC_STAT</b> for a detailed bitmask.
<b>RPL_SET_x</b>	
0x50	Illegal bank selected
0x60	Change refused. Only lower privileges are allowed.
<b>RPL_GET</b>	
0x00	No-access
0x01	Read-only
0x02	No restriction due to invalid CRC checksum
0x03	No restriction
0x50	Illegal bank selected
<b>PRESET (all), FORCE (all)</b>	
0xFE	Command rejected as it would modify data in a write-protected bank
<b>AUTO_ADJ_ANA, AUTO_ADJ_DIG, AUTO_READJ_DIG, AUTO_ADJ_ECC</b>	
0x80	Timeout. Either speed too low or settings too high. Increase speed or decrease <b>AC_COUNT</b> to prevent timeout.
<b>AUTO_ADJ_ECC (continuing)</b>	
0x81	Rotation reversal detected
0x82	Rotation too fast or position error occurred
<b>others</b>	
0x00	No error code available

Table 68: Command Status error-code

**ABSOLUTE DATA INTERFACE (ADI)**

Key features:

- Absolute data interface (ADI) master
- Read revolution counter (RC), i.e., multiturn (MT) position data, from external slave(s)
- SSI protocol with synchronization (SB), error (EB) and warning bits (WB)
- Slave position data (RC + SB) as one word (synced) or parts of words (unsynced)
- Access of raw slave position data

ADI key parameters	
Parameter	Value
Clock frequency	150 kHz or 1500 kHz
Frame repetition period	1.5 ms (normal operation) 0.2 ms (fast startup)
Slave timeout	11.5 μs ... 40 μs
RC data length per slave	1 ... 32 bit (single-slave operation) 1 ... 5 bit (multi-slave operation)
Note	See also operating requirements for detailed timing information.

Table 69: ADI key parameters

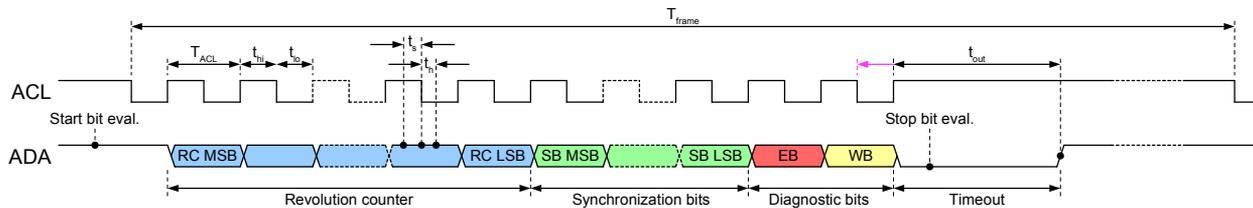


Figure 44: SSI protocol timing

**Interface Enable**

An internal revolution counter (RC) with a configurable absolute data interface (ADI) master is implemented to (periodically) read multiturn (MT) position data from an external sensor (slave) via the synchronous serial interface (SSI) protocol. Here, pin ACL is the clock output and pin ADA is the data input, respectively. The number of clock pulses depends on the configuration of the ADI master. The interface is enabled via bit [ADI\\_CFG\(7\)](#). If this bit is 0, only the internal revolution counter (RC) is active.

ADI_CFG(7)		Addr. 0x0, 0x0C; bit 7	default: 0
Code	Function		
0	Interface disabled (ACL=const=1), only counting		
1	Interface enabled		
Note	In case the bit becomes 0 while the interface is active, the communication is still completed, but the data is discarded.		

Table 70: ADI Enable

When the ADI master is enabled, the received position data is synchronized to the singleturn position to compensate for mechanical misalignment and SSI communication delays. The synchronized data is then compared to the internal revolution counter, which has been initialized during startup. In case of a position mismatch an error, [DIAG\(21\)](#), is reported and stored in the status register.

**Single- and Multi-Slave Operation**

Besides the classical single-slave operation, the ADI master also supports multi-slave operation, which is used to read in the raw (unsynchronized) position of multiple slave devices of a gear system and perform the synchronization among the slaves with the programmed synchronization bit length inside iC-PZ. Single- or multi-slave operation is configured via parameter [ADI\\_MSO](#). In case a gear system with multiple slaves already provides a position that is synchronized among the slaves and only needs to be synchronized to the singleturn, [ADI\\_MSO = 000](#) has to be used. That is because from the master's point of view, this system behaves like a single-slave system.

ADI_MSO(2:0)		Addr. 0x0, 0x0B; bit 7:5	default: 000
Code	Number of ADI slaves		
000	1 (Single-slave operation)		
001	2		
...	...		
110	7		
111	8		

Table 71: Multi-Slave Operation

The revolution counter (multiturn position) bit length per slave is implicitly configured via parameter [MT\\_PDL](#), see Table 33. Here, [MT\\_PDL](#) always defines the total revolution counter bit length of the system. Hence, for single-slave operation ([ADI\\_MSO = 000](#)) the revolution

counter bit length per slave exactly corresponds to the value of **MT\_PDL**. For multi-slave operation (**ADI\_MSO** ≠ 000) however, the value of **MT\_PDL** corresponds to the revolution counter bit length per slave multiplied by the number of slaves. In other words, valid values for **MT\_PDL** depend on **ADI\_MSO**. The maximum revolution counter bit length per slave is limited to 5. The allowed configurations for all supported systems are given in the Table below. Multi-slave operation is visualized at the end of this chapter in Figure 46 and Figure 47.

ADI_MSO	Slaves	Valid MT_PDL	Resulting RC bit length per slave
000	1	1..32	1..32
001	2	2, 4, 6, 8, 10	1, 2, 3, 4, 5
010	3	3, 6, 9, 12, 15	1, 2, 3, 4, 5
011	4	4, 8, 12, 16, 20	1, 2, 3, 4, 5
100	5	5, 10, 15, 20, 25	1, 2, 3, 4, 5
101	6	6, 12, 18, 24, 30	1, 2, 3, 4, 5
110	7	7, 14, 21, 28	1, 2, 3, 4
111	8	8, 16, 24, 32	1, 2, 3, 4

Table 72: Revolution counter bit length per slave

### Synchronization

The ADI master implements a ±1 position data synchronization of two devices, i.e., the position of the so-called "sync-slave" is synchronized to the position of the so-called "sync-master". In a single-slave system there are only two devices: The singleturn device acts as the sync-master, while the multiturn device is the sync-slave. In a multi-slave (gear) system adjacent ADI slaves have to be synchronized as well: Here, the faster turning device acts as the sync-master, while the slower turning device is the sync-slave. The tolerable phase shift values in the following tables refer to a mechanical revolution of the sync-master device. Due to the ±1 synchronization principle, the optimal mounting phase shift between sync-master and sync-slave corresponds to 0 °m (center of the tolerable phase shift).

The synchronization bit length per slave is configured via parameter **ADI\_SBL**. For the synchronization to the singleturn parameter **ADI\_OS** can be used to compensate for a non-ideal mounting phase shift by shifting the received slave position according to **ADI\_OS** inside the master device. If the mounting phase shift is unknown, one can determine it by reading parameter **ADI\_SB** and comparing it to 4 MSBs of the singleturn position.

ADI_SBL(2:0)		Addr. 0x0, 0x0A; bit 2:0	default: 001
Code	Synchronization bit length per slave	Tolerable phase shift	
<b>Single-slave operation (ADI_MSO = 000) only.</b>			
000	0	Sync. disabled	
<b>Single- and multi-slave operation</b>			
001	1	± 90 °m	
010	2	± 135 °m	
011	3	± 157.5 °m	
100	4	± 168.75 °m	
others	invalid		
Note	Disabling the synchronization also disables the internal revolution counter, meaning the raw data received from the ADI slave can be accessed. This is only recommended for test or adjustment purposes. Increasing the revolution counter bit length in the ADI master by the number of synchronization bits of the slave grants access to the full, non-synchronized position data of the slave (RC + SB).		

Table 73: Synchronization Bit Length per Slave

ADI_OS(4:0)		Addr. 0x0, 0x0A; bit 7:3	default: 00000
Code	Synchronization offset		
00000	0 °m		
00001	11.25 °m		
...	...		
01110	157.5 °m		
01111	168.75 °m		
10000	-180 °m		
10001	-168.75 °m		
...	...		
11110	-22.5 °m		
11111	-11.25 °m		
Note	The synchronization offset is only applied to the synchronization of revolution counter to singleturn and not for the synchronization among slaves in multi-slave operation.		

Table 74: Synchronization Offset (electrical alignment slave vs. master)

ADI_SB(3:0)	Addr. 0x5E; bit 3:0	read-only
Output of up to 4 synchronization bits, as received from the ADI slave (left aligned).		

Table 75: Received Synchronization Bits (read-only)

### Error and Warning Bits

For diagnostic purposes error and warning bits can be transmitted in the protocol besides the position data. The error bit length per slave is defined by parameter **ADI\_EBL**, the polarity can be configured by parameter **ADI\_EBP**. For multi-slave systems there are two different transmission schemes for the diagnostic bits, which are illustrated in Figure 46.

<b>ADI_EBL(3:0)</b>		Addr. 0x0, 0x0B; bit 3:0	default: 0001
Code	Error bit length per slave		
<b>Single-slave operation (ADI_MSO = 000)</b>			
0000	0		
0001	1		
...	...		
0111	7		
1000	8		
others	invalid		
<b>Multi-slave operation (ADI_MSO ≠ 000)</b>			
0000	0		
0001	1 (individual diagnostic bit transmission)		
0010	1 (gathered diagnostic bit transmission)		
others	invalid		
Note	See Figure 46 for the difference between individual and gathered diagnostic bit transmission.		

Table 76: Error Bit Length per Slave

<b>ADI_EBP</b>		Addr. 0x0, 0x0B; bit 4	default: 0
Code	Function		
0	Slave diagnostic bit polarity active low		
1	Slave diagnostic bit polarity active high		

Table 77: Diagnostic Bit Polarity

<b>ADI_CFG(5)</b>		Addr. 0x0, 0x0C; bit 5	default: 0
Code	Function		
0	No warning bit(s) in ADI data stream		
1	Evaluate warning bit(s) in ADI data stream		
Note	For ADI_CFG(5) = 1 the warning bit length corresponds to the configured error bit length.		

Table 78: Usage of Warning Bit(s)

### Interface Settings

Register ADI\_CFG contains additional important ADI configuration parameters, as described in the following tables.

<b>ADI_CFG(8)</b>		Addr. 0x0, 0x0D; bit 0	default: 0
Code	Function		
0	Slave position data in binary code		
1	Slave position data in gray code		
Note	The data format refers to the position data only (revolution counter + synchronization bits) and not the diagnostic bits.		

Table 79: ADI Slave Position Data Format

<b>ADI_CFG(6)</b>			Addr. 0x0, 0x0C; bit 6	default: 0
Code	Function	Typ. frequency		
0	Slow ACL mode	150 kHz		
1	Fast ACL mode	1500 kHz		

Table 80: ACL Clock Operating Mode

<b>ADI_CFG(4)</b>		Addr. 0x0, 0x0C; bit 4	default: 1
Code	Function		
0	No cyclic counter verification (ACL=const=1)		
1	Cyclic counter verification vs. external data		

Table 81: Cyclic Counter Verification vs. external data

<b>ADI_CFG(3)</b>		Addr. 0x0, 0x0C; bit 3	default: 0
Code	Function		
0	Always keep internal counter		
1	Always load internal counter with external data		
Note	In case of ADI_CFG(3) = 1 the internal counter is always loaded with the available external data, but there is one exception: For ADI_CFG(2) = 1 the internal counter is kept in case of a single error event (ADA stuck-at, counter verification or error bit set).		

Table 82: External Data Priority

<b>ADI_CFG(2)</b>		Addr. 0x0, 0x0C; bit 2	default: 0
Code	Function		
0	Report single error events immediately		
1	Ignore single error events and keep internal counter value		
Note	In case of ADI_CFG(2) = 1, errors are only reported in case of two consecutive erroneous communications.		

Table 83: Double Error Messaging

<b>ADI_CFG(1)</b>		Addr. 0x0, 0x0C; bit 1	default: 0
Code	Function		
0	Default frame repetition period during startup (1500 µs)		
1	Modified frame repetition period during startup (200 µs)		
Note	For fast slaves a low frame repetition period during startup can be used to speed up the startup phase. It is recommended to configure the fast ACL mode here. Otherwise, the SSI communication might already exceed the target frame repetition period of 200 µs. In this case the actual frame repetition period is extended by iC-PZ to ensure a minimum waiting time of 50 µs between ACL pulse trains.		

Table 84: Fast Startup Enable

### Startup

During system startup (after reset via power-cycle, NRES pin low or reset command) and if the interface is enabled ( $ADI\_CFG(7) = 1$ ), the ADI master cyclically tries to read data from the ADI slave(s). In case the slave(s) is(are) not yet ready (indicated by  $ADA = \text{const} = 1$  or  $ADA = \text{const} = 0$ , which is recommended), the ADI master will stay in the startup phase and wait for the slave(s). If the ADI slave(s) does(do) not become ready, the startup phase is aborted after 100 ms and an error is reported. Note: The ADI master ensures a minimum idle time of 50  $\mu\text{s}$ , before it initiates the first communication at the beginning of the startup phase.

Once valid communication is established with the ADI slave(s), the startup phase is not immediately left. To successfully finish the startup phase, two consecutive communications have to be valid (no protocol error), the internal counter versus external data verification has to be ok and no error bit in the protocol must be set. In case all conditions are met, the startup phase is left and the interface continues with normal operation.

Besides the system startup, the ADI master's startup phase can also be triggered by disabling and re-enabling the interface via  $ADI\_CFG(7)$  or by executing command  $ADI\_RESET$ . Note that in this case there is

no termination criterion, i.e., the master will stay in the startup phase forever, if no valid communication can be established with the slave(s). The startup phase has in this case successfully finished, once  $DIAG(21) = 0$  (Int./ext. RC position comparison) after clearing the status register.

The frame repetition period during startup can be lowered to 200  $\mu\text{s}$  via  $ADI\_CFG(1)$ . For fast slaves this might be useful to speed up the startup phase.

### Diagnosis

In addition to the synchronized position verification,  $DIAG(21)$ , the received error and warning bits,  $DIAG(19:16)$  and  $DIAG(20)$ , the ADI master also checks the following SSI protocol conditions:

1. ADA line is 1, right before the first ACL falling edge (start bit)  $\rightarrow$  Verifies that the last frame finished correctly and detects ADA stuck-at-zero,  $DIAG(22)$ .
2. ADA line is 0, right after the last ACL rising edge (stop bit)  $\rightarrow$  Verifies the correct timeout and protocol length and detects ADA stuck-at-one,  $DIAG(23)$ .

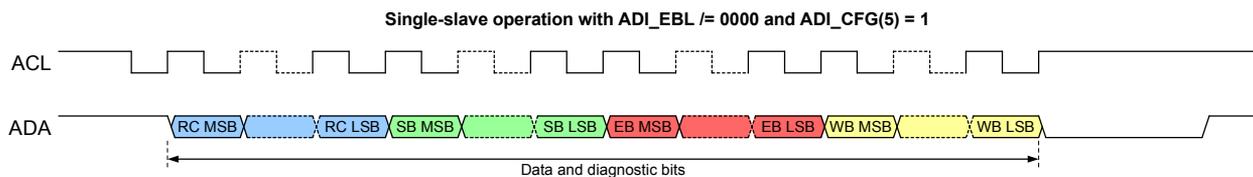


Figure 45: Single-slave operation – exemplary SSI protocol format

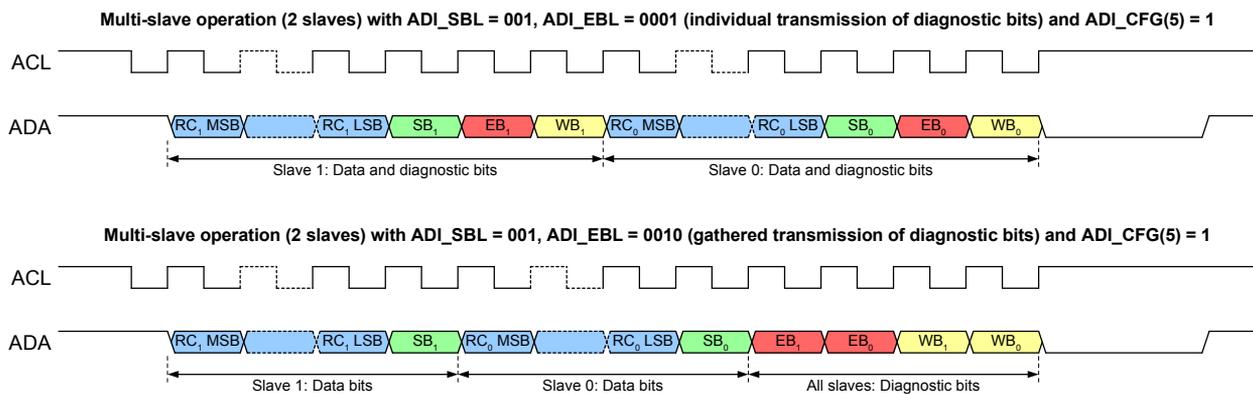


Figure 46: Multi-slave operation – exemplary SSI protocol formats

# iC-PZ Series

HIGH-RESOLUTION REFLECTIVE ABSOLUTE ENCODER



Rev F1, Page 59/85

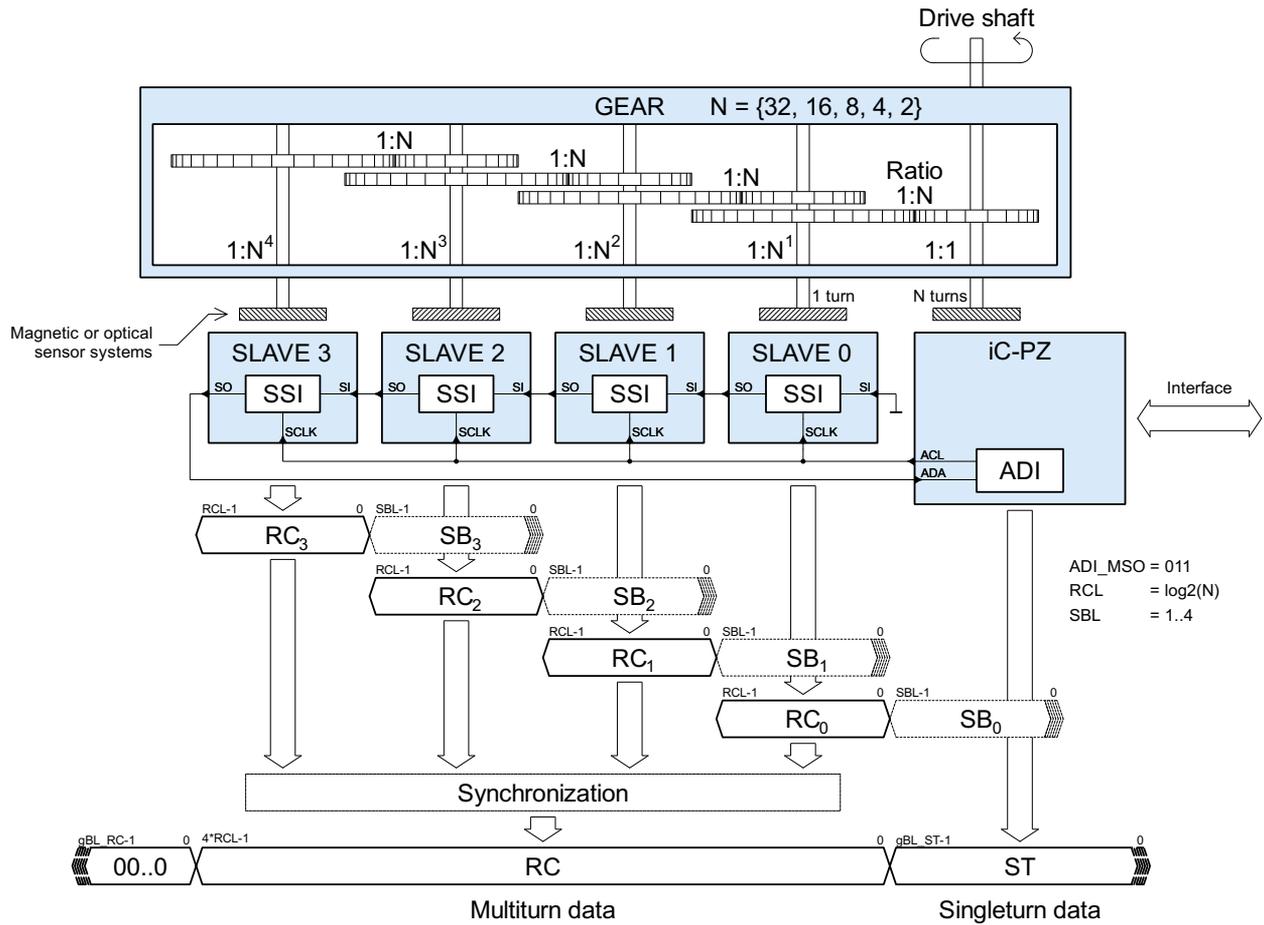


Figure 47: Multi-slave operation – application example with 4 slaves

### I2C MASTER

The implemented I2C master is connected to at least one I2C slave, the EEPROM. Besides, it is possible to communicate with up to four additional slaves connected to the I2C master. The I2C communication can be initiated via BiSS or SPI.

The applied I2C clock frequency is selected via [I2C\\_F\\_x](#). Up to 100 kHz and up to 400 kHz are supported. The individual I2C slave address has to be specified in [I2C\\_DEV\\_ID\\_x](#). Only 7-bit addresses are supported. Additionally, the memory architecture of the individual I2C slave has to be defined via [I2C\\_T\\_x](#). 8-bit or 16-bit devices are supported.

#### Communication with 8-bit devices

When using BiSS to initiate the I2C communication, the general procedure is quite similar to common on-chip register access. However, the I2C transmission takes additional processing time that has to be taken into account. No register accesses are allowed until the requested data has been fully received.

When using SPI to initiate the I2C communication, specific opcodes are available for that purpose. Refer to SPI SLAVE on page 48 for details.

#### Communication with 16-bit devices

The procedure described for 8-bit devices is still applicable. However, transferred data is buffered in [I2C\\_DATA\\_x\(15:0\)](#). When reading, the received byte must be ignored. Data will be available in [I2C\\_DATA\\_x\(15:0\)](#) after transmission has been completed. When writing, data to be transmitted has to be written to [I2C\\_DATA\\_x\(15:0\)](#) first. Then the appropriate procedures have to be executed, ignoring the data that is sent.

See Figures 48 and 49 for details of the supported I2C transmission modes.

**Note:** Before initiating any I2C communication, the active bank [BSEL](#) has to be set according to the address space used by the I2C slave. Refer to MEMORY ORGANIZATION on page 22 for details.

<b>I2C_F_0</b>	Addr. 0xA, 0x04; bit 0	default: 0
<b>I2C_F_1</b>	Addr. 0xA, 0x05; bit 0	default: 0
<b>I2C_F_2</b>	Addr. 0xA, 0x06; bit 0	default: 0
<b>I2C_F_3</b>	Addr. 0xA, 0x07; bit 0	default: 0
Code	Description	
0	Up to 100 kHz I2C frequency	
1	Up to 400 kHz I2C frequency	

Table 85: I2C Frequency

<b>I2C_DEV_ID_0</b>	Addr. 0xA, 0x00; bit 7:0	default: 0x00
<b>I2C_DEV_ID_1</b>	Addr. 0xA, 0x01; bit 7:0	default: 0x00
<b>I2C_DEV_ID_2</b>	Addr. 0xA, 0x02; bit 7:0	default: 0x00
<b>I2C_DEV_ID_3</b>	Addr. 0xA, 0x03; bit 7:0	default: 0x00
Code	Description	
0x80..0x9F, 0xB0..0xFF	available 7-bit I2C device ID, left-aligned, LSB is not used	
others	forbidden	

Table 86: I2C Device ID

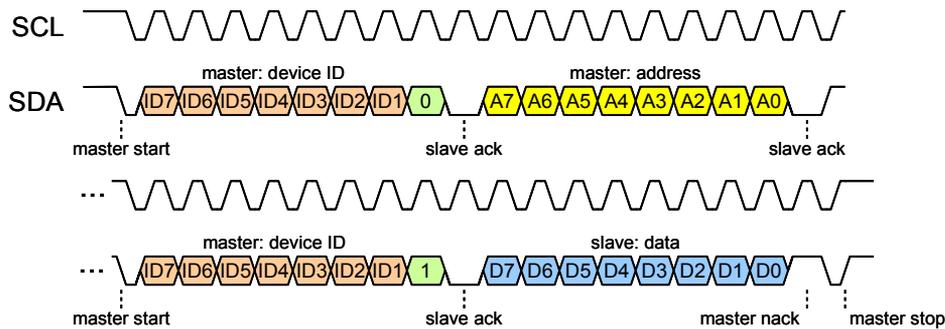
<b>I2C_T_0</b>	Addr. 0xA, 0x04; bit 7:1	default: 0x00
<b>I2C_T_1</b>	Addr. 0xA, 0x05; bit 7:1	default: 0x00
<b>I2C_T_2</b>	Addr. 0xA, 0x06; bit 7:1	default: 0x00
<b>I2C_T_3</b>	Addr. 0xA, 0x07; bit 7:1	default: 0x00
Code	Description	
0x00	8-bit access (devices like EEPROMs)	
0x01	16-bit access (devices like temperature sensors)	

Table 87: I2C Slave Architecture Type

<b>I2C_DATA_0(7:0)</b>	Addr. 0x60; bit 7:0	default: 0x00
<b>I2C_DATA_0(15:8)</b>	Addr. 0x61; bit 7:0	default: 0x00
<b>I2C_DATA_1(7:0)</b>	Addr. 0x62; bit 7:0	default: 0x00
<b>I2C_DATA_1(15:8)</b>	Addr. 0x63; bit 7:0	default: 0x00
<b>I2C_DATA_2(7:0)</b>	Addr. 0x64; bit 7:0	default: 0x00
<b>I2C_DATA_2(15:8)</b>	Addr. 0x65; bit 7:0	default: 0x00
<b>I2C_DATA_3(7:0)</b>	Addr. 0x66; bit 7:0	default: 0x00
<b>I2C_DATA_3(15:8)</b>	Addr. 0x67; bit 7:0	default: 0x00
Value	Description	
	I2C data for communication with 16-bit devices ( <a href="#">I2C_T_x</a> = 0x01)	

Table 88: I2C Data

### I<sup>2</sup>C register read (8 bit)



### I<sup>2</sup>C register write (8 bit)

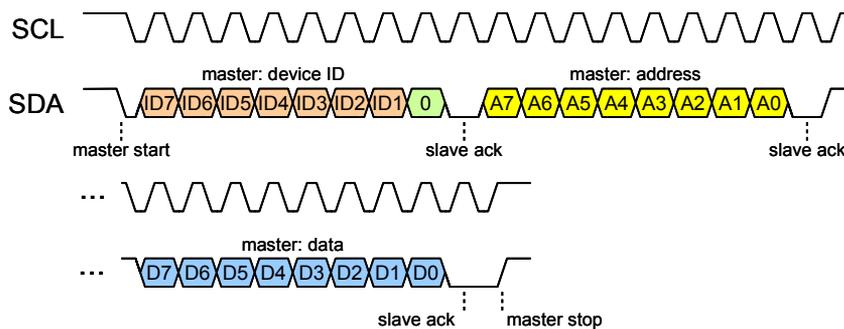
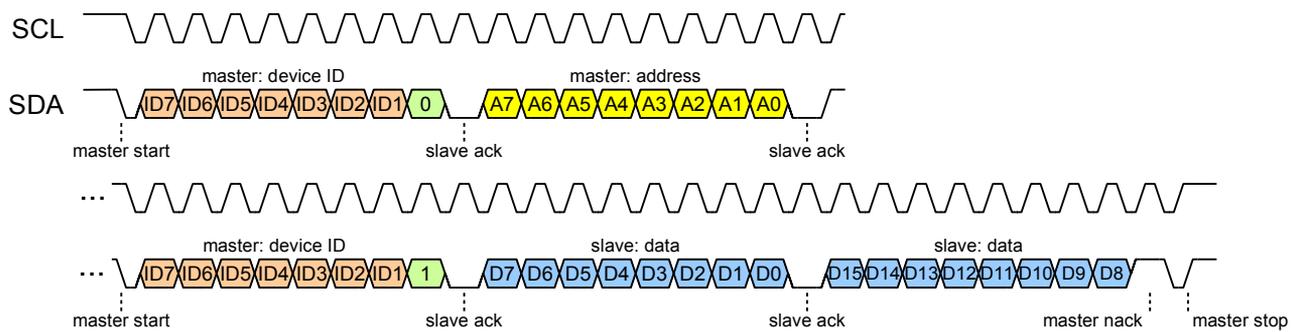


Figure 48: I2C 8-bit data protocol

### I<sup>2</sup>C register read (16 bit)



### I<sup>2</sup>C register write (16 bit)

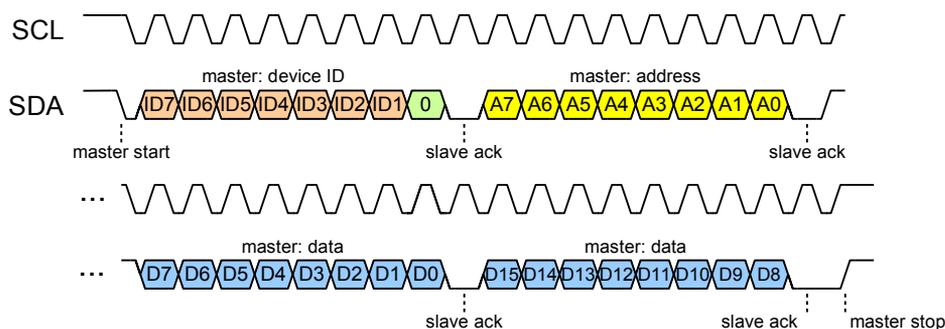


Figure 49: I2C 16-bit data protocol

### GPIO

Two General Purpose Input/Output (GPIO) pins are provided by iC-PZ. Via [GPIOx\\_CFG\(1:0\)](#) each pin can be configured as input, push-pull output or open-drain in-out versus GNDIO or VDDIO. A graphical overview of the GPIO signal path is shown in Figure 50.

GPIO0_CFG(1:0)		Addr. 0x9, 0x08; bit 1:0	default: 01
GPIO1_CFG(1:0)		Addr. 0x9, 0x08; bit 5:4	default: 00
Code	Description		
00	Input only with pull-up		
01	Open-drain (hard 0) and input with pull-up		
10	Open-drain (hard 1) and input with pull-down		
11	Push-pull (and input)		

Table 89: GPIO Output Pad Configuration

#### GPIO as Input

When configured as input, the current state of GPIO(x) can be read from [GPIO\\_IN\(x\)](#). Additionally, an input can be used to trigger [DIAG\(26\)](#) or [DIAG\(27\)](#), which is useful to detect an error/warning from an external source. In [GPIOx\\_DIAG](#) the polarity triggering the error/warning is defined.

GPIO_IN(1:0)		Addr. 0x4D; bit 3:2	default: n/a
Bit	Description		
1:0	Reads the current state of pins GPIO(1:0). The parameter is read-only		

Table 90: GPIO Input State

#### GPIO as Output

When configured as output, the source controlling the logical pin state is selected via [GPIOx\\_SEL](#). The output can either be controlled through setting the corresponding bit of [GPIO\\_OUT\(1:0\)](#), or using the state of specific events logged in [DIAG](#). The bitmask [GPIOx\\_M](#) is used to select or deselect events for that purpose. In [GPIOx\\_DIAG](#) the polarity indicating the error/warning is defined.

GPIO0_SEL		Addr. 0x9, 0x08; bit 3	default: 0
GPIO1_SEL		Addr. 0x9, 0x08; bit 7	default: 0
Code	Description		
0	Pin GPIO(x) is driven by output of <a href="#">DIAG</a> masked with <a href="#">GPIOx_M</a>		
1	Pin GPIO(x) is driven by register <a href="#">GPIO_OUT(x)</a>		

Table 91: GPIO Output Selection

GPIO_OUT(1:0)		Addr. 0x4D; bit 1:0	default: 00
Bit	Description		
1:0	Defines output of pins GPIO(x), if selected by <a href="#">GPIOx_SEL</a> . Can be written to directly, or command <a href="#">GPIO_OUTx_SETx</a> can be sent		

Table 92: GPIO Output State

GPIO0_DIAG		Addr. 0x9, 0x08; bit 2	default: 0
GPIO1_DIAG		Addr. 0x9, 0x08; bit 6	default: 0
Code	Description		
0	For input: 0 is interpreted as error/warning in <a href="#">DIAG</a> For output: error is output as 0		
1	For input: 1 is interpreted as error/warning in <a href="#">DIAG</a> For output: error is output as 1		

Table 93: GPIO Diagnosis Polarity

GPIO0_M(7:0)		Addr. 0x9, 0x00; bit 7:0	default: 0x01000000
GPIO0_M(15:8)		Addr. 0x9, 0x01; bit 7:0	default: 0x00000000
GPIO0_M(23:16)		Addr. 0x9, 0x02; bit 7:0	default: 0x00000000
GPIO0_M(31:24)		Addr. 0x9, 0x03; bit 7:0	default: 0x00000000
GPIO1_M(7:0)		Addr. 0x9, 0x04; bit 7:0	default: 0x00000000
GPIO1_M(15:8)		Addr. 0x9, 0x05; bit 7:0	default: 0x00000000
GPIO1_M(23:16)		Addr. 0x9, 0x06; bit 7:0	default: 0x00000000
GPIO1_M(31:24)		Addr. 0x9, 0x07; bit 7:0	default: 0x00000000
Bit	Description		
31:0	Selected bits of <a href="#">DIAG</a> to be forwarded		

Table 94: GPIO(0/1) Bit Mask

**Note:** By default GPIO(0) is configured as open-drain output. After system startup has been completed successfully, GPIO(0) is set from hard 0 to pull-up 1.

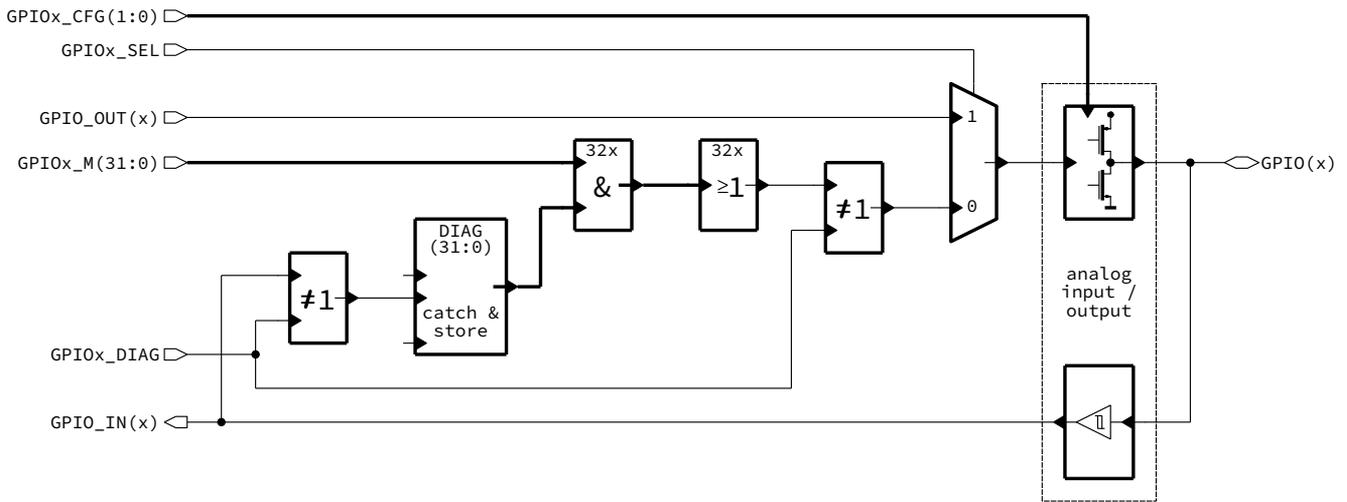


Figure 50: GPIO datapath overview

### TEMPERATURE SENSOR

An on-chip temperature sensor with a resolution of 0.1 °C is integrated in iC-PZ. The temperature value is stored in **TEMP**. Refer to ELECTRICAL CHARACTERISTICS T01 for the temperature range covered.

but the upper four bits are omitted. **TEMP\_LT\_x** defines whether an upper or lower limit is set. If the temperature value is outside those limits at any time, an error/warning is reported in **DIAG**.

<b>TEMP(7:0)</b>		Addr. 0x4E; bit 7:0	read-only
<b>TEMP(15:8)</b>		Addr. 0x4F; bit 7:0	read-only
Code	Value		
Signed: 2K	$\frac{\text{value}}{10} \text{ } ^\circ\text{C}$		
...	...		
0xFE70	-40.0 °C		
...	...		
0xFFFF	- 0.1 °C		
0x0000	0.0 °C		
0x0001	0.1 °C		
...	...		
0x0578	140.0 °C		
...	...		

Table 95: Temperature

<b>TEMP_L_1(7:0)</b>		Addr. 0xD,0x00; bit 7:0	default: 0x578
<b>TEMP_L_1(11:8)</b>		Addr. 0xD,0x01; bit 3:0	
<b>TEMP_L_2(7:0)</b>		Addr. 0xD,0x02; bit 7:0	default: 0xE70
<b>TEMP_L_2(11:8)</b>		Addr. 0xD,0x03; bit 3:0	
Code	Value		
	Coding equivalent to <b>TEMP</b> , omitting bits 15:12		
...	...		
0xE70	-40.0 °C		
...	...		
0xFFF	- 0.1 °C		
0x000	0.0 °C		
0x001	0.1 °C		
...	...		
0x578	140.0 °C		
...	...		

Table 96: Temperature Limit 1/2



**TEMP** is latched by reading its lowest address 0x4E so that all corresponding register values are related to the same request. Reading this parameter from lowest to highest address is mandatory.

Additionally, two temperature limits can be set using **TEMP\_L\_x**. The limit uses the same coding as **TEMP**,

<b>TEMP_LT_1</b>		Addr. 0xD, 0x04; bit 0	default: 0
<b>TEMP_LT_2</b>		Addr. 0xD, 0x04; bit 1	default: 1
Code	Value		
0	Upper limit: will trigger diagnosis error/warning if temperature exceeds limit.		
1	Lower limit: will trigger diagnosis error/warning if temperature falls below limit.		

Table 97: Temperature Limit Type 1/2

### DIAGNOSIS

An extensive diagnosis and error/warning reporting mechanism is provided by iC-PZ. A lot of parameters are monitored and recapped in the **DIAG** registers. They are caught and stored, even if occurring as a single event for a very short period of time. By masking specific bits in **DIAG**, the errors and warnings being reported in **ERR** and **WARN** can be selected individually.

For each serial interface an individual error/warning configuration can be defined. Refer to BiSS SLAVE, SSI SLAVE, and SPI SLAVE for details. An overview of the diagnosis datapath is shown in Figure 51.

Writing to **DIAG** will set the individually addressed bits. This can be used for testing purposes.

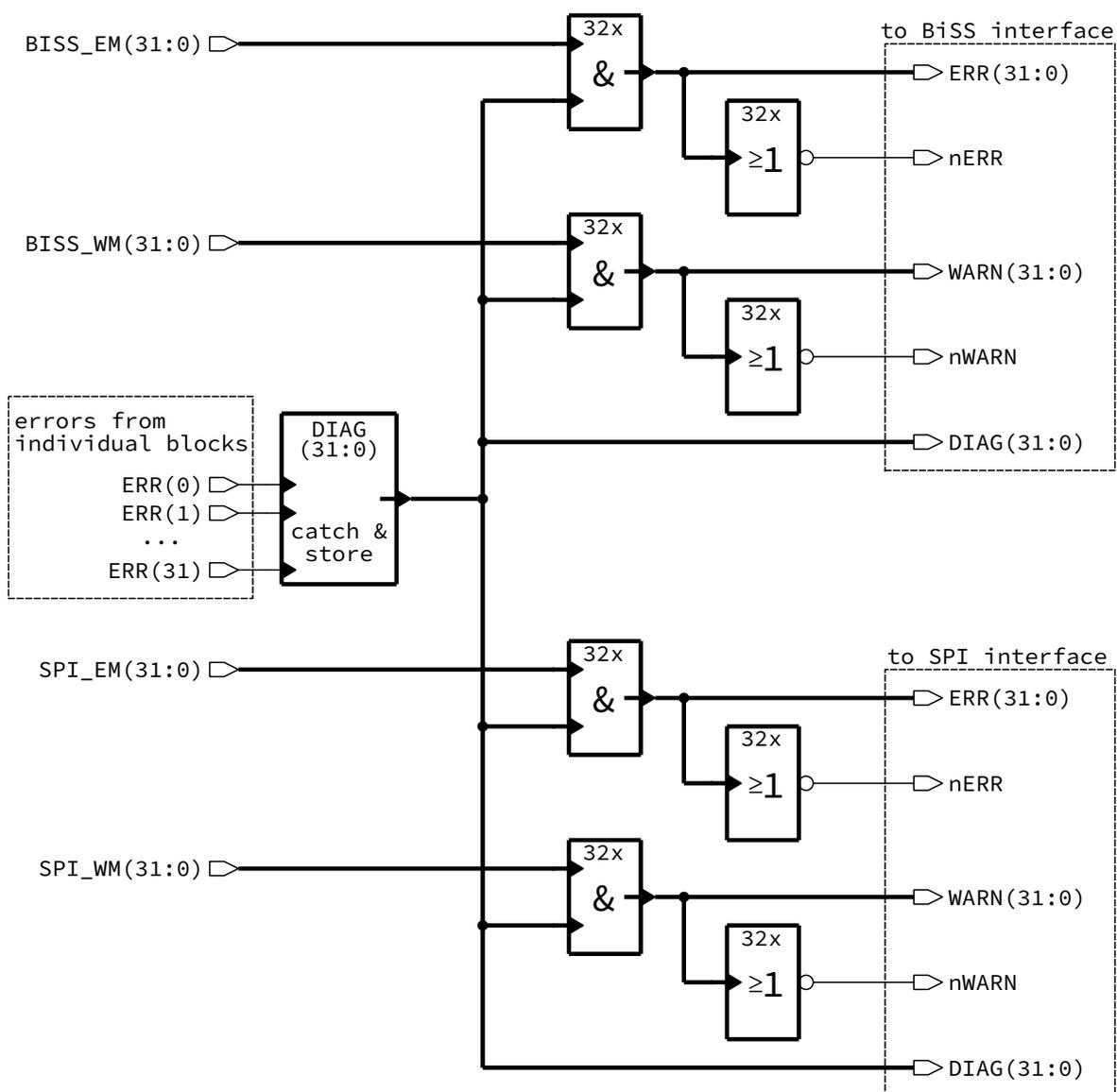


Figure 51: Diagnosis datapath overview

DIAG(7:0)	Addr. 0x68; bit 7:0	default:
DIAG(15:8)	Addr. 0x69; bit 7:0	0x00000000
DIAG(23:16)	Addr. 0x6A; bit 7:0	
DIAG(31:24)	Addr. 0x6B; bit 7:0	
Bit	Description	Error Condition
<b>Analog blocks:</b>		
0	Digital photo-amplifiers in saturation	Illumination of at least one of the digital photo-diodes is too high, digital photocurrent-amplifier saturates
1	LED current low	LED current smaller than 50% of the nominal current of square or sum control
2	Temperature sensor not in steady state	Temperature sensor has not yet found the actual chip temperature
3	VDDIO voltage low	Voltage at VDDIO below threshold set by VDDIOSEL as defined in Elec. Char. P05
4	Interpolator error	Sinusoidal (analog) interpolator position is not within $\pm 22.5^\circ$ of the digital interpolator position
6	ABZ-Interface not ready	The ABZ-Interface has not yet reached the current position. Relevant at startup only. This error is not latched and automatically cleared when the position is reached.
7	UVW-Interface not ready	The UVW-Interface has not yet reached the current position. Relevant at startup only. This error is not latched and automatically cleared when the position is reached.
<b>Digital blocks:</b>		
8	Position filter: alpha overflow	Acceleration register alpha exceeds an internally defined limit resulting in new filter startup ( $\approx 256 \mu\text{s}$ )
9	Position filter: omega overflow	Velocity register omega exceeds an internally defined limit resulting in new filter startup ( $\approx 256 \mu\text{s}$ )
10	Digital photo-amplifiers not in steady state	Digital photo-amplifiers did not yet startup successfully
11	PRC synchronization failed	PRC mismatching tolerance set in <a href="#">RAN_TOL</a> exceeded. Most likely the PRC track was not sampled correctly. See chapter ADJUSTMENT DIGITAL for details on setting up the sampling.
12	Analog adjustment parameter at boundary value	Parameter <a href="#">COS_OFFS</a> , <a href="#">SIN_OFFS</a> , <a href="#">SC_GAINS</a> or <a href="#">SC_PHASES</a> reached minimum or maximum value, or dynamic part of one of these registers is unexpected high ( $\pm 12.5\%$ of the range)
13	Digital adjustment parameter at boundary value	Parameter <a href="#">AI_PHASES</a> or <a href="#">AI_SCALES</a> reached minimum or maximum value, or dynamic part of one of these registers is unexpected high ( $\pm 12.5\%$ of the range)
14	Temperature limit 1	Chip temperature exceeds / falls below temperature limit defined by <a href="#">TEMP_L_1</a> and <a href="#">TEMP_LT_1</a>
15	Temperature limit 2	Chip temperature exceeds / falls below temperature limit defined by <a href="#">TEMP_L_2</a> and <a href="#">TEMP_LT_2</a>
<b>Absolute data interface (ADI):</b>		
19:16	Multiturn error bit(s) read from slave(s)	Mapping of multiturn error bit(s) as set and transmitted by multiturn slave(s). Error bits of slaves 0..2 are individually mapped to diagnosis bits 16..18, while error bits of slaves 3..7 are reduced (OR-operation) to diagnosis bit 19.
20	Multiturn warning bit(s) read from slave(s)	Mapping of multiturn warning bit(s) as set and transmitted by multiturn slave(s). Warning bits of all slaves are reduced (OR-operation) to this single diagnosis bit
21	Multiturn position comparison failed	Mismatch of internally counted revolution counter (RC) with external (synchronized) RC from multiturn interface
22	Pin ADA stuck-at-0	SSI data line pin (ADA) is stuck-at-0 → No communication is possible
23	Pin ADA stuck-at-1	SSI data line pin (ADA) is stuck-at-1 → No communication is possible
<b>Digital control:</b>		
24	System startup in progress	System is starting up and not yet ready
25	EEPROM CRC error	At least one CRC in the EEPROM is invalid (or EEPROM communication failed)
26	Pin GPIO(0) as error input	Pin GPIO(0) is used as external error input. Error is reported if $\text{GPIO}(0) \neq \text{GPIO0\_DIAG}$
27	Pin GPIO(1) as error input	Pin GPIO(1) is used as external error input. Error is reported if $\text{GPIO}(1) \neq \text{GPIO1\_DIAG}$
28	System startup aborted due to timeout	Timeout reached while waiting for functional block to become ready at system startup
31:29	User available	User-defined diagnosis register bits. Can be set and reset by any serial interface.

Table 98: Diagnosis

### Errors & Warnings

Individual error and warning bits can be cleared by writing the corresponding bits to **ERR** or **WARN**. If an error is cleared, the warning is also affected and the other way around. All error and warning bits are cleared by either writing 0xFFFFFFFF to **ERR** or **WARN** or executing command **SCLEAR**. Clearing errors or warnings affects all interfaces at the same time.

**Note:** The EEPROM CRC error bit in **DIAG(25)** cannot be cleared externally. For that purpose either one of the commands **CONF\_READ** and **CONF\_READ\_ALL** has to be executed or a power-on reset has to be performed.

<b>ERR(7:0)</b>	Addr. 0x6C; bit 7:0	default:
<b>ERR(15:8)</b>	Addr. 0x6D; bit 7:0	0x00000000
<b>ERR(23:16)</b>	Addr. 0x6E; bit 7:0	
<b>ERR(31:24)</b>	Addr. 0x6F; bit 7:0	
Bit	Description	
31:0	Errors as indicated by <b>DIAG(31:0)</b> and masked with <b>BISS_EM(31:0)</b> / <b>SPI_EM(31:0)</b> . Each interface reads its individually selected errors only.	

Table 99: Error

<b>WARN(7:0)</b>	Addr. 0x70; bit 7:0	default:
<b>WARN(15:8)</b>	Addr. 0x71; bit 7:0	0x00000000
<b>WARN(23:16)</b>	Addr. 0x72; bit 7:0	
<b>WARN(31:24)</b>	Addr. 0x73; bit 7:0	
Bit	Description	
31:0	Warnings as indicated by <b>DIAG(31:0)</b> and masked with <b>BISS_EM(31:0)</b> / <b>SPI_EM(31:0)</b> . Each interface reads its individually selected warnings only.	

Table 100: Warning

### CRC Status

Invalid CRC values of a bank are marked as 1 in **CRC\_STAT**. After startup, **CRC\_STAT** is automatically updated. Banks that could not be read from the EEPROM due to communication problems, are marked as invalid as well. **CRC\_STAT** = 0x0000 indicates that all CRC values are valid. By using the command **CRC\_CALC**, correct CRC values for all banks are calculated and marked as valid in **CRC\_STAT**.

Example:

**CRC\_STAT** = 0b0000000000001001 indicates invalid CRC values for bank 0x0 and bank 0x3.

<b>CRC_STAT(7:0)</b>	Addr. 0x74; bit 7:0	default: 0xFFFF,
<b>CRC_STAT(15:8)</b>	Addr. 0x75; bit 7:0	read-only
Bit	Value	
0:14	Register banks 0x0 to 0xE	
15	EEPROM protection CRC	

Table 101: CRC Status

### SYSTEM CONFIGURATION

#### Revision & Identification

**ID**, **REV**, and **SYS** are defined by iC-Haus and are read-only.

<b>ID(7:0)</b>	Addr. 0x48; bit 7:0	read-only
<b>ID(15:8)</b>	Addr. 0x49; bit 7:0	read-only
<b>ID(23:16)</b>	Addr. 0x4A; bit 7:0	read-only
<b>ID(31:24)</b>	Addr. 0x4B; bit 7:0	read-only
Individual chip identification number from iC-Haus		

Table 102: ID

<b>REV(3:0)</b>		Addr. 0x4C; bit 3:0	read-only
Code	Value		
0x0	iC-PZxxxx_0		
0x1	iC-PZxxxx_Z		
0x2	iC-PZxxxx_Y		
0x3	iC-PZxxxx_X		

Table 103: Chip Revision

<b>SYS(3:0)</b>		Addr. 0x4C; bit 7:4	read-only
Code	Value		
0x7	7-bit system, native for iC-PZ0974 rotary Ø9 mm		
0x8	8-bit system, native for iC-PZ2656 rotary Ø26 mm		
0xF	15-bit system, native for iC-PZ205 linear with length up to 6.71m		

Table 104: Chip System

#### System Definition

If the resolution provided by the system is not equal to **SYS**, in case the FLEXCODE® feature is used, **SYS\_OVR** has to be set accordingly. The effective chip system **SYS\_eff** is defined as:

$$\text{SYS\_eff} = \text{SYS} \quad \text{for } \text{SYS\_OVR} = 0$$

$$\text{SYS\_eff} = \text{SYS\_OVR} \quad \text{for } \text{SYS\_OVR} \neq 0$$

<b>SYS_OVR(3:0)</b>		Addr. 0x0, 0x07; bit 7:4	0x0
Code	Value		
0x0	no override: <b>SYS_eff</b> equals system-setting <b>SYS</b>		
others	override: <b>SYS_eff</b> equals <b>SYS_OVR</b>		
	iC-PZ0974: 0x7		
	iC-PZ2656: 0x7 ... 0x9		
	iC-PZ205: 0x9 ... 0xF		

Table 105: Chip System Override



If **SYS\_OVR** is adapted, that change has to be written to the EEPROM and the system must then be rebooted.

In case discs/scales with inverted and/or flipped codes are used, **CD\_INV** and **CD\_FLIP** can adjust iC-PZ accordingly. Both corrections can be used at the same time.

<b>CD_INV</b>		Addr. 0x0, 0x07; bit 0	default: 0
Code	Value		
0	Code on disc/scale not inverted (default)		
1	Code on disc/scale inverted (reflective/non-reflective)		

Table 106: Inverted Code Correction

<b>CD_FLIP</b>		Addr. 0x0, 0x07; bit 1	default: 0
Code	Value		
0	Code on disc/scale not flipped (default)		
1	Code on disc/scale flipped (mirrored left-to-right)		

Table 107: Flipped Code Correction

### BiSS Profile & Identifier

[EDS\\_BANK](#), [BISS\\_PROFILE\\_ID](#), [SERIAL](#), [DEV\\_ID](#), and [MFG\\_ID](#) are available as editable parameters in bank 0xE and as read-only registers in the direct access registers 0x40..0x7F. Both are mapped to the same internal memory, so that their content is identical. After editing the parameters in bank 0xE, it can be good practice to protect the registers from being accessed by using command [RPL\\_SET\\_RO](#) (read-only) or [RPL\\_SET\\_NA](#) (no access). As last step, the bank then has to be written to the external EEPROM sending command [CONF\\_WRITE](#).

<b>EDS_BANK(7:0)</b>	Addr. 0x41; bit 7:0	read-only
<b>EDS_BANK_X(7:0)</b>	Addr. 0xE, 0x01; bit 7:0	
Value	Pointer to BiSS EDS bank	

Table 108: BiSS EDS Bank

<b>BISS_PROFILE_ID_1(7:0)</b>	Addr. 0x42; bit 7:0	read-only
<b>BISS_PROFILE_ID_0(7:0)</b>	Addr. 0x43; bit 7:0	read-only
<b>BISS_PROFILE_ID_1_X(7:0)</b>	Addr. 0xE, 0x02; bit 7:0	
<b>BISS_PROFILE_ID_0_X(7:0)</b>	Addr. 0xE, 0x03; bit 7:0	
Value	BiSS profile ID	

Table 109: BiSS Profile ID

<b>SERIAL_3(7:0)</b>	Addr. 0x44; bit 7:0	read-only
<b>SERIAL_2(7:0)</b>	Addr. 0x45; bit 7:0	read-only
<b>SERIAL_1(7:0)</b>	Addr. 0x46; bit 7:0	read-only
<b>SERIAL_0(7:0)</b>	Addr. 0x47; bit 7:0	read-only
<b>SERIAL_3_X(7:0)</b>	Addr. 0xE, 0x04; bit 7:0	
<b>SERIAL_2_X(7:0)</b>	Addr. 0xE, 0x05; bit 7:0	
<b>SERIAL_1_X(7:0)</b>	Addr. 0xE, 0x06; bit 7:0	
<b>SERIAL_0_X(7:0)</b>	Addr. 0xE, 0x07; bit 7:0	
Value	Module manufacturer serial number	

Table 110: Serial Number

<b>DEV_ID_5(7:0)</b>	Addr. 0x78; bit 7:0	read-only
<b>DEV_ID_4(7:0)</b>	Addr. 0x79; bit 7:0	read-only
<b>DEV_ID_3(7:0)</b>	Addr. 0x7A; bit 7:0	read-only
<b>DEV_ID_2(7:0)</b>	Addr. 0x7B; bit 7:0	read-only
<b>DEV_ID_1(7:0)</b>	Addr. 0x7C; bit 7:0	read-only
<b>DEV_ID_0(7:0)</b>	Addr. 0x7D; bit 7:0	read-only
<b>DEV_ID_5_X(7:0)</b>	Addr. 0xE,0x08; bit 7:0	0x50 ('P')
<b>DEV_ID_4_X(7:0)</b>	Addr. 0xE,0x09; bit 7:0	0x5A ('Z')
<b>DEV_ID_3_X(7:0)</b>	Addr. 0xE,0x0A; bit 7:0	SYS & REV
<b>DEV_ID_2_X(7:0)</b>	Addr. 0xE,0x0B; bit 7:0	BISS_ST_DL
<b>DEV_ID_1_X(7:0)</b>	Addr. 0xE,0x0C; bit 7:0	BISS_MT_DL
<b>DEV_ID_0_X(7:0)</b>	Addr. 0xE,0x0D; bit 7:0	CRC16 & ENSOL & CRCS
Value	BiSS device ID	

Table 111: BiSS Device ID

<b>MFG_ID_1(7:0)</b>	Addr. 0x7E; bit 7:0	read-only
<b>MFG_ID_0(7:0)</b>	Addr. 0x7F; bit 7:0	read-only
<b>MFG_ID_1_X(7:0)</b>	Addr. 0xE,0x0E; bit 7:0	0x69 ('i')
<b>MFG_ID_0_X(7:0)</b>	Addr. 0xE,0x0F; bit 7:0	0x43 ('C')
Value	BiSS manufacturer ID	

Table 112: BiSS Manufacturer ID

### FLEXCODE®

With the FlexCode® feature the available iC-PZ devices can be used to realize rotative absolute encoder systems with arbitrary code disc diameters:

Device	Covered code disc diameters
iC-PZ2656	16.2 mm ... 44.6 mm
iC-PZ205	44.6 mm ... up to linear 6.71 m

Table 113: FlexCode® supported disc diameters

The FlexCode® system is set up with **FCL** and **FCS**. Additionally, overriding the chip system via **SYS\_OVR** might be necessary. Two examples of FlexCode®-systems are given in Table 116. Parameters and settings for specific code disc diameters are provided at [support@ichaus.de](mailto:support@ichaus.de).

FCL(7:0)	Addr. 0x8, 0x00; bit 7:0	default:
FCL(14:8)	Addr. 0x8, 0x01; bit 6:0	0x0000
Code	Value	
0x0000	FlexCode® disabled	
others	See Table 116 for example	

Table 114: FlexCode® Length

FCS(7:0)	Addr. 0x8, 0x02; bit 7:0	default:
FCS(14:8)	Addr. 0x8, 0x03; bit 6:0	0x0000
See Table 116 for example		

Table 115: FlexCode® Identifier

Code disc	Device	SYS_OVR	FCL	FCS
PZ08S, ∅ 44 mm	iC-PZ2656	9	446	216
Other FlexCode®-systems on request.				

Table 116: FlexCode®-system example

Independent of the number of increments on the code disc, the digital interfaces BiSS, SSI, and SPI always output data as fully coded powers of 2, i.e., the binary singleturn value range is defined by **SYS\_eff** bits for the PRC track (MSBs) plus the interpolated bits (LSBs).

Example:

iC-PZ2656 with PZ08S is a FlexCode®-system with **SYS\_eff** = 9 and **FCL** = 446 (number of increments on the code disc). During a full mechanical revolution of 360°m, the 9 MSBs of the output singleturn position take values between 0 and 511.

### Specification Modifications for FlexCode®

In a FlexCode®-system, the hysteresis values given in °e in Table 22 change according to

$$\frac{\text{FCL}}{2^{\text{SYS\_eff}}} \cdot \frac{2 \cdot \text{ABZ\_HYS}}{2^{14}} \cdot 360^\circ \text{e}$$

The maximum rotary speed defined in Table 145 is

$$\text{RPM} \leq \frac{14.4 \cdot 10^6}{\text{FCL}}$$

Minimum speed and time required for analog and digital autocalibration change as follows:

analog (see page 73):

$$n_{\min} = \frac{2^{\text{AC\_COUNT}}}{\text{FCL}} \cdot 1.25 \frac{1}{\text{s}}$$

$$t_{\text{rot}} \approx \frac{2^{\text{AC\_COUNT}}}{\text{FCL}} \cdot \frac{\text{AC\_SEL1} - \text{AC\_SEL2}}{n}$$

digital (see page 75):

$$n_{\min} = \frac{2^{\text{AC\_COUNT}}}{\text{FCL}} \cdot 1.5 \frac{1}{\text{s}}$$

$$t_{\text{rot}} \approx \frac{2^{\text{AC\_COUNT}}}{\text{FCL}} \cdot \frac{\text{AC\_SEL1} - \text{AC\_SEL2}}{\frac{3}{4} \cdot n}$$

### ADJUSTMENT ANALOG

To achieve best interpolation results, the signal quality can be optimized on-chip via analog adjustment. Typical errors such as offset between the positive and the negative phases of cosine and sine, amplitude mismatch between cosine and sine, and incorrect phase shift between cosine and sine can be compensated. Good practice is to use the autocalibrations on command, however entering parameters manually is possible. In general, the raw signals to be adjusted can be described as

$$\begin{aligned} \text{COS} &= \text{PCOS} - \text{NCOS} \\ &= \text{AMP\_COS} \cdot \sin(\omega t + \text{PH\_COS}) + \text{OS\_COS} \end{aligned}$$

$$\begin{aligned} \text{SIN} &= \text{PSIN} - \text{NSIN} \\ &= \text{AMP\_SIN} \cdot \sin(\omega t + \text{PH\_SIN}) + \text{OS\_SIN} \end{aligned}$$

#### Offset

The cosine signals before and after the offset has been adjusted are shown in Figure 52. By setting **COS\_OFF** to a positive value, the DC value of the differential signal is increased. The applied offset is independent of the actual signal amplitudes. For the sine signals this is applied identically.

$$\text{COS} = \cos(\omega t) + \text{OS\_COS} + \text{COROS(COS)}$$

To compensate the offset, a DC value COROS(COS) has to be added in the opposite direction of the measured offset OS\_COS:

$$\text{COROS(COS)} = - \text{OS\_COS}$$

<b>COS_OFF(1:0)</b>	Addr. 0x1, 0x00; bit 7:6	default: 0x000
<b>COS_OFF(9:2)</b>	Addr. 0x1, 0x01; bit 7:0	
<b>SIN_OFF(1:0)</b>	Addr. 0x1, 0x02; bit 7:6	default: 0x000
<b>SIN_OFF(9:2)</b>	Addr. 0x1, 0x03; bit 7:0	
Code	COROS(COS) COROS(SIN)	
Signed (2K)	$\text{COS\_OFF} \cdot 0.235 \text{ mV}$ $\text{SIN\_OFF} \cdot 0.235 \text{ mV}$	
0x000	0.000 mV	
0x001	0.235 mV	
0x002	0.470 mV	
...	...	
0x1FE	119.765 mV	
0x1FF	120.000 mV	
0x200	-120.000 mV (equal to 0x201)	
0x201	-120.000 mV	
0x202	-119.765 mV	
...	...	
0x3FE	-0.470 mV	
0x3FF	-0.235 mV	

Table 117: Adjustment Offset (static)

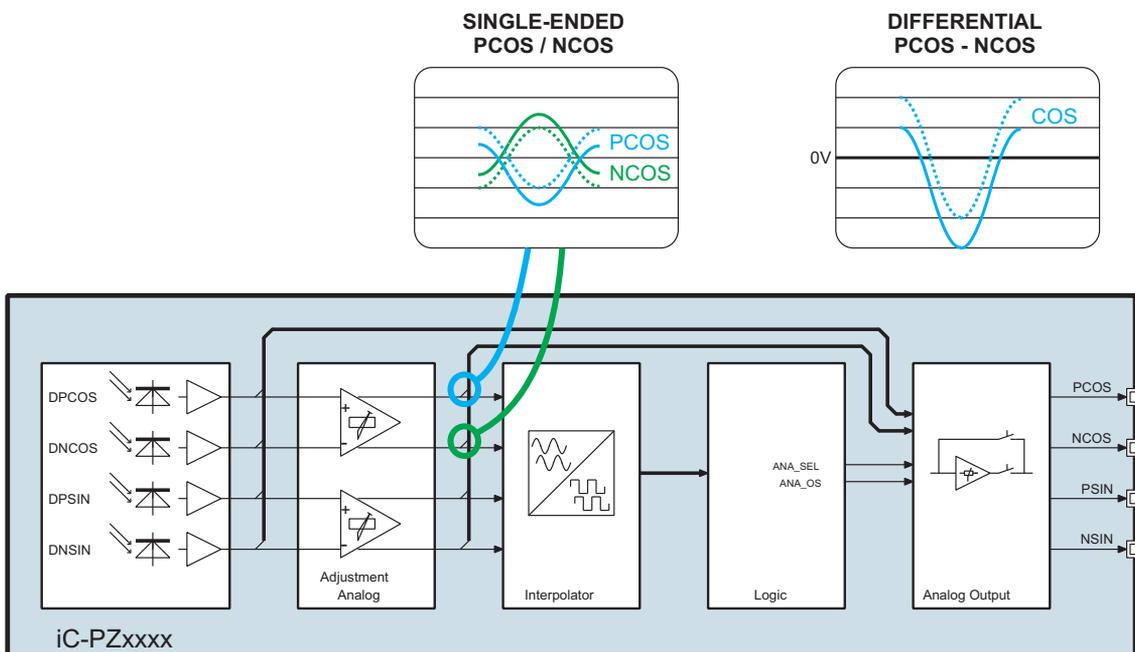


Figure 52: Signals before (solid) and after (dashed) Offset Adjustment with a positive DC value

### Amplitude

All four phases of the cosine and sine signals before and after amplitude adjustment are shown in Figure 53. By changing **SC\_GAIN**, the amplitudes of both cosine and sine are changed.

$$\begin{aligned} \text{COS} &= \text{COS\_GAIN} \cdot \text{AMP\_COS} \cdot \cos(\omega t) \\ \text{SIN} &= \text{SIN\_GAIN} \cdot \text{AMP\_SIN} \cdot \sin(\omega t) \end{aligned}$$

Higher **SC\_GAIN** increases the amplitude of the cosine signals and decreases the amplitude of the sine signals.

The amplitude correction factor CFA is defined as

$$\text{CFA} = \frac{\text{COS\_GAIN}}{\text{SIN\_GAIN}}$$

To correct amplitude mismatch, the correction factor CFA can be calculated as

$$\text{CFA} = \frac{\text{AMP\_SIN}}{\text{AMP\_COS}}$$

**Note:** AMP\_SIN and AMP\_COS are the **uncorrected** amplitudes for **SC\_GAIN** = 0x000.

<b>SC_GAIN(1:0)</b>		Addr. 0x1, 0x04; bit 7:6	default: 0x000
<b>SC_GAIN(9:2)</b>		Addr. 0x1, 0x05; bit 7:0	
Code	CFA		
Signed (2K)	$\left( \frac{14}{11} \right)^{\frac{\text{SC\_GAIN}}{511}}$		
0x200	0.7857 (equal to 0x201)		
0x201	0.7857		
0x202	0.7861		
...	...		
0x3FF	0.9995		
0x000	1.0000		
0x001	1.0005		
...	...		
0x1FE	1.2721		
0x1FF	1.2727		

Table 118: Adjustment cosine-to-sine amplitude ratio (static)

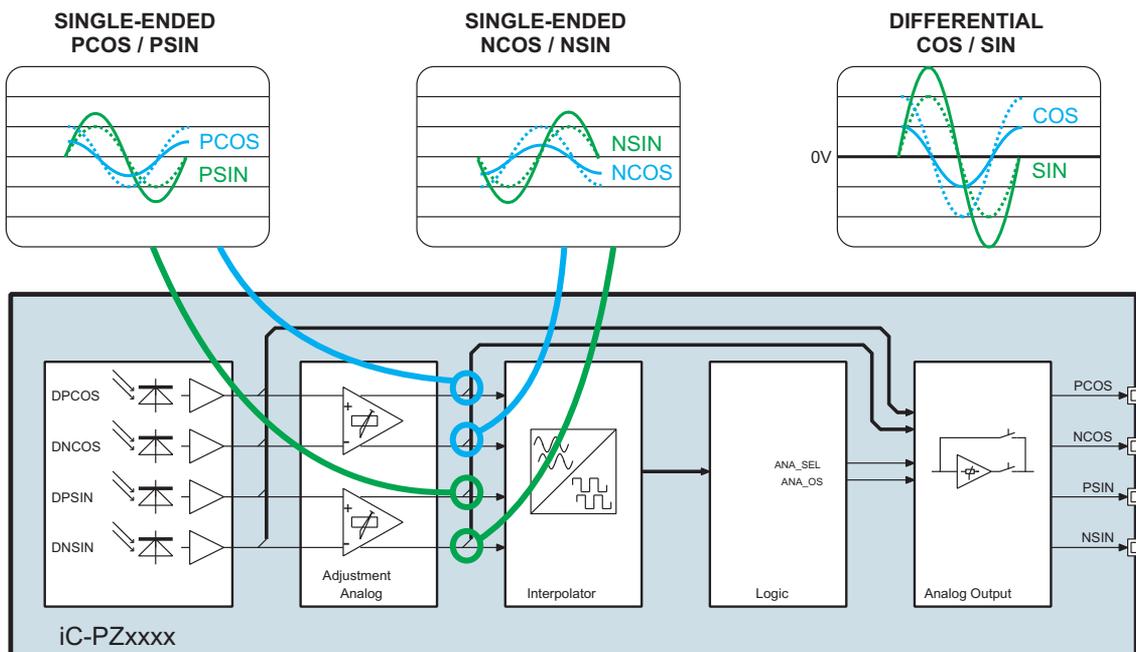


Figure 53: Signals before (solid) and after (dashed) Amplitude Adjustment with a positive value

### Phase

The differential cosine and sine signals before and after phase adjustment are shown in Figure 54.

In ideal systems with no phase error, PH\_COS = 90°e and PH\_SIN = 0°e, the phase error can be described as

$$PH\_ERR = 90^\circ e - (PH\_COS - PH\_SIN) = 0^\circ e$$

In this example, the phase difference between cosine and sine is larger than 90°e:

$$PH\_COS - PH\_SIN > 90^\circ e$$

$$PH\_ERR = 90^\circ e - (PH\_COS - PH\_SIN) < 0^\circ e$$

By changing SC\_PHASE, both the phase of cosine and sine are changed:

$$PH\_ERR = 90^\circ e - (PH\_COS - PH\_SIN) + CORPH$$

To adjust the phase between cosine and sine, SC\_PHASE has to be increased until PH\_ERR = 0°e:

$$CORPH = PH\_COS - PH\_SIN - 90^\circ e$$

SC_PHASE(1:0) Addr. 0x1, 0x06; bit 7:6 default: 0x000	
SC_PHASE(9:2) Addr. 0x1, 0x07; bit 7:0	
Code	CORPH
Signed (2K)	$\frac{SC\_PHASE}{511} \cdot 11.4^\circ e$
0x200	-11.400°e (equal to 0x201)
0x201	-11.400°e
0x202	-11.378°e
...	...
0x3FF	-0.022°e
0x000	0.000°e
0x001	0.022°e
...	...
0x1FE	11.378°e
0x1FF	11.400°e

Table 119: Adjustment phase between cosine and sine (static)

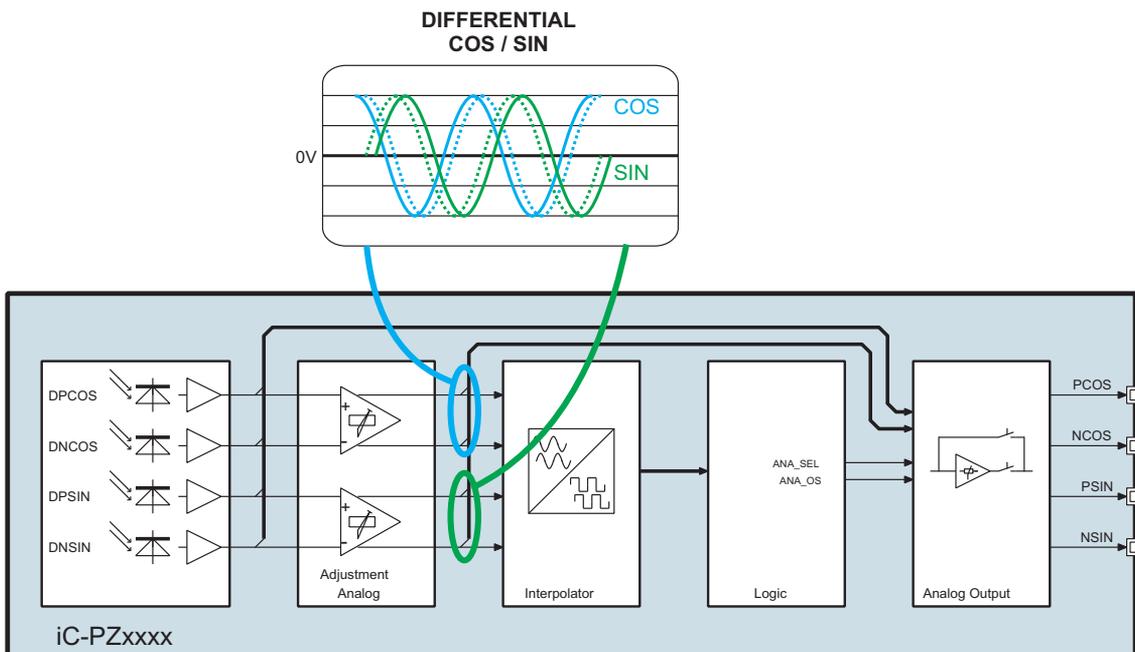


Figure 54: Signals before (solid) and after (dashed) Phase Adjustment with a positive value

### Adjustment Analog, Static

[COS\\_OFF](#), [SIN\\_OFF](#), [SC\\_GAIN](#) and [SC\\_PHASE](#) can be calculated automatically by executing command [AUTO\\_ADJ\\_ANA](#). There are several settings available to individually adapt the autocalibration procedure, however using the default values for [AC\\_SEL1](#), [AC\\_SEL2](#), [AC\\_COUNT](#) and [AC\\_ETO](#) is highly recommended.

When executing command [AUTO\\_ADJ\\_ANA](#), the system must be rotating/moving. If constant speed is applied, the minimum speed can be calculated with the formulas in Table 124. There is also an example given that contains typical values for a rotary system with a code disc of 26 mm diameter and a linear system, both with the default values for [AC\\_SEL1](#), [AC\\_SEL2](#), [AC\\_COUNT](#) and [AC\\_ETO](#).

The time required for the analog autocalibration with constant speed can be calculated with the formulas in Table 125. There is also an example given that contains typical values for systems described above and rotating/moving at 6 times the minimum speed. The time required can be reduced by increasing the speed. However, very high speeds may reduce the adjustment quality. For very slow systems, the minimum speed can be reduced by factor 10 activating [AC\\_ETO](#).

**Note:** In FlexCode®-systems, the minimum speed and the time required for the analog autocalibration change as described in FLEXCODE® on page 69. For short linear applications, the calibration process can be performed by moving the sensor back and forth.



For frequencies  $f(\sin)$  above approx. 50 kHz, the analog adjustment is internally disabled.

rotary	$n_{\min} = 2^{\text{AC\_COUNT} - \text{SYS\_eff}} \cdot 1.25 \frac{1}{\text{s}}$	$n_{\min, \text{typ}, \varnothing 26} = 1.25 \frac{1}{\text{s}} = 75 \text{ RPM}$
linear	$v_{\min} = 2^{\text{AC\_COUNT}} \cdot 2.5 \cdot 10^{-4} \frac{\text{m}}{\text{s}}$	$v_{\min, \text{typ}} = 6.4 \frac{\text{cm}}{\text{s}}$

Table 124: Minimum speed required for analog autocalibration with default settings

rotary	$t_{\text{rot}} \approx 2^{\text{AC\_COUNT} - \text{SYS\_eff}} \cdot \frac{\text{AC\_SEL1} - \text{AC\_SEL2}}{n}$	$t_{\text{rot, typ}, \varnothing 26, 450 \text{ RPM}} \approx 2.0 \text{ s}$
linear	$t_{\text{lin}} \approx 2^{\text{AC\_COUNT}} \cdot \frac{\text{AC\_SEL1} - \text{AC\_SEL2}}{4883 \frac{1}{\text{m}} \cdot v}$	$t_{\text{lin, typ}, 38.4 \text{ cm/s}} \approx 2.0 \text{ s}$

Table 125: Time required for analog autocalibration with default settings

AC_SEL1(3:0)		Addr. 0x5C; bit 3:0	default: 0xF
Code	Description		
0xF	Best but longest autocalibration		
..	..		
0x0	Worst but fastest autocalibration		

Table 120: Autocalibration Select Start

AC_SEL2(3:0)		Addr. 0x5C; bit 7:4	default: 0x0
Code	Description		
0x0	Best but longest autocalibration		
..	..		
0xE	Worst but fastest autocalibration		
0xF	No autocalibration		

Table 121: Autocalibration Select End

AC_COUNT(3:0)		Addr. 0x5D; bit 3:0	default: 0x8
Bit	Value		
3:0	Samples per autocalibration step, logarithmic		

Table 122: Autocalibration Count

AC_ETO		Addr. 0x5D; bit 7	default: 0
Code	Description		
0	Regular timeout for analog/digital autocalibration		
1	Extended timeout for analog/digital autocalibration: minimum speed required is reduced by factor 10		

Table 123: Autocalibration Extended Timeout

**Note:** [AC\\_SEL1](#), [AC\\_SEL2](#), [AC\\_COUNT](#) and [AC\\_ETO](#) are shared among all autocalibrations.

### Adjustment Analog, Dynamic

Besides the static analog adjustment, iC-PZ is able to correct signal drift during operation, e. g. caused by temperature. The sensitivity against dynamically occurring effects is selected individually for each parameter via [SC\\_OFF\\_SEL](#), [SC\\_GAIN\\_SEL](#) and [SC\\_PHASE\\_SEL](#). Those values are not affecting the static values set for [COS\\_OFF](#), [SIN\\_OFF](#), [SC\\_GAIN](#) and [SC\\_PHASE](#).

<b>SC_OFF_SEL</b> Addr. 0x2, 0x00; bit 3:0    default: 0x0 (3:0)	
Code	Function
0x0	Disabled
0x1	Lowest Dynamic
...	... (logarithmic)
0x7	Moderate Dynamic
...	... (logarithmic)
0xF	Highest Dynamic

Table 126: Dynamic analog offset adjustment select

<b>SC_GAIN_SEL</b> Addr. 0x2, 0x00; bit 7:4    default: 0x0 (3:0)	
Code	Function
0x0	Disabled
0x1	Lowest Dynamic
...	... (logarithmic)
0x7	Moderate Dynamic
...	... (logarithmic)
0xF	Highest Dynamic

Table 127: Dynamic analog amplitude-ratio adjustment select

<b>SC_PHASE_SEL</b> Addr. 0x2, 0x01; bit 3:0    default: 0x0 (3:0)	
Code	Function
0x0	Disabled
0x1	Lowest Dynamic
...	... (logarithmic)
0x7	Moderate Dynamic
...	... (logarithmic)
0xF	Highest Dynamic

Table 128: Dynamic analog phase adjustment select

### Adjustment Analog, Applied Correction

[COS\\_OFFS](#), [SIN\\_OFFS](#), [SC\\_GAINS](#) and [SC\\_PHASES](#) are giving information about the currently applied correction values of both static and dynamic analog adjustment.

**Note:** [COS\\_OFFS](#), [SIN\\_OFFS](#), [SC\\_GAINS](#) and [SC\\_PHASES](#) are read-only and can not be stored in the EEPROM. When the dynamic adjustment is disabled, its correction value is set to 0 and only the static adjustment is effective.

<b>COS_OFFS(1:0)</b> Addr. 0x1, 0x20; bit 7:6    default: 0x000	
<b>COS_OFFS(9:2)</b> Addr. 0x1, 0x21; bit 7:0	
<b>SIN_OFFS(1:0)</b> Addr. 0x1, 0x22; bit 7:6    default: 0x000	
<b>SIN_OFFS(9:2)</b> Addr. 0x1, 0x23; bit 7:0	
Code	Value
	Identical to <a href="#">COS_OFF</a> and <a href="#">SIN_OFF</a>

Table 129: Adjustment offset (static+dynamic)

<b>SC_GAINS(1:0)</b> Addr. 0x1, 0x24; bit 7:6    default: 0x000	
<b>SC_GAINS(9:2)</b> Addr. 0x1, 0x25; bit 7:0	
Code	Value
	Identical to <a href="#">SC_GAIN</a>

Table 130: Adjustment cosine-to-sine amplitude ratio (static+dynamic)

<b>SC_PHASES</b> Addr. 0x1, 0x26; bit 7:6    default: 0x000 (1:0)	
<b>SC_PHASES</b> Addr. 0x1, 0x27; bit 7:0 (9:2)	
Code	Value
	Identical to <a href="#">SC_PHASE</a>

Table 131: Adjustment phase between cosine and sine (static+dynamic)

[COS\\_OFFS](#), [SIN\\_OFFS](#), [SC\\_GAINS](#) and [SC\\_PHASES](#) are latched by reading their lowest address 0x20, 0x22, 0x24 and 0x26 so that all corresponding register values are related to the same request. Reading those parameters from lowest to highest register address is mandatory.



### ADJUSTMENT DIGITAL

#### Adjustment Digital, Static

The initial digital autocalibration is done by executing command `AUTO_ADJ_DIG`. `AI_PHASE` and `AI_SCALE` are calculated automatically, so that even significant misalignments between incremental and absolute track can be compensated.

**Note:** Changing `AI_PHASE` and/or `AI_SCALE` affects position calculation so that current offsets `ST_OFF`, `MT_OFF`, `ABZ_OFF` and `UVW_OFF` as well as eccentricity compensation parameters may become invalid.

Command `AUTO_READJ_DIG` can be used to compensate minor changes in alignment for a system that has been calibrated before. The current singleturn position will not be affected and all offsets stay valid. There are several settings available to individually adapt the autocalibration procedure, however using the default values for `AC_SEL1`, `AC_SEL2`, `AC_COUNT` and `AC_ETO` is highly recommended.

**Note:** `AC_SEL1`, `AC_SEL2`, `AC_COUNT` and `AC_ETO` are shared among all autocalibrations.

When executing command `AUTO_ADJ_DIG` or `AUTO_READJ_DIG`, the system must be rotating/moving. If constant speed is applied, the minimum speed can be calculated with the formulas in Table 134. There is also an example given that contains typical values for a rotary system with a code disc of 26 mm diameter and a linear system, both with the default values for `AC_SEL1`, `AC_SEL2`, `AC_COUNT` and `AC_ETO`.

The time required for the digital autocalibration with constant speed can be calculated with the formulas in Table 135. There is also an example given that contains typical values for systems described above and rotating/moving at 5 times the minimum speed. The time required can be reduced by increasing the speed. However, very high speeds may reduce the adjustment quality. For very slow systems, the minimum speed required can be reduced by factor 10 activating `AC_ETO`.

**Note:** In FlexCode®-systems, the minimum speed and the time required for digital autocalibration change as described in FLEXCODE® on page 69. For short linear applications, the calibration process can be performed by moving the sensor back and forth.

<b>AI_PHASE(1:0)</b>	Addr. 0x1, 0x08; bit 7:6	default: 0x000
<b>AI_PHASE(9:2)</b>	Addr. 0x1, 0x09; bit 7:0	
Code	Phase adjustment value	
Signed (2K)	$\frac{AI\_PHASE}{512} \cdot 180^\circ e$	

Table 132: Adjustment phase error (static)

<b>AI_SCALE(0)</b>	Addr. 0x1, 0x0A; bit 7	default: 0x000
<b>AI_SCALE(8:1)</b>	Addr. 0x1, 0x0B; bit 7:0	
Code	Scale adjustment factor	
Signed (2K)	$1 + \frac{AI\_SCALE}{1820}$	

Table 133: Adjustment scale error (static)

rotary	$n_{min} = 2^{AC\_COUNT - SYS\_eff} \cdot 1.5 \frac{1}{s}$	$n_{min,typ,\varnothing 26} = 1.5 \frac{1}{s} = 90 \text{ RPM}$
linear	$v_{min} = 2^{AC\_COUNT} \cdot 3 \cdot 10^{-4} \frac{m}{s}$	$v_{min,typ} = 7.68 \frac{cm}{s}$

Table 134: Minimum speed required for digital autocalibration with default settings

rotary	$t_{rot} \approx 2^{AC\_COUNT - SYS\_eff} \cdot \frac{AC\_SEL1 - AC\_SEL2}{\frac{3}{4} \cdot n}$	$t_{rot,typ,\varnothing 26,450 \text{ RPM}} \approx 2.67 \text{ s}$
linear	$t_{lin} \approx 2^{AC\_COUNT} \cdot \frac{AC\_SEL1 - AC\_SEL2}{3662 \frac{1}{m} \cdot v}$	$t_{lin,typ,38.4 \text{ cm/s}} \approx 2.73 \text{ s}$

Table 135: Time required for digital autocalibration with default settings

### Adjustment Digital, Dynamic

Besides the static digital adjustment, iC-PZ is able to compensate misalignments during operation. The sensitivity against dynamically occurring effects is selected individually for each parameter via **AI\_P\_SEL** and **AI\_S\_SEL**. Those values are not affecting the static values set for **AI\_PHASE** and **AI\_SCALE**.

<b>AI_P_SEL(3:0)</b> Addr. 0x2, 0x03; bit 3:0 default: 0x0	
Code	Phase adjustment value
0x0	Disabled
0x1	Lowest Dynamic
...	... (logarithmic)
0x7	Moderate Dynamic
...	... (logarithmic)
0xF	Highest Dynamic

Table 136: Dynamic adjustment phase select

<b>AI_S_SEL(3:0)</b> Addr. 0x2, 0x03; bit 7:4 default: 0x0	
Code	Scale adjustment value
0x0	Disabled
0x1	Lowest Dynamic
...	... (logarithmic)
0x7	Moderate Dynamic
...	... (logarithmic)
0xF	Highest Dynamic

Table 137: Dynamic adjustment scale select

### Adjustment Digital, Applied Correction

**AI\_PHASES** and **AI\_SCALES** are giving information about the currently applied correction values of both static and dynamic digital adjustment.

**Note:** **AI\_PHASES** and **AI\_SCALES** are read-only and can not be stored in the EEPROM. When the dynamic adjustment is disabled, its correction value is set to 0 and only the static adjustment is effective.

<b>AI_PHASES(1:0)</b> Addr. 0x1, 0x28; bit 7:6 read-only	
<b>AI_PHASES(9:2)</b> Addr. 0x1, 0x29; bit 7:0	
Code	Phase adjustment in °e (rounded)
	Identical to <b>AI_PHASE</b>

Table 138: Adjustment phase error (static+dynamic)

<b>AI_SCALES(0)</b> Addr. 0x1, 0x2A; bit 7 read-only	
<b>AI_SCALES(8:1)</b> Addr. 0x1, 0x2B; bit 7:0	
Code	Description
	Identical to <b>AI_SCALE</b>

Table 139: Adjustment scale error (static+dynamic)

**AI\_PHASES** and **AI\_SCALES** are latched by reading their lowest address 0x28 and 0x2A so that all corresponding register values are related to the same request. Reading those parameters from lowest to highest register address is mandatory.



### ADJUSTMENT ECCENTRICITY

Code discs mounted eccentric are providing a long-wave sinusoidal position error over a full revolution. Most of this error can be compensated by the eccentricity adjustment of the iC-PZ. The correction values **ECC\_AMP** and **ECC\_PHASE** are calculated on-chip by executing command **AUTO\_ADJ\_ECC**. The autocalibration is performed over  $2^{AC\_COUNT}$  revolutions. Using the default value for **AC\_COUNT** is highly recommended. Before starting the autocalibration, the eccentricity correction has to be switched off via **ECC\_EN**.

**Note:** **AC\_COUNT** is shared among all autocalibrations.

The minimum speed for the eccentricity autocalibration given in Table 143 is independent of the code disc diameter.

The time required for the eccentricity autocalibration procedure with constant speed can be calculated with the formula in Table 144. There is also an example given that contains a typical value for a system rotating/moving at 30 times the minimum speed. The time required can be reduced by increasing the speed.



Rotating at constant speed and at steady state is crucial when eccentricity autocalibration is performed.

<b>ECC_AMP(7:0)</b>		Addr. 0x2, 0x04; bit 7:0	default:
<b>ECC_AMP(15:8)</b>		Addr. 0x2, 0x05; bit 7:0	0x0000000
<b>ECC_AMP(23:16)</b>		Addr. 0x2, 0x06; bit 7:0	
<b>ECC_AMP(31:24)</b>		Addr. 0x2, 0x07; bit 7:0	
Code	Eccentricity amplitude value		
Unsigned	$1.407 \cdot 10^{-9} \cdot r_{opt\_AB} \cdot \mathbf{ECC\_AMP}$ , with $r_{opt\_AB}(\varnothing 26mm) = 10700 \mu m$ $r_{opt\_AB}(\varnothing 09mm) = 3600 \mu m$		

Table 140: Eccentricity amplitude error

<b>ECC_PHASE(5:0)</b>		Addr. 0x2, 0x08; bit 7:2	default: 0x0000
<b>ECC_PHASE(13:6)</b>		Addr. 0x2, 0x09; bit 7:0	
Code	Eccentricity phase value		
Signed (2K)	$\frac{360^\circ m}{2^{14}} \cdot \mathbf{ECC\_PHASE}$		

Table 141: Eccentricity phase error

<b>ECC_EN</b>		Addr. 0x2, 0x0A; bit 0	default: 0
Code	Value		
0	Eccentricity correction off		
1	Eccentricity correction on		

Table 142: Enable eccentricity correction

rotary	$n_{min} = 0.5 \frac{1}{s}$	$n_{min,all} = 0.5 \frac{1}{s} = 30 \text{ RPM}$
--------	-----------------------------	--

Table 143: Minimum speed for eccentricity autocalibration

rotary	$t_{rot} \approx 2^{AC\_COUNT} \cdot \frac{1}{n}$	$t_{rot,typ,all,900RPM} \approx 17 \text{ s}$
--------	---	---

Table 144: Time required for eccentricity autocalibration with default settings

# iC-PZ Series

HIGH-RESOLUTION REFLECTIVE ABSOLUTE ENCODER



Rev F1, Page 78/85

## SAFETY ADVICE

Depending on the mode of operation, these devices emit highly concentrated visible blue light which can be hazardous to the human eye.

Products which incorporate these devices have to follow the safety precautions given in IEC 60825-1 and IEC 62471.

## HANDLING ADVICE

Because of the specific housing materials and geometries used, these LED devices are sensitive to rough handling or assembly and can thus be easily damaged

or may fail in regard to their electro-optical operation. Excessive mechanical stress or load on the LED surface or to the glass windows must be avoided.

### DEVICE OVERVIEW

#### Rotative

Device	CPR native	Resolution		Code Disc		Typ. I(VDDA) / mA			Max. RPM <sup>2</sup>
		bit (CPR+IPO)	arcsec	P/O Code	Type	LED off	1.5 mm <sup>1</sup>	2.0 mm <sup>1</sup>	
iC-PZ0974	74	< 21	1.08	PZ07S	M	18.4	21.6	23.3	190 000
iC-PZ2656	256	22 (8+14)	0.31	PZ03S	P	18.4	21.6	23.3	56 250
iC-PZ2656	256	22 (8+14)	0.31	PZ03S	M	18.4	21.6	23.3	56 250
iC-PZ2656	446	< 23	0.18	PZ08S	M	18.4	21.6	23.3	32 200
iC-PZ205	1024	24 (10+14)	0.08	PZ05S	M	18.4	24.0	27.4	14 000

#### Linear

Device	Max. Len. m	Resolution		Code Disc		Typ. I(VDDA) / mA			Max. Speed m/s
		bit	nm	P/O Code	Type	LED off	1.5 mm <sup>1</sup>	2.0 mm <sup>1</sup>	
iC-PZ205	6.7	29 (15+14)	12.5	PZ01L		18.4	25.1	29.4	50

Type M = Metal  
Type P = Polycarbonate  
Type F = Film  
Type [ ] = Glass

Device availability on request.

Table 145: Device overview

<sup>1</sup> Air gap (iC to code disc / linear scale), I(VDDA) without analog output buffer

<sup>2</sup> In FlexCode®-systems, Max. RPM changes as described in chapter FLEXCODE®.

### DESIGN REVIEW: Notes On Chip Functions

iC-PZxxxx Z		
No.	Function, Parameter/Code	Description and Application Hints
1	ABZ Generator	Z signal level changes during configuration read from external EEPROM if ABZ_CFG(1) = 1.
2	UVW Generator	UVW signal levels not determined during startup. Valid UVW signal output 50 µs after DIAG(24) = 1 (RDY).
3	System Definition	For SYS_OVR ≠ 0x0, system startup may finish incompletely. In this case, a system reboot is required, e. g. via pin NRES, command REBOOT, or power cycle.
4	I2C	I2C-interface suits single-master applications only.
5	Feature FLEXCODE® Feature DISC/SCALE CORRECTION Feature ABZ/UVW ready Parameter LED_CONST Command CHIP_REV Default value IPO_FILT1	Feature/parameter/command available with chip release Y.

Table 146: Notes on chip functions regarding iC-PZxxxx chip revision Z

iC-PZxxxx Y		
No.	Function, Parameter/Code	Description and Application Hints
3	System Definition	For SYS_OVR ≠ 0x0, system startup may finish incompletely. In this case, a system reboot is required, e. g. via pin NRES, command REBOOT, or power cycle.
4	I2C	I2C-interface suits single-master applications only.
6	Adjustment Analog	Command AUTO_ADJ_ANA requires SC_OFF_SEL, SC_GAIN_SEL or SC_PHASE_SEL ≠ 0.

Table 147: Notes on chip functions regarding iC-PZxxxx chip revision Y

iC-PZxxxx X		
No.	Function, Parameter/Code	Description and Application Hints
		None at time of release.

Table 148: Notes on chip functions regarding iC-PZxxxx chip revision X

### REVISION HISTORY

Rel.	Rel. Date <sup>1</sup>	Chapter	Modification	Page
A1	2019-04-18	...	Initial release	all

Rel.	Rel. Date <sup>1</sup>	Chapter	Modification	Page
B1	2019-11-08	FEATURES	Features updated	1
		BLOCK DIAGRAM	Updated	1
		DESCRIPTION	Description updated	2
		PACKAGING INFORMATION	Added reference for C1V8	5
		ABSOLUTE MAXIMUM RATINGS	G006/G007: Added parameters	7
		ELECTRICAL CHARACTERISTICS	003: Changed max. value 217: Added parameter 222/224: Added parameter 226/227: Changed Conditions and max. value 229/230: Changed parameter description 231: Added parameter 3: Renamed section 505: Corrected definition C02: Replaced SYS with SYS_eff T03: Added min./max. values W01: Renamed parameter, updated Conditions W02: Removed parameter tstart()fast Z01-Z02: Updated values Z05-Z07: Updated values	8-11
		OPERATING REQUIREMENTS: Supply Voltages	Chapter added	12
		OPERATING REQUIREMENTS: SSI Slave	I206: Changed max. value	14
		OPERATING REQUIREMENTS: SSI Slave	I202-I204: Changed min. value	14
		OPERATING REQUIREMENTS: SPI Slave	I311: Changed max. value	15
		EEPROM SELECTION	Chapter updated	18
		INTERFACE PORTS	Chapter renamed	19
		PCB EXAMPLES	Chapter added for PCB Examples	20
		BANK ORGANIZATION, EEPROM & RPL	Renamed section "I2C Devices" to "Banks for I2C Devices"	21
		REGISTER MAP	Updated	23-28
		GENERAL	FILT1 setting for normal operation changed to 0xEA	29
		LED POWER CONTROL	Parameter LED_CONST added	31
		TEMPERATURE SENSOR	Chapter renamed	35
		ABSOLUTE DATA INTERFACE (ADI)	Chapter renamed	36
		ABZ GENERATOR	Chapter updated	41-45
		UVW GENERATOR	Chapter updated	46-48
		POSITION OFFSET	Figure added	49
		COMMANDS	Added command "CHIP_REV"	61
		IDs, SERIAL NUMBERS & SYSTEM DEFINITION	Definition of SYS_eff added Note added	65
		FLEXCODE®	Chapter added for FlexCode®	67
		DISC/SCALE CORRECTION	Chapter added for Disc/Scale Correction	68
		DIAGNOSIS	Diagnosis bits 6, 7 added	68
		ADJUSTMENT ANALOG	Notes added Updated formulas for autocalibration speed and time	71-76
		ADJUSTMENT DIGITAL	Note added Updated formulas for autocalibration speed and time	77
		DEVICE OVERVIEW	PZ0974 added	82

Rel.	Rel. Date <sup>1</sup>	Chapter	Modification	Page
C1	2020-07-08	all	Updated descriptions	all
		BLOCK DIAGRAM	Renamed function blocks: 'Random Evaluation' → 'Singleturn Position Evaluation' 'Disc/Scale Correction' → 'System Configuration' 'Position Offset' → 'Position Settings'	1

	DESCRIPTION	Added 'General notice on application-specific programming'	2
	PACKAGING INFORMATION	Added Sensor Layout and AOI Criteria	5
	ELECTRICAL CHARACTERISTICS	215: Moved typ. value to min. value P03: Changed max. value W01: Changed typ. value Renamed 'Random' → 'PRC'	9-12
	OPERATING REQUIREMENTS	Reworked Figure 1: Supply voltages at startup	13
	EEPROM SELECTION	Added requirement: 11-bit addressing scheme	19
	STARTUP	Removed bullet point about uProcessor watchdog	19
	CIRCUIT DESIGN PROPOSALS	Renamed chapter (former: PCB EXAMPLES)	21
	MEMORY ORGANIZATION	Renamed chapter (former: BANK ORGANIZATION, EEPROM & RPL)	22
	REGISTER MAP	Removed IPO_MODE, IPO_RESMIN, IPO_RESMAX, IPO_HYS and RAN_VAL Added AC_ETO Renamed sector 'Random Evaluation' → 'Singleturn Position Evaluation'	23-28
	INTERFACE PORTS CONFIGURATION	Renamed chapter (former: GENERAL) Moved subsections about interpolator to new chapter 'INTERPOLATOR' Moved subsection about position data length to new chapter 'POSITION SETTINGS' Renamed subsection 'VDDIO voltage selection' → 'VDDIO Monitoring'	29
	INTERPOLATOR	Added chapter Removed interpolator legacy mode	31
	SINGLETURN POSITION EVALUATION	Renamed chapter (former: RANDOM EVALUATION) Removed RAN_VAL and timing diagram Changed description of RAN_TOL and RAN_FLD	31
	ANALOG OUTPUT	Corrected LED_CUR = 0 to LED_CONST = 0 in note box	32
	POSITION SETTINGS	Renamed chapter (former: POSITION OFFSET) Included ST_PDL and MT_PDL of former chapter GENERAL	41
	BISS SLAVE	Removed note on SSI_GRAY affecting BiSS single cycle data Removed Figure: BiSS protocol showing Control Communication Removed subsection BiSS Commands	43-44
	SPI SLAVE	Reworked SPI Operation Codes: 0x81 Read Registers (former: Register Read (Continuous)) 0xCF Write Registers (former: Register Write (Continuous)) 0xA6 Read Position (former: Position Read) 0xD9 Write Command (former: Write Command) 0x9C Read Status (former: Read Status) 0x97 Request Data From I2C Slave (former: Register Read (Single)) 0xD2 Transmit Data To I2C Slave (former: Register Write (Single)) 0xAD Get I2C Transmission Info (former: Register Status/Data) 0xB0 Activate Slave In Chain (former: Activate)	46-50
	COMMANDS	Added commands MTST_PRESET_STORE, MT_PRESET_STORE, ABZ_PRESET_STORE, UVW_PRESET_STORE	51
	ABSOLUTE DATA INTERFACE (ADI)	Added note on ADI_CFG bit 0	55
	I2C MASTER	Added Figure 47: I2C protocol	59
	DIAGNOSIS	Renamed DIAG bit 11 'Random synchronization failed' → 'PRC synchronization failed'	63
	SYSTEM CONFIGURATION	Added chapter (former: IDs, SERIAL NUMBERS & SYSTEM DEFINITION and DISC/SCALE CORRECTION) Changed default values of DEV_ID and MFG_ID Changed note on SYS_OVR Changed SYS_OVR overriding SYS_eff only, not affecting SYS anymore Changed naming order of BISS_PROFILE_ID, SERIAL, DEV_ID and MFG_ID	65-66
	ADJUSTMENT ANALOG	Added AC_ETO Removed analog adjustment requiring SC_OFF_SEL, SC_GAIN_SEL or SC_PHASE_SEL to be non-zero Corrected minimum speed to 1.25 1/s in Table 131 Changed recommended values to default	68-72
	ADJUSTMENT DIGITAL	Added note on AC_SEL1, AC_SEL2, AC_COUNT and AC_ETO Changed recommended values to default	73-74
	ADJUSTMENT ECCENTRICITY	Corrected calculation of ECC_AMP Added note on AC_COUNT Changed recommended values to default Changed RPM to 900 in Table 150	75
	DEVICE OVERVIEW	Updated	77

Rel.	Rel. Date <sup>1</sup>	Chapter	Modification	Page
D1	2020-10-23	ELECTRICAL CHARACTERISTICS	004: Changed typ. value Added filter settings to C01 and C02 Updated Z03 to more than 2 slaves in SPI daisy chain	9-12
		BiSS Slave	Changed I105 to include Start Bit Delay	14
		SPI Slave	Added I312 and I313	16

		EEPROM SELECTION	Updated description	19
		INTERFACE PORTS CONFIGURATION	Corrected default value of VDDIOSEL	29
		BISS SLAVE	Updated description regarding total data length	43
		SPI SLAVE	Updated description regarding total data length Reworked description for SPI opcode 0xB0	47, 50
		I2C MASTER	Changed available I2C_DEV_ID range	58
		FLEXCODE®	Removed specification modification for ABZ_PER Added description and example for output data value range	67
		ORDERING INFORMATION	Updated Evaluation Kits	82

Rel.	Rel. Date <sup>1</sup>	Chapter	Modification	Page
E1	2021-06-14	FEATURES	Added assembly tolerances to features	1
		PACKAGING INFORMATION	Updated link to customer information 'Optical Selection Criteria'	5
		PACKAGE DIMENSIONS	Updated drawing	7
		THERMAL DATA	T03: Updated link to customer information 'Handling and Soldering Conditions'	8
		ELECTRICAL CHARACTERISTICS	Added 'GNDA = GNDIO = 0V' to operating conditions Updated 'VDDIO = 2.25...VDDA' in operating conditions 007: Added note on linear applications C03: Added parameter fout() S05/S06: Corrected unit Renamed Item No. of Temperature Sensor to U Z01: Added 'Permissible' Z02: Added 'Permissible' Z03: Added 'Permissible'	9ff
		SPI Slave	I306: Corrected parameter description	16
		EEPROM SELECTION	Added I2C device address 0x50	19
		REGISTER MAP	Changed ADI_CFG(0) to reserved[0]	23
		ABZ GENERATOR	Updated example for ABZ_PER Added example for $v_{max}$ Corrected description of DIAG(6) to 'not(ABZ_RDY)' Corrected description of DIAG(24) to 'not(RDY)'	33, 37
		UVW GENERATOR	Corrected description of DIAG(7) to 'not(UVW_RDY)'	40
		POSITION SETTINGS	Added note for properly setting MT_PDL with external MT slaves	41
		BISS SLAVE	Updated BiSS slave performance table Added BiSS protocol commands Added warning when sending short BiSS frames < 6 MA pulses	43, 45
		SSI SLAVE	Added warning when sending short SSI frames < 6 MA pulses	47
		SPI SLAVE	Renamed opcode 0x9C 'Read Status' → 'Read Diagnosis' Renamed opcode 0xAD 'Get I2C Transmission Info' → 'Get Transaction Info' Reworked description of opcode 0xAD 'Get Transaction Info'	48ff
		ABSOLUTE DATA INTERFACE (ADI)	Removed note on ADI_CFG(0)	57
		TEMPERATURE SENSOR	Added instruction for reading TEMP	63
		ADJUSTMENT ANALOG	Corrected description of SC_GAIN 0x001 to '1.0005' Note on linear applications Added instruction for reading COS_OFFS, SIN_OFFS, SC_GAINS and SC_PHASES	71,73,74
		ADJUSTMENT DIGITAL	Added note on position validity when changing AI_PHASE and/or AI_SCALE Note on linear applications Added instruction for reading AI_PHASES and AI_SCALES	75f

Rel.	Rel. Date <sup>1</sup>	Chapter	Modification	Page
F1	2023-04-20	DEVICE OVERVIEW	Corrected PZ07S disc M-Type	79
		THERMAL DATA	MSL3	8
		ABSOLUTE MAXIMUM RATINGS	ESD description	8
		ELECTRICAL CHARACTERISTICS	Renamed parameters in "Temperature Sensor" from Uxx to Txx	11
		CIRCUIT DESIGN PROPOSALS	EEPROM Advice	21
		ORDERING INFORMATION	Corrected Eval-Kit Code Disc PZ03PS	85

<sup>1</sup> Release Date format: YYYY-MM-DD

# iC-PZ Series

## HIGH-RESOLUTION REFLECTIVE ABSOLUTE ENCODER



Rev F1, Page 84/85

iC-Haus expressly reserves the right to change its products, specifications and related supplements (together the Documents). A Datasheet Update Notification (DUN) gives details as to any amendments and additions made to the relevant Documents on our internet website [www.ichaus.com/DUN](http://www.ichaus.com/DUN) and is automatically generated and shall be sent to registered users by email.

Copying – even as an excerpt – is only permitted with iC-Haus' approval in writing and precise reference to source.

The data and predicted functionality is intended solely for the purpose of product description and shall represent the usual quality and behaviour of the product. In case the Documents contain obvious mistakes e.g. in writing or calculation, iC-Haus reserves the right to correct the Documents and no liability arises insofar that the Documents were from a third party view obviously not reliable. There shall be no claims based on defects as to quality and behaviour in cases of insignificant deviations from the Documents or in case of only minor impairment of usability.

No representations or warranties, either expressed or implied, of merchantability, fitness for a particular purpose or of any other nature are made hereunder with respect to information/specification resp. Documents or the products to which information refers and no guarantee with respect to compliance to the intended use is given. In particular, this also applies to the stated possible applications or areas of applications of the product.

iC-Haus products are not designed for and must not be used in connection with any applications where the failure of such products would reasonably be expected to result in significant personal injury or death (*Safety-Critical Applications*) without iC-Haus' specific written consent. Safety-Critical Applications include, without limitation, life support devices and systems. iC-Haus products are not designed nor intended for use in military or aerospace applications or environments or in automotive applications unless specifically designated for such use by iC-Haus.

iC-Haus conveys no patent, copyright, mask work right or other trade mark right to this product. iC-Haus assumes no liability for any patent and/or other trade mark rights of a third party resulting from processing or handling of the product and/or any other use of the product.

Software and its documentation is provided by iC-Haus GmbH or contributors "AS IS" and is subject to the ZVEI General Conditions for the Supply of Products and Services with iC-Haus amendments and the ZVEI Software clause with iC-Haus amendments ([www.ichaus.com/EULA](http://www.ichaus.com/EULA)).

# iC-PZ Series

HIGH-RESOLUTION REFLECTIVE ABSOLUTE ENCODER



Rev F1, Page 85/85

## ORDERING INFORMATION

Type	Package	Options	Order Designation
iC-PZ2656	32-pin optoQFN, 5 mm x 5 mm, 0.9 mm thickness RoHS compliant		iC-PZ2656 oQFN32-5x5
iC-PZ0974	32-pin optoQFN, 5 mm x 5 mm, 0.9 mm thickness RoHS compliant		iC-PZ0974 oQFN32-5x5
iC-PZ205	32-pin optoQFN, 5 mm x 5 mm, 0.9 mm thickness RoHS compliant		iC-PZ205 oQFN32-5x5
Evaluation Kit	Kit with Reflective Encoder IC (approx. 61mm x 64 mm), Code Disc PZ03PS		iC-PZ2656 EVAL PZ1M
	Kit with Reflective Encoder IC (approx. 61mm x 64 mm), Linear Scale PZ01L		iC-PZ205 EVAL PZ1M LIN
	Kit with Reflective Encoder IC (approx. 61mm x 64 mm), Code Disc PZ05SM		iC-PZ205 EVAL PZ1M ROT
	Kit with Reflective Encoder IC (approx. 61mm x 64 mm), Code Disc PZ07SM		iC-PZ0974 EVAL PZ1M
Motherboard	Adapter PCB (approx. 80 mm x 110 mm)		iC-PZ EVAL PZ2D

Please send your purchase orders to our order handling team:

**Fax: +49 (0) 61 35 - 92 92 - 692**

**E-Mail: [dispo@ichaus.com](mailto:dispo@ichaus.com)**

For technical support, information about prices and terms of delivery please contact:

**iC-Haus GmbH**  
**Am Kuemmerling 18**  
**D-55294 Bodenheim**  
**GERMANY**

**Tel.: +49 (0) 61 35 - 92 92 - 0**  
**Fax: +49 (0) 61 35 - 92 92 - 192**  
**Web: <https://www.ichaus.com>**  
**E-Mail: [sales@ichaus.com](mailto:sales@ichaus.com)**

**Appointed local distributors: [https://www.ichaus.com/sales\\_partners](https://www.ichaus.com/sales_partners)**