APPLICATIONS

The AMIS−30624/NCV70624 is ideally suited for small positioning applications. Target markets include: automotive (headlamp alignment, HVAC, idle control, cruise control), industrial equipment (lighting, fluid control, labeling, process control, XYZ tables, robots) and building automation (HVAC, surveillance, satellite dish, renewable energy systems). Suitable applications typically have multiple axes or require mechatronic solutions with the driver chip mounted directly on the motor.

Table 1. ORDERING INFORMATION

Part No.	Peak Current	End Market/Version	Package*	Shipping†
AMIS30624C6244G	800 mA	Industrial High Voltage Version	SOIC−20 (Pb−Free)	Tube/Tray
AMIS30624C6244RG	800 mA		SOIC−20 (Pb−Free)	Tape & Reel
AMIS30624C6245G	800 mA		NQFP−32 (7 x 7 mm) (Pb−Free)	Tube/Tray
AMIS30624C6245RG	800 mA		NQFP−32 (7 x 7 mm) (Pb−Free)	Tape & Reel
NCV70624DW010G	800 mA	Automotive High Temperature Version	SOIC−20 (Pb−Free)	Tube/Tray
NCV70624DW010R2G	800 mA		SOIC−20 (Pb−Free)	Tape & Reel

*For additional information on our Pb−Free strategy and soldering details, please download the ON Semiconductor Soldering and Mounting Techniques Reference Manual, SOLDERRM/D.

†For information on tape and reel specifications, including part orientation and tape sizes, please refer to our Tape and Reel Packaging Specification Brochure, BRD8011/D.

QUICK REFERENCE DATA

Table 2. ABSOLUTE MAXIMUM RATINGS

Parameter		Min	Max	Unit
V_{BB}, V_{HW}, V_{SWI}	Supply voltage, hardwired address and SWI pins	−0.3	+40 (Note 2)	V
T_J	Junction temperature range (Note 3)	−50	+175	°C
T_{st}	Storage temperature	−55	+160	°C
V_{esd} (Note 4)	Human Body Model (HBM) Electrostatic discharge voltage on pins	−2	+2	kV
	Machine Model (MM) Electrostatic discharge voltage on pins	−200	+200	V

Stresses exceeding Maximum Ratings may damage the device. Maximum Ratings are stress ratings only. Functional operation above the Recommended Operating Conditions is not implied. Extended exposure to stresses above the Recommended Operating Conditions may affect device reliability.

2. For limited time: V_{BB} < 0.5 s, SWI and HW pins <1.0 s.
3. The circuit functionality is not guaranteed.
4. HBM according to AEC−Q100: EIA−JESD22−A114−B (100 pF via 1.5 kΩ) and MM according to AEC−Q100: EIA−JESD22−A115−A.

Table 3. OPERATING RANGES

Parameter		Min	Max	Unit
V_{BB}	Supply voltage	+6.5	+29	V
T_J	Operating temperature range	−40	+165	°C

Table of Contents

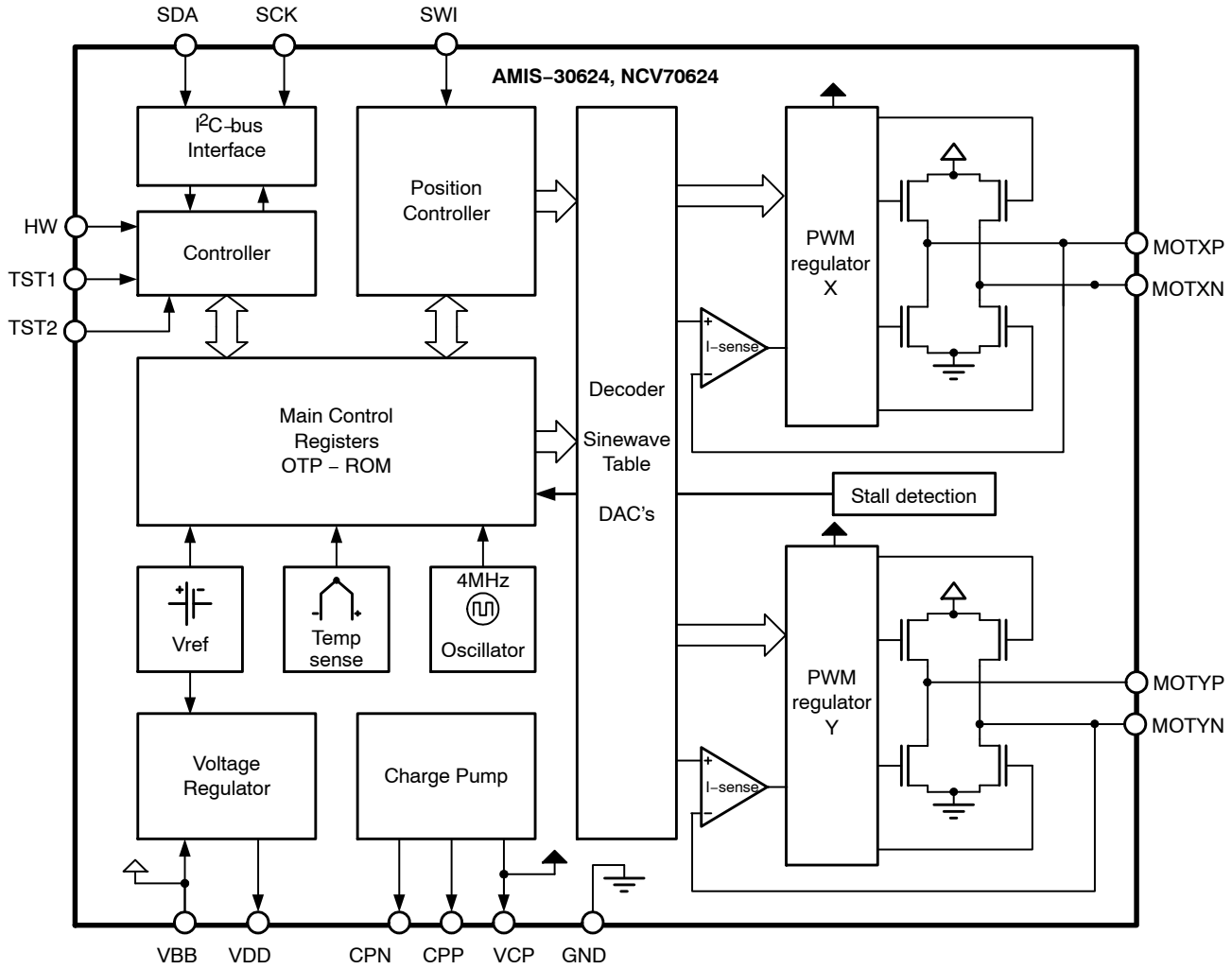


Figure 1. Block Diagram

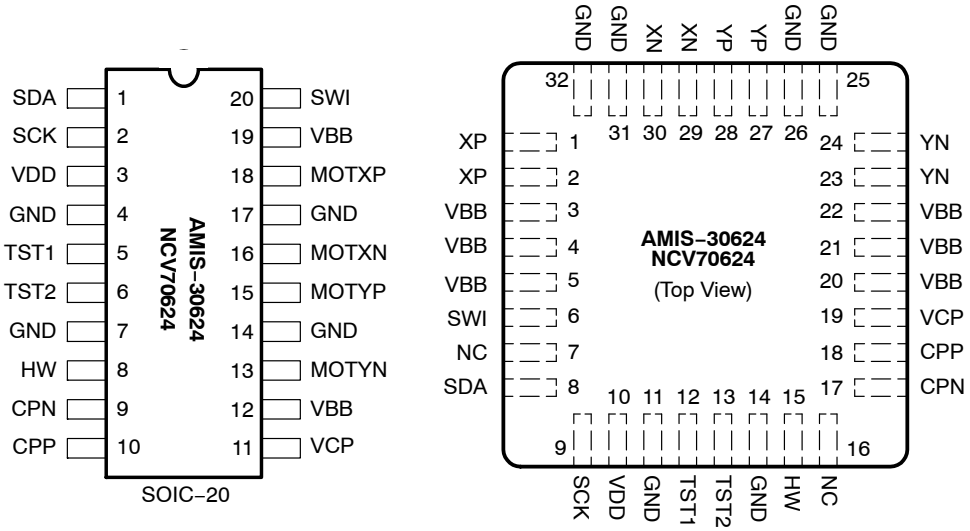


Figure 2. SOIC−20 and NQFP−32 Pin−out

Table 4. PIN DESCRIPTION

Pin Name	Pin Description	SOIC−20	NQFP−32
SDA	I^2C serial data line	1	8
SCK	I^2C serial clock line	2	9
V_{DD}	Internal supply (needs external decoupling capacitor)	3	10
GND	Ground, heat sink	4, 7, 14, 17	11, 14, 25, 26, 31, 32
TST1	Test pin (to be tied to ground in normal operation)	5	12
TST2	Test pin (to be left open in normal operation: internally pulled up)	6	13
HW	Hard wired address bit	8	15
CPN	Negative connection of pump−capacitor (charge pump)	9	17
CPP	Positive connection of pump−capacitor (charge pump)	10	18
VCP	Charge−pump filter−capacitor	11	19
V_{BB}	Battery voltage supply	12, 19	3, 4, 5, 20, 21, 22
MOTYN	Negative end of phase Y coil	13	23, 24
MOTYP	Positive end of phase Y coil	15	27, 28
MOTXN	Negative end of phase X coil	16	29, 30
MOTXP	Positive end of phase X coil	18	1, 2
SWI	Switch input	20	6
NC	Not connected (to be tied to ground)		7, 16

PACKAGE THERMAL RESISTANCE

The AMIS−30624/NCV70624 is available in SOIC−20 or optimized NQFP−32 packages. For cooling optimizations, the NQFP has an exposed thermal pad which has to be soldered to the PCB ground plane. The ground plane needs thermal vias to conduct the head to the bottom layer. Figures 3 and 4 give examples for good power distribution solutions.

For precise thermal cooling calculations the major thermal resistances of the devices are given. The thermal media to which the power of the devices has to be given are:

- Static environmental air (via the case)
- PCB board copper area (via the device pins and exposed pad)

The thermal resistances are presented in Table 5: DC Parameters.

The major thermal resistances of the device are the Rth from the junction to the ambient (Rthja) and the overall Rth from the junction to the leads (Rthjp).

The NQFP device is designed to provide superior thermal performance. Using an exposed die pad on the bottom surface of the package is mainly contributing to this performance. In order to take full advantage of the exposed pad, it is most important that the PCB has features to conduct heat away from the package. A thermal grounded pad with thermal vias can achieve this.

In the table below, one can find the values for the Rthja and Rthjp, simulated according to the JESD−51 norm:

Package	Rth Junction−to−Leads and Exposed Pad − Rthjp	Rth Junction−to−Leads Rthjp	Rth Junction−to−Ambient Rthja (1S0P)	Rth Junction−to−Ambient Rthja (2S2P)
SOIC−20		19	62	39
NQFP−32	0,95		60	30

The Rthja for 2S2P is simulated conform to JESD−51 as follows:

- A 4−layer printed circuit board with inner power planes and outer (top and bottom) signal layers is used
- Board thickness is 1.46 mm (FR4 PCB material)
- The 2 signal layers: 70 μm thick copper with an area of 5500 mm^2 copper and 20% conductivity
- The 2 power internal planes: 36 μm thick copper with an area of 5500 mm^2 copper and 90% conductivity

The Rthja for 1S0P is simulated conform to JESD−51 as follows:

- A 1−layer printed circuit board with only 1 layer
- Board thickness is 1.46 mm (FR4 PCB material)
- The layer has a thickness of 70 μm copper with an area of 5500 mm^2 copper and 20% conductivity

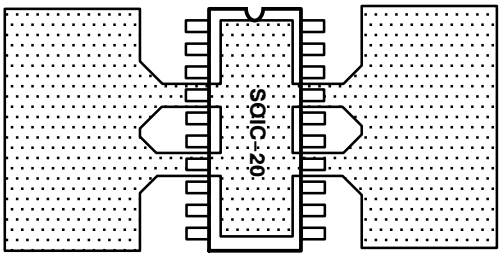


Figure 3. Example of SOIC−20 PCB Ground Plane Layout (preferred layout at top and bottom)

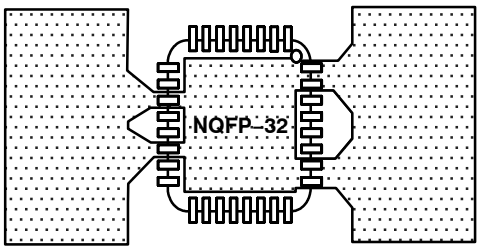


Figure 4. Example of NQFP−32 PCB Ground Plane Layout (preferred layout at top and bottom)

DC PARAMETERS

The DC parameters are guaranteed overtemperature and V_{BB} in the operating range, unless otherwise specified. Convention: currents flowing into the circuit are defined as positive.

Table 5. DC PARAMETERS

Symbol	Pin(s)	Parameter	Test Conditions	Min	Typ	Max	Unit
MOTORDRIVER							
$I_{MSmax,Peak}$	MOTXP MOTXN MOTYP MOTYN	Max current through motor coil in normal operation	V_{BB} = 14 V		800		mA
$I_{MSmax,RMS}$		Max rms current through coil in normal operation	V_{BB} = 14 V		570		mA
I_{MSabs}		Absolute error on coil current (Note 5)	V_{BB} = 14 V	−10		10	%
I_{MSrel}		Matching of X & Y coil currents	V_{BB} = 14 V	−7	0	7	%
$R_{DS(on)}$		On resistance for each motor pin at I_{MSmax} (Note 6)	V_{BB} = 12 V, T_j = 50°C		0.50	1	Ω
			V_{BB} = 8 V, T_j = 50°C		0.55	1	Ω
			V_{BB} = 12 V, T_j = 150°C		0.70	1	Ω
			V_{BB} = 8 V, T_j = 150°C		0.85	1	Ω
I_{MSL}		Pulldown current	HiZ mode, V_{BB} = 7.8 V		2		mA
I²C SERIAL INTERFACE							
V_{IL}	SDA SCK	Input level low (Note 11)		−0.5		0.3 * V_{DD}	V
V_{IH}		Input level high (Note 12)		0.7 * V_{DD}		V_{DD} + 0.5	V
V_{nL}		Noise margin at the LOW level for each connected device (including hysteresis)		0.1 * V_{DD}			V
V_{nH}		Noise margin at the HIGH level for each connected device (including hysteresis)		0.2 * V_{DD}			
THERMAL WARNING & SHUTDOWN							
T_{tw}		Thermal warning (Notes 7 and 8)		138	145	152	°C
T_{tsd}		Thermal shutdown (Note 9)			T_{tw} + 10		°C
T_{low}		Low temperature warning (Note 9)			T_{tw} − 155		°C
SUPPLY AND VOLTAGE REGULATOR							
V_{bbOTP}	V_{BB}	Supply voltage for OTP zapping (Note 10)		9.0		10.0	V
UV_1		Stop voltage high threshold		7.8	8.4	8.9	V
UV_2		Stop voltage low threshold		7.1	7.5	8.0	V
I_{bat}		Total current consumption	Unloaded outputs V_{BB} = 29 V		3.50	10.0	mA

5. Tested in production for 800 mA, 400 mA, 200 mA and 100 mA current settings for both X and Y coil.
6. Not measured in production. Guaranteed by design.
7. Parameter guaranteed by trimming relevant OTP's in production test at 143°C (±5°C) and V_{BB} = 14 V.
8. No more than 100 cumulated hours in life time above Tw.
9. Thermal shutdown and low temperature warning are derived from thermal warning. Guaranteed by design.
10. A buffer capacitor of minimum 100 μF is needed between V_{BB} and GND. Short connections to the power supply are recommended.
11. If input voltages < − 0.3 V, than a resistor between 22 Ω to 100 Ω needs to be put in series.
12. If the I²C−bus is operated in Fast Mode V_{IHmin} = 0.7 * V_{DD}.

Table 5. DC PARAMETERS

Symbol	Pin(s)	Parameter	Test Conditions	Min	Typ	Max	Unit
SUPPLY AND VOLTAGE REGULATOR							
V_{DD}	V_{DD}	Regulated internal supply (Note 13)	8 V < V_{BB} < 29 V	4.75	5	5.50	V
$V_{ddReset}$		Digital supply reset level @ power down (Note 14)				4.5	V
I_{ddLim}		Current limitation	Pin shorted to ground V_{BB} = 14 V			45	mA
SWITCH INPUT AND HARDWIRE ADDRESS INPUT							
$Rt_{\_OFF}$	SWI HW	Switch OPEN resistance (Note 15)		10			kΩ
$Rt_{\_ON}$		Switch ON resistance (Note 15)	Switch to GND or V_{BB}			2	kΩ
V_{bb_sw}		V_{BB} range for guaranteed operation of SWI and HW		6		29	V
I_{lim_sw}		Current limitation	Short to GND or V_{bat} V_{BB} = 29 V	20	30	45	mA
TEST PIN							
V_{ihigh}	TST	Input level high	V_{BB} = 14 V	0.7 * V_{dd}			V
V_{ilow}		Input level low	V_{BB} = 14 V			0.3 * V_{dd}	V
HW_{hyst}		Hysteresis	V_{BB} = 14 V	0.075 * V_{dd}			V
CHARGE PUMP							
V_{cp}	VCP	Output voltage	6 V ≤ V_{BB} ≤ 14 V		2 * V_{BB} − 2.5		V
			14 V ≤ V_{BB} ≤ 30 V	V_{BB} + 10		V_{BB} + 15	V
C_{buffer}		External buffer capacitor		220		470	nF
C_{pump}	CPP CPN	External pump capacitor		220		470	nF
MOTION QUALIFICATION MODE OUTPUT (Note 16)							
V_{OUT}	SWI	Output voltage swing	TestBemf I^2C command		0 − 4,85		V
R_{OUT}		Output impedance	Service mode I^2C command		2		kΩ
Av		Gain = V_{SWI} / V_{BEMF}	Service mode I^2C command		0.50		
PACKAGE THERMAL RESISTANCE VALUES							
Rth_{ja}	SO	Thermal resistance junction to ambient (2S2P)	Simulated conform JEDEC JESD51		39		K/W
Rth_{jp}	SO	Thermal resistance junction to leads			19		K/W
Rth_{ja}	NQ	Thermal resistance junction to ambient (2S2P)			30		K/W
Rth_{jp}	NQ	Thermal resistance junction to leads and exposed pad			0.95		K/W

13. Pin V_{DD} must not be used for any external supply
14. The RAM content will not be altered above this voltage.
15. External resistance value seen from pin SWI or HW, including 1 kΩ series resistor. For the switch OPEN, the maximum allowed leakage current is represented by a minimum resistance seen from the pin.
16. Not applicable for “Product Versions NCV70624DW010G, NCV70624DW010R2G”

AC PARAMETERS

The AC parameters are guaranteed for temperature and V_{BB} in the operating range unless otherwise specified.

Table 6. AC PARAMETERS

Symbol	Pin(s)	Parameter	Test Conditions	Min	Typ	Max	Unit
POWERUP							
T_{pu}		Power−up time	Guaranteed by design			10	ms
INTERNAL OSCILLATOR							
f_{osc}		Frequency of internal oscillator	V_{BB} = 14 V	3.6	4.0	4.4	MHz
I²C TRANSCEIVER (STANDARD MODE)							
f_{SCL}	SDA SCK	SCL clock frequency				100	kHz
$t_{HD,START}$		Hold time (repeated) START condition. After this period the first clock pulse is generated.		4.0			μs
t_{LOW}		LOW period of the SCK clock		4.7			μs
t_{HIGH}		HIGH period of the SCK clock		4.0			μs
$t_{SU,START}$		Set−up time for a repeated START condition		4.7			μs
$t_{HD,DATA}$		Data hold time for I²C bus devices		0 (Note 18)		3.45 (Note 19)	μs
$t_{SU,DATA}$		Data set−up time		250			ns
t_R		Rise time of SDA and SCK signals				1.0	μs
t_F		Fall time of SDA and SCK signals				0.3	μs
$t_{SU,STOP}$		Set−up time for STOP condition		4.0			μs
t_{BUF}		Bus free time between STOP and START condition		4.7			μs
I²C TRANSCEIVER (FAST MODE)							
f_{SCL}	SDA SCK	SCL clock frequency				360	kHz
$t_{HD,START}$		Hold time (repeated) START condition. After this period the first clock pulse is generated.		0.6			μs
t_{LOW}		LOW period of the SCK clock		1.3			μs
t_{HIGH}		HIGH period of the SCK clock		0.6			μs
$t_{SU,START}$		Set−up time for a repeated START condition		0.6			μs
$t_{HD,DATA}$		Data hold time for I²C bus devices		0 (Note 18)		0.9 (Note 19)	μs
$t_{SU,DATA}$		Data set−up time		100 (Note 20)			ns
t_R		Rise time of SDA and SCK signals		20 + 0.1 C_B		300	ns
t_F		Fall time of SDA and SCK signals		20 + 0.1 C_B		300	ns
$t_{SU,STOP}$		Set−up time for STOP condition		0.6			μs
t_{BUF}		Bus free time between STOP and START condition		1.3			μs

17. The maximum number of connected I²C devices is dependent on the number of available addresses and the maximum bus capacitance to still guarantee the rise and fall times of the bus signals.
18. An I²C device must internally provide a hold time of at least 300 ns for the SDA signal (referred to the V_{IHmin} of the SCL signal) to bridge the undefined region of the falling edge of SCL.
19. The maximum $t_{HD,DAT}$ has only to be met if the device does not stretch the LOW period (t_{LOW}) of the SCL signal.
20. A Fast−mode I²C−bus device can be used in a standard−mode I²C bus system, but the requirement $t_{SU,DATA} \geq 250$ ns must than be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line trmax + $t_{SU,DATA}$ = 1000 + 250 = 1250 ns (according to the standard−mode I²C−bus specification) before the SCL line is released.

Table 6. AC PARAMETERS

Symbol	Pin(s)	Parameter	Test Conditions	Min	Typ	Max	Unit
SWITCH INPUT AND HARDWIRE ADDRESS INPUT							
T_{sw}	SWI HW	Scan pulse period (Note 21)	V_{BB} = 14 V		1024		µs
T_{sw_on}		Scan pulse duration (Note 21)	V_{BB} = 14 V		128		µs
MOTORDRIVER							
F_{pwm}	MOTxx	PWM frequency (Note 21)	PWMfreq = 0 (Note 22)	20.6	22.8	25.0	kHz
			PWMfreq = 1 (Note 22)	41.2	45.6	50.0	kHz
F_{jit_depth}		PWM jitter modulation depth	PWMJen = 1 (Note 22)		10		%
T_{brise}		Turn−on transient time	Between 10% and 90%		140		ns
T_{bfall}		Turn−off transient time			130		ns
T_{stab}		Run current stabilization time (Note 21)		29	32	35	ms
CHARGE PUMP							
f_{CP}	CPN CPP	Charge pump frequency (Note 21)	V_{BB} = 14 V		250		kHz

21. Derived from the internal oscillator
22. See SetMotorParam and PWM Regulator

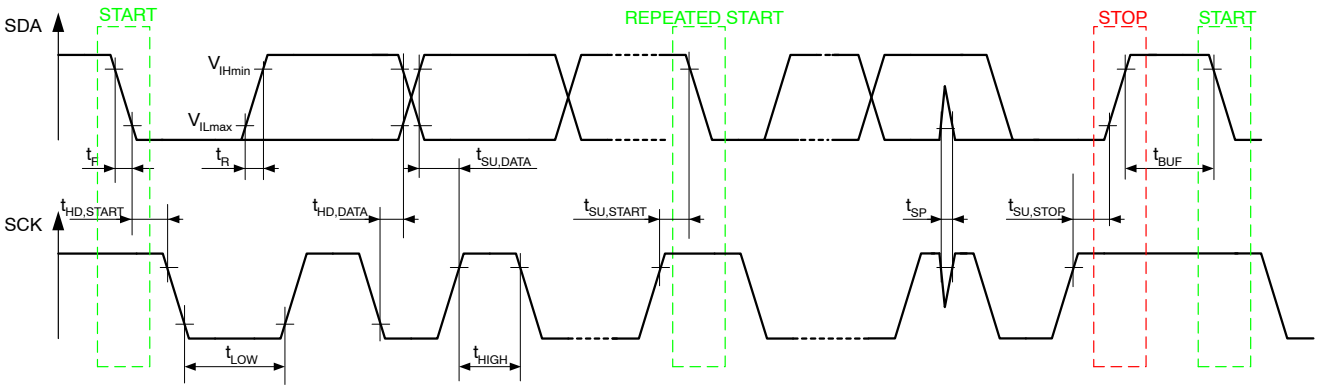


Figure 5. I²C Timing Diagrams

Typical Application

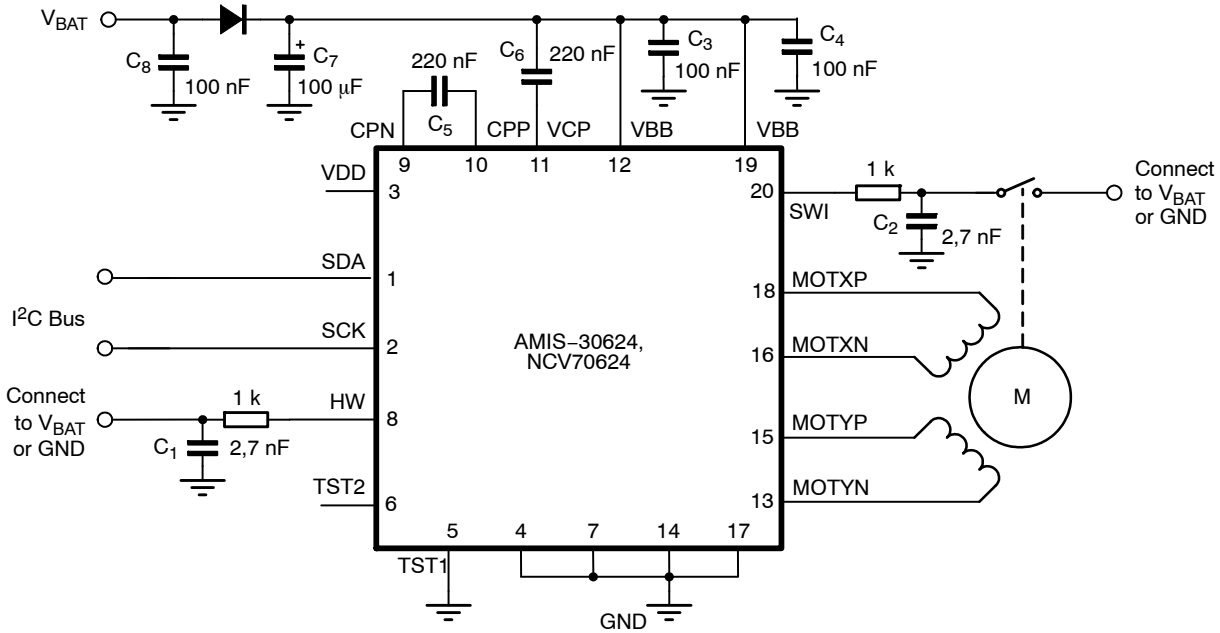


Figure 6. Typical Application Diagram for SO Device

NOTES: All resistors are ± 5%, 1/4 W
C_1, C_2 minimum value is 2.7 nF, maximum value is 10 nF
Depending on the application, the ESR value and working voltage of C_7 must be carefully chosen
C_3 and C_4 must be close to pins V_{BB} and GND
C_5 and C_6 must be as close as possible to pins CPN, CPP, VCP, and V_{BB} to reduce EMC radiation
C_9 must be a ceramic capacitor to assure low ESR

POSITIONING PARAMETERS

Stepping Modes

One of four possible stepping modes can be programmed:

- Half−stepping
- 1/4 micro−stepping
- 1/8 micro−stepping
- 1/16 micro−stepping

Maximum Velocity

For each stepping mode, the maximum velocity Vmax can be programmed to 16 possible values given in the table below.

The accuracy of Vmax is derived from the internal oscillator. Under special circumstances it is possible to change the Vmax parameter while a motion is ongoing. All 16 entries for the Vmax parameter are divided into four groups. When changing Vmax during a motion the application must take care that the new Vmax parameter stays within the same group.

Table 7. MAXIMUM VELOCITY SELECTION TABLE

Vmax Index				Stepping Mode			
Hex	Dec	Vmax (full step/s)	Group	Half−stepping (half−step/s)	1/4th Micro−stepping (micro−step/s)	1/8th Micro−stepping (micro−step/s)	1/16th Micro−stepping (micro−step/s)
0	0	99	A	197	395	790	1579
1	1	136	B	273	546	1091	2182
2	2	167		334	668	1335	2670
3	3	197		395	790	1579	3159
4	4	213		425	851	1701	3403
5	5	228		456	912	1823	3647
6	6	243		486	973	1945	3891
7	7	273	C	546	1091	2182	4364
8	8	303		607	1213	2426	4852
9	9	334		668	1335	2670	5341
A	10	364		729	1457	2914	5829
B	11	395		790	1579	3159	6317
C	12	456		912	1823	3647	7294
D	13	546	D	1091	2182	4364	8728
E	14	729		1457	2914	5829	11658
F	15	973		1945	3891	7782	15564

Minimum Velocity

Once the maximum velocity is chosen, 16 possible values can be programmed for the minimum velocity Vmin. The table below provides the obtainable values in full−step/s. The accuracy of Vmin is derived from the internal oscillator.

Table 8. OBTAINABLE VALUES IN FULL−STEP/s FOR THE MINIMUM VELOCITY

Vmin Index		Vmax	Vmax (Full−step/s)															
			A	B						C						D		
Hex	Dec	Factor	99	136	167	197	213	228	243	273	303	334	364	395	456	546	729	973
0	0	1	99	136	167	197	213	228	243	273	303	334	364	395	456	546	729	973
1	1	1/32	3	4	5	6	6	7	7	8	8	10	10	11	13	15	19	27
2	2	2/32	6	8	10	11	12	13	14	15	17	19	21	23	27	31	42	57
3	3	3/32	9	12	15	18	19	21	22	25	27	31	32	36	42	50	65	88
4	4	4/32	12	16	20	24	26	28	30	32	36	40	44	48	55	65	88	118
5	5	5/32	15	21	26	31	32	35	37	42	46	51	55	61	71	84	111	149
6	6	6/32	18	25	31	36	39	42	45	50	55	61	67	72	84	99	134	179
7	7	7/32	21	30	36	43	46	50	52	59	65	72	78	86	99	118	156	210
8	8	8/32	24	33	41	49	52	56	60	67	74	82	90	97	113	134	179	240
9	9	9/32	28	38	47	55	59	64	68	76	84	93	101	111	128	153	202	271
A	10	10/32	31	42	51	61	66	71	75	84	93	103	113	122	141	168	225	301
B	11	11/32	34	47	57	68	72	78	83	93	103	114	124	135	156	187	248	332
C	12	12/32	37	51	62	73	79	85	91	101	113	124	135	147	170	202	271	362
D	13	13/32	40	55	68	80	86	93	98	111	122	135	147	160	185	221	294	393
E	14	14/32	43	59	72	86	93	99	106	118	132	145	158	172	198	237	317	423
F	15	15/32	46	64	78	93	99	107	113	128	141	156	170	185	214	256	340	454

NOTES: The Vmax factor is an approximation.
In case of motion without acceleration (**AccShape = 1**) the length of the steps = 1/**Vmin.** In case of accelerated motion (**AccShape** = 0) the length of the first step is shorter than 1/**Vmin** depending of **Vmin, Vmax** and **Acc**.

Acceleration and Deceleration

Sixteen possible values can be programmed for Acc (acceleration and deceleration between Vmin and Vmax). The table below provides the obtainable values in full−step/s^2. One observes restrictions for some combinations of acceleration index and maximum speed (gray cells).

The accuracy of Acc is derived from the internal oscillator.

Table 9. ACCELERATION AND DECELERATION SELECTION TABLE

Vmax (FS/s) →		99	136	167	197	213	228	243	273	303	334	364	395	456	546	729	973
↓ Acc Index		Acceleration (Full−step/s^2)															
Hex	Dec																
0	0	49							106						473		
1	1	218													735		
2	2	1004															
3	3	3609															
4	4	6228															
5	5	8848															
6	6	11409															
7	7	13970															
8	8	16531															
9	9	14785	19092														
A	10		21886														
B	11		24447														
C	12		27008														
D	13		29570														
E	14		29570						34925								
F	15								40047								

The formula to compute the number of equivalent full−steps during acceleration phase is:

$$Nstep = \frac{Vmax^2 - Vmin^2}{2 \times Acc}$$

Positioning

The position programmed in command `SetPosition` is given as a number of (micro−)steps. According to the chosen stepping mode, the position words must be aligned as described in the table below. When using command `GotoSecurePosition`, data is automatically aligned.

Table 10. POSITION WORD ALIGNMENT

Stepping Mode	Position Word: `Pos[15:0]`																Shift
1/16th	**S**	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	**LSB**	No shift
1/8th	**S**	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	**LSB**	0	1−bit left ⇔ ×2
1/4th	**S**	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	**LSB**	0	0	2−bit left ⇔ ×4
Half−stepping	**S**	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	**LSB**	0	0	0	3−bit left ⇔ ×8
SecurePosition	**S**	B9	B8	B7	B6	B5	B4	B3	B2	B1	**LSB**	0	0	0	0	0	No shift

NOTES: LSB: Least Significant Bit
S: Sign bit

Position Ranges

A position is coded by using the binary two's complement format. According to the positioning commands used and to the chosen stepping mode, the position range will be as shown in the following table.

Table 11. POSITION RANGE

Command	Stepping Mode	Position Range	Full Range Excursion	Number of Bits
SetPosition	Half−stepping	−4096 to +4095	8192 half−steps	13
	1/4th micro−stepping	−8192 to +8191	16384 micro−steps	14
	1/8th micro−stepping	−16384 to +16383	32768 micro−steps	15
	1/16th micro−stepping	−32768 to +32767	65536 micro−steps	16

When using the command SetPosition, although coded on 16 bits, the position word will have to be shifted to the left by a certain number of bits, according to the stepping mode.

Secure Position

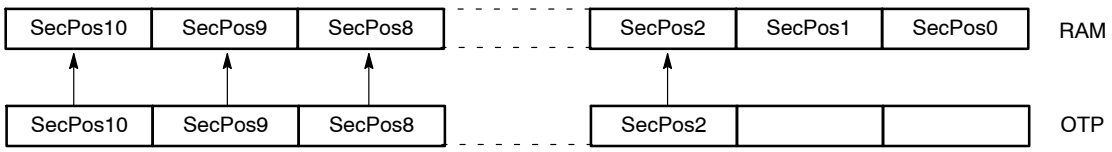
A secure position can be programmed. It is coded in 11−bits, thus having a lower resolution than normal positions, as shown in the following table. See also command GotoSecurePosition.

Table 12. SECURE POSITION

Stepping Mode	Secure Position Resolution
Half−stepping	4 half−steps
1/4th micro−stepping	8 micro−steps (1/4th)
1/8th micro−stepping	16 micro−steps (1/8th)
1/16th micro−stepping	32 micro−steps (1/16th)

Important

NOTES: The secure position is disabled in case the programmed value is the reserved code "10000000000" (0x400 or most negative position).
The resolution of the secure position is limited to 9 bit at start−up. The OTP register is copied in RAM as illustrated below. The RAM bits SecPos1 and SecPos0 are set to 0.



Shaft

A shaft bit, which can be programmed in OTP or with command SetMotorParam, defines whether a positive motion is a clockwise (CW) or counter−clockwise rotation (CCW) (an outer or an inner motion for linear actuators):

- Shaft = 0 ⇒ MOTXP is used as positive pin of the X coil, while MOTXN is the negative one.
- Shaft = 1 ⇒ opposite situation

Exception: in RunVelocity mode, the shaft bit has no function. In this mode the rotational direction is always CW or CCW, which is only determined by the motor wiring.

STRUCTURAL DESCRIPTION

See also the Block Diagram in Figure 1.

Stepper Motordriver

The Motordriver receives the control signals from the control logic. The main features are:

- Two H−bridges, designed to drive a stepper motor with two separated coils. Each coil (X and Y) is driven by one H−bridge, and the driver controls the currents flowing through the coils. The rotational position of the rotor, in unloaded condition, is defined by the ratio of current flowing in X and Y. The torque of the stepper motor when unloaded is controlled by the magnitude of the currents in X and Y.
- The control block for the H−bridges, including the PWM control, the synchronous rectification and the internal current sensing circuitry.
- The charge pump to allow driving of the H−bridges' high side transistors.
- Two pre−scale 4−bit DAC's to set the maximum magnitude of the current through X and Y.

- Two DAC's to set the correct current ratio through X and Y.

Battery voltage monitoring is also performed by this block, which provides the required information to the control logic part. The same applies for detection and reporting of an electrical problem that could occur on the coils or the charge pump.

Control Logic (Position Controller and Main Control)

The control logic block stores the information provided by the I^2C interface (in a RAM or an OTP memory) and digitally controls the positioning of the stepper motor in terms of speed and acceleration, by feeding the right signals to the motordriver state machine.

It will take into account the successive positioning commands to properly initiate or stop the stepper motor in order to reach the set point in a minimum time.

It also receives feedback from the motordriver part in order to manage possible problems and decide on internal actions and reporting to the I^2C interface.

Motion Detection

Motion detection is based on the back−emf generated internally in the running motor. When the motor is blocked, e.g. when it hits the end position, the velocity, and as a result also the generated back−emf, is disturbed. The AMIS−30624/NCV70624 senses the back−emf, calculates a moving average and compares the value with two independent threshold levels. If the back−emf disturbance is bigger than the set threshold, the running motor is stopped.

Miscellaneous

The AMIS−30624/NCV70624 also contains the following:

- An internal oscillator, needed for the control logic handler as well as the control logic and the PWM control of the motordriver.
- An internal trimmed voltage source for precise referencing.
- A protection block featuring a thermal shutdown and a power−on−reset circuit.
- A 5 V regulator (from the battery supply) to supply the internal logic circuitry.

FUNCTIONS DESCRIPTION

This chapter describes the following functional blocks in more detail:

- Position controller
- Main control and register, OTP memory + ROM
- Motordriver

The Motion detection and I^2C controller are discussed in separate chapters.

Position Controller

Positioning and Motion Control

A positioning command will produce a motion as illustrated in Figure 7. A motion starts with an acceleration phase from minimum velocity (Vmin) to maximum velocity (Vmax) and ends with a symmetrical deceleration. This is defined by the control logic according to the position required by the application and the parameters programmed by the application during the configuration phase. The current in the coils is also programmable.

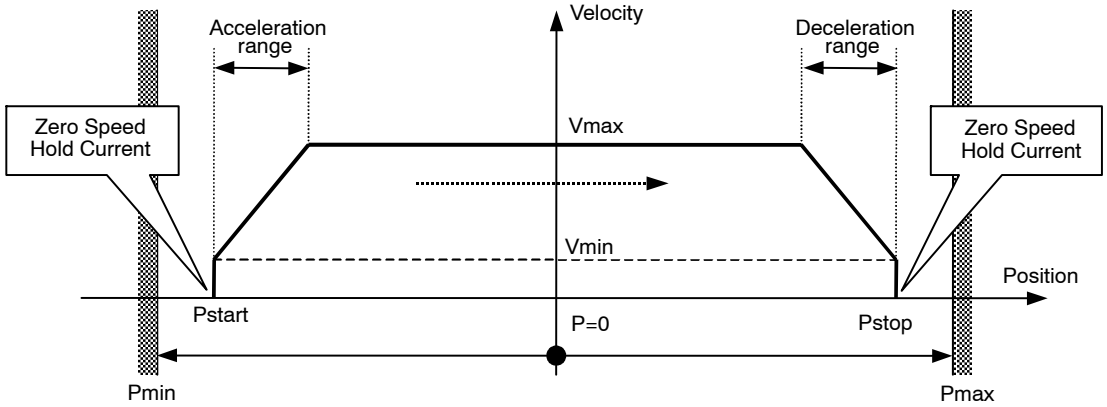


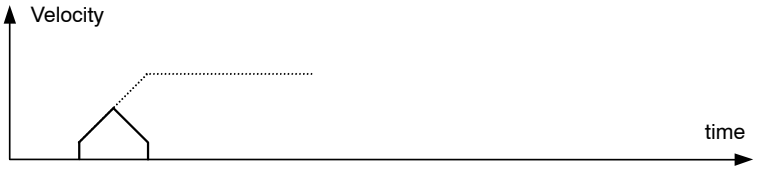
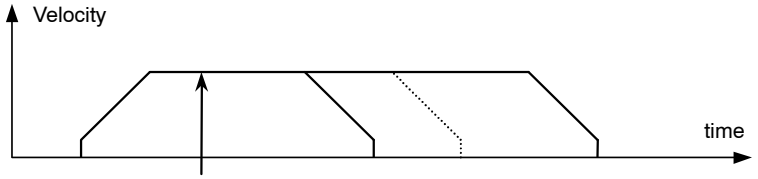
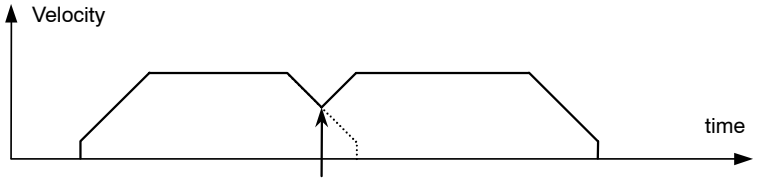
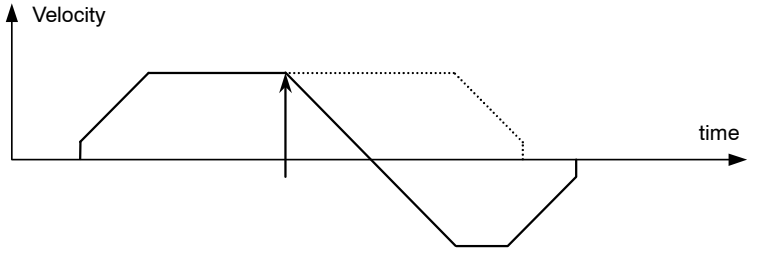
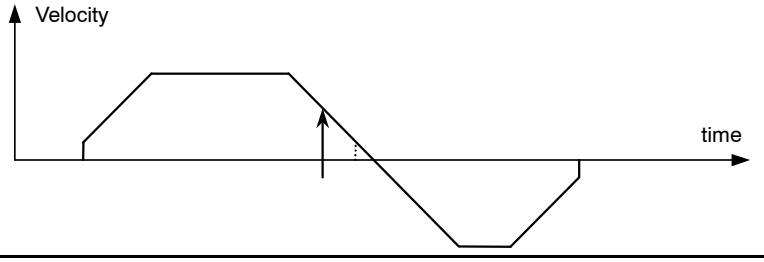
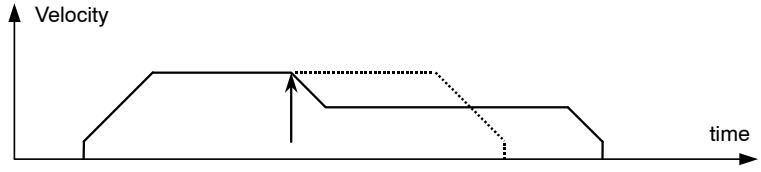
Figure 7. Positioning and Motion Control

Table 13. POSITION RELATED PARAMETERS

Parameter	Reference
Pmax − Pmin	See Positioning
Zero Speed Hold Current	See Ihold
Maximum Current	See Irun
Acceleration and Deceleration	See Acceleration and Deceleration
Vmin	See Minimum Velocity
Vmax	See Maximum Velocity

Different positioning examples are shown in the table below.

Table 14. POSITIONING EXAMPLES

Short motion.	
New positioning command in same direction, shorter or longer, while a motion is running at maximum velocity.	
New positioning command in same direction while in deceleration phase (Note 23) **Note:** there is no wait time between the deceleration phase and the new acceleration phase.	
New positioning command in reverse direction while motion is running at maximum velocity.	
New positioning command in reverse direction while in deceleration phase.	
New velocity programming while motion is running.	

23. Reaching the end position is always guaranteed, however velocity rounding errors might occur after consecutive accelerations during a deceleration phase. The velocity rounding error will be removed at Vmin (e.g. at end of acceleration or when AccShape=1).

Dual Positioning

A SetDualPosition command allows the user to perform a positioning using two different velocities. The first motion is done with the specified Vmin and Vmax velocities in the SetDualPosition command, with the acceleration (deceleration) parameter already in RAM, to a position Pos1[15:0] also specified in SetDualPosition.

Then a second relative motion to a physical position Pos1[15:0] + Pos2[15:0] is done at the specified Vmin velocity in the SetDualPosition command (no acceleration). Once the second motion is achieved, the ActPos register is reset to zero, whereas TagPos register is not changed.

When the Secure position is enabled, after the dual positioning, the secure positioning is executed. The figure below gives a detailed overview of the dual positioning function. After the dual positioning is executed an internal flag is set to indicate the AMIS−30624/NCV70624 is referenced.

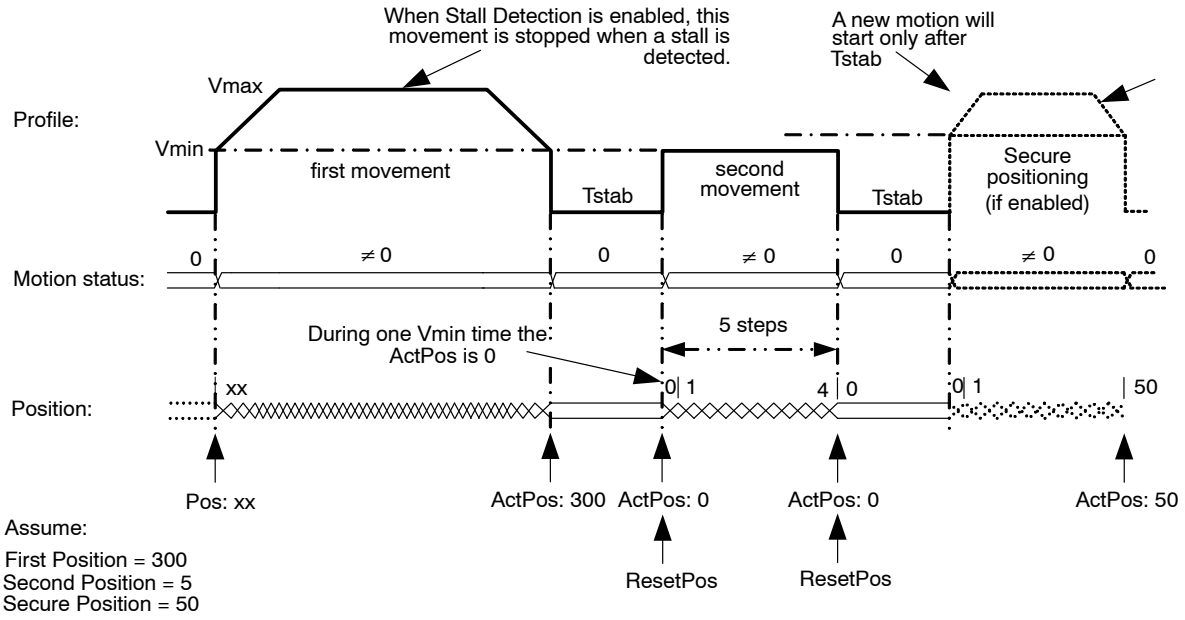


Figure 8. Dual Positioning

Remark: This operation cannot be interrupted or influenced by any further command unless the occurrence of the conditions driving to a motor shutdown or by a HardStop command. Sending a SetDualPosition command while a motion is already ongoing is not recommended.

24. The priority encoder is describing the management of states and commands.
25. A DualPosition sequence starts by setting TagPos buffer register to SecPos value, provided secure position is enabled otherwise TagPos is reset to zero. If a SetPosition(Short) command is issued during a DualPosition sequence, it will be kept in the position buffer memory and executed afterwards. This applies also for the command GotoSecurePosition.
26. Commands such as GetFullStatus1 or GetFullStatus2 will be executed while a Dual Positioning is running.
27. The Pos1, Pos2, Vmax and Vmin values programmed in a SetDualPosition command apply only for this sequence. All other motion parameters are used from the RAM registers (programmed for instance by a former SetMotorParam command). After the DualPosition motion is completed, the former Vmin and Vmax become active again.
28. Commands ResetPosition, SetDualPosition, and SoftStop will be ignored while a DualPosition sequence is ongoing, and will not be executed afterwards.
29. Recommendation: a SetMotorParam command should not be sent during a SetDualPosition sequence: all the motion parameters defined in the command, except Vmin and Vmax, become active immediately.

Position Periodicity

Depending on the stepping mode the position can range from −4096 to +4095 in half−step to −32768 to +32767 in 1/16th micro−stepping mode. One can project all these positions lying on a circle. When executing the command SetPosition, the position controller will set the movement direction in such a way that the traveled distance is minimal.

The figure below illustrates that the moving direction going from ActPos = +30000 to TagPos = −30000 is clockwise.

If a counter clockwise motion is required in this example, several consecutive SetPosition commands can be used. One could also use for larger movements the command RunVelocity.

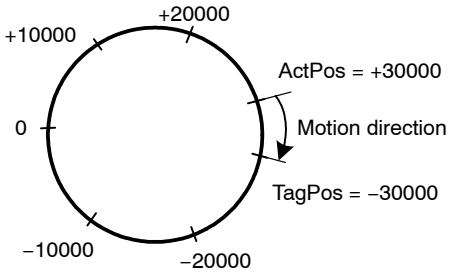


Figure 9. Motion Direction is Function of Difference between ActPos and TagPos

Hardwired Address HW

In the drawing below, a simplified schematic diagram is shown of the HW comparator circuit.

The HW pin is sensed via 2 switches. The DriveHS and DriveLS control lines are alternatively closing the top and bottom switch connecting HW pin with a current to resistor converter. Closing S_{TOP} (DriveHS = 1) will sense a current to GND. In that case the top I→ R converter output is low, via the closed passing switch S_{PASS_T} this signal is fed to the "R" comparator which output HW_Cmp is high. Closing bottom switch S_{BOT} (DriveLS = 1) will sense a current to V_{BAT}. The corresponding I → R converter output is low and via S_{PASS_B} fed to the comparator. The output HW_Cmp will be high.

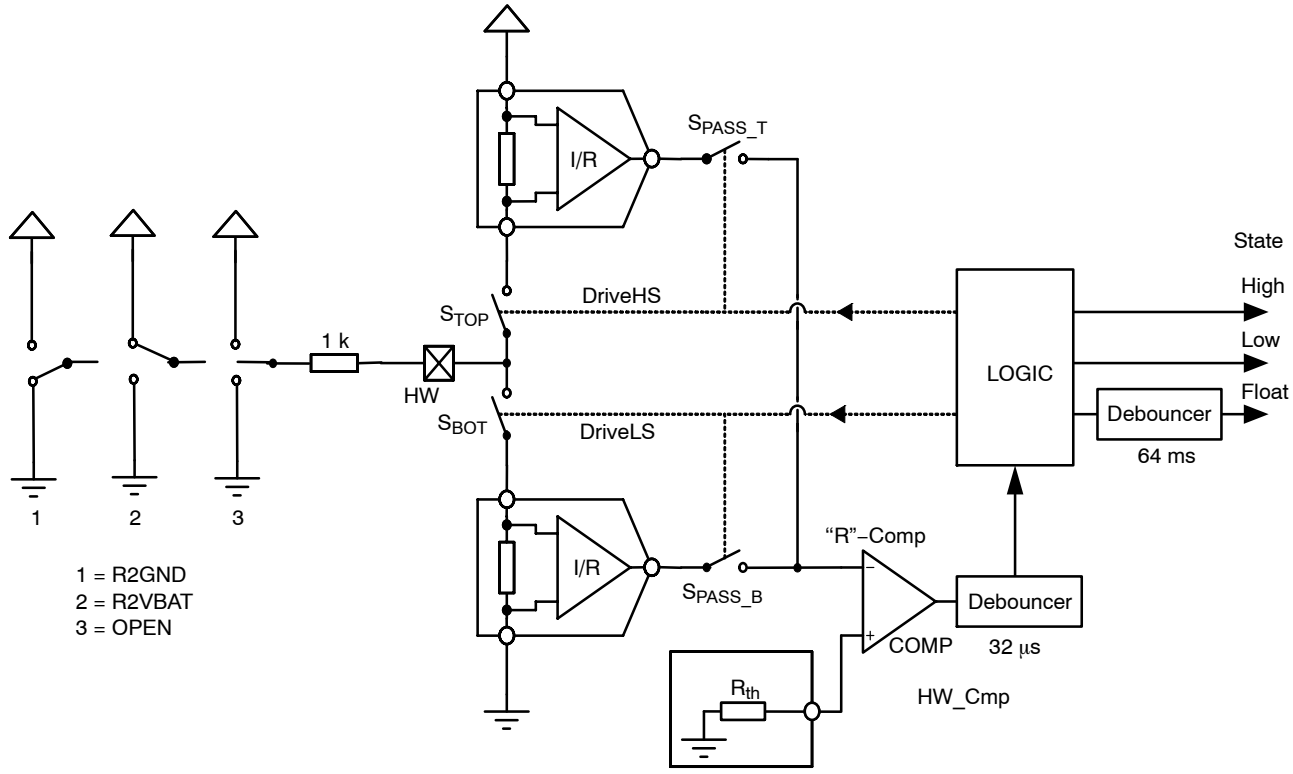


Figure 10. Simplified Schematic Diagram of the HW Comparator

3 cases can be distinguished (see also Figure 10 above):

- HW is connected to ground: R2GND or drawing 1
- HW is connected to VBAT: R2VBAT or drawing 2
- HW is floating: OPEN or drawing 3

Table 15. STATE DIAGRAM OF THE HW COMPARATOR

Previous State	DriveLS	DriveHS	HW_Cmp	New State	Condition	Drawing
Float	1	0	0	Float	R2GND or OPEN	1 or 3
Float	1	0	1	High	R2VBAT	2
Float	0	1	0	Float	R2VBAT or OPEN	2 or 3
Float	0	1	1	Low	R2GND	1
Low	1	0	0	Low	R2GND or OPEN	1 or 3
Low	1	0	1	High	R2VBAT	2
Low	0	1	0	Float	R2VBAT or OPEN	2 or 3
Low	0	1	1	Low	R2GND	1
High	1	0	0	Float	R2GND or OPEN	1 or 3
High	1	0	1	High	R2VBAT	2
High	0	1	0	High	R2VBAT or OPEN	2 or 3
High	0	1	1	Low	R2GND	1

The logic is controlling the correct sequence in closing the switches and in interpreting the 32 μs debounced HW_Cmp output accordingly. The output of this small state−machine is corresponding to:

- High or address = 1
- Low or address = 0
- Floating

As illustrated in the table above (Table 15), the state is depending on the previous state, the condition of the 2 switch controls (DriveLS and DriveHS) and the output of HW_Cmp. Figure 11 shows an example of a practical case where a connection to VBAT is interrupted.

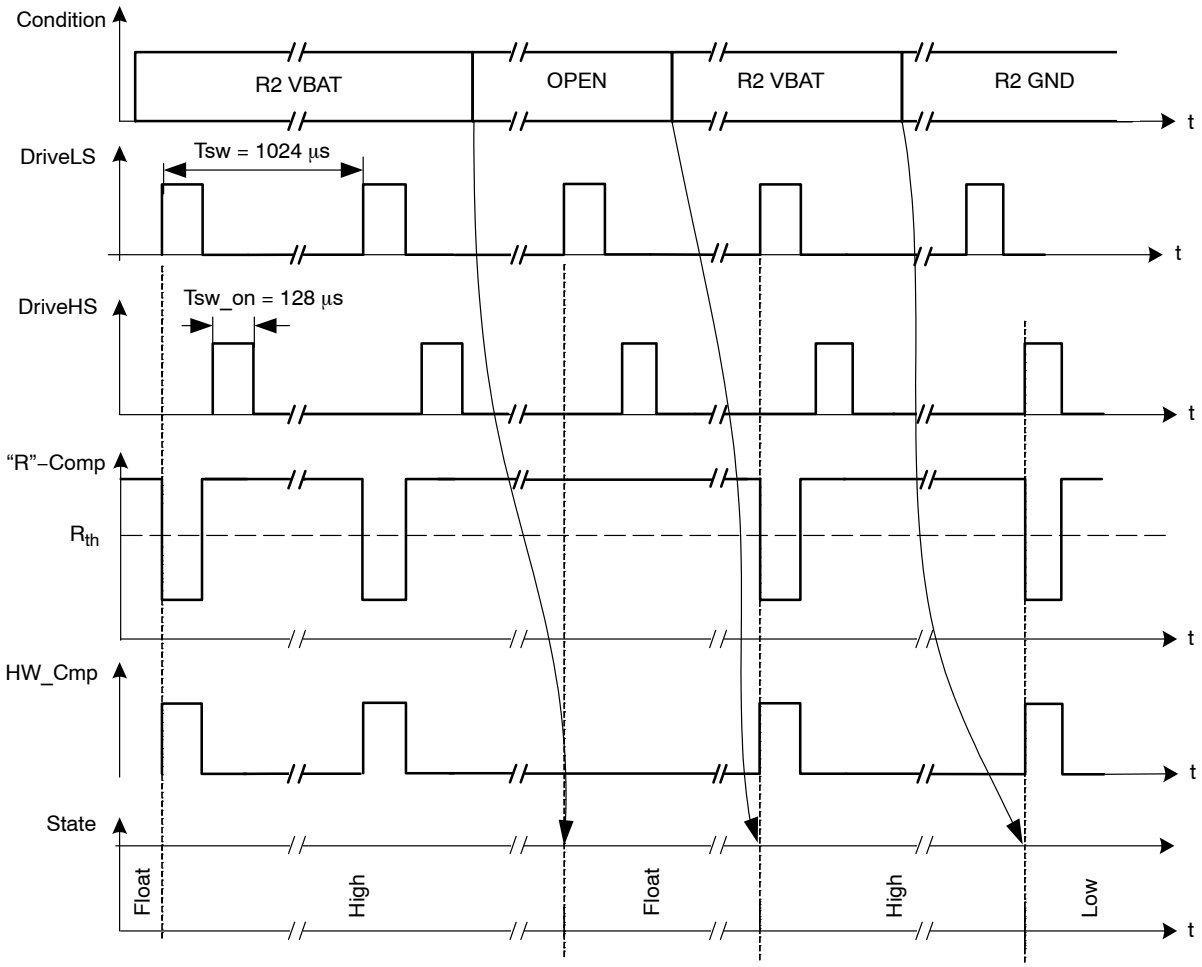


Figure 11. Timing Diagram Showing the Change in States for HW Comparator

R2VBAT

A resistor is connected between VBAT and HW. Every 1024 μs S_{BOT} is closed and a current is sensed. The output of the I ⇒ R converter is low and the HW_Cmp output is high. Assuming the previous state was floating, the internal logic will interpret this as a change of state and the new state will be high (see also Table 15). The next time S_{BOT} is closed the same conditions are observed. The previous state was high so based on Table 15 the new state remains unchanged. This high state will be interpreted as HW address = 1.

OPEN

In case the HW connection is lost (broken wire, bad contact in connector) the next time S_{BOT} is closed, this will be sensed. There will be no current, the output of the corresponding I ⇒ R converter is high and the HW_Cmp will be low. The previous state was high. Based in Table 15 one can see that the state changes to float. This will trigger a motion to secure position after a debounce time of 64 ms, which prevents false triggering in case of micro−interruptions of the power supply.

R2GND

If a resistor is connected between HW and the GND, a current is sensed every 1024 μs when S_{TOP} is closed. The output of the top I ⇒ R converter is low and as a result the HW_Cmp output switches to high. Again based on the stated diagram in Table 15 one can see that the state will change to Low. This low state will be interpreted as HW address = 0.

External Switch SWI

As illustrated in Figure 12 the SWI comparator is almost identical to HW. The major difference is in the limited number of states. Only open or closed is recognized leading to respectively ESW = 0 and ESW = 1.

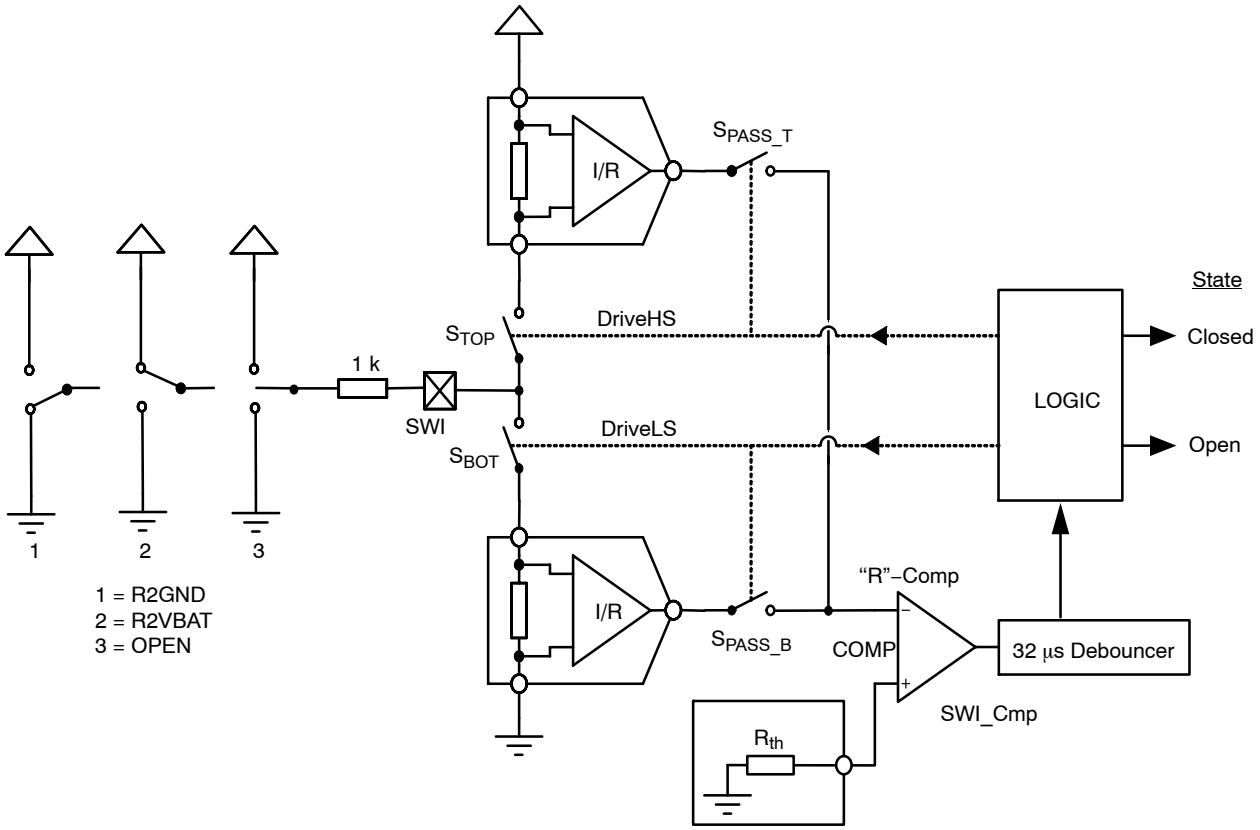


Figure 12. Simplified Schematic Diagram of the SWI Comparator

As illustrated in the drawing above, a change in state is always synchronized with DriveHS or DriveLS. The same synchronization is valid for updating the internal position register. This means that after every current pulse (or closing of S_{TOP} or S_{BOT}) the state of the position switch together with the corresponding position is memorized.

The `FullStatus1` command reads back the `<ActPos>` register and the status of ESW. In this way the master node may get synchronous information about the state of the switch together with the position of the motor. See Table 16 below.

Table 16. GetFullStatus1 I²C COMMAND

GetFullStatus1 Response Frame									
		Structure							
Byte	Content	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Address	1	1	OTP3	OTP2	OTP1	OTP0	HW	1
1	Address	1	1	1	OTP3	OTP2	OTP1	OTP0	HW
2	Data 1	Irun[3:0]				Ihold[3:0]			
3	Data 2	Vmax[3:0]				Vmin[3:0]			
4	Data 3	AccShape	StepMode[1:0]		Shaft	Acc[3:0]			
5	Data 4	VddReset	StepLoss	ElDef	UV2	TSD	TW	Tinfo[1:0]	
6	Data 5	Motion[2:0]			ESW	OVC1	OVC2	Stall	CPFail
7	Data 6	1	1	1	1	1	1	1	1
8	Data 7	AbsThr[3:0]				DelThr[3:0]			

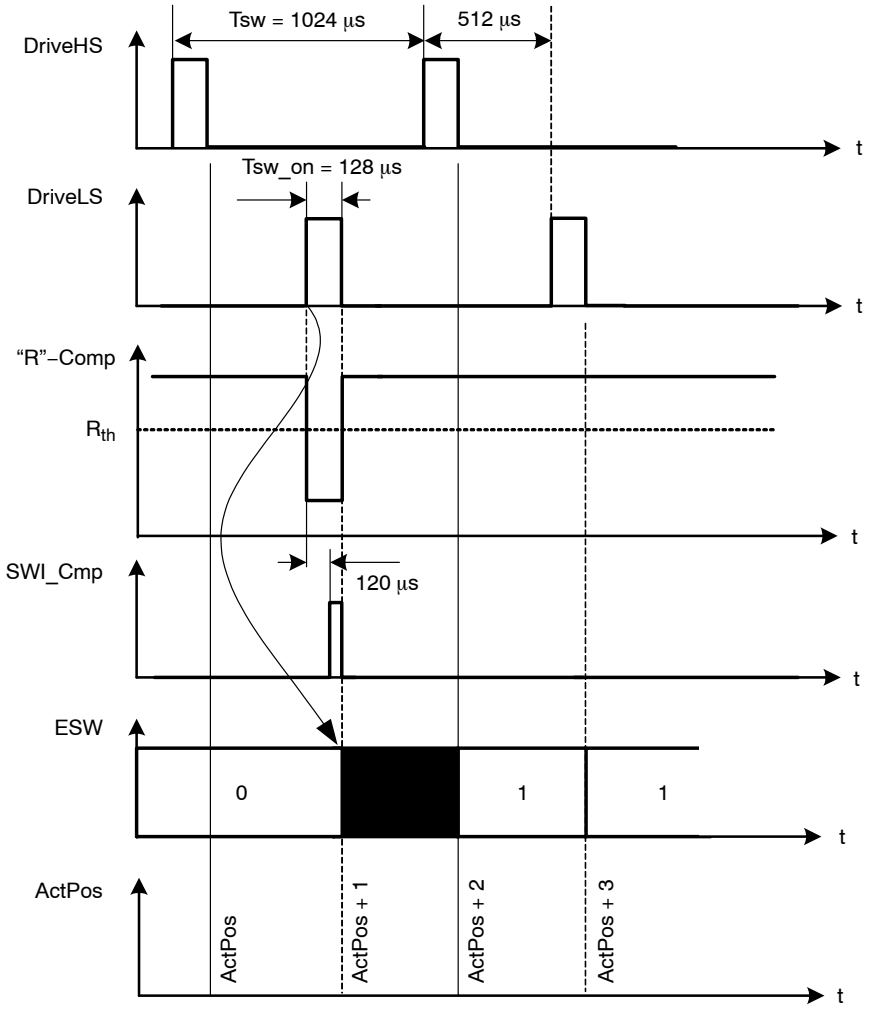


Figure 13. Simplified Timing Diagram Showing the Change in States for SWI Comparator

Main Control and Register, OTP memory + ROM

Power−up Phase

Power−up phase of the AMIS−30624/NCV70624 will not exceed 10 ms. After this phase, the AMIS−30624/NCV70624 is in standby mode, ready to receive I^2C messages and execute the associated commands. After power−up, the registers and flags are in the reset state, while some of them are being loaded with the OTP memory content (see Table 19: RAM Registers).

Reset

After power−up, or after a reset occurrence (e.g. a micro−cut on pin V_{BB} has made V_{DD} to go below VddReset level), the H−bridges will be in high−impedance mode, and the registers and flags will be in a predetermined position. This is documented in Table 19: RAM Registers and Table 20: Flags Table.

Soft−stop

A soft−stop is an immediate interruption of a motion, but with a deceleration phase. At the end of this action, the register `<TagPos>` is loaded with the value contained in register `<ActPos>`, see Table 19: Ram Registers). The circuit is then ready to execute a new positioning command, provided thermal and electrical conditions allow for it.

Thermal Shutdown Mode

When thermal shutdown occurs, the circuit performs a <SoftStop> command and goes to motor shutdown mode (see Figure 14: State Diagram Temperature Management).

Temperature Management

The AMIS−30624/NCV70624 monitors temperature by means of two thresholds and one shutdown level, as illustrated in the state diagram and illustration of Figure 14: State Diagram Temperature Management below. The only condition to reset flags <TW> and <TSD> (respectively thermal warning and thermal shutdown) is to be at a temperature lower than Ttw and to get the occurrence of a GetFullStatus1 I^2C frame.

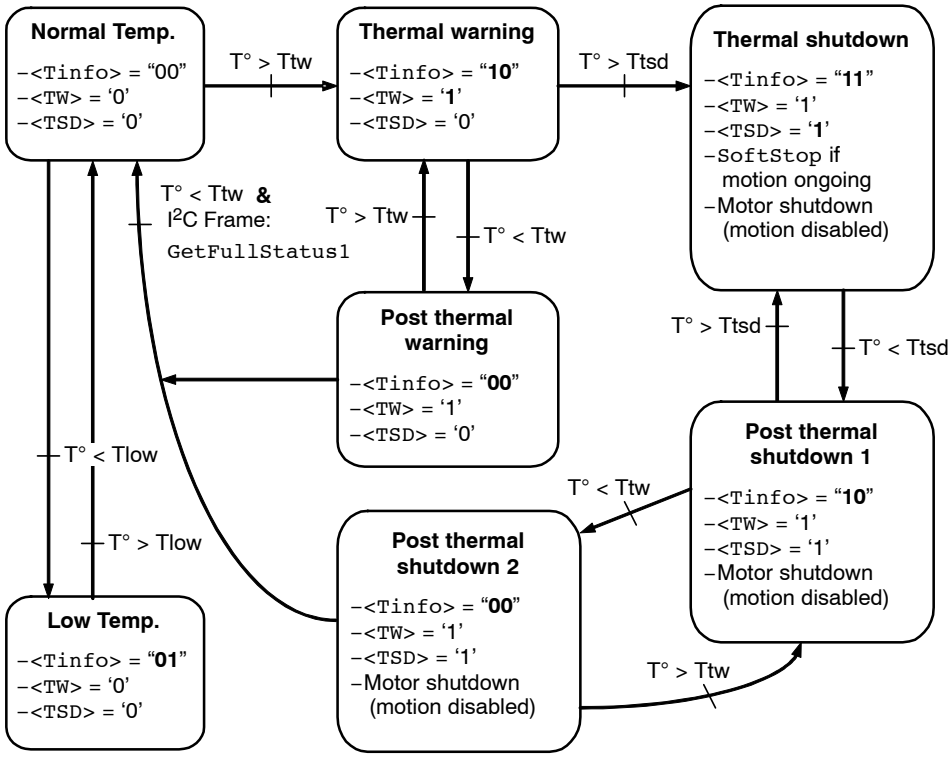


Figure 14. State Diagram Temperature Management

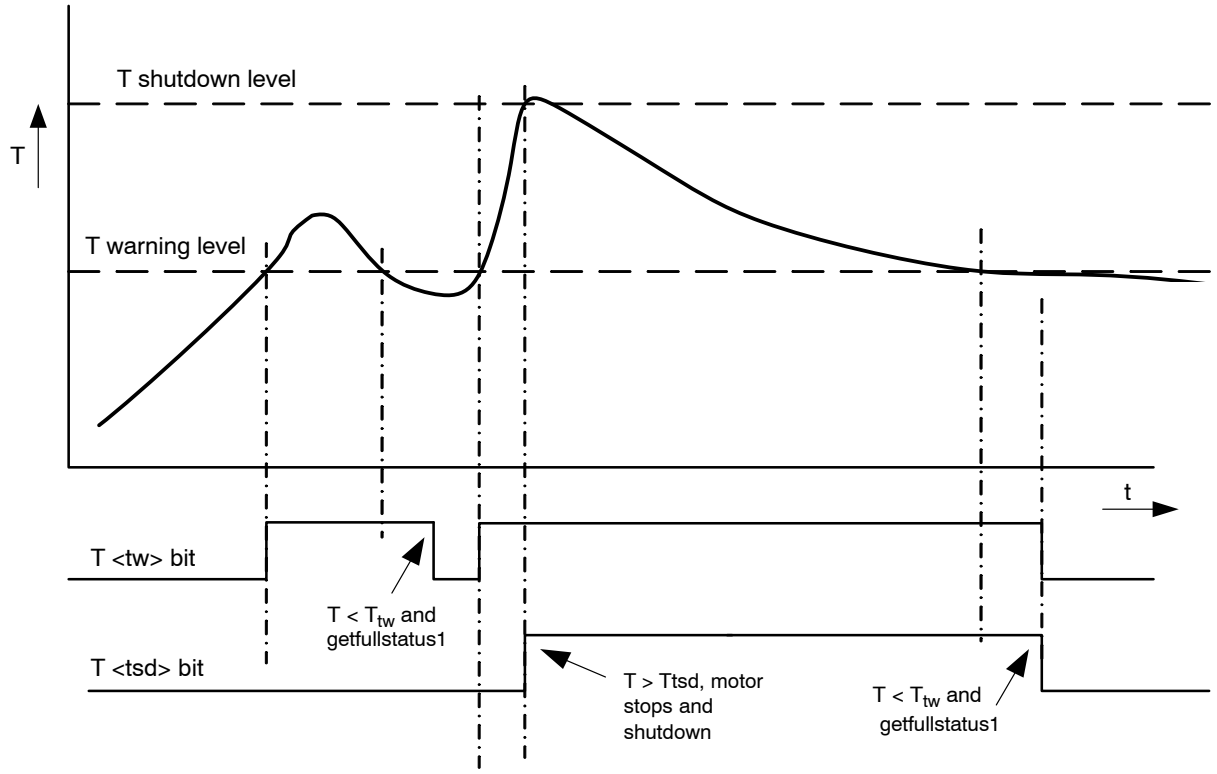


Figure 15. Illustration of Thermal Management Situation

Autarkic Functionality in Under−Voltage Condition

Battery Voltage Management

The AMIS−30624/NCV70624 monitors the battery voltage by means of one threshold and one shutdown level. The only condition to reset flags <UV2> and <StepLoss> is to recover by a battery voltage higher than UV1 and to receive a GetFullStatus1 command.

Autarkic Function

The device enters states <HardUnder> (see Figure 16), followed by <ShutUnder> when V_{BB} is below the UV2 level or <CPFail> = 1. The motion is stopped immediately and Target Position (TagPos) is kept and not overwritten by Actual Position (ActPos). The motor is in HiZ state and the flags <UV2> and <Steploss> are set to inform the master that the voltage has dropped below UV2 or the charge pump voltage has dropped below the level of the charge pump comparator and loss of steps is possible.

- If in this state V_{BB} becomes > UV1 within 15 seconds, then AMIS−30624/NCV70624 returns to <Stopped> state. From there, it resumes the interrupted motion and accepts updates of the target position by means of the commands SetPosition and GotoSecurePosition, even if the <UV2> flag, the <CPFail> flag and <Steploss> flags are NOT cleared.
- If however the V_{BB} voltage remains below UV2 level or the charge pump voltage level is below the charge pump comparator for more than 15 seconds, then the device will enter <Shutdown> state and the target position is overwritten by Actual Position. This state can be exited only if V_{BB} is > UV1, the charge pump voltage is above the charge pump comparator voltage and an incoming command GetFullStatus1 is received.

Important Notes:

1. In the case of Autarkic positioning, care needs to be taken because accumulated steploss can cause a significant deviation between physical and stored actual position.
2. The SetDualPosition command will only be executed after clearing the <UV2>, CPFail and <Steploss> flags.
3. RAM reset occurs when Vdd < VddReset (digital Power−On−Reset level).
4. The Autarkic function remains active as long as V_{DD} > VddReset.

OTP Register

OTP Memory Structure

The table below shows how the parameters to be stored in the OTP memory are located.

Table 17. OTP MEMORY STRUCTURE

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x00	OSC3	OSC2	OSC1	OSC0	IREF3	IREF2	IREF1	IREF0
0x01	0	TSD2	TSD1	TSD0	BG3	BG2	BG1	BG0
0x02	AbsThr3	AbsThr2	AbsThr1	AbsThr0	PA3	PA2	PA1	PA0
0x03	Irun3	Irun2	Irun1	Irun0	Ihold3	Ihold2	Ihold1	Ihold0
0x04	Vmax3	Vmax2	Vmax1	Vmax0	Vmin3	Vmin2	Vmin1	Vmin0
0x05	SecPos10	SecPos9	SecPos8	Shaft	Acc3	Acc2	Acc1	Acc0
0x06	SecPos7	SecPos6	SecPos5	SecPos4	SecPos3	SecPos2		
0x07	DelThr3	DelThr2	DelThr1	DelThr0	StepMode1	StepMode0	LOCKBT	LOCKBG

Parameters stored at address 0x00 and 0x01 and bit <LOCKBT> are already programmed in the OTP memory at circuit delivery. They correspond to the calibration of the circuit and are just documented here as an indication.

Each OTP bit is at ‘0’ when not zapped. Zapping a bit will set it to ‘1’. Thus only bits having to be at ‘1’ must be zapped. Zapping of a bit already at ‘1’ is disabled. Each OTP byte will be programmed separately (see command SetOTPparam). Once OTP programming is completed, bit <LOCKBG> can be zapped to disable future zapping, otherwise any OTP bit at ‘0’ could still be zapped by using a SetOTPparam command.

Table 18. OTP OVERWRITE PROTECTION

Lock Bit	Protected Bytes
LOCKBT (factory zapped before delivery)	0x00 to 0x01
LOCKBG	0x00 to 0x07

The command used to load the application parameters via the I^{2}C bus in the RAM prior to an OTP Memory programming is SetMotorParam. This allows for a functional verification before using a SetOTPparam command to program and zap separately one OTP memory byte. A GetOTPparam command issued after each SetOTPparam command allows verifying the correct byte zapping.

Note: Zapped bits will become active only after a power cycle. After programming the I^{2}C bits the power cycle has to be performed first to guarantee further communication with the device.

Application Parameters Stored in OTP Memory

Except for the physical address <PA[3:0]> these parameters, although programmed in a non−volatile memory can still be overridden in RAM by a I^{2}C writing operation.

PA[3:0] In combination with hired wired (HW) address, it forms the physical address AD[6:0] of the stepper−motor. Up to 32 stepper motors can theoretically be connected to the same I^{2}C bus.

AbsThr[3:0] Absolute threshold used for the motion detection

Index	AbsThr				AbsThr level (V) (*)
0	0	0	0	0	Disable
1	0	0	0	1	0.5
2	0	0	1	0	1.0
3	0	0	1	1	1.5
4	0	1	0	0	2.0
5	0	1	0	1	2.5
6	0	1	1	0	3.0
7	0	1	1	1	3.5
8	1	0	0	0	4.0
9	1	0	0	1	4.5
A	1	0	1	0	5.0
B	1	0	1	1	5.5
C	1	1	0	0	6.0
D	1	1	0	1	6.5
E	1	1	1	0	7.0
F	1	1	1	1	7.5

(*) Not tested in production. Values are approximations.

DelThr[3:0] Relative threshold used for the motion detection

Index	DelThr				DelThr Level (V) (*)
0	0	0	0	0	Disable
1	0	0	0	1	0.25
2	0	0	1	0	0.50
3	0	0	1	1	0.75
4	0	1	0	0	1.00
5	0	1	0	1	1.25
6	0	1	1	0	1.50
7	0	1	1	1	1.75
8	1	0	0	0	2.00
9	1	0	0	1	2.25
A	1	0	1	0	2.50
B	1	0	1	1	2.75
C	1	1	0	0	3.00
D	1	1	0	1	3.25
E	1	1	1	0	3.50
F	1	1	1	1	3.75

(*) Not tested in production. Values are approximations.

Irun[3:0] Current amplitude value to be fed to each coil of the stepper−motor. The table below provides the 16 possible values for <IRUN>.

Index	Irun				Run Current (mA)
0	0	0	0	0	59
1	0	0	0	1	71
2	0	0	1	0	84
3	0	0	1	1	100
4	0	1	0	0	119
5	0	1	0	1	141
6	0	1	1	0	168
7	0	1	1	1	200
8	1	0	0	0	238
9	1	0	0	1	283
A	1	0	1	0	336
B	1	0	1	1	400
C	1	1	0	0	476
D	1	1	0	1	566
E	1	1	1	0	673
F	1	1	1	1	800

Ihold[3:0] Hold current for each coil of the stepper−motor. The table below provides the 16 possible values for <IHOLD>.

Index	Ihold				Hold Current (mA)
0	0	0	0	0	59
1	0	0	0	1	71
2	0	0	1	0	84
3	0	0	1	1	100
4	0	1	0	0	119
5	0	1	0	1	141
6	0	1	1	0	168
7	0	1	1	1	200
8	1	0	0	0	238
9	1	0	0	1	283
A	1	0	1	0	336
B	1	0	1	1	400
C	1	1	0	0	476
D	1	1	0	1	566
E	1	1	1	0	673
F	1	1	1	1	0

Note: When the motor is stopped, the current is reduced from <IRUN> to <IHOLD>. In the case of 0 mA hold current (1111 in the hold current table), the following sequence is applied:

1. The current is first reduced to 59 mA (corresponding to 0000 value in the table).
2. The PWM regulator is switched off; the bottom transistors of the bridges are grounded.

StepMode Setting of step modes.

StepMode		Step Mode
0	0	1/2 stepping
0	1	1/4 stepping
1	0	1/8 stepping
1	1	1/16 stepping

Shaft This bit distinguishes between a clock−wise or counter−clock−wise rotation. The shaft bit is not working in RunVelocity mode.

SecPos[10:2] Secure Position of the stepper−motor. This is the position to which the motor is driven in case of a HW pin connection is lost. If <SecPos[10:2]> = “100 0000 00xx”, secure positioning is disabled; the stepper−motor will be kept in the position occupied at the moment these events occur.

Note: The Secure Position is coded on 11 bits only, providing actually the most significant bits of the position, the non coded least significant bits being set to ‘0’. The Secure Position in OTP has only 9 bits. The two least significant bits are loaded as ‘0’ to RAM when copied from OTP.

`Vmax[3:0]` Maximum velocity

Index	Vmax				Vmax(full step/s)	Group
0	0	0	0	0	99	A
1	0	0	0	1	136	B
2	0	0	1	0	167	
3	0	0	1	1	197	
4	0	1	0	0	213	
5	0	1	0	1	228	
6	0	1	1	0	243	
7	0	1	1	1	273	C
8	1	0	0	0	303	
9	1	0	0	1	334	
A	1	0	1	0	364	
B	1	0	1	1	395	
C	1	1	0	0	456	
D	1	1	0	1	546	D
E	1	1	1	0	729	
F	1	1	1	1	973	

Vmin[3:0] Minimum velocity.

Index	Vmin				Vmax Factor
0	0	0	0	0	1
1	0	0	0	1	1/32
2	0	0	1	0	2/32
3	0	0	1	1	3/32
4	0	1	0	0	4/32
5	0	1	0	1	5/32
6	0	1	1	0	6/32
7	0	1	1	1	7/32
8	1	0	0	0	8/32
9	1	0	0	1	9/32
A	1	0	1	0	10/32
B	1	0	1	1	11/32
C	1	1	0	0	12/32
D	1	1	0	1	13/32
E	1	1	1	0	14/32
F	1	1	1	1	15/32

`Acc[3:0]` Acceleration and deceleration between Vmax and Vmin.

Index	Acc				Acceleration (Full−step/s^2)
0	0	0	0	0	49 (*)
1	0	0	0	1	218 (*)
2	0	0	1	0	1004 .
3	0	0	1	1	3609 .
4	0	1	0	0	6228 .
5	0	1	0	1	8848 .
6	0	1	1	0	11409 .
7	0	1	1	1	13970 .
8	1	0	0	0	16531 .
9	1	0	0	1	19092 (*)
A	1	0	1	0	21886 (*)
B	1	0	1	1	24447 (*)
C	1	1	0	0	27008 (*)
D	1	1	0	1	29570 (*)
E	1	1	1	0	34925 (*)
F	1	1	1	1	40047 (*)

(*) restriction on speed

Table 19. RAM REGISTERS

Register	Mnemonic	Length (bit)	Related Commands	Comment	Reset State
Actual position	ActPos	16	GetFullStatus2 GotoSecurePos ResetPosition	16−bit signed	
Last programmed Position	Pos/TagPos	16/11	GetFullStatus2 GotoSecurePos ResetPosition SetPosition	16−bit signed or 11−bit signed for half stepping (see Positioning)	
Acceleration shape	AccShape	1	GetFullStatus1 SetMotorParam ResetToDefault	'0' ⇒ normal acceleration from Vmin to Vmax '1' ⇒ motion at Vmin without acceleration	'0'
Coil peak current	Irun	4	GetFullStatus1 SetMotorParam ResetToDefault	Operating current See look−up table Irun	From OTP memory
Coil hold current	Ihold	4	GetFullStatus1 SetMotorParam ResetToDefault	Standstill current See look−up table Ihold	
Minimum Velocity	Vmin	4	GetFullStatus1 SetMotorParam ResetToDefault	See Section Minimum Velocity See look−up table Vmin	
Maximum Velocity	Vmax	4	GetFullStatus1 SetMotorParam ResetToDefault	See Section Maximum Velocity See look−up table Vmax	
Shaft	Shaft	1	GetFullStatus1 SetMotorParam ResetToDefault	Direction of movement	
Acceleration/ deceleration	Acc	4	GetFullStatus1 SetMotorParam ResetToDefault	See Section Acceleration See look−up table Acc	
Secure Position	SecPos	11	GetFullStatus2 SetMotorParam ResetToDefault	Target position when HW connection fails; 11 MSB's of 16−bit position (LSB's fixed to '0')	
Stepping mode	StepMode	2	GetFullStatus1 SetStallParam ResetToDefault	See Section Stepping Modes See look−up table StepMode	
Stall detection absolute threshold	AbsThr	4	GetFullStatus1 SetStallParam ResetToDefault		
Stall detection delta threshold	DelThr	4	GetFullStatus1 SetStallParam ResetToDefault		
Stall detection delay	FS2StallEn	3	GetFullStatus2 SetStallParam	Delays the stall detection after acceleration	'000'
Stall detection sampling	MinSamples	3	GetFullStatus2 SetStallParam	Duration of the zero current step in number of PWM cycles.	'000'
PWM Jitter	PWMJEn	1	GetFullStatus2 SetStallParam	'1' means jitter is added	'0'
100% duty cycle Stall Enable	DC100StEn	1	GetFullStatus2 SetStallParam	'1' means stall detection is enabled in case PWM regulator runs at δ = 100%	'0'
PWM frequency	PWMFreq	1	SetMotorParam	'0' means ~ 22 KHz, '1' means ~ 44 KHz	'0'

30. A ResetToDefault command will act as a reset of the RAM content, except for ActPos and TagPos, which are registers that are not modified. Therefore, the application should not send a ResetToDefault during a motion, to avoid any unwanted change of parameter.

Table 20. FLAGS TABLE

Flag	Mnemonic	Length (bit)	Related Commands	Comment	Reset State
Charge pump failure	CPFail	1	GetFullStatus1	'0' = charge pump OK '1' = charge pump failure Resets only after GetFullStatus1	'0'
Electrical defect	ElDef	1	GetFullStatus1	<OVC1> or <OVC2> or 'open−load on coil X' or 'open−load on coil XY or <CPFail> Resets only after GetFullStatus1	'0'
External switch status	ESW	1	GetFullStatus1	'0' = open '1' = close	'0'
Electrical flag	HS	1	Internal use	<CPFail> or <UV2> or <ElDef> or <VDDreset>	'0'
Motion status	Motion	3	GetFullStatus1	"x00" = Stop "001" = inner (CCW) motion acceleration "010" = inner (CCW) motion deceleration "011" = inner (CCW) motion max. speed "101" = outer (CW) motion acceleration "110" = outer (CW) motion deceleration "111" = outer (CW) motion max. speed	"000"
Over current in coil X	OVC1	1	GetFullStatus1	'1' = over current; reset only after GetFull−Status1	'0'
Over current in coil Y	OVC2	1	GetFullStatus1	'1' = over current; reset only after GetFull−Status1	'0'
Secure position enabled	SecEn	1	Internal use	'0' if <SecPos> = "100 0000 0000" '1' otherwise	n.a.
Step loss	StepLoss	1	GetFullStatus1	'1' = step loss due to under voltage, over current, open circuit or stall; Resets only after GetFull−Status1	'1'
Delta High Stall	DelStallHi	1	GetFullStatus2	'1' = Vbemf > Ubemf + DeltaThr	'0'
Delta Low Stall	DelStallLo	1	GetFullStatus2	'1' = Vbemf < Ubemf − DeltaThr	'0'
Absolute Stall	AbsStall	1	GetFullStatus2	'1' = Vbemf < AbsThr	'0'
Stall	Stall	1	GetFullStatus1		'0'
Motor stop	Stop	1	Internal use		'0'
Temperature info	Tinfo	2	GetFullStatus1	"00" = normal temperature range "01" = low temperature warning "10" = high temperature warning "11" = motor shutdown	"00"
Thermal shutdown	TSD	1	GetFullStatus1	'1' = shutdown ($T_j > T_{tsd}$) Resets only after GetFullStatus1 and if <Tinfo> = "00"	'0'
Thermal warning	TW	1	GetFullStatus1	'1' = over temperature ($T_j > T_{tw}$) Resets only after GetFullStatus1 and if <Tinfo> = "00"	'0'
Battery stop voltage	UV2	1	GetFullStatus1	'0' = V_{BB} > UV2 '1' = V_{BB} ≤ UV2 Resets only after GetFullStatus1	'0'
Digital supply reset	VddReset	1	GetActualPos GetStatus GetFullStatus1	Set at '1' after power−up of the circuit. If this was due to a supply micro−cut, it warns that the RAM contents may have been lost; can be reset to '0' with a Get(Full)Status1 command	'1'

Priority Encoder

The table below describes the simplified state management performed by the main control block.

Table 21. PRIORITY ENCODER

State →	Standby	Stopped	GotoPos	DualPosition	SoftStop	HardStop	ShutDown	HardUnder	ShutUnder
Command ↓		**Motor Stopped, Ihold in Coils**	**Motor Motion Ongoing**	**No Influence on RAM and Tag-Pos**	**Motor Decelerating**	**Motor Forced to Stop**	**Motor Stopped, H−bridges in Hi−Z**		
GetOTPparam		OTP refresh; I²C slave response	OTP refresh; I²C slave response	OTP refresh; I²C slave response	OTP refresh; I²C slave response	OTP refresh; I²C slave response	OTP refresh; I²C slave response		
GetFullStatus1 [attempt to clear all flags] (Note 31)		I²C slave response	I²C slave response	I²C slave response	I²C slave response	I²C slave response	I²C slave response; if (<TSD> or <ElFlag> = '0' then →**Stopped**		
GetFullStatus2		I²C slave response	I²C slave response	I²C slave response	I²C slave response	I²C slave response	I²C slave response		
ResetToDefault [ActPos and TagPos are not altered]		OTP refresh; OTP to RAM; AccShape reset	OTP refresh; OTP to RAM; AccShape reset	OTP refresh; OTP to RAM; AccShape reset (Note 33)	OTP refresh; OTP to RAM; AccShape reset	OTP refresh; OTP to RAM; AccShape reset	OTP refresh; OTP to RAM; AccShape reset		
SetMotorParam [Master takes care about proper update]	RAM update	RAM update	RAM update	RAM update	RAM update	RAM update	RAM update	RAM update	RAM update
ResetPosition		<TagPos> and <ActPos> reset					<TagPos> and <ActPos> reset		<TagPos> and <ActPos> reset
SetPosition		<TagPos> updated; →**GotoPos**	<TagPos> updated	<TagPos> updated					
SetPositionShort		<TagPos> updated; →**GotoPos**	<TagPos> updated	<TagPos> updated					
GotoSecPosition		If <SecEn> = '1' then <TagPos> = <SecPos>; →**GotoPos**	If <SecEn> = '1' then <TagPos> = <SecPos>	If <SecEn> = '1' then <TagPos> = <SecPos>					
DualPosition		→DualPosition							
SoftStop			→SoftStop						
HardStop			→HardStop	→HardStop	→HardStop				
V_{BB} < UV2 and t > 15 seconds		→HardUnder	→HardUnder	→HardStop	→HardUnder				
V_{BB} < UV2 and t < 15 seconds									→Stopped
<ElDef> = '1' ⇒ <HS> = '1'		→Shutdown	→HardStop; <StepLoss> = '1'	→HardStop; <StepLoss> = '1'	→HardStop; <StepLoss> = '1'				→Shutdown
Thermal shutdown [<TSD> = '1']		→Shutdown	→SoftStop	→SoftStop					→Shutdown
Motion finished		n.a.	→Stopped	→Stopped	→Stopped; <TagPos> = <ActPos>	→Stopped; <TagPos> = <ActPos>	n.a.		

With the Following Color Code:

Command Ignored | Transition to Another State | Master is responsible for proper update (see Note 36)

NOTE: See table notes on the following page.

31. <ElFlag> = <CPFail> or <UV2> or <ElDef> or <VDDreset>
32. After power−on−reset, the <Standby> state is entered.
33. A DualPosition sequence runs with a separate set of RAM registers. The parameters that are not specified in a DualPosition command are loaded with the values stored in RAM at the moment the DualPosition sequence starts. <AccShape> is forced to '1' during second motion. <AccShape> at '0' will be taken into account after the DualPosition sequence. A GetFullStatus1 command will return the default parameters for <Vmax> and <Vmin> stored in RAM.
34. Shutdown state can be left only when <TSD> and <HS> flags are reset.
35. Flags can be reset only after the master could read them via a GetFullStatus1 command, and provided the physical conditions allow for it (normal temperature, correct battery voltage and no electrical or charge pump defect).
36. A SetMotorParam command sent while a motion is ongoing (state <GotoPos>) should not attempt to modify <Acc> and <Vmin> values. This can be done during a DualPosition sequence since this motion uses its own parameters, the new parameters will be taken into account at the next SetPosition command.
37. <SecEn> = '1' when register <SecPos> is loaded with a value different from the most negative value (i.e. different from 0x400 = "100 0000 0000").
38. <Stop> flag allows distinguishing whether state <Stopped> was entered after HardStop/SoftStop or not. <Stop> is set to '1' when leaving state <HardStop> or <SoftStop> and is reset during first clock edge occurring in state <Stopped>.
39. While in state <Stopped>, if <ActPos> → <TagPos> there is a transition to state <GotoPos>. This transition has the lowest priority, meaning that <Stop>, <TSD>, etceteras are first evaluated for possible transitions.
40. If <StepLoss> is active, then SetPosition and GotoSecurePosition commands **are not** ignored. <StepLoss> can only be cleared by a GetFullStatus1 command.

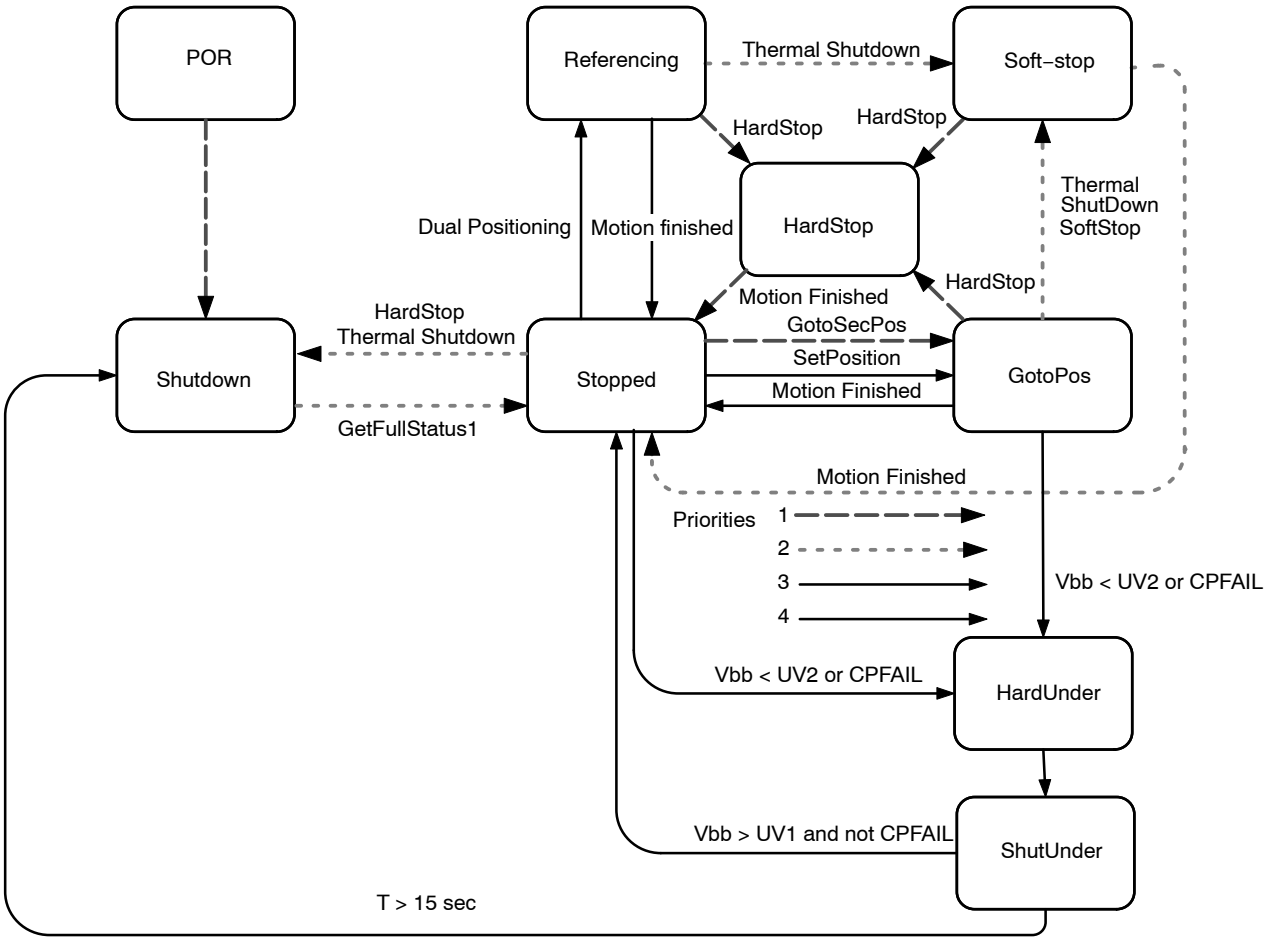


Figure 16. Simplified State Diagram

Motordriver

Current Waveforms in the Coils

Figure 17 below illustrates the current fed to the motor coils by the motordriver in half−step mode.

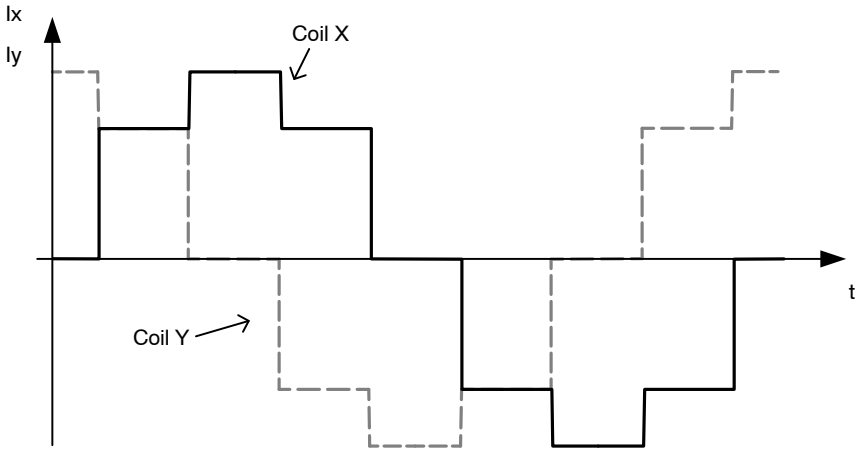


Figure 17. Current Waveforms in Motor Coils X and Y in Halfstep Mode

Whereas Figure 18 below shows the current fed to the coils in 1/16th micro stepping (1 electrical period).

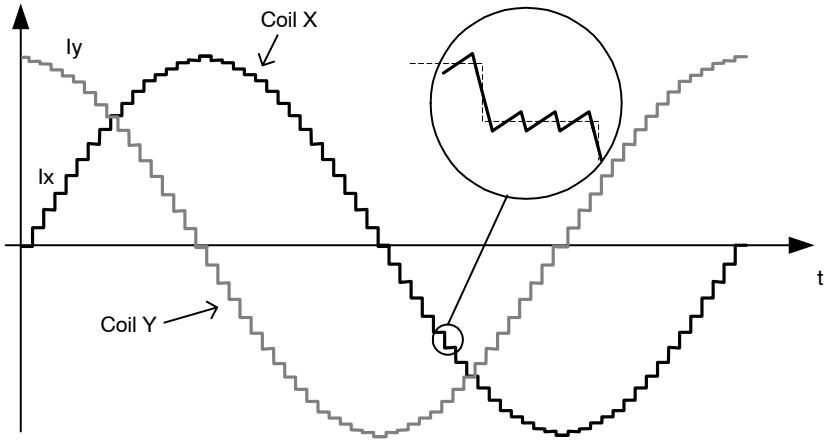


Figure 18. Current Waveforms in Motor Coils X and Y in 1/16th Micro−Step Mode

PWM Regulation

In order to force a given current (determined by <Irun> or <Ihold> and the current position of the rotor) through the motor coil while ensuring high energy transfer efficiency, a regulation based on PWM principle is used. The regulation loop performs a comparison of the sensed output current to an internal reference, and features a digital regulation generating the PWM signal that drives the output switches. The zoom over one micro−step in the Figure 18 above shows how the PWM circuit performs this regulation. To reduce the current ripple, a higher PWM frequency is selectable. The RAM register PWMfreq is used for this.

Table 22. PWM FREQUENCY SELECTION

PWMfreq	Applied PWM Frequency
0	22,8 kHz
1	45,6 kHz

PWM Jitter

To lower the power spectrum for the fundamental and higher harmonics of the PWM frequency, jitter can be added to the PWM clock. The RAM register <PWMJEn> is used for this.

Table 23. PWM JITTER SELECTION

PWMJEn	Status
0	Single PWM frequency
1	Added jitter to PWM frequency

Motor Starting Phase

At motion start, the currents in the coils are directly switched from **<Ihold>** to **<Irun>** with a new sine/cosine ratio corresponding to the first half (or micro−) step of the motion.

Motor Stopping Phase

At the end of the deceleration phase, the currents are maintained in the coils at their actual DC level (hence keeping the sine/cosine ratio between coils) during the stabilization time t_{stab} (see AC Table). The currents are then set to the hold values, respectively **Ihold** x sin(TagPos) and **Ihold** x cos(TagPos), as illustrated below. A new positioning order can then be executed.

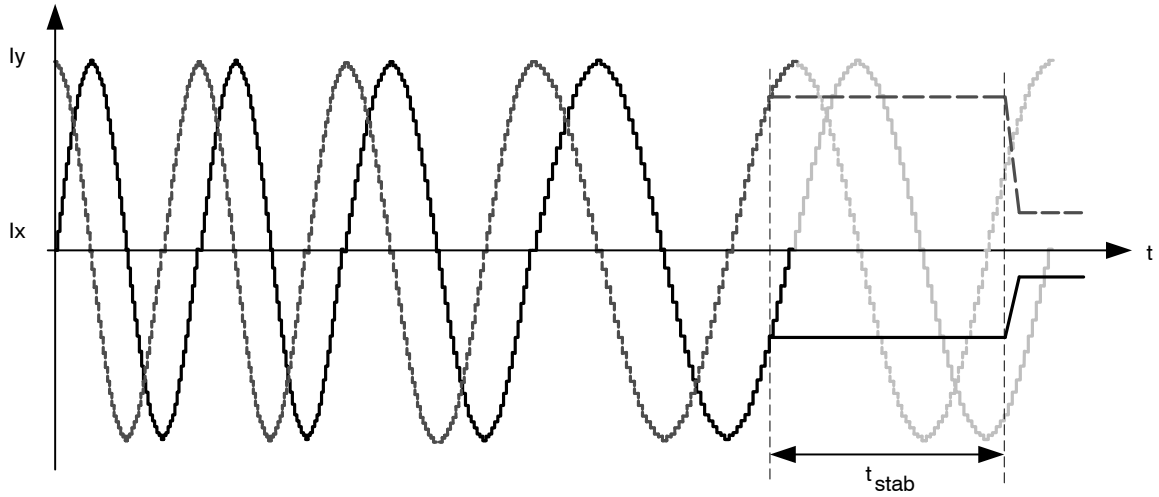


Figure 19. Motor Stopping Phase

Charge Pump Monitoring

If the charge pump voltage is not sufficient for driving the high side transistors (due to failure), an internal HardStop command is issued. This is acknowledged to the master by raising flag <CPFail> (available with command GetFullStatus1).

In case this failure occurs while a motion is ongoing, the flag <StepLoss> is also raised.

Electrical Defect on Coils, Detection and Confirmation

The principle relies on the detection of a voltage drop on at least one transistor of the H−bridge. Then the decision is taken to open the transistors of the defective bridge.

This allows the detection the following short circuits:

- External coil short circuit
- Short between one terminal of the coil and Vbat or GND

One cannot detect an internal short in the motor.

Open circuits are detected by 100% PWM duty cycle value during one electrical period with duration, determined by Vmin.

Table 24. ELECTRICAL DEFECT DETECTION

Pins	Fault Mode
Yi or Xi	Short−circuit to GND
Yi or Xi	Short−circuit to Vbat
Yi or Xi	Open
Y1 and Y2	Short circuited
X1 and X2	Short circuited
Xi and Yi	Short circuited

Motor Shutdown Mode

A motor shutdown occurs when:

- The chip temperature rises above the thermal shutdown threshold Ttsd (see Thermal Shutdown Mode).
- The battery voltage goes below UV2 for longer than 15 seconds (see Battery Voltage Management).
- The charge pump voltage goes below the charge pump comparator level for more than 15 seconds.
- Flag <ElDef> = ‘1’, meaning an electrical problem is detected on one or both coils, e.g. a short circuit.

A motor shutdown leads to the following:

- H−bridges in high impedance mode.
- The <TagPos> register is loaded with the <ActPos>, except in autarkic states.

The conditions to get out of a motor shutdown mode are:

- Reception of a GetFullStatus1 command AND
- The four above causes are no longer detected

This leads to H−bridges going in Ihold mode. Hence, the circuit is ready to execute any positioning command.

This can be illustrated in the following sequence given as an application example. The master can check whether there is a problem or not and decide which application strategy to adopt.

Table 25. Example of Possible Sequence used to Detect and Determine Cause of Motor Shutdown

Tj ≥ Tsd or V_{BB} ≤ UV2 (>15s) or <ElDef> = '1' or <CPFail> = '1' (>15s) ↓	SetPosition frame ↓	GetFullStatus1 frame ↓	GetFullStatus1 frame ↓...
− The circuit is driven in motor shutdown mode − The application is not aware of this	− The position set−point is updated by the I^2C Master − Motor shutdown mode ⇒ no motion − The application is still unaware	− The application is aware of a problem	− Possible confirmation of the problem
		− Reset <TW> or <TSD> or <UV2> or <StepLoss> or <ElDef> or <CPFail> by the application − Possible new detection of over temperature or low voltage or electrical problem ⇒ Circuit sets <TW> or <TSD> or <UV2> or <StepLoss> or <ElDef> or <CPFail> again at '1'	

Important: While in shutdown mode, since there is no hold current in the coils, the mechanical load can cause a step loss, which indeed cannot be flagged by the AMIS−30624/NCV70624.

Note: The Priority Encoder is describing the management of states and commands.

Warning: The application should limit the number of consecutive GetFullStatus1 commands to try to get the AMIS−30624/NCV70624 out of shutdown mode when this proves to be unsuccessful, e.g. there is a permanent defect. The reliability of the circuit could be altered since GetFullStatus1 attempts to disable the protection of the H−bridges.

Motion Detection

Motion detection is based on the back emf generated internally in the running motor. When the motor is blocked, e.g. when it hits the end−stop, the velocity and as a result also the generated back emf, is disturbed. The AMIS−30624/NCV70624 senses the back emf, calculates a moving average and compares the value with two independent threshold levels: Absolute threshold (AbsThr[3:0]) and Delta threshold (<DelThr[3:0]>). Instructions for correct use of these two levels in combination with three additional parameters (<MinSamples>, <FS2StallEn> and <DC100StEn>) are available in a dedicated Application Note "Robust Motion Control with AMIS−3062x Stepper Motor Drivers".

If the motor is accelerated by a pulling or propelling force and the resulting back emf increases above the Delta threshold (+ΔTHR), then <DelStallHi> is set. When the motor is slowing down and the resulting back emf decreases below the Delta threshold (−ΔTHR), then <DelStallLo> is set. When the motor is blocked and the velocity is zero after the acceleration phase, the back emf is low or zero. When this value is below the Absolute threshold, <AbsStall> is set. The <Stall> flag is the OR function of <DelStallLo> OR <DelStallHi> OR <AbsStall>.

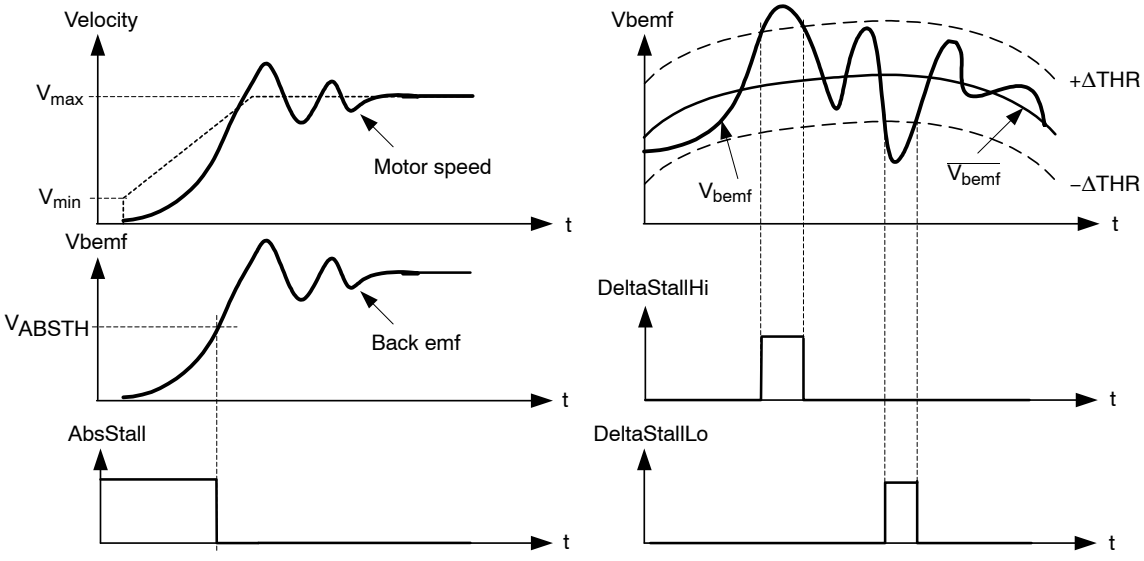


Figure 20. Triggering of the Stall Flags in Function of Measured Backemf and the Set Threshold Levels

Table 26. TRUTH TABLE

Condition	<DelStallLo>	<DelStallHi>	<AbsStall>	<Stall>
Vbemf < Average − DelThr	1	0	0	1
Vbemf > Average + DelThr	0	1	0	1
Vbemf < AbsThr	0	0	1	1

By design, the motion will only be detected when the motor is running at the maximum velocity, not during acceleration or deceleration.

If the motor is positioning when Stall is detected, an (internal) hardstop of the motor is generated and the <StepLoss> and <Stall> flags are set. These flags can only be reset by sending a GetFullStatus1 command.

If Stall appears during DualPosition then the first phase is cancelled (via internal hardstop) and after timeout Tstab (see AC table) the second phase at Vmin starts.

When the <Stall> flag is set the position controller will generate an internal HardStop. As a consequence also the <Steploss> flag will be set. The position in the internal counter will be copied to the <ActPos> register. All flags can be read out with the GetFullStatus1 command.

Important Remark (limited to motion detection flags / parameters):

Using GetFullStatus1 will read **AND** clear the following flags: <Steploss>, <Stall>, <AbsStall>, <DelStallLo> and <DelStallHi>. New positioning is possible and the <ActPos> register will be further updated.

Motion detection is disabled when the RAM registers <AbsThr[3:0]> and <DelThr[3:0]> are zero. Both levels can be programmed using the I^2C command SetStallParam in the registers <AbsThr[3:0]> and <DelThr[3:0]>. Also the OTP register <AbsThr[3:0]> and <DelThr[3:0]> can be set using the I^2C command SetOTPParam. These values are copied in the RAM registers during power on reset.

Table 27.
ABSOLUTE AND DELTA THRESHOLD SETTINGS

AbsThr Index	AbsThr Level (V) (*)		DelThr Index	DelThr Level (V) (*)
0	Disable		0	Disable
1	0.5		1	0.25
2	1.0		2	0.50
3	1.5		3	0.75
4	2.0		4	1.00
5	2.5		5	1.25
6	3.0		6	1.50
7	3.5		7	1.75
8	4.0		8	2.00
9	4.5		9	2.25
A	5.0		A	2.50
B	5.5		B	2.75
C	6.0		C	3.00
D	6.5		D	3.25
E	7.0		E	3.50
F	7.5		F	3.75

(*) Not tested in production. Values are approximations.

MinSamples

<MinSamples[2:0]> is a programmable delay timer. After the zero crossing is detected, the delay counter is started. After the delay time−out (t_{delay}) the back−emf sample is taken. For more information please refer to the Application Note “Robust Motion Control with AMIS−3062x Stepper Motor Drivers”.

Table 28. BACK EMF SAMPLE DELAY TIME

Index	MinSamples[2:0]	t_{DELAY} (µs)
0	000	87
1	001	130
2	010	174
3	011	217
4	100	261
5	101	304
6	110	348
7	111	391

FS2StallEn

If <AbsThr> or <DelThr> <> 0 (i.e. motion detection is enabled), then stall detection will be activated AFTER the acceleration ramp + an additional number of full−steps, according to the following table:

Table 29. ACTIVATION DELAY OF MOTION DETECTION

Index	FS2StallEn[2:0]	Delay (Full Steps)
0	000	0
1	001	1
2	010	2
3	011	3
4	100	4
5	101	5
6	110	6
7	111	7

DC100StEn

When a motor with large bemf is operated at high speed and low supply voltage, then the PWM duty cycle can be as high as 100%. This indicates that the supply is too low to generate the required torque and might also result in erroneously triggering the stall detection. The bit <DC100StEn> enables stall detection when duty cycle is 100%. For more information please refer to the Application Note “Robust Motion Control with AMIS−3062x Stepper Motor Drivers”.

Motion Qualification Mode (*)

This mode is useful to debug motion parameters and to verify the stability of stepper motor systems. The motion qualification mode is entered by means of the I^2C command TestBemf. The SWI pin will be converted into an analogue output on which the Back EMF integrator output can be measured. Once activated, it can only be stopped after a POR. During the Back emf observation, reading of the SWI state is internally forbidden.

(*) Note: Not applicable for product versions NCV70624DW010G, NCV70624DW010R2G.

More information is available in the Application Note “Robust Motion Control with AMIS−3062x Stepper Motor Drivers”.

I²C BUS DESCRIPTION

General Description

AMIS−30624/NCV70624 uses a simple bi−directional 2−wire bus for efficient inter−ic control. This bus is called the Inter IC or I²C−bus.

Features include:

- Only two bus lines are required; a serial data line (SDA) and a serial clock line (SCK).
- Each device connected to the bus is software addressable by a unique address and simple master/slave relationships exists at all times; master can operate as master−transmitter or as master receiver.
- Serial, 8−bit oriented, bi−directional data transfers can be made up to 400 kb/s.
- On−chip filtering rejects spikes on the bus data line to preserve data integrity.
- No need to design bus interfaces because I²C−bus interface is already integrated on−chip.
- IC's can be added to or removed from a system without affecting any other circuits on the bus.

Concept

The I²C−bus consists of two wires, serial data (SDA) and serial clock (SCK), carrying information between the devices connected on the bus. Each device connected to the bus is recognized by a unique address and operates as either a transmitter or receiver, depending on the function of the device. AMIS−30624/NCV70624 can both receive and transmit data. In addition to transmitters and receivers, devices can also be considered as masters or slaves when performing data transfers. AMIS−30624/NCV70624 is a slave device. See Table 31.

Table 30. DEFINITION OF I²C−BUS TERMINOLOGY

Term	Description
Transmitter	The device which sends data on the bus
Receiver	The device which receives data from the bus
Master	The device which initiates a transfer, generates clock signals and terminates a transfer
Slave	The devices addressed by a master
Synchronization	Procedure to synchronizer the clock signals of two or more devices

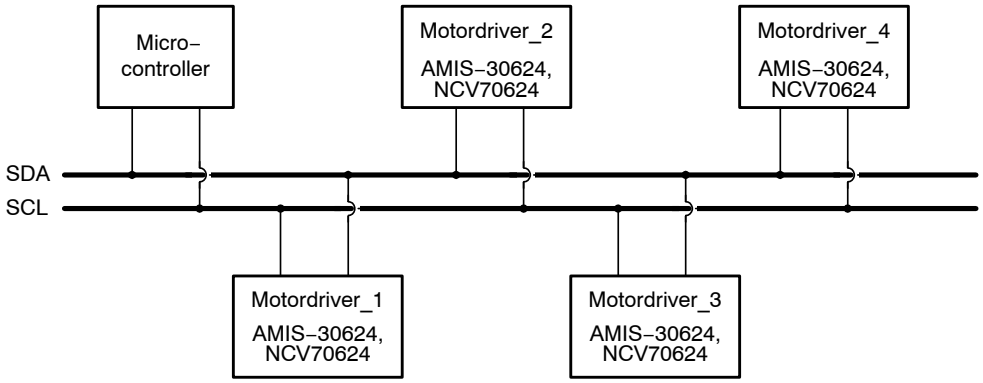


Figure 21. Example of an I²C−bus Configuration Using One Microcontroller and Four Slaves

Figure 21 highlights the master−slave and receiver−transmitter relationships to be found on the I²C−bus. It should be noted that these relationships are not permanent but only depend on the direction of data transfer at that time. The transfer of data would proceed as follows:

1. Suppose the microcontroller wants to send information to motordriver_1:
   - ♦ Microcontroller (master) addresses motordriver_1 (slave)
   - ♦ Microcontroller (master−transmitter) sends data to motordriver_1 (slave−receiver)
   - ♦ Microcontroller terminates the transfer
2. If the microcontroller wants to receive information from motordriver_2:
   - ♦ Microcontroller (master) addresses motordriver_2 (slave)
   - ♦ Microcontroller (master−receiver) receives data from motordriver_2 (slave−transmitter)
   - ♦ Microcontroller terminates the transfer

Even in this case the master generates the timing and terminates the transfer.

Generation of the signals on the I²C−bus is always the responsibility of the master device. It generates its own clock signal when transferring data on the bus. Bus clock signals from a master can only be altered when they are stretched by a slow slave device holding−down the clock line.

General Characteristics

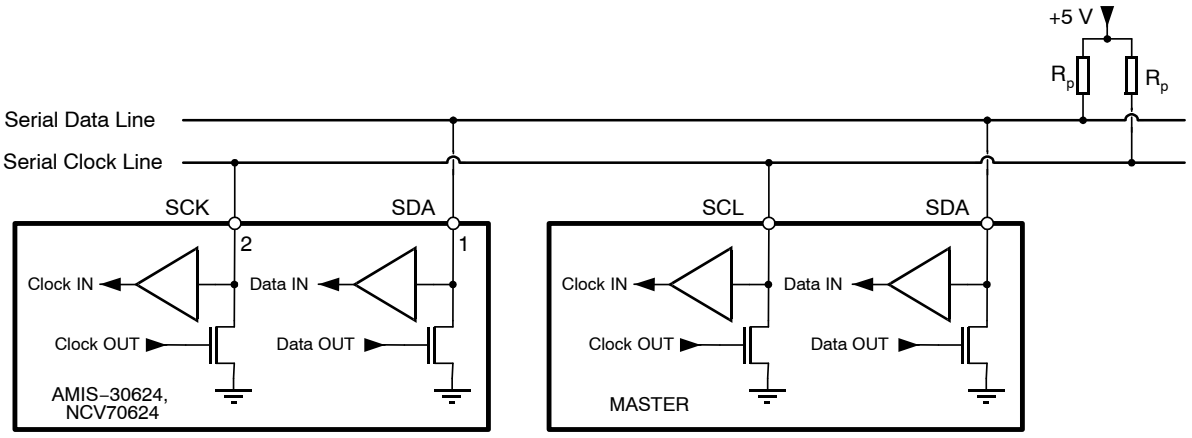


Figure 22. Connection of a Device to the I²C−bus

Both SDA and SCK are bi−directional lines connected to a positive supply voltage via a pull−up resistor (see Figure 22). When the bus is free both lines are HIGH. The output stages of the devices connected to the bus must have an open drain to perform the wired−AND function. Data on the I²C−bus can be transferred up to 400 kb/s in fast mode. The number of interfaces connected to the bus is dependent on the maximum bus capacitance limit (See C_B in Table 6) and the available number of addresses.

Bit Transfer

The levels for logic ‘0’ (LOW) and ‘1’ (HIGH) are not fixed in the I²C standard but dependent on the used V_{DD} level. Using AMIS−30624/NCV70624, the levels are specified in Table 5. One clock pulse is generated for each data bit transferred.

Data Validity

The data on the SDA line must be stable during the HIGH period of the clock. The HIGH or LOW state of the data line can only change when the clock signal on the SCL line is LOW (See Figure 23).

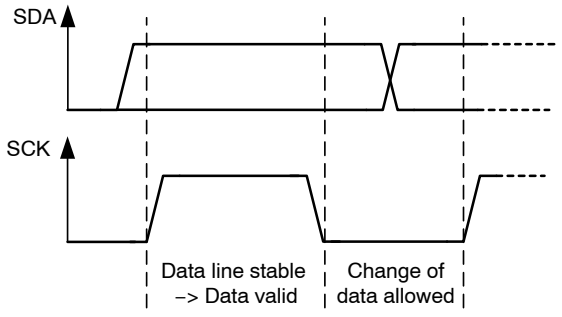


Figure 23. Bit Transfer on the I²C−bus

START and STOP Conditions

Within the procedure of the I²C−bus, unique situations arise, which are defined as START (S) and STOP (P) conditions (See Figure 24).

A HIGH to LOW transition on the SDA line while SCK is HIGH is one such unique case. This situation indicates a START condition. LOW to HIGH transition on the SDA line while SCK is HIGH defines a STOP condition.

START and STOP conditions are always generated by the master. The bus is considered to be busy after the START condition. The bus is considered to be free again a certain time after the STOP condition. The bus free situation is specified as t_{BUF} in Table 6.

The bus stays busy if a repeated START (Sr) is generated instead of a STOP condition. In this respect, the START (S) and repeated START (Sr) conditions are functionally identical (See Figure 25). The symbol S will be used to represent START and repeated START, unless otherwise noted.

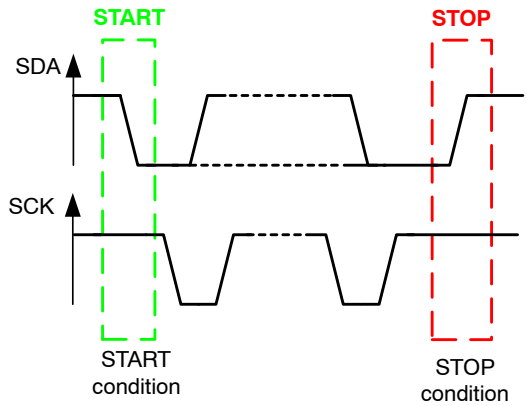


Figure 24. START and STOP Conditions

Transferring Data

Byte Format

Every byte put on the SDA line must be 8−bits long. The number of bytes that can be transmitted per transfer to AMIS−30624/NCV70624 is restricted to eight. Each byte has to be followed by an acknowledge bit. Data is transferred with the most significant bit (MSB) first (See Figure 25). If a slave can't receive or transmit another complete byte of data, it can hold the clock line SCK LOW to force the master into a wait state. Data transfer then continues when the slave is ready for another byte of data and releases clock line SCK.

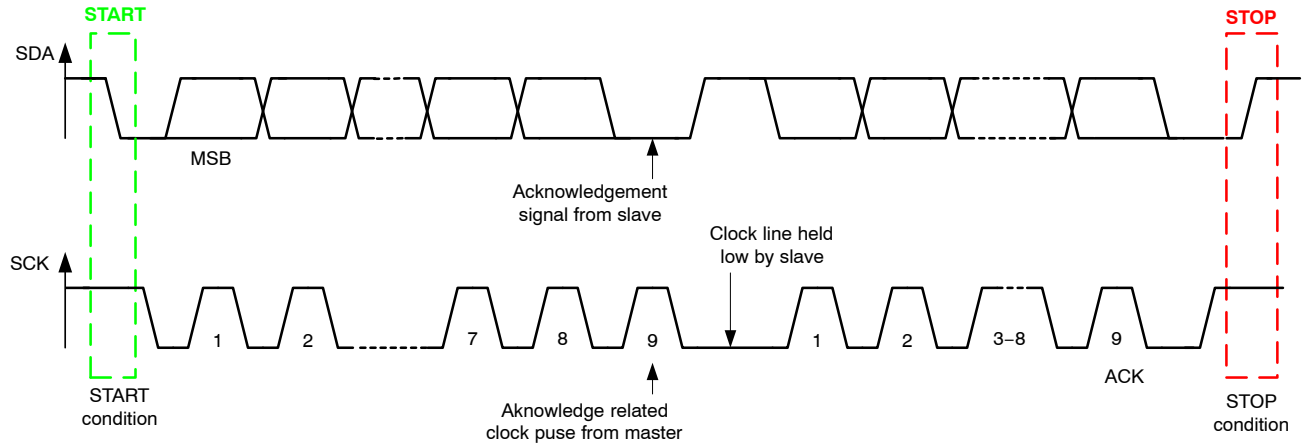


Figure 25. Data Transfer on the I²C−bus

Acknowledge

Data transfer with acknowledge is obligatory. The acknowledge−related clock pulse is generated by the master. The transmitter releases the SDA line (HIGH) during the acknowledge clock pulse. The receiver must pull down the SDA line during the acknowledge clock pulse so that it remains stable LOW during the HIGH period of this clock pulse (see Figure 26). Of course, set−up and hold times must also taken into account (see Table 6). When AMIS−30624/NCV60624 doesn't acknowledge the slave address, the data line will be left HIGH. The master can than generate either a STOP condition to abort the transfer, or a repeated START condition to start a new transfer.

If AMIS−30624/NCV60624 as slave−receiver does acknowledge the slave address but later in the transfer cannot receive any more data bytes, this is indicated by generating a not−acknowledge on the first byte to follow. The master generates than a STOP or a repeated START condition.

If a master−receiver is involved in the transfer, it must signal the end of data to the slave−transmitter by not generating an acknowledge on the last byte that was clocked out of the slave. AMIS−30624/NCV70624 as slave−transmitter shall release the data line to allow the master to generate STOP or repeated START condition.

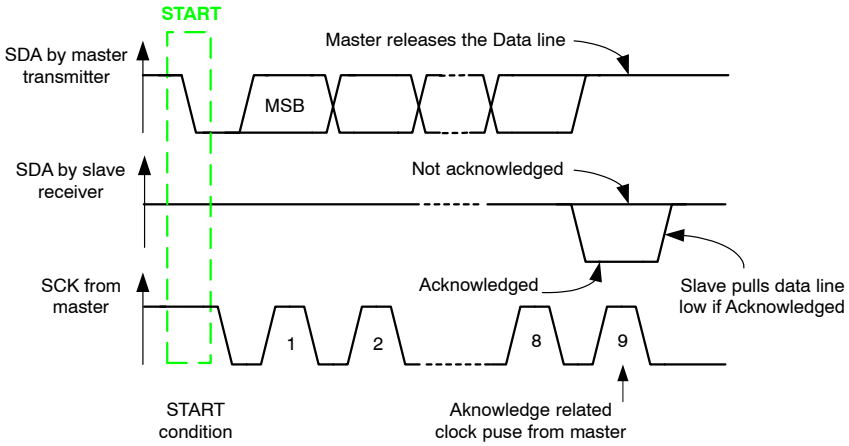


Figure 26. Acknowledge on the I²C−bus

Clock Generation

The master generates the clock on the SCK line to transfer messages on the I^2C−bus. Data is only valid during the HIGH period of the clock.

Data Formats with 7−bit Addresses

Data transfers follow the format shown in Figure 27. After the START condition (S), a slave address is sent. This address is 7−bit long followed by an eighth bit which is a data direction bit (R/W) − a ‘zero’ indicates a transmission (WRITE), a ‘one’ indicates a request for data (READ). A data transfer is always terminated by a STOP condition (P) generated by the master.

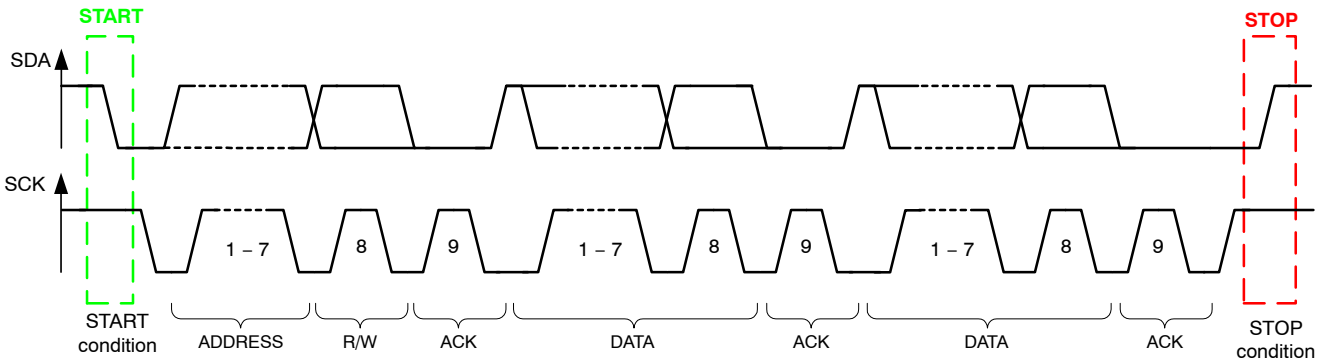


Figure 27. A Complete Data Transfer

However, if a master still wishes to communicate on the bus, it can generate a repeated START (Sr) and address another slave without first generating a STOP condition. Various combinations of read/write formats are then possible within such a transfer.

Data Transfer Formats

Writing Data to AMIS−30624/NCV70624

When writing to AMIS−30624/NCV70624, the master−transmitter transmits to slave−receiver and the transfer direction is not changed. A complete transmission consists of:

- Start condition
- The slave address (7−bit)
- Read/Write bit (‘0’ = write)
- Acknowledge bit
- Any further data bytes are followed by an acknowledge bit. The acknowledge bit is used to signal a correct reception of the data to the transmitter. In this case the AMIS−30624/NCV70624 pulls the SDA line to ‘0’. The AMIS−30624/NCV70624 reads the incoming data at SDA on every rising edge of the SCK signal
- Stop condition to finish the transmission

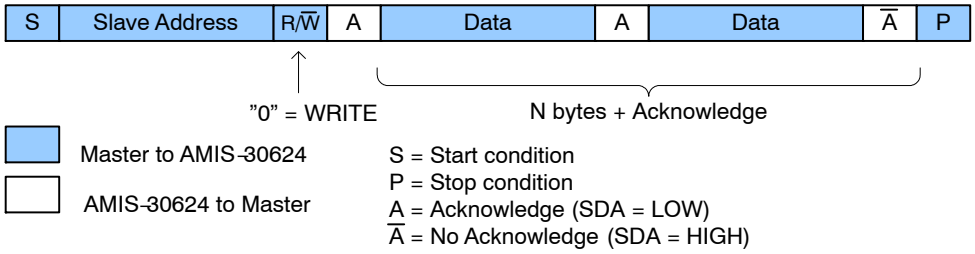


Figure 28. Master Writing Data to AMIS−30624/NCV70624

Some commands for the AMIS−30624/NCV70624 are supporting eight bytes of data, other commands are transmitting two bytes of data. See Table 31.

Reading Data to AMIS−30624/NCV70624

When reading data from AMIS−30624/NCV70624 two transmissions are needed:

1. The first transmission consists of two bytes of data:
   - The first byte contains the slave address and the write bit.
   - The second byte contains the address of an internal register in the AMIS−30624/NCV70624. This internal register address is stored in the circuit RAM.

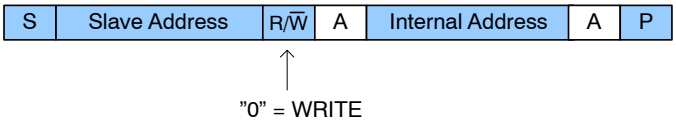


Figure 29. Master Reading Data from AMIS−30624/NCV70624: First Transmission is Addressing

2. The second transmission consists of the slave address and the read bit. Then the master can read the data bits on the SDA line on every rising edge of signal SCK. After each byte of data the master has to acknowledge correct data reception by pulling SDA LOW. The last byte is not acknowledged by the master and therefore the slave knows the end of transmission.

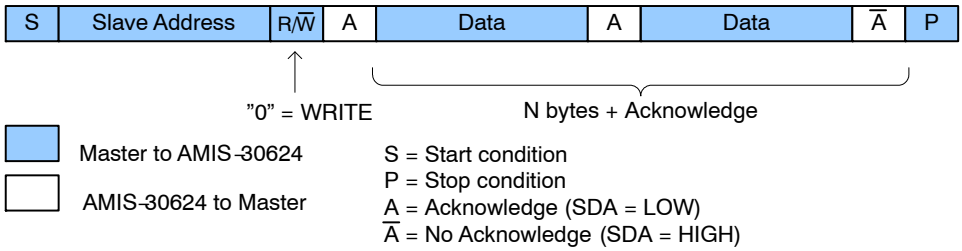


Figure 30. Master Reading Data from AMIS−30624/NCV70624: Second Transmission is Reading Data

Notes:
1. Each byte is followed by an acknowledgment bit as indicated by the A or Ā in the sequence.
2. I²C−bus compatible devices must reset their bus logic on receipt of a START condition such that they all anticipate the sending of a slave address, even if these START conditions are not positioned according to the proper format.
3. A START condition immediately followed by a STOP condition (void message) is an illegal format.

7−bit Addressing

The addressing procedure for the I²C−bus is such that the first byte after the START condition usually determines which slave will be selected by the master. The exception is the general call address which can call all devices. When this address is used all devices should respond with an acknowledge. The second byte of the general call address then defines the action to be taken.

Definition of Bits in the First Byte

The first seven bits of the first byte make up the slave address. The eighth bit is the least significant bit (LSB). It determines the direction of the message. If the LSB is a “zero” it means that the master will write information to a selected slave. A “one” in this position means that the master will read information from the slave. When an address is sent, each device in a system compares the first seven bits after the START condition with its address. If they match, the device considers itself addressed by the master as a slave−receiver or slave−transmitter, depending on the R/W bit.

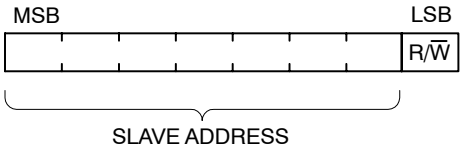


Figure 31. First Byte after START Procedure

AMIS−30624/NCV70624 is provided with a physical address in order to discriminate this circuit from other circuits on the I²C bus. This address is coded on seven bits (two bits being internally hardwired to ‘1’), yielding the theoretical possibility of 32 different circuits on the same bus. It is a combination of four OTP memory bits (OTP Memory Structure OPEN) and of the externally hardwired address bits (pin HW). HW must either be connected to ground or to V_{bat}. When HW is not connected and is left floating, correct functionality of the positioner is not guaranteed. The motor will be driven to the programmed secure position (See Hardwired Address – OPEN).

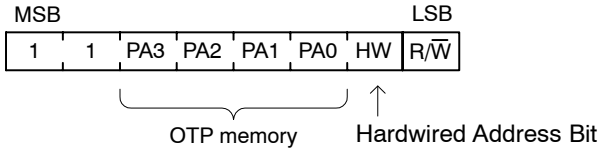


Figure 32. First Byte after START Procedure

General Call Address

The AMIS−30624/NCV70624 supports also a “general call” address “000 0000”, which can address all devices. When this address is used all devices should respond with an acknowledge. The second byte of the general call address then defines the action to be taken.

I²C APPLICATION COMMANDS

Introduction

Communications between the AMIS−30624/NCV70624 and a 2−wire serial bus interface master takes place via a large set of commands.

Reading commands are used to:

- Get actual status information, e.g. error flags
- Get actual position of the stepper motor
- Verify the right programming and configuration of the AMIS−30624/NCV70624.

Writing commands are used to:

- Program the OTP memory
- Configure the positioner with motion parameters (max/min speed, acceleration, stepping mode, etc.)
- Provide target positions to the Stepper motor

The I²C−bus master will have to use commands to manage the different application tasks the AMIS−30624/NCV70624 can feature. The commands summary is given in Table 31.

Commands Table

Table 31. I²C COMMANDS WITH CORRESPONDING ROM POINTER

Command Mnemonic	Function	Command Byte	
		Binary	Hexadecimal
GetFullStatus1	Returns complete status of the chip	“1000 0001”	0x81
GetFullStatus2	Returns actual, target and secure position	“1111 1100”	0xFC
GetOTPParam	Returns OTP parameter	“1000 0010”	0x82
GotoSecurePosition	Drives motor to secure position	“1000 0100”	0x84
HardStop	Immediate full stop	“1000 0101”	0x85
ResetPosition	Sets actual position to zero	“1000 0110”	0x86
ResetToDefault	Overwrites the chip RAM with OTP contents	“1000 0111”	0x87
SetDualPosition	Drives the motor to two different positions with different speed	“1000 1000”	0x88
SetMotorParam	Sets motor parameter	“1000 1001”	0x89
SetOTP	Zaps the OTP memory	“1001 0000”	0x90
SetPosition	Programs a target and secure position	“1000 1011”	0x8B
SetStallParam	Sets stall parameters	“1001 0110”	0x96
SoftStop	Motor stopping with deceleration phase	“1000 1111”	0x8F
Runvelocity	Drives motor continuously	“1001 0111”	0x97
TestBemf	Outputs Bemf voltage on pin SWI	“1001 1111”	0x9F

These commands are described hereafter, with their corresponding I²C frames. Refer to Data Transfer Formats for more details. A color coding is used to distinguish between master and slave parts within the frames. An example is shown below.

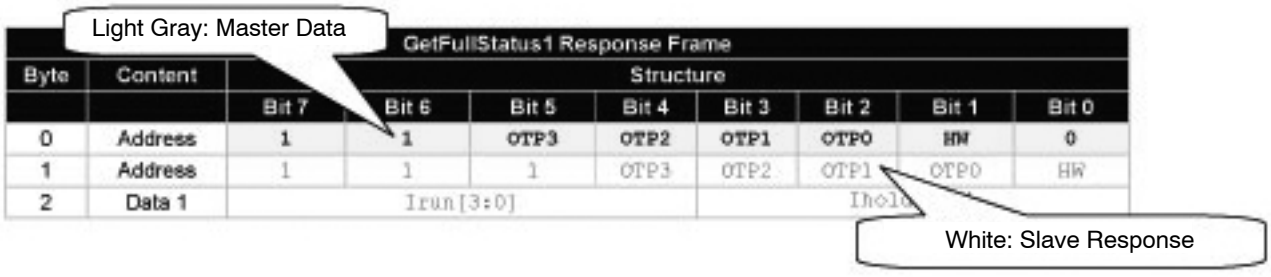


Figure 33. Color Code Used in the Definition of I²C Frames

Application Commands

GetFullStatus1

This command is provided to the circuit by the master to get a complete status of the circuit and of the stepper motor. Refer to Tables 19 and 20 to see the meaning of the parameters sent back to the I^2C master.

Note: A `GetFullStatus1` command will attempt to reset flags `<TW>`, `<TSD>`, `<UV2>`, `<ElDef>`, `<StepLoss>`, `<CPFail>`, `<OVC1>`, `<OVC2>`, and `<VddReset>`.

GetFullStatus1 corresponds to the following I^2C command frame:

Table 32. GetFullStatus1 COMMAND FRAME

Byte	Content	Structure							
		Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	Address	**1**	**1**	**OTP3**	**OTP2**	**OTP1**	**OTP0**	**HW**	**0**
1	Command	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**1**

Table 33. GetFullStatus1 RESPONSE FRAME

Byte	Content	Structure							
		Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	Address	**1**	**1**	**OTP3**	**OTP2**	**OTP1**	**OTP0**	**HW**	**1**
1	Address	1	1	1	OTP3	OTP2	OTP1	OTP0	HW
2	Data 1	Irun[3:0]				Ihold[3:0]			
3	Data 2	Vmax[3:0]				Vmin[3:0]			
4	Data 3	AccShape	StepMode[1:0]		Shaft	Acc[3:0]			
5	Data 4	VddReset	StepLoss	ElDef	UV2	TSD	TW	Tinfo[1:0]	
6	Data 5	Motion[2:0]			ESW	OVC1	OVC2	Stall	CPFail
7	Data 6	1	1	1	1	1	1	1	1
8	Data 7	AbsThr[3:0]				DelThr[3:0]			

Where:

OTP(n)	OTP address bits PA[3:0]
HW	Hardwired address bit
Irun[3:0]	Operating current in the motor coil
Ihold[3:0]	Standstill current in the motor coil
Vmax[3:0]	Maximum velocity
Vmin[3:0]	Minimum velocity
AccShape	Enables motion without acceleration
StepMode[1:0]	Step mode definition
Shaft	Direction of movement
Acc[3:0]	Acceleration form minimum to maximum velocity
VddReset	Reset of digital supply
StepLoss	Step loss occurred
ElDef	Electrical defect
UV2	Battery under voltage detected
TSD	Thermal shutdown
TW	Thermal warning
Tinfo[1:0]	Temperature Info
Motion[2:0]	Motion status
ESW	External switch status
OVC1	Over current in X−coil detected
OVC2	Over current in Y−coil detected
Stall	Stall detected
CPFail	Charge pump failure
AbsThr[3:0]	Stall detection absolute threshold
DelThr[3:0]	Stall detection delta threshold

GetFullStatus2

This command is provided to the circuit by the master to get the actual, target and secure position of the stepper motor. Both the actual and target position are returned in signed two's complement 16−bit format. Secure position is coded in 10−bit format. According to the programmed stepping mode the LSBs of `ActPos[15:0]` and `TagPos[15:0]` may have no meaning and should be assumed to be '0'. This command also gives additional information concerning stall detection. Refer to Tables 19 and 20 to see the meaning of the parameters sent back to the I^2C master.

GetFullStatus2 corresponds to the following I2C command frame:

Table 34. GetFullStatus2 COMMAND FRAME

Byte	Content	Structure							
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Address	1	1	OTP3	OTP2	OTP1	OTP0	HW	0
1	Command	1	1	1	1	1	1	0	0

Table 35. GetFullStatus2 RESPONSE FRAME

Byte	Content	Structure							
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Address	1	1	OTP3	OTP2	OTP1	OTP0	HW	1
1	Address	1	1	1	OTP3	OTP2	OTP1	OTP0	HW
2	Data 1	ActPos[15:8]							
3	Data 2	ActPos[7:0]							
4	Data 3	TagPos[15:8]							
5	Data 4	TagPos[7:0]							
6	Data 5	SecPos[7:0]							
7	Data 6	FS2StallEn[2:0]			1	DC100	SecPos[10:8]		
8	Data 7	AbsStall	DelStallLo	DelStallHi	MinSamples[2:0]			DC100StEn	PWMJEn

Where:

OTP(n)	OTP address bits PA[3:0]
HW	Hardwired address bit
ActPos[15:0]	Actual position
TagPos[15:0]	Target position
SecPos[10:0]	Secure position
FS2StallEn[2:0]	Number of full steps after stall detection is enabled
DC100	Flag indicating PWM is at 100 percent duty cycle
AbsStall	Stall detected because the absolute threshold is not reached
DelStallLo	Stall detected because the delta threshold is under crossed
DelStallHi:	Stall detected because the delta threshold is crossed
MinSamples[2:0]	Back−emf sampling delay time
DC100StEn	Enables the switch off of stall detection when DC100 = 1
PWMJEn	PWM jitter enable

GetOTPParam

This command is provided to the circuit by the I^{2}C master to read the content of the OTP memory. More information can be found in OTP Memory Structure corresponds to the following I^{2}C command frame:.

GetOTPParam

Table 36. GetOTPParam COMMAND FRAME

Byte	Content	Structure							
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Address	1	1	OTP3	OTP2	OTP1	OTP0	HW	0
1	Command	1	0	0	0	0	0	1	0

Table 37. GetOTPParam RESPONSE FRAME

Byte	Content	Structure							
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Address	1	1	OTP3	OTP2	OTP1	OTP0	HW	1
1	OTP byte 0	OTP byte @0x00							
2	OTP byte 1	OTP byte @0x01							
3	OTP byte 2	OTP byte @0x02							
4	OTP byte 3	OTP byte @0x03							
5	OTP byte 4	OTP byte @0x04							
6	OTP byte 5	OTP byte @0x05							
7	OTP byte 6	OTP byte @0x06							
8	OTP byte 7	OTP byte @0x07							

GotoSecurePosition

This command is provided by the I^{2}C master to one or all the stepper motors to move to the secure position `SecPos[10:0]`. See the priority encoder corresponds to the following I^{2}C command frame: description for more details. The priority encoder table also acknowledges the cases where a `GotoSecurePosition` command will be ignored.

GotoSecurePosition

Table 38. GotoSecurePosition COMMAND FRAME

Byte	Content	Structure							
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Address	1	1	OTP3	OTP2	OTP1	OTP0	HW	0
1	Command	1	0	0	0	0	1	0	0

HardStop

This command will be internally triggered when an electrical problem is detected in one or both coils, leading to shutdown mode. If this occurs while the motor is moving, the <StepLoss> flag is raised to allow warning of the I²C master at the next GetStatus1 command that steps may have been lost. Once the motor is stopped, ActPos register is copied into TagPos register to ensure keeping the stop position. The I²C master for some safety reasons can also issue a HardStop command.

HardStop corresponds to the following I²C command frame:

Table 39. HardStop COMMAND FRAME

Byte	Content	Structure							
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Address	1	1	OTP3	OTP2	OTP1	OTP0	HW	0
1	Command	1	0	0	0	0	1	0	1

ResetPosition

This command is provided to the circuit by the I²C master to reset ActPos and TagPos registers to zero. This can be helpful to prepare for instance a relative positioning.

ResetPosition corresponds to the following I²C command frame:

Table 40. ResetPosition COMMAND FRAME

Byte	Content	Structure							
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Address	1	1	OTP3	OTP2	OTP1	OTP0	HW	0
1	Command	1	0	0	0	0	1	1	0

ResetToDefault

This command is provided to the circuit by the I²C master in order to reset the whole slave node into the initial state. ResetToDefault will, for instance, overwrite the RAM with the reset state of the registers parameters (see Table 19). This is another way for the I²C master to initialize a slave node in case of emergency, or simply to refresh the RAM content.

Note: ActPos and TagPos are not modified by a ResetToDefault command.

Important: Care should be taken not to send a ResetToDefault command while a motion is ongoing, since this could modify the motion parameters in a way forbidden by the position controller.

ResetToDefault corresponds to the following I²C command frame:

Table 41. ResetToDefault COMMAND FRAME

Byte	Content	Structure							
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Address	1	1	OTP3	OTP2	OTP1	OTP0	HW	0
1	Command	1	0	0	0	0	1	1	1

RunVelocity

This command is provided to the circuit by the I²C master in order to put the motor in continuous motion state.

RunVelocity corresponds to the following I²C command frame:

Table 42. RunVelocity COMMAND FRAME

Byte	Content	Structure							
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Address	1	1	OTP3	OTP2	OTP1	OTP0	HW	0
1	Command	1	0	0	1	0	1	1	1

SetDualPosition

This command is provided to the circuit by the I²C master in order to perform a positioning of the motor using two different velocities. See Section Dual Positioning.

Note: This sequence cannot be interrupted by another positioning command.

Important: If for some reason `ActPos` equals `Pos1[15:0]` at the moment the `SetDualPosition` command is issued, the circuit will enter in deadlock state. Therefore, the application should check the actual position by a `GetFullStatus2` corresponds to the following I²C command frame command prior to starting a dual positioning. Another solution may consist of programming a value out of the stepper motor range for `Pos1[15:0]`. For the same reason `Pos2[15:0]` should not be equal to `Pos1[15:0].`

SetDualPosition

Table 43. SetDualPosition COMMAND FRAME

Byte	Content	Structure							
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Address	**1**	**1**	**OTP3**	**OTP2**	**OTP1**	**OTP0**	**HW**	**0**
1	Command	**1**	**0**	**0**	**0**	**1**	**0**	**0**	**0**
2	Data 1	1	1	1	1	1	1	1	1
3	Data 2	1	1	1	1	1	1	1	1
4	Data 3	Vmax[3:0]				Vmin[3:0]			
5	Data 4	Pos1[15:8]							
6	Data 5	Pos1[7:0]							
7	Data 6	Pos2[15:8]							
8	Data 7	Pos2[7:0]							

Where:

Vmax[3:0]	Max. velocity for first motion
Vmin[3:0]	Min. velocity for first motion and velocity for the second motion
Pos1[15:0]	First position to be reached during the first motion
Pos2[15:0]	Relative position of the second motion

SetStallParam

This command sets the motion detection parameters and the related stepper motor parameters, such as the minimum and maximum velocity, the run− and hold current, acceleration and step−mode. See Motion Detection corresponds to the following I²C command frame for the meaning of these parameters.

SetStallParam

Table 44. SetStallParam COMMAND FRAME

Byte	Content	Structure							
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Address	1	1	OTP3	OTP2	OTP1	OTP0	HW	0
1	Command	1	0	0	1	0	1	1	0
2	Data 1	1	1	1	1	1	1	1	1
3	Data 2	1	1	1	1	1	1	1	1
4	Data 3	Irun[3:0]				Ihold[3:0]			
5	Data 4	Vmax[3:0]				Vmin[3:0]			
6	Data 5	MinSamples[2:0]			Shaft	Acc[3:0]			
7	Data 6	AbsThr[3:0]				DelThr[3:0]			
8	Data 7	FS2StallEn[2:0]			AccShape	StepMode[1:0]		DC100StEn	PWMJEn

SetMotorParam

This command is provided to the circuit by the I²C master to set the values for the stepper motor parameters (listed below) in RAM. Refer to Table 19 to see the meaning of the parameters sent by the I²C master.

Important: If a `SetMotorParam` occurs while a motion is ongoing, it will modify at once the motion parameters (see Position Controller corresponds to the following I²C command frame:). Therefore the application should not change parameters other than `Vmax` while a motion is running, otherwise correct positioning cannot be guaranteed.

SetMotorParam

Table 45. SetMotorParam COMMAND FRAME

Byte	Content	Structure							
		Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	Address	**1**	**1**	**OTP3**	**OTP2**	**OTP1**	**OTP0**	**HW**	**0**
1	Command	**1**	**0**	**0**	**0**	**1**	**0**	**0**	**1**
2	Data 1	1	1	1	1	1	1	1	1
3	Data 2	1	1	1	1	1	1	1	1
4	Data 3	Irun[3:0]				Ihold[3:0]			
5	Data 4	Vmax[3:0]				Vmin[3:0]			
6	Data 5	SecPos[10:8]			Shaft	Acc[3:0]			
7	Data 6	SecPos[7:0]							
8	Data 7	1	PWMfreq	1	AccShape	StepMode[1:0]		1	PWMJEn

SetOTPParam

This command is provided to the circuit by the I²C master to program and zap the OTP data `D[7:0]` in OTP address `OTPA[2:0]`.

Important: This command must be sent under a specific V_{BB} voltage value. See parameter V_{BB}OTP in Table 5. This is a mandatory condition to ensure reliable zapping.

SetOTPParam corresponds to the following I²C command frame:

Table 46. SetOTPParam COMMAND FRAME

Byte	Content	Structure							
		Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	Address	**1**	**1**	**OTP3**	**OTP2**	**OTP1**	**OTP0**	**HW**	**0**
1	Command	**1**	**0**	**0**	**1**	**0**	**0**	**0**	**0**
2	Data 1	1	1	1	1	1	1	1	1
3	Data 2	1	1	1	1	1	1	1	1
4	Data 3	1	1	1	1	1	OTPA[2:0]		
5	Data 4	D[7:0]							

Where:
OTPA[2:0]: OTP address
D[7:0]: Corresponding OTP data

SetPosition

This command is provided to the circuit by the I²C master to drive the motor to a given absolute position. See Positioning (see Priority Encoder) for more details. The priority encoder table acknowledges the cases where a `SetPosition` command will be ignored.

SetPosition corresponds to the following I²C command frame:

Table 47. SetPosition COMMAND FRAME

Byte	Content	Structure							
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Address	1	1	OTP3	OTP2	OTP1	OTP0	HW	0
1	Command	1	0	0	0	1	0	1	1
2	Data 1	1	1	1	1	1	1	1	1
3	Data 2	1	1	1	1	1	1	1	1
4	Data 3	Pos[15:8]							
5	Data 4	Pos[7:0]							

Where:

Pos [15:0] Signed 16−bit position set−point for motor.

SoftStop

This command will be internally triggered when the chip temperature rises above the thermal shutdown threshold (see Table 5 and the Temperature Management Section). It provokes an immediate deceleration to `Vmin` (see Minimum Velocity corresponds to the following I²C command frame:) followed by a stop, regardless of the position reached. Once the motor is stopped, `TagPos` register is overwritten with value in `ActPos` register to ensure keeping the stop position. The I²C Master for some safety reasons can also issue a SoftStop command.

SoftStop

Table 48. SoftStop COMMAND FRAME

Byte	Content	Structure							
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Address	**1**	**1**	**OTP3**	**OTP2**	**OTP1**	**OTP0**	**HW**	**0**
1	Command	**1**	**0**	**0**	**0**	**1**	**1**	**1**	**1**

TestBemf

This command is provided to the circuit by the I²C master in order to output the Bemf integrator output to the SWI output of the chip. Once activated, it can be stopped only after POR. During the Bemf observation, reading of the SWI state is internally forbidden.

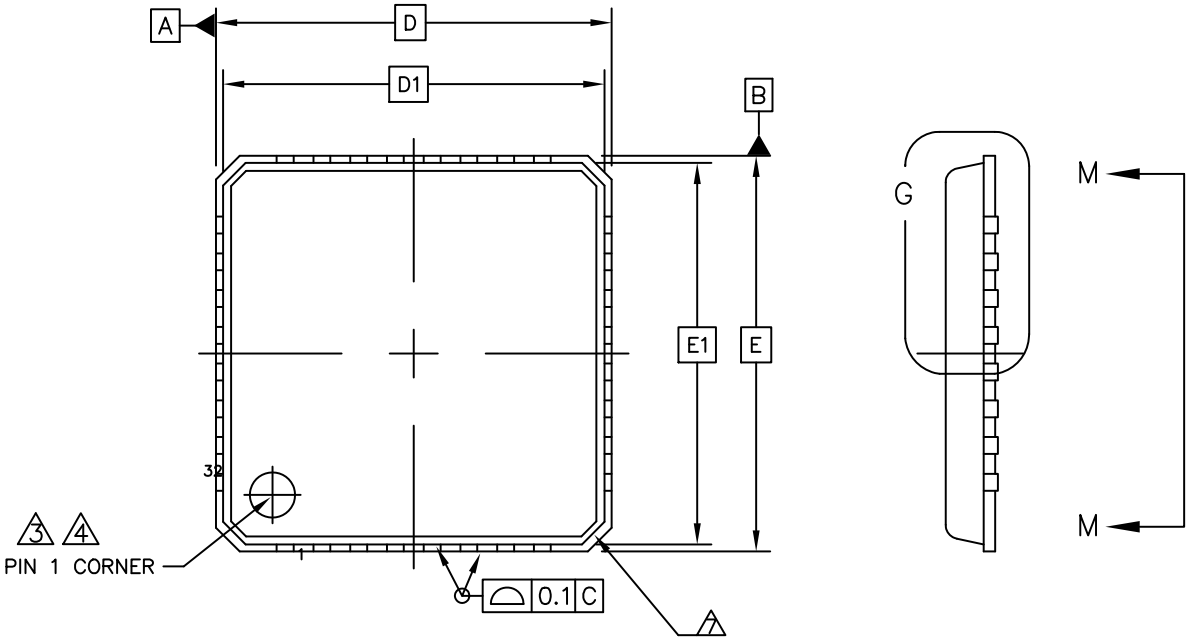
TestBemf corresponds to the following I²C command frame:

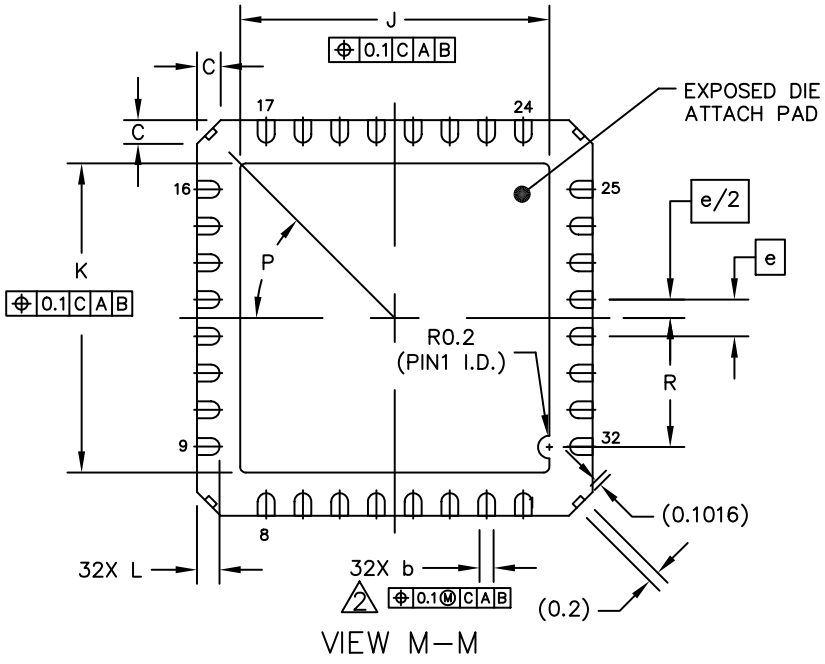
Table 49. TestBemf COMMAND FRAME

Byte	Content	Structure							
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Address	**1**	**1**	**OTP3**	**OTP2**	**OTP1**	**OTP0**	**HW**	**0**
1	Command	**1**	**0**	**0**	**1**	**1**	**1**	**1**	**1**

QFN32, 7x7
CASE 560AA
ISSUE A

DATE 23 SEP 2015



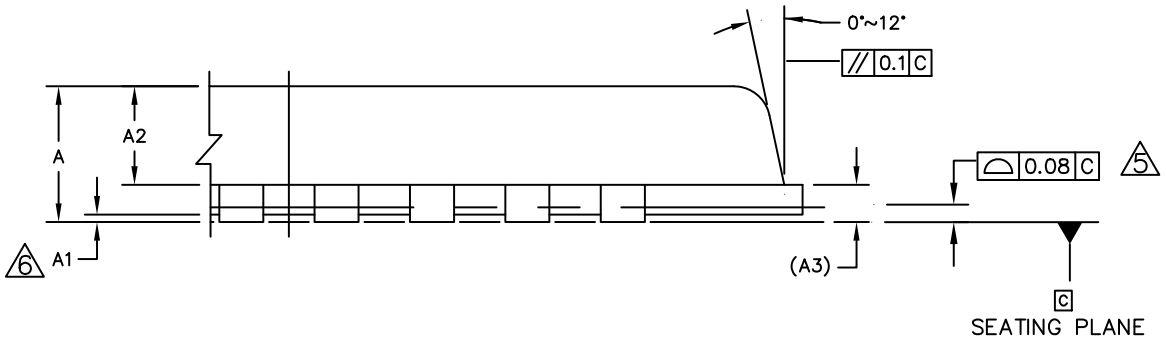


VIEW M–M

DOCUMENT NUMBER:	**98AON30885E**	Electronic versions are uncontrolled except when accessed directly from the Document Repository. Printed versions are uncontrolled except when stamped "CONTROLLED COPY" in red.	
DESCRIPTION:	**QFN32, 7X7**		**PAGE 1 OF 2**

QFN32, 7x7
CASE 560AA
ISSUE A

DATE 23 SEP 2015



DETAIL G
VIEW ROTATED 90° CLOCKWISE

DIM	MIN	NOM	MAX
A	0.8		0.9
A1	0	0.02	0.05
A2	0.576	0.615	0.654
A3		0.203 REF.	
b	0.25	0.3	0.35
C	0.24	0.42	0.6
D		7 BSC	
D1		6.75 BSC	
E		7 BSC	
E1		6.75 BSC	
e		0.65 BSC	
J	5.37	5.47	5.57
K	5.37	5.47	5.57
L	0.35	0.4	0.45
P		45° REF	
R	2.185		2.385

NOTES

1. DIE THICKNESS ALLLOWABLE IS 0.305mm MAXIMUM (.012 INCHES MAXIMUM)

2. DIMENSION APPLIES TO PLATED TERMINAL AND IS MEASURED BETWEEN 0.2 AND 0.25mm FROM TERMINAL TIP.

3. THE PIN #1 IDENTIFIER MUST BE PLACED ON THE TOP SURFACE OF THE PACKAGE BY USING INDENTATION MARK OR OTHER FEATURE OF PACKAGE BODY.

4. EXACT SHAPE AND SIZE OF THIS FEATURE IS OPTIONAL.

5. APPLIED FOR EXPOSED PAD AND TERMINALS. EXCLUDE EMBEDDING PART OF EXPOSED PAD FROM MEASURING.

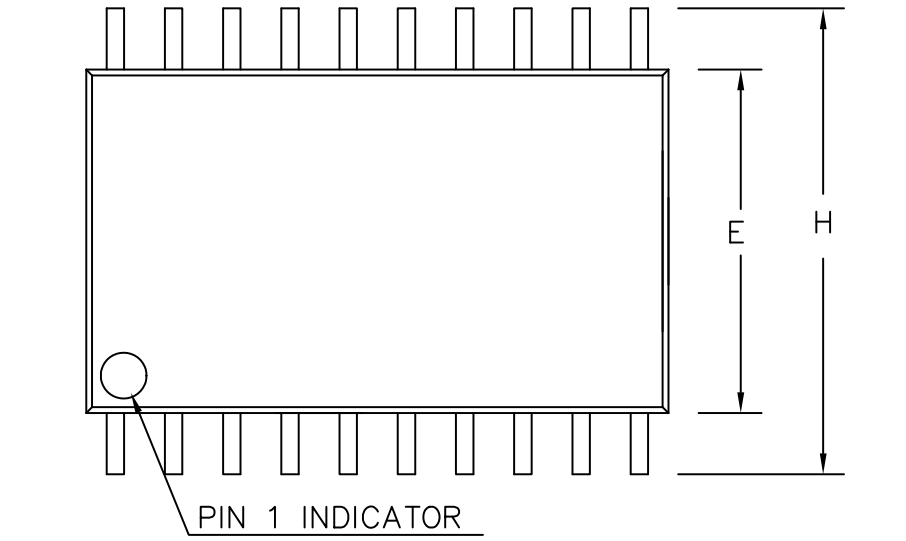
6. APPLIED ONLY TO TERMINALS.

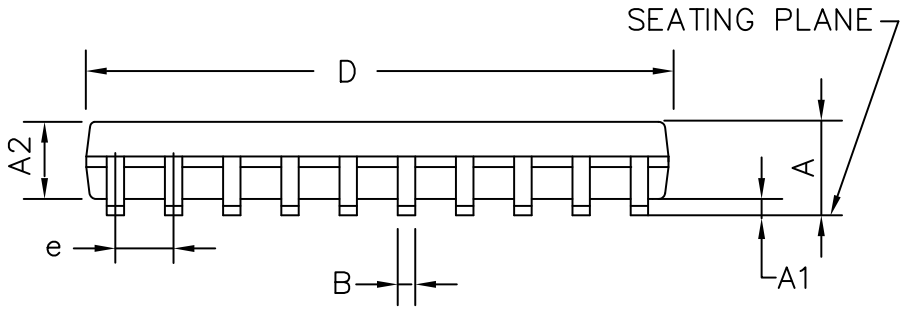
7. EXACT SHAPE OF EACH CORNER IS OPTIONAL.

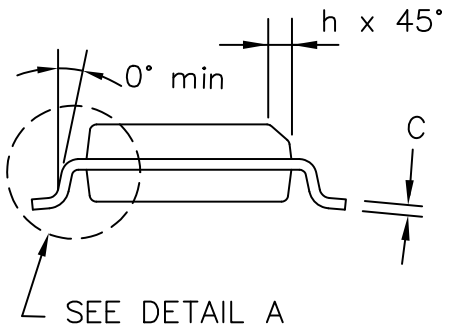
UNIT	DIMENSION AND TOLERANCES	REFERENCE DOCUMENT
MM	ASME_Y14.5M	JEDEC–MO–220_REV.F

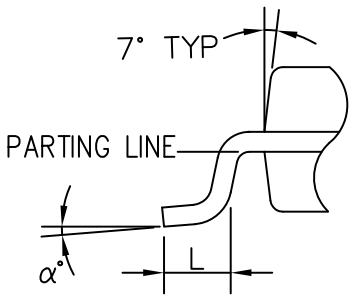
SOIC 20 W
CASE 751AQ−01
ISSUE O

DATE 19 JUN 2008









DETAIL A

DIMENSIONS IN INCHES			
SYMBOL	MIN.	NOM.	MAX.
A	0.093	0.099	0.104
A1	0.004	0.008	0.012
A2	0.088	0.094	0.100
B	0.013	0.016	0.020
C	0.0090	0.0100	0.0125
D	0.496	0.503	0.510
E	0.292	0.296	0.299
e	.050 BSC.		
H	0.394	0.402	0.419
h	0.010	0.015	0.019
L	0.016	0.033	0.050
α	0°	5°	8°

DOCUMENT NUMBER:	**98AON30891E**	Electronic versions are uncontrolled except when accessed directly from the Document Repository. Printed versions are uncontrolled except when stamped "CONTROLLED COPY" in red.
DESCRIPTION:	**SOIC 20 W**	**PAGE 1 OF 1**

PUBLICATION ORDERING INFORMATION

LITERATURE FULFILLMENT:
Email Requests to: orderlit@onsemi.com

onsemi Website: www.onsemi.com

TECHNICAL SUPPORT
North American Technical Support:
Voice Mail: 1 800−282−9855 Toll Free USA/Canada
Phone: 011 421 33 790 2910

Europe, Middle East and Africa Technical Support:
Phone: 00421 33 790 2910
For additional information, please contact your local Sales Representative

◊

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

onsemi:

AMIS30624C6244RG AMIS30624C6245RG AMIS30624C6245G AMIS30624C6244G