AMIS-30622 I^{2}C Microstepping Motordriver

Data Sheet

General Description

The AMIS-30622 is a member of a stepper motordriver family with position controller and control/diagnostics interface integrated in one single chip.
The family consists of two products:

- AMIS-30621 with LIN interface, ready to build dedicated mechatronics solutions connected remotely with a LIN master.
- AMIS-30622 with SERIAL interface, ready to act as peripheral device next to a microcontroller.

The chip receives high-level positioning instructions through the interface and subsequently drives the motor coils until the desired position is reached. The on-chip position controller is configurable (OTP and Interface) for different motor types, positioning ranges and parameters for speed, acceleration and deceleration.
The AMIS-30622 acts as a slave on the bus and the master can fetch specific status information like actual position, error flags, etc. from each individual slave node.

Features

Motordriver

- Microstepping (1/2, 1/4, 1/8, 1/16)
- Low resonance & noise
- High resolution
- Programmable peak current up to 800mA
- 20kHz PWM current-control
- Automatic selection of fast & slow decay mode
- Internal fly-back FETs
- Fully integrated current sense
- 8V-29V supply voltage
- Automotive compliant
- Full diagnostics and status information

Controller with RAM and OTP memory

- Position controller
- Configurable speeds, acceleration and deceleration
- Flexible hold-current
- Movement/position sensor-input
- Optional stall detection

Serial interface

- 2-wire serial interface
- 5V microcontroller compatible
- Up to 32 node addresses
- 5V regulator with wake-up on LIN activity

Protection

- Over-current protection
- Under-voltage management
- Over-voltage protection
- High-temp warning and shutdown
- Low-temp warning
- LIN bus short-circuit protection to supply & ground

Power Saving

- Power-down supply current <50µA
- 5V regulator with wake-up on LIN activity

EMI compatibility

- Power drivers with slope control

Applications and Benefits

The AMIS-30622 is ideally suited for small positioning applications. Target markets include: automotive (headlamp alignment, HVAC, idle control, cruise), industrial equipment (lighting, fluid control, labeling, process, XYZ tables) and building automation (HVAC, surveillance, satellite dish positioning). Suitable applications typically have multiple axes or require mechatronic solutions with the driver chip mounted directly on the motor.

The high abstraction level of the products’ command set reduces the load of the processor on the master side. Scaling of the application towards number of axes is straight-forward: hardware - and software designs are extended in a modular way, without severely effecting the demands on the master microcontroller. The bus structure simplifies PCB track-layout and/or wiring architectures.

Microstepping operation removes the design trade-off between minimal operation speed and avoiding the risk of noise and step-loss due to resonance phenomena. The stall-detection feature (optional) offers silent, yet accurate position-calibrations during the referencing run and allows semi-closed loop operation when approaching the mechanical end-stops.

All these benefits result in reduced system-cost and time-to-market and improved technical performance.

Ordering Information

Part N°	**AMIS-30622**
Package	**SOIC-20**
Peak Current	**800mA**
Temp. Range	**-40°C...125°C**

Table of Contents

Document History

Version	Date of Version
2.0	March 4th, 2003
2.1	April 2nd, 2003
2.2	April 3rd, 2003
2.3	May 28th, 2003

1.0 Quick Reference Data

1.1. Absolute Maximum Ratings

Parameter		Min	Max	Unit
Vbb	Supply voltage	-0.3	+40[1]	V
Tamb	Ambient temperature under bias [2]	-50	+150	°C
Tst	Storage temperature	-55	+160	°C
Vesd	Electrostatic discharge voltage on all pins [3] -2	+2	kV	

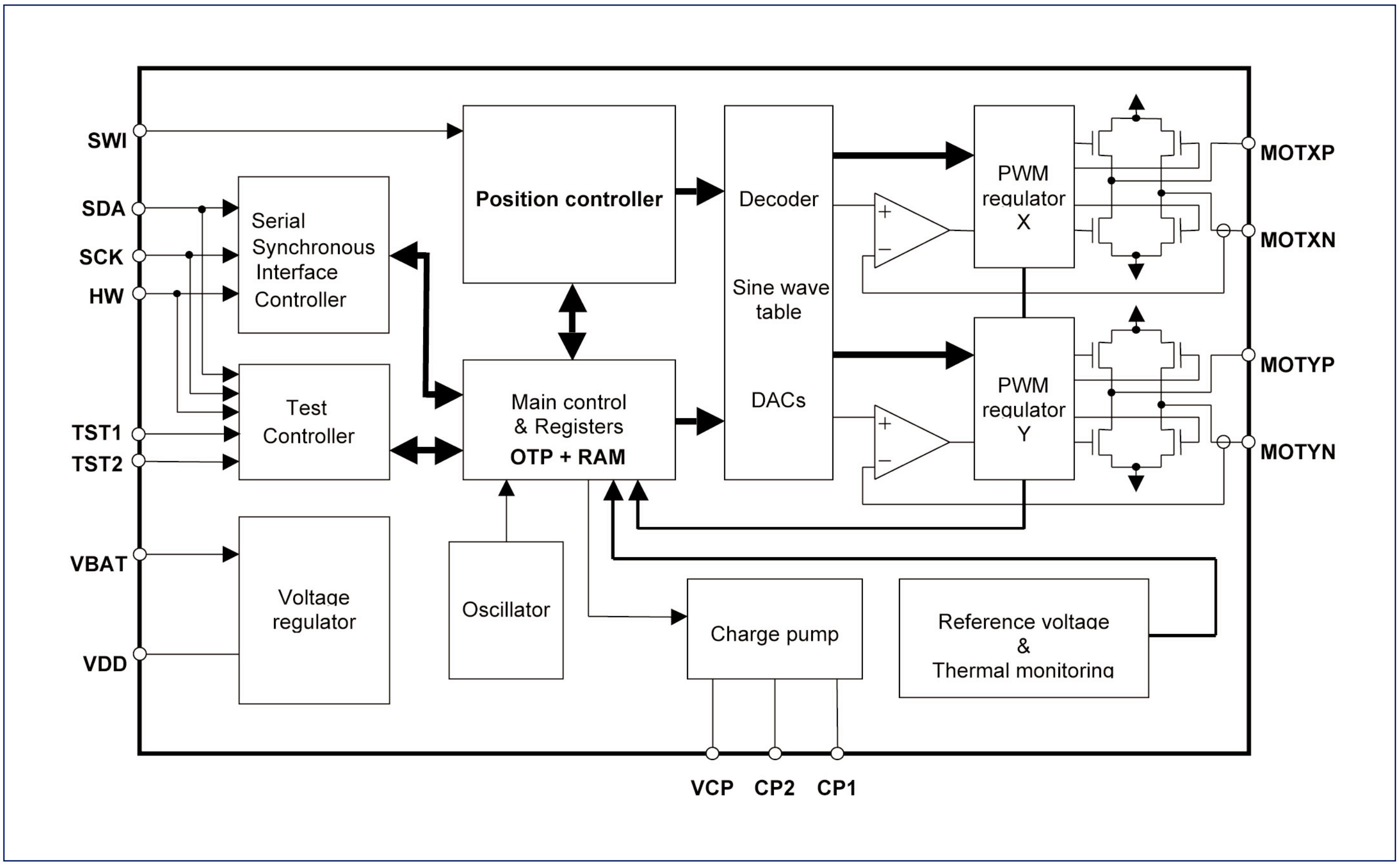
Notes
(1) For limited time: < 0.5 s.
(2) The circuit functionality is not guaranteed.
(3) Human body model (100 pF via 1.5 kΩ,
according to MIL std. 883E, method 3015.7).

1.2. Operating Ranges

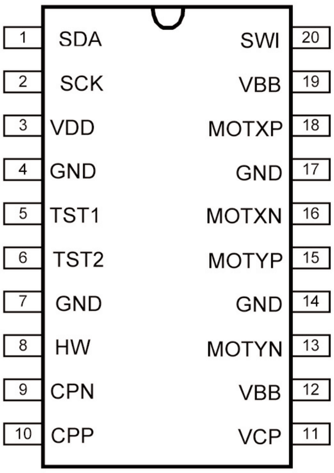
Parameter			Min	Max	Unit
Vbb	Supply voltage [1]		+6.5	+29	V
Top	Operating temperature range	Vbb ≤ 18V	-40	+125	°C
		Vbb ≤ 29V	-40	+85	°C

Notes
(1) Motordriver is disabled when Vbb < 8.9V.

2.0 Block Diagram



3.0 Pin-out



SOIC-20

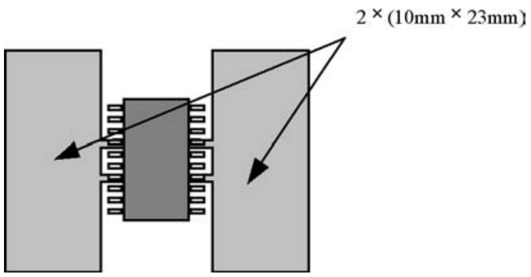
Pin Name	Pin Description	SOIC-20
SDA	Serial I/O	1
SCK	Clock for SDA	2
VDD	Internal supply (needs external decoupling capacitor)	3
GND	Ground, heat sink	4,7,14,17
TST1	Test pin (to be tied to ground in normal operation)	5
TST2	Test pin (to be left open in normal operation)	6
HW	Hardwired address bit	8
CPN	Negative connection of pump-capacitor (charge pump)	9
CPP	Positive connection of pump-capacitor (charge pump)	10
VCP	Charge-pump filter-capacitor	11
VBB	Battery voltage supply	12, 19
MOTYN	Negative end of phase Y coil	13
MOTYP	Positive end of phase Y coil	15
MOTXN	Negative end of phase X coil	16
MOTXP	Positive end of phase X coil	18
SWI	Switch input	20

4.0 Package Thermal Resistance

4.1. SO20

The junction-case thermal resistance is 28°C/W, leading to a junction-ambient thermal resistance of 63°C/W, with the PCB ground plane layout condition given on the figure beside, and with:

- PCB thickness = 1.6mm
- 1 layer
- Copper thickness = 35µm



5.0 DC Parameters

The DC parameters are given for Vbb and temperature in their operating ranges.
Convention: currents flowing in the circuit are defined as positive.

Symbol Motordriver	Pin(s)	Parameter	Test Conditions	Min	Typ	Max	Unit
IMSmax Peak		Max current trough motor coil in normal operation			800		mA
IMSmax RMS	MOTXP MOTXN	Max RMS current trough coil in normal operation			570		mA
IMSabs	MOTYP	Absolute error on coil current		-10		10	%
IMSrel	MOTYN	Error on current ratio Icoilx / Icoily		-7		7	%
RDSon		On resistance for each pin (including bond wire)	To be confirmed by characterization			1	Ω
IMSL		Pull down current	HZ mode		1		mA
Thermal Warning & Shutdown							
T_{tw}		Thermal warning		138	145	152	°C
T_{tsd}		Thermal shutdown [1]			Ttw+10		°C
T_{low}		Low temperature warning			Ttw-155		°C
Supply & Voltage Regulator							
Vbb		Nominal operating supply range [2]		6.5		18	V
VbbOTP		Supply voltage for OTP zapping		9.0		10.0	V
UV1	VBB	Stop voltage high threshold				9.8	V
UV2		Stop voltage low threshold		8.0	8.5	9.0	V
Ibat		Total current consumption	Unloaded outputs		10		mA
Vdd		Internal regulated output [3]	8V < Vbb < 18V Cload = 1µF (+100nF cer.)	4.75	5	5.35	V
IddStop	VDD	Digital current consumption	Vbb < UV2		2		mA
VddReset		Digital supply reset level [4]				4.4	V
IddLim		Current limitation	Pin shorted to ground			40	mA
Switch Input and Hardwire Address Input							
Rt_OFF		Switch OFF resistance [5]	Switch to Gnd or Vbat,	10			kΩ
Rt_ON		Switch ON resistance [5]				2	kΩ
Vbb_sw	SWI HW	Vbb range for guaranteed operation of SWI and HW		6		29	V
Vmax_sw		Maximum voltage	T < 1s			40V	V
Ilim_sw		Current limitation	Short to Gnd or Vbat		30		mA
Serial Interface Pins							
V_{IL}		Input level low [6]		-0.5		0.3 Vdd	V
V_{IH}		Input level high [7]		0.7 Vdd		Vdd + 0.5	V
V_{nL}	SDA SCK	Noise margin at the LOW level for each connected device (including hysteresis)		$0.1V_{DD}$			V
V_{nH}		Noise margin at the HIGH level for each connected device (including hysteresis)		$0.2V_{DD}$			V
Charge Pump							
Vcp	VCP	Output voltage	Vbb > 15V Vbb > 8V	Vbb+10 Vbb+5.8V	Vbb+12.5	Vbb+15	V V
Cbuffer		External buffer capacitor		220		470	nF
Cpump	CPP CPN	External pump capacitor		220		470	nF

Notes

[1] No more than 100 cumulated hours in life time above Ttsd.

[2] Communication over serial bus is operating. Motordriver is disabled when Vbb < UV2.

[3] Pin VDD must not be used for any external supply.

[4] The RAM content will not be altered above this voltage.

[5] External resistance value seen from pin SWI or HW, including 1kΩ series resistor.

[6] If input voltage ≤ 0.3V, then a resistor of 22 to 100Ω must be added in series.

[7] In case 100kHz ≤ f_{SCL} ≤ 360kHz V_{IH} min = $0.7V_{DD}$.

6.0 AC Parameters

The AC parameters are given for Vbb and temperature in their operating ranges.

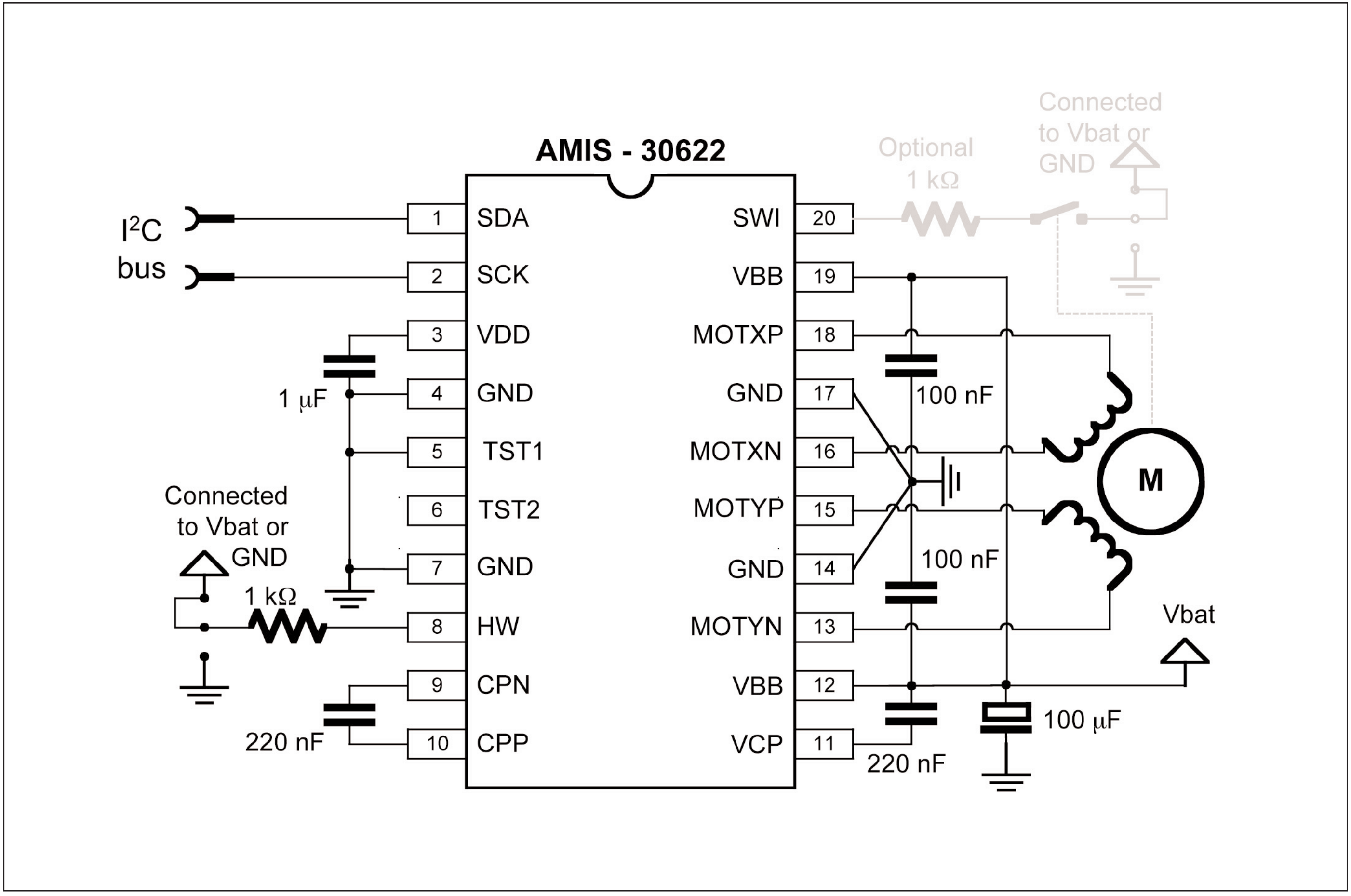
Symbol	Pin(s)	Parameter	Test Conditions	Min	Typ	Max	Unit
Power-up							
T_{pu}		Power-up time				10	ms
Internal Oscillator							
f_{osc}		Frequency of internal oscillator		3.6	4.0	4.4	MHz
I²C Transceiver							
f_{SCL}		SCL clock frequency	f_{SCL} ≤ 100kHz	0		100	kHz
			f_{SCL} ≤ 360kHz	0		360	kHz
$t_{HD;STA}$		Hold time (repeated) START condition. After this period, the first clock pulse is generated.	f_{SCL} ≤ 100kHz	4.0			µs
			f_{SCL} ≤ 360kHz	0.6			µs
t_{LOW}		LOW period of the SCK clock	f_{SCL} ≤ 100kHz	4.7			µs
			f_{SCL} ≤ 360kHz	1.3			µs
t_{HIGH}		HIGH period of the SCK clock	f_{SCL} ≤ 100kHz	4.0			µs
			f_{SCL} ≤ 360kHz	0.6			µs
$t_{SU;STA}$		Set-up time for a repeated START condition	f_{SCL} ≤ 100kHz	4.7			µs
			f_{SCL} ≤ 360kHz	0.6			µs
$t_{SU;DAT}$		Data set-up time	f_{SCL} ≤ 100kHz	250			ns
			f_{SCL} ≤ 360kHz	100			ns
t_r	SDA	Rise time of both SDA and SCK signals [1]	f_{SCL} ≤ 100kHz			1000	ns
	SCK		f_{SCL} ≤ 360kHz	20+0.1C_b		300	ns
t_f		Fall time of both SDA and SCK signals [1]	f_{SCL} ≤ 100kHz			300	ns
			f_{SCL} ≤ 360kHz	20+0.1C_b		300	ns
$t_{SU;STO}$		Set-up time for STOP condition	f_{SCL} ≤ 100kHz	4.0			µs
			f_{SCL} ≤ 360kHz	0.6			µs
t_{BUF}		Bus free time between a STOP and START condition	f_{SCL} ≤ 100kHz	4.7			µs
			f_{SCL} ≤ 360kHz	1.3			µs
t_{SP}		Pulse width of spikes which must be suppressed by the input filter		50			ns
C_b		Capacitive load for each bus line				400	pF
C_i		Capacitance for each I/O pin				10	pF
Switch Input and Hardwire Address Input							
T_{sw}	SWI	Scan pulse period [2]			1024		µs
T_{sw_on}	HW	Scan pulse duration			1/16		Tsw
Motordriver							
f_{pwm}		PWM frequency [2]		18	20	22	kHz
T_{brise}	MOTxx	Turn-on transient time	Between 10% and 90%		350		ns
T_{bfall}		Turn-off transient time			250		ns
Charge Pump							
f_{CP}	CPN CPP	Charge pump frequency [2]			250		kHz

Notes

[1] C_b = total capacitance of one bus line in pF.

[2] Derived from the internal oscillator.

7.0 Typical Application



Notes

(1) Optionally an external switch to Vbat or GND can be connected to the SWI pin.

(2) Resistors tolerance: ± 5%.

(3) Depending on the application the ESR value of the 1μF and 100μF capacitors must be carefully chosen. The working voltage of the 100μF capacitor depends on the maximum Vbat value.

(4) 100nF capacitors must be close to pins VBB and VDD.

(5) 220nF capacitors must be as close as possible to pins CPN, CPP, VCP, and VBB to reduce EMC radiation.

(6) If SDA and/or SCK input voltage ≤ 0.3V, then a resistor of 22 to 100Ω must be added in series.

8.0 Positioning Data

8.1 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

8.2 Maximum Velocity

For each stepping mode, 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.

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

8.3 Minimum Velocity

Once Vmax is chosen, 16 possible values can be programmed for Vmin. The table below provides the obtainable values in Full-step/s. The accuracy of Vmin is derived from the internal oscillator.

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

Notes

(1) The Vmax factor is an approximation.

(2) 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**.

8.4 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/s². One observes restrictions for some combination of acceleration index and maximum speed (gray cells).
The accuracy of **`Acc`** is derived from the internal oscillator.

Vmax (FS/s) ’	99	136	167	197	213	228	243	273	303	334	364	395	456	546	729	973
ACC Index ↓	Acceleration (Full Step/s²)															
0	49							106						473		
1	218													735		
2	1004															
3	3609															
4	6228															
5	8848															
6	11409															
7	13970															
8	16531															
9	14785	19092														
10		21886														
11		24447														
12		27008														
13		29570														
14	29570							34925								
15								40047								

The formula to compute the number of equivalent Full-step during acceleration phase is:

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

8.5 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.

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 ↔ x2
1/4th	S	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	LSB	0	0	2-bit left ↔ x4
Half-stepping	S	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	LSB	0	0	0	3-bit left ↔ x8
SecurePosition	S	B9	B8	B7	B6	B5	B4	B3	B2	B1	LSB	0	0	0	0	0	No shift

8.5.1 Position Ranges

A position is coded by using the binary two’s complement format. According to the positioning commands which are used (see § 9.2.2.10 Application Commands) and to the chosen stepping mode, the position range will be as shown in the table below.

Command	Stepping Mode	Position Range	Full Range	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 on the left by a certain number of bits, according to the chosen stepping mode.

8.5.2 Secure Position

A secure position can be programmed. It is coded on 11-bit, thus having a lower resolution than normal positions, as shown in the table below. See command **`GotoSecurePosition`**.

Stepping Mode	Secure Position Resoultion
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 note

The secure position is disabled in case the programmed value is the reserved code “10000000000”
(most negative position).

8.5.3 Shaft

A shaft bit can be programmed to define whether a positive motion is an outer or an inner motion:

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

9.0 Functional Description

9.1 Structure Description

9.1.1 Stepper Motordriver

The motordriver receives the control signals from the control logic. It mainly features:

- Two H-bridges designed to drive a two separated coils stepper motor. 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 DACs to set the maximum magnitude of the current through X and Y.
- Two DACs to set the correct current ratio through X and Y.

Battery voltage monitoring is also performed by this block, which provides needed 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.

9.1.2 Control Logic (Position Controller and Main Control)

The control logic block stores the information provided by the I²C interface (in the RAM or 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 initiate or stop properly 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 about internal actions and reporting to the I²C interface.

9.1.3 Miscellaneous

The AMIS-30622 also implements the followings:

- An internal oscillator, needed for the Control logic and for 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 5V regulator (from the battery supply) to supply the internal logic circuitry.

9.2 Functions Description

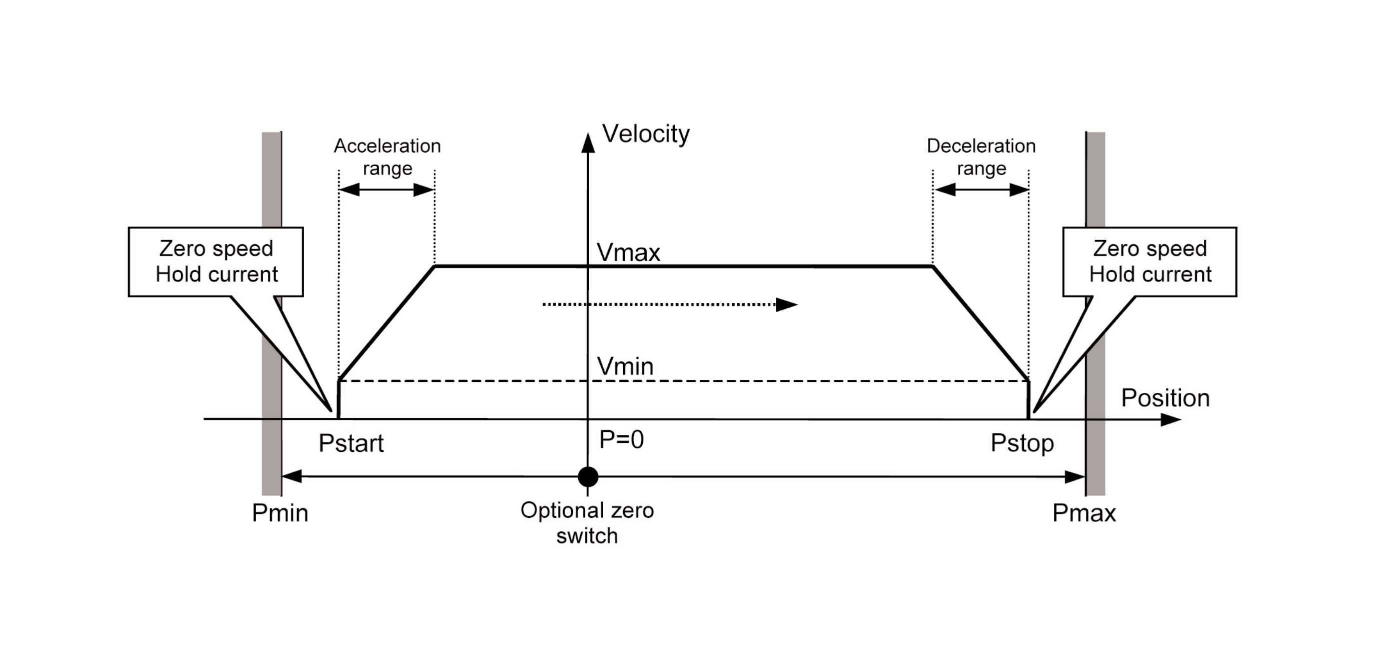
This chapter describes the four most important blocks:

- Position controller
- Main control and register, OTP memory + RAM
- Motordriver
- I²C controller

9.2.1 Position Controller

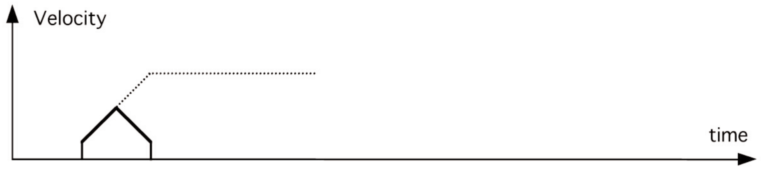
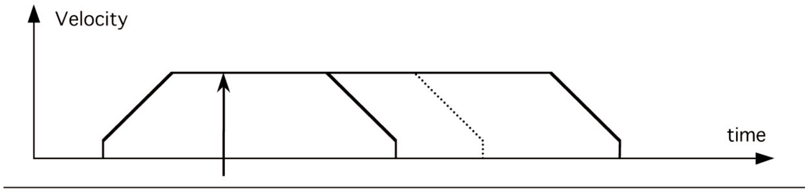
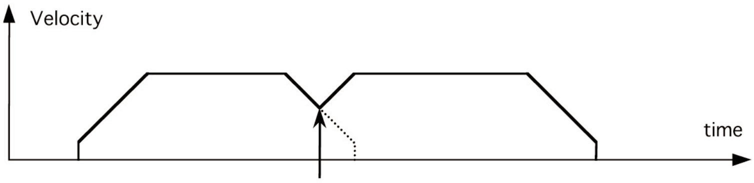
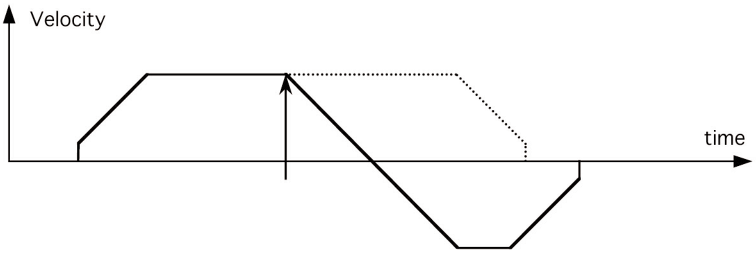
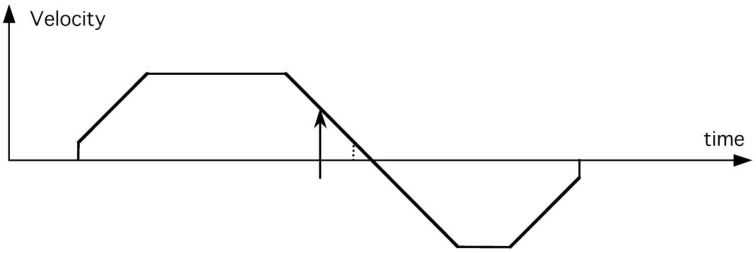
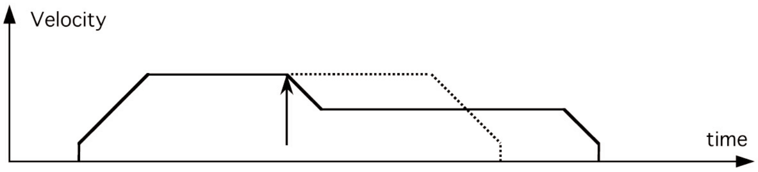
9.2.1.1 Positioning and Motion Control

A positioning command will produce a motion as illustrated below. 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 to the parameters programmed by the application during configuration phase. The current in the coils is also programmable.



Parameter	Value
Pmax - Pmin	See § 8.5
Zero Speed Hold Current	See § 9.2.2.12 (**Ihold**)
Maximum Current	See § 9.2.2.12 (**Irun**)
Acceleration and Deceleration	See § 8.4
Vmin	See § 8.3
Vmax	See § 8.2

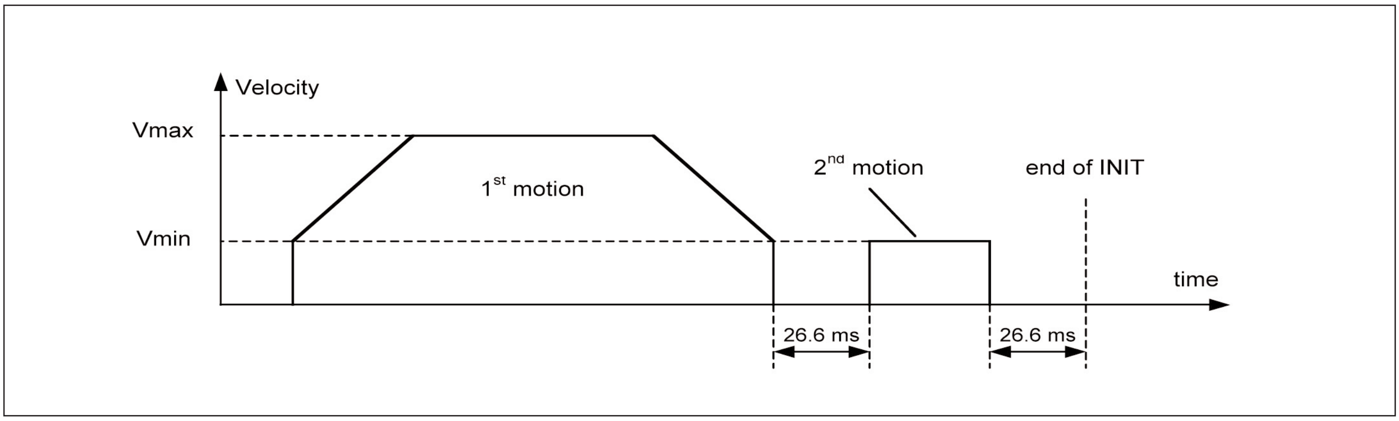
Different positioning examples are shown in the table below.

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 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	

9.2.1.2 Position Initialization

After power-up or when a Vdd reset has been acknowledged to the master, a position initialization of the stepper motor can be requested by the application, by use of the **RunInit** command (see § 0). The position initialization is performed by the position controller under the control of the Main control block. This operation cannot be interrupted or influenced by any further command. A position initialization can only be interrupted by the occurrence of the conditions driving to a motor shutdown (see § 9.2.2.7) or by a **HardStop** command. On the other hand, sending a **RunInit** command while a motion is already ongoing is not recommended.

A position initialization consists of two successive motions, as illustrated below.



The first motion is done with the specified **Vmin** and **Vmax** velocities in the **RunInit** command, with the acceleration (deceleration) parameter already in RAM, to a position **Pos1[15:0]** also specified in **RunInit**. The goal here is to perform a motion large enough to reach a stall position (considered to be the reference position).

Then a second motion to a position **Pos2[15:0]** is done at the specified **Vmin** velocity in the **RunInit** command (no acceleration). The purpose of this second motion is to confirm with a low velocity the positioning of the motor at the stall position, assuming that the stepper motor may have bounced against the stall position. Therefore, **Pos2** should only be a few half or micro steps further than **Pos1**, in order to perform a displacement of at least one electrical period.

Once the second motion is achieved, the **ActPos** register (see § 0) is reset to zero, to set the reached position as the reference position, whereas **TagPos** register is not changed.

Notes

(1) The priority encoder (see 9.2.2.11 Priority Encoder) is describing the management of states and commands. The notes below are to be considered illustrative.

(2) The last **SetPosition** command issued during an initialization sequence will be kept in memory and executed afterwards. This applies also for the commands **SetMotorParam** and **GotoSecurePosition**.

(3) Commands such as **GetActualPos** or **GetStatus** will be executed while the position initialization is running.

(4) An initialization sequence starts by setting **TagPos** register to **SecPos** value, provided secure position is enabled otherwise **TagPos** is reset to zero.

(5) The acceleration/deceleration value applied during an initialization sequence is the one stored in RAM before the **RunInit** command is sent. The same applies for **Shaft** bit, but not for **Irun**, **Ihold** and **StepMode**, which can be changed during an initialization sequence.

(6) The **Pos1**, **Pos2**, **Vmax** and **Vmin** values programmed in a **RunInit** command apply only for this initialization sequence. All further positioning will use the parameters stored in RAM (programmed for instance by a former **SetMotorParam** command).

(7) Commands **ResetPosition**, **RunInit** and **SoftStop** will be ignored while an initialization sequence is ongoing, and will not be executed afterwards.

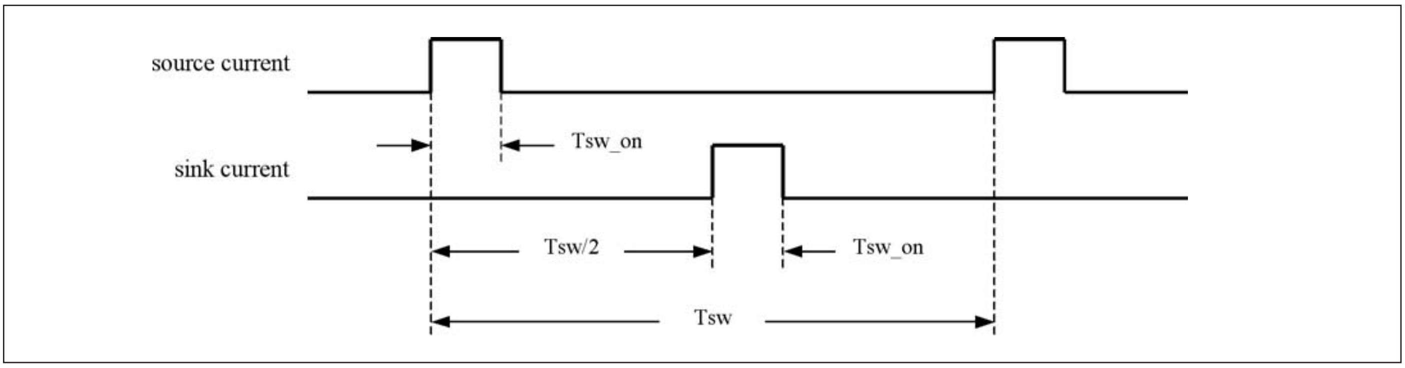
(8) A **SetMotorParam** command should not be sent during an initialization sequence.

(9) If for some reason **ActPos** equals **Pos1[15:0]** at the moment the **RunInit** command is issued, the circuit will enter in deadlock state. Therefore, the application should check the actual position by a **GetPosition** or a **GetFullStatus** command prior to an initialization. Another solution may consist in programming a value out of the stepper motor range for **Pos1[15:0]**.

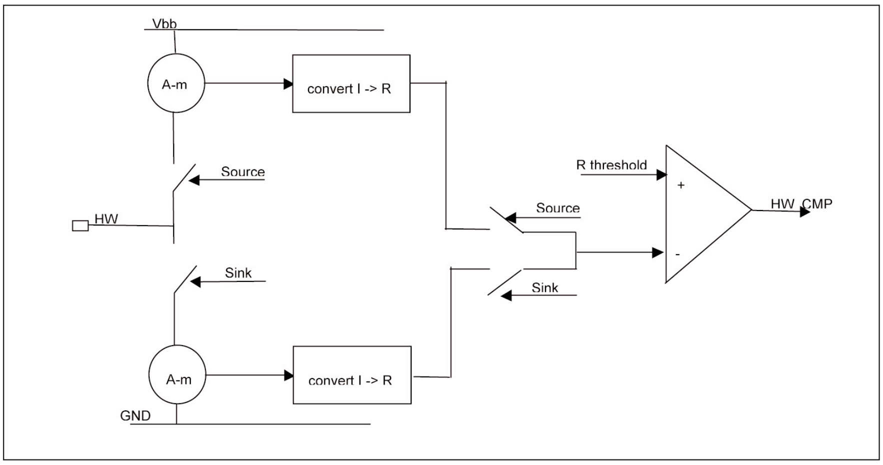
9.2.1.3 External Switch and HW Pin

Pin SWI and hardwired address pin HW (see § 9.2.4.3 Physical Address) will alternatively attempt to source and sink a current in/from the external switch (see application schematic) to test whether it is ON or OFF. This current is set around 10mA when a 1kΩ external series resistor is used.

This can be represented by the following time diagram (the timings are given in § 6).



If the switch is detected ON (closed), then the flag **<ESW>** is raised. The status of this flag can be read by the application via a **GetActualPos** or a **GetFullStatus1** reading frame. At the falling edge of every current pulse (at around 1kHz), the stepper motor actual position is refreshed (register ActPos, see § 9.2.2.9), so that the master node may get synchronous information about the state of the switch together with the position of the motor. The position is then given with an accuracy of ± 1 half-step (or micro-step, depending of the programmed stepping mode). The block diagram below shows how this function is implemented for HW.



With the following truth table:

State	Sink	Source	New State
Float	1	0	Float
Float	1	0	HWHi
Float	0	1	Float
Float	0	1	HWLo
HWLo	1	0	HWLo
HWLo	1	0	HWHi
HWLo	0	1	Float
HWLo	0	1	HWLo
HWHi	1	0	Float
HWHi	1	0	HWHi
HWHi	0	1	HWHi
HWHi	0	1	HWLo

HWHi	address = "1"
HWLo	address = "0"

note that e.g. if HW is connected to GND, LS-part will report "float" while HS-part will report "low resistance detected".

Note
If HW is detected to be floating, motion to the secure position is performed.

9.2.2 Main Control and Register, OTP Memory + RAM

9.2.2.1 Power-up Phase

Power-up phase of the AMIS-30622 will not exceed 10ms. After this phase, the AMIS-30622 is in Shutdown mode, ready to receive I²C messages and to execute the associated commands. After power-up, the registers and flags are in the Reset state, some of them being loaded with the OTP memory content (see § 9.2.2.13 OTP Memory Structure)

9.2.2.2 Reset State

After power-up, or after a reset occurrence (e.g. a micro cut on pin VBB has made Vdd 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. See also § 9.2.2.7 Motor Shutdown Mode and Digital Supply Reset in § 9.2.2.9 Flags Table.

9.2.2.3 Soft Stop

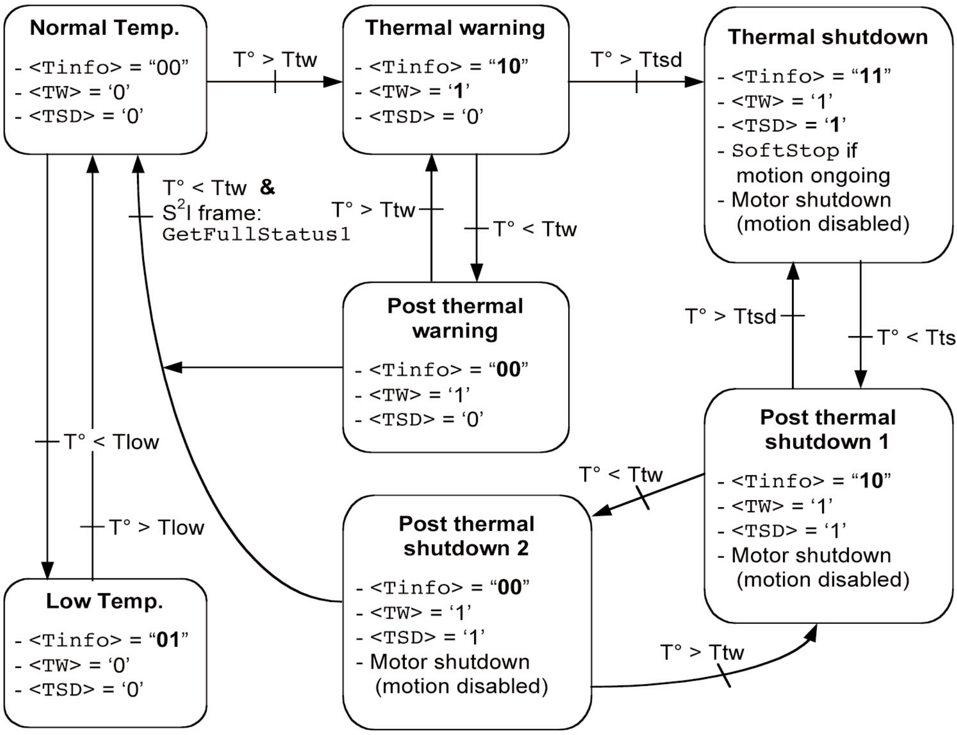
A Soft Stop is an immediate interruption of a motion, but with a deceleration phase. At the end of this action, the register **`TagPos`** is loaded with the value contained in register **`ActPos`** to avoid an attempt of the circuit to achieve the motion (see § 9.2.2.9 Flags Table). The circuit is then ready to execute a new positioning command, provided thermal and electrical conditions allow for it.

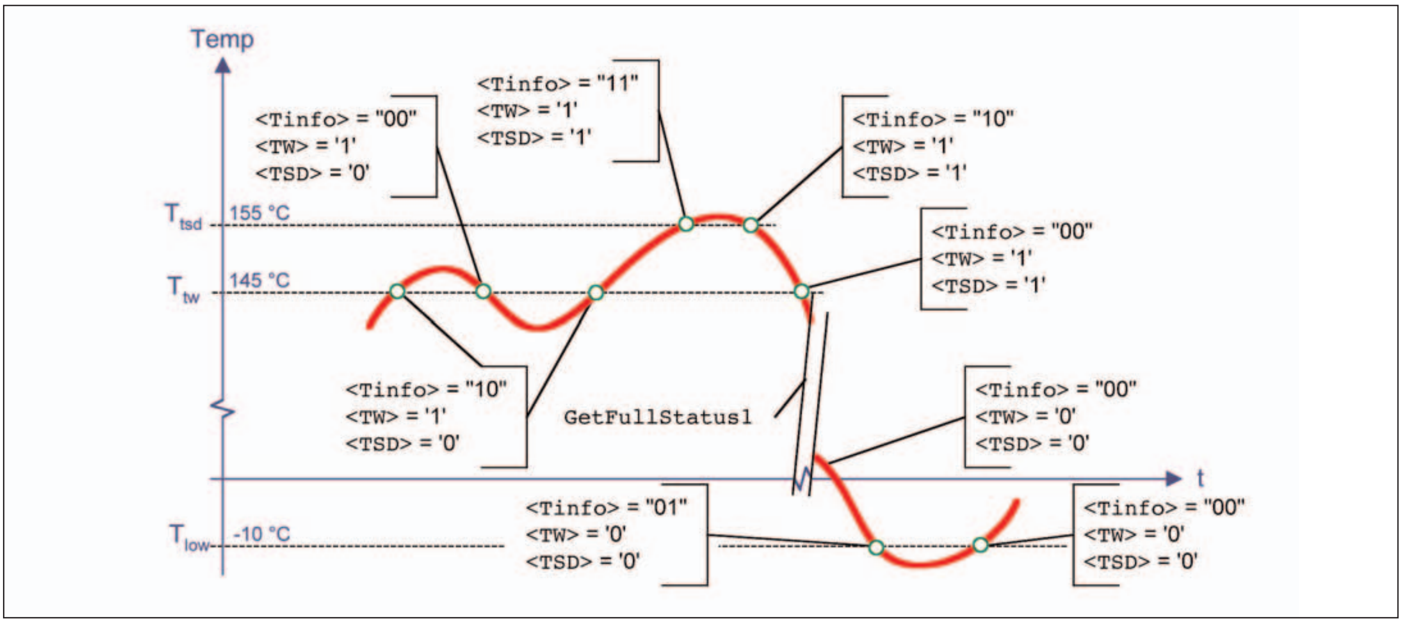
9.2.2.4 Thermal Shutdown Mode

When thermal shutdown occurs, the circuit performs a **`SoftStop`** command and goes to motor shutdown mode (see below).

9.2.2.5 Temperature Management

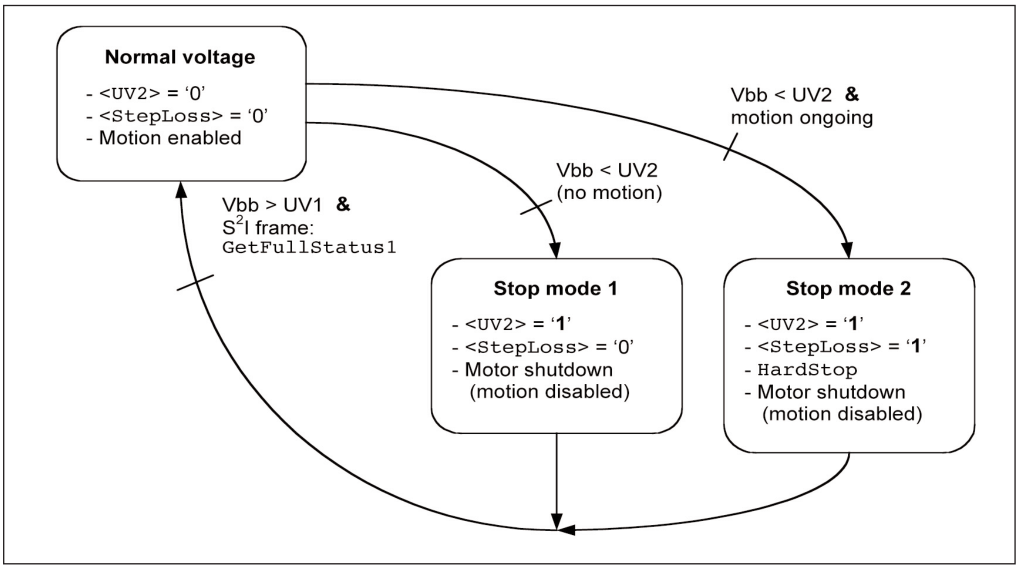
The AMIS-30622 monitors temperature by mean of two thresholds and one shutdown level, as illustrated in the state diagram below. The only condition to reset flags **`<TW>`** and **`<TSD>`** (respectively Thermal Warning and Thermal Shutdown) is to be at a temperature lower than Ttw and to get the occurrence of a **`GetFullStatus1`** command.

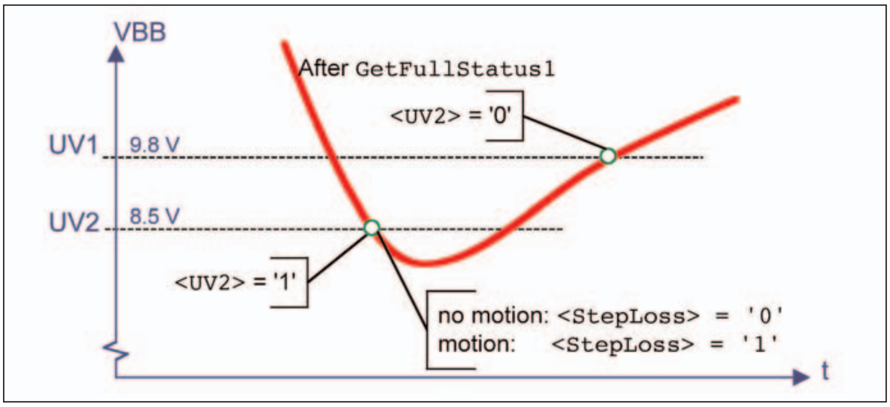




9.2.2.6 Battery Voltage Management

The AMIS-30622 monitors the battery voltage by mean of one threshold and one shutdown level, as illustrated in the state diagram below. The only condition to reset flags **<UV2>** and **<StepLoss>** is to recover a battery voltage higher than UV1 and to receive a **GetFullStatus1** command.





9.2.2.7 Motor Shutdown Mode

A motor shutdown occurs when:
1. The chip temperature rises above the thermal shutdown threshold T_{tsd} (see § 5 DC-Parameters).
2. The battery voltage goes below UV2 (see § 5 DC Parameters).
3. Flag **`<ElDef>`** = '1', meaning an electrical problem is detected on one or both coils.
4. Flag **`<CPFail>`** = '1', meaning there is a charge pump failure.

A motor shutdown leads to the followings:
- H-bridges in high impedance mode.
- The **`TagPos`** register is loaded with the **`ActPos`** (to avoid any motion after leaving the motor shutdown mode)

The I^{2}C interface remains active, being able to receive orders or send status.

The conditions to get out of a motor shutdown mode are:
- Reception of a **`GetFullStatus1`** command AND
- The four above causes are no more detected.

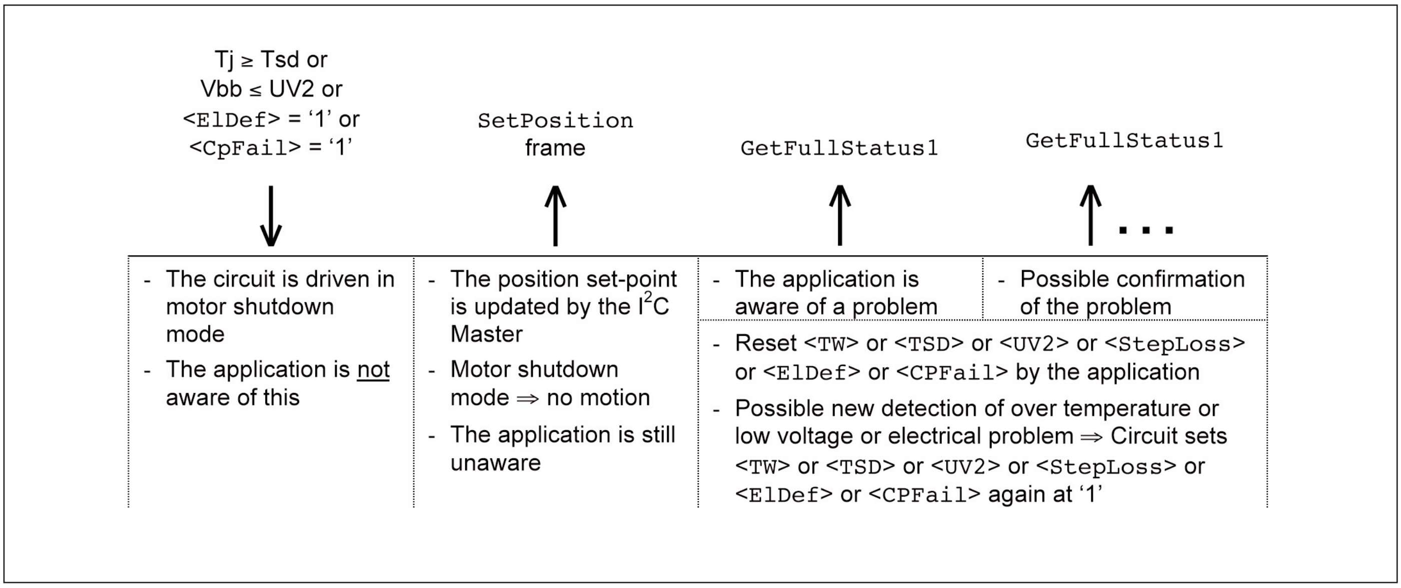
Which leads to H-bridges in Ihold mode. Hence the circuit is ready to execute any positioning command.

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

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-30622.

Warning
The application should limit the number of consecutive **`GetFullStatus1`** commands to try to get the AMIS-30622 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.

Note
The priority encoder (see § 9.2.2.11 Priority Encoder) is describing the management of states and commands. The table above is to be considered illustrative.



Tj ≥ Tsd or
Vbb ≤ UV2 or
`<ElDef>` = '1' or
`<CpFail>` = '1'

- The circuit is driven in motor shutdown mode
- The application is not aware of this

`SetPosition` frame

- The position set-point is updated by the I^{2}C Master
- Motor shutdown mode ⇒ no motion
- The application is still unaware

`GetFullStatus1`

- The application is aware of a problem

`GetFullStatus1` ...

- 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'

9.2.2.8 RAM Registers

Register	Mnemonic	Length (bit)	Related Commands	Comment	Reset State
Actual Position	**ActPos**	16	**GetActualPos** **GetFullStatus2** **GotoSecurePos** **ResetPosition**	- 16-bit signed	(1)
Last Programmed Position	**Pos/** **TagPos**	16/11	**GetFullStatus2** **GotoSecurePos** **ResetPosition** **SetPosition**	- 16-bit signed or - 11-bit signed for half stepping (see § 8.5)	
Acceleration Shape	**AccShape**	1	**GetFullStatus**[1] **ResetToDefault**[1] **SetMotorParam**	'0' → normal acceleration from Vmin to Vmax '1' → motion at Vmin without acceleration	'0'
Coil peak Current	**Irun**	4	**GetFullStatus**[1] **ResetToDefault**[1] **SetMotorParam**	Operating current (see § 9.2.2.12)	From OTP memory
Coil Hold Current	**Ihold**	4	**GetFullStatus** **ResetToDefault**[1] **SetMotorParam**	Standstill current (see § 9.2.2.12)	
Minimum Velocity	**Vmin**	4	**GetFullStatus**[1] **ResetToDefault**[1] **SetMotorParam**	See § 8.3 and § 9.2.2.12 (look-up table)	
Maximum Velocity	**Vmax**	4	**GetFullStatus**[1] **ResetToDefault**[1] **SetMotorParam**	See § 8.2 and § 9.2.2.12 (look-up table)	
Shaft	**Shaft**	1	**GetFullStatus**[1] **ResetToDefault**[1] **SetMotorParam**	Direction of movement for positive velocity	
Acceleration/ Deceleration	**Acc**	4	**GetFullStatus**[1] **ResetToDefault**[1] **SetMotorParam**	See § 8.4 and § 9.2.2.12 (look-up table)	
Secure Position	**SecPos**	11	**GetFullStatus2** **ResetToDefault**[1] **SetMotorParam**	Target position when I^2C connection fails; 11 MSBs of 16-bit position (LSBs fixed to '0')	
Stepping Mode	**StepMode**	2	**GetFullStatus**[1] **ResetToDefault**[1] **SetMotorParam**	See § 8.1 and § 9.2.2.12	

Note

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

9.2.2.9 Flags Table

Register	Mnemonic	Length (bit)	Related Commands	Comment	Reset State
Charge Pump Failure	**CPFail**	1	**GetFullStatus**	'0' = charge pump OK '1' = charge pump failure reset only after **GetFullStatus1**	'0'
Electrical Defect	**ElDef**	1	**GetActualPos** **GetStatus** **GetFullStatus1**	<**OVC1**> or <**OVC2**> or <**open circuit 1**> or <**open circuit 2**> or <**CPFail**> resets only after **GetFullStatus1**	'1'
External Switch Status	**ESW**	1	**GetActualPos** **GetStatus** **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 motion acceleration "010" = inner motion deceleration "011" = inner motion max. speed "101" = outer motion acceleration "110" = outer motion deceleration "111" = outer motion max. speed	"000"
Over Current in Coil X	**OVC1**	1	**GetFullStatus1**	'1' = over current reset only after **GetFullStatus1**	'1'
Over Current in Coil Y	**OVC2**	1	**GetFullStatus1**	'1' = over current reset only after **GetFullStatus1**	'1'
Secure Position Enabled	**SecEn**	1	Internal use	'0' if **SecPos** = "100 0000 0000" '1' otherwise	n.a.
Step Loss	**StepLoss**	1	**GetActualPos** **GetStatus** **GetFullStatus1**	'1' = step loss due to under voltage, over current or open circuit	'1'
Motor Stop	**Stop**	1	Internal use	See § 9.2.2.11	'0'
Temperature Info	**Tinfo**	2	**GetActualPos** **GetStatus** **GetFullStatus1**	"00" = normal temperature range "01" = low temperature warning "10" = high temperature warning "11" = motor shutdown	"00"
Thermal Shutdown	**TSD**	1	**GetActualPos** **GetStatus** **GetFullStatus1**	'1' = shutdown (> 155°C typ.) reset only after **GetFullStatus1** and if **<Tinfo>** = "00"	'0'
Thermal Warning	**TW**	1	**GetActualPos** **GetStatus** **GetFullStatus1**	'1' = over temp. (> 145°C) reset only after **GetFullStatus1** and if **<Tinfo>** = "00"	'0'
Battery Stop Voltage	**UV2**	1	**GetActualPos** **GetStatus** **GetFullStatus1**	'0' = Vbb > UV2 '1' = Vbb ≤ UV2 reset 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, may have been lost; it warns that the RAM contents can be reset to '0' with a **GetFullStatus1** command.	'1'

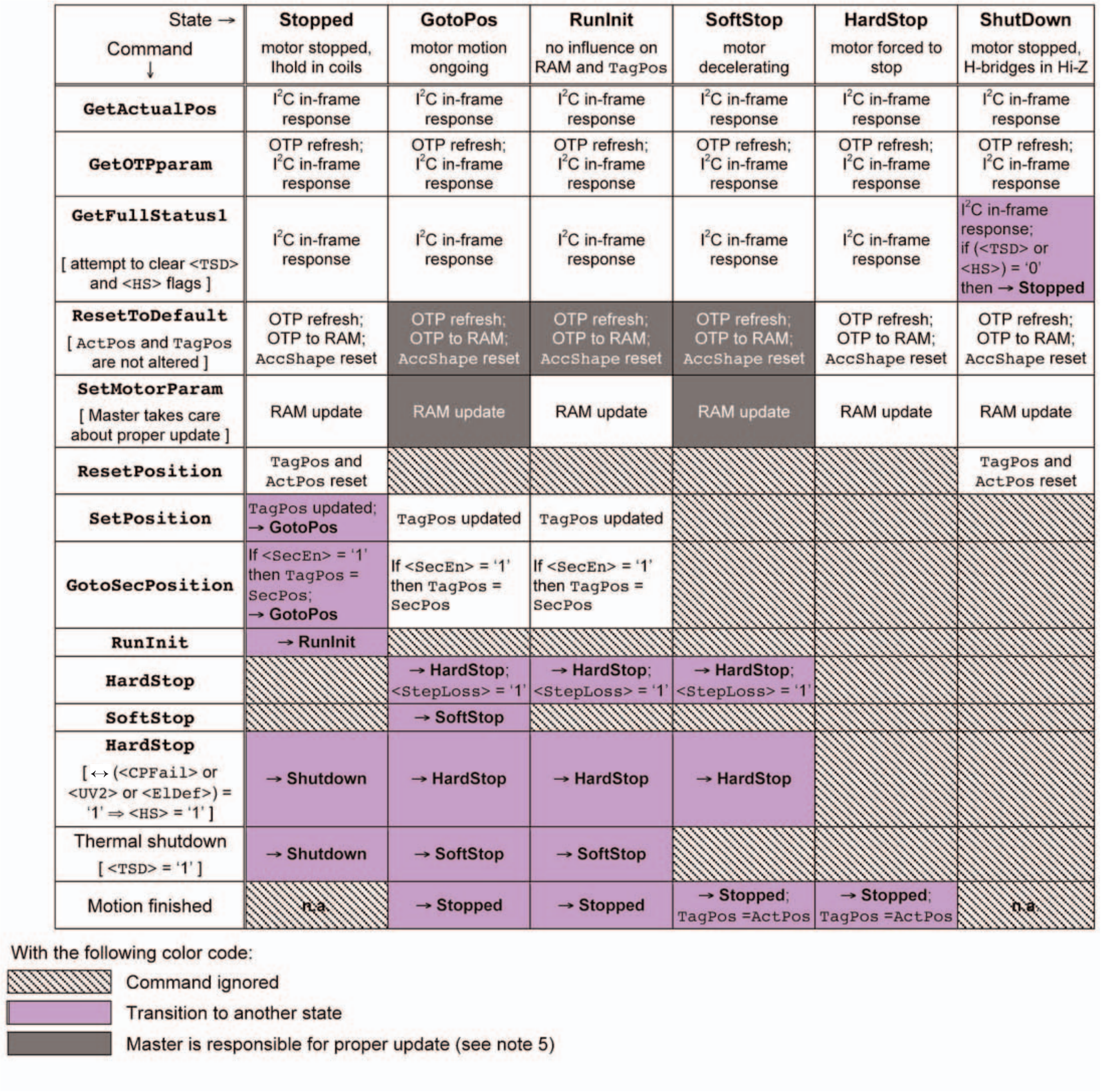
9.2.2.10 Application Commands

The I^{2}C Master will have to use commands to manage the different application tasks the AMIS-30622 can feature. The commands summary is given in the table below.

Command Mnemonic	Function
`GetFullStatus1`	Returns complete status of the chip
`GetFullStatus2`	Returns actual, target and secure position
`GetOTPParam`	Returns OTP parameters
`GotoSecurePosition`	Drives motor to secure position
`HardStop`	Immediate full stop
`ResetPosition`	Sets actual position to zero
`ResetToDefault`	Overwrites the chip RAM with OTP contents
`RunInit`	Reference Search
`SetMotorParam`	Sets motor parameter
`SetOTP`	Zaps the OTP memory
`SetPosition`	Programmes a target
`SoftStop`	Motor stopping with deceleration phase

9.2.2.11 Priority Encoder

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



State → Command ↓	Stopped motor stopped, Ihold in coils	GotoPos motor motion ongoing	RunInit no influence on RAM and TagPos	SoftStop motor decelerating	HardStop motor forced to stop	ShutDown motor stopped, H-bridges in Hi-Z
GetActualPos	I^{2}C in-frame response	I^{2}C in-frame response	I^{2}C in-frame response	I^{2}C in-frame response	I^{2}C in-frame response	I^{2}C in-frame response
GetOTPparam	OTP refresh; I^{2}C in-frame response	OTP refresh; I^{2}C in-frame response	OTP refresh; I^{2}C in-frame response	OTP refresh; I^{2}C in-frame response	OTP refresh; I^{2}C in-frame response	OTP refresh; I^{2}C in-frame response
GetFullStatus1 [attempt to clear <TSD> and <HS> flags]	I^{2}C in-frame response	I^{2}C in-frame response	I^{2}C in-frame response	I^{2}C in-frame response	I^{2}C in-frame response	I^{2}C in-frame response; if (<TSD> or <HS>) = '0' then → Stopped
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	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
ResetPosition	TagPos and ActPos reset	▨	▨	▨	▨	TagPos and ActPos reset
SetPosition	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	▨	▨	▨
RunInit	→ RunInit	▨	▨	▨	▨	▨
HardStop	▨	→ HardStop; <StepLoss> = '1'	→ HardStop; <StepLoss> = '1'	→ HardStop; <StepLoss> = '1'	▨	▨
SoftStop	▨	→ SoftStop	▨	▨	▨	▨
HardStop [↔ (<CPFail> or <UV2> or <ElDef>) = '1' ⇒ <HS> = '1']	→ Shutdown	→ HardStop	→ HardStop	→ HardStop	▨	▨
Thermal shutdown [<TSD> = '1']	→ Shutdown	→ SoftStop	→ SoftStop	▨	▨	▨
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 5)

Notes

(1) After Power-on-reset, the Shutdown state is entered. The Shutdown state can only be left after **GetFullStatus1** command (so that the Master could read the <**VddReset**> flag).

(2) A RunInit sequence runs with a separate set of RAM registers. The parameters that are not specified in a RunInit command are loaded with the values stored in RAM at the moment the RunInit sequence starts. **AccShape** is forced to '1' during second motion even if a **ResetToDefault** command is issued during a RunInit sequence, in which case **AccShape** at '0' will be taken into account after the RunInit sequence. A **GetFullStatus1** command will return the default parameters for **Vmax** and **Vmin** stored in RAM.

(3) Shutdown state can be left only when <**TSD**> and <**HS**> flags are reset.

(4) 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).

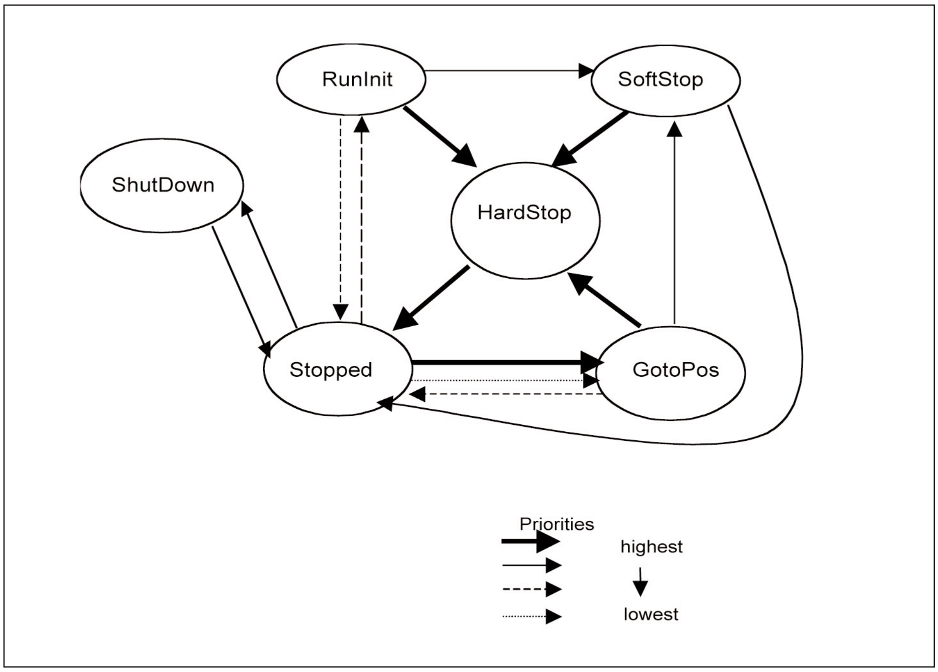
(5) A **SetMotorParam** command sent while a motion is ongoing (state GotoPos) should not attempt to modify **Acc** and **Vmin** values. This can be done during a RunInit sequence since this motion uses its own parameters, the new parameters will be taken into account at the next **SetPosition** command.

(6) <**SecEn**> = '1' when register **SecPos** is loaded with a value different from the most negative value (i.e. different from 0x400 = "100 0000 0000").

(7) <**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.

(8) While in state Stopped, if ActPos ≠ TagPos there is a transition to state GotoPos. This transition has the lowest priority, meaning that <**Stop**>, <**TSD**>, etc. are first evaluated for possible transitions.

(9) If <StepLoss> is active, then **SetPosition** and **GotoSecurePosition** commands are ignored (they will not modify **TagPos** register whatever the state), and motion to secure position is forbidden. Other command like **RunInit** or **ResetPosition** will be executed if allowed by current state. <**StepLoss**> can only be cleared by a **GetFullStatus1** command.



9.2.2.12 Application Parameters Stored in OTP Memory

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

`AD[4:0]` Physical address of the stepper motor. Up to 32 Stepper-motors can theoretically be connected to the same I^{2}C bus.

`Irun[3:0]` Peak current value to be fed to each coil of the stepper motor. The table to the right provides the 16 possible values for **`IRUN`**.

	Irun			Peak current (mA)
0	0	0	0	59
0	0	0	1	71
0	0	1	0	84
0	0	1	1	100
0	1	0	0	119
0	1	0	1	141
0	1	1	0	168
0	1	1	1	200
1	0	0	0	238
1	0	0	1	283
1	0	1	0	336
1	0	1	1	400
1	1	0	0	476
1	1	0	1	566
1	1	1	0	673
1	1	1	1	800

`Ihold[3:0]` Hold current for each coil of the stepper motor. The table to the right provides the 16 possible values for **`IHOLD`**.

	Ihold			Hold current (mA)
0	0	0	0	59
0	0	0	1	71
0	0	1	0	84
0	0	1	1	100
0	1	0	0	119
0	1	0	1	141
0	1	1	0	168
0	1	1	1	200
1	0	0	0	238
1	0	0	1	283
1	0	1	0	336
1	0	1	1	400
1	1	0	0	476
1	1	0	1	566
1	1	1	0	673
1	1	1	1	800

`StepMode` Indicator of stepping mode to be used.

StepMode		Step mode
0	0	Half stepping
0	1	1/4 micro step
1	0	1/8 micro step
1	1	1/16 micro step

Shaft Indicator of Reference Position. If **Shaft** = '0', the reference position is the maximum inner position, whereas if **Shaft** = '1', the reference position is the maximum outer position.

SecPos[10:0] Secure Position of the stepper motor. This is the position to which the motor is driven in case of a **GotoSecurePosition** command, or if the HW-pin is disconnected from Vbat or Gnd.

If **SecPos[10:0]** = "100 0000 0000", this means that Secure Position is disabled, e.g. the steppermotor will be kept in the position occupied at the moment these events occur.

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'.

Vmax[3:0] **Vmin[3:0]** **Acc[3:0]** Maximum velocity, minimum velocity and acceleration of the stepper motor are programmed by coding the respective Vmax, Vmin and Acc parameters index as defined in § 8 Positioning Data.

Code				Parameter Index
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

9.2.2.13 OTP Memory Structure

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

OTP Address	7	6	5	4	3	2	1	0
0x00	OSC3	OSC2	OSC1	OSC0	IREF3	IREF2	IREF1	IREF0
0x01		TSD2	TSD1	TSD0	BG3	BG2	BG1	BG0
0x02					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	SecPos1	SecPos0
0x07					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 OPT 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 using a SetOTPparam command could still zap any OTP bit at '0'.

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

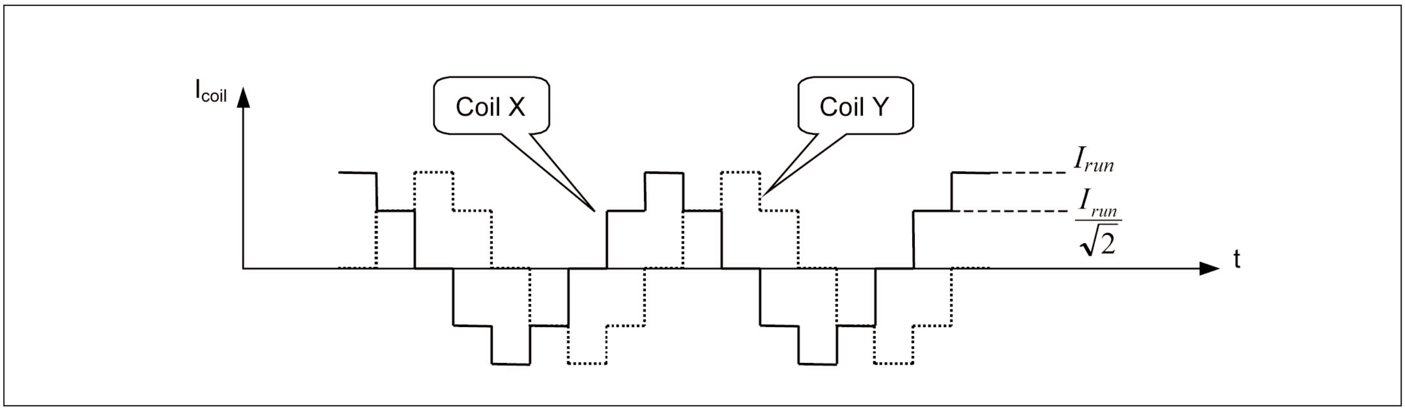
The command used to load the application parameters via the I²C bus in the RAM prior to an OTP Memory programming is **SetMotorParam**. This allows for a functional verification before using a **SetOTPparam** command to program and zap separately one OTP memory byte. A **GetOTPparam** command issued after each **SetOTPparam** command allows verifying the correct byte zapping.

Note
Zapped bits will really be "active" after a **GetOTPparam** or a **ResetToDefault** command or after a power-up.

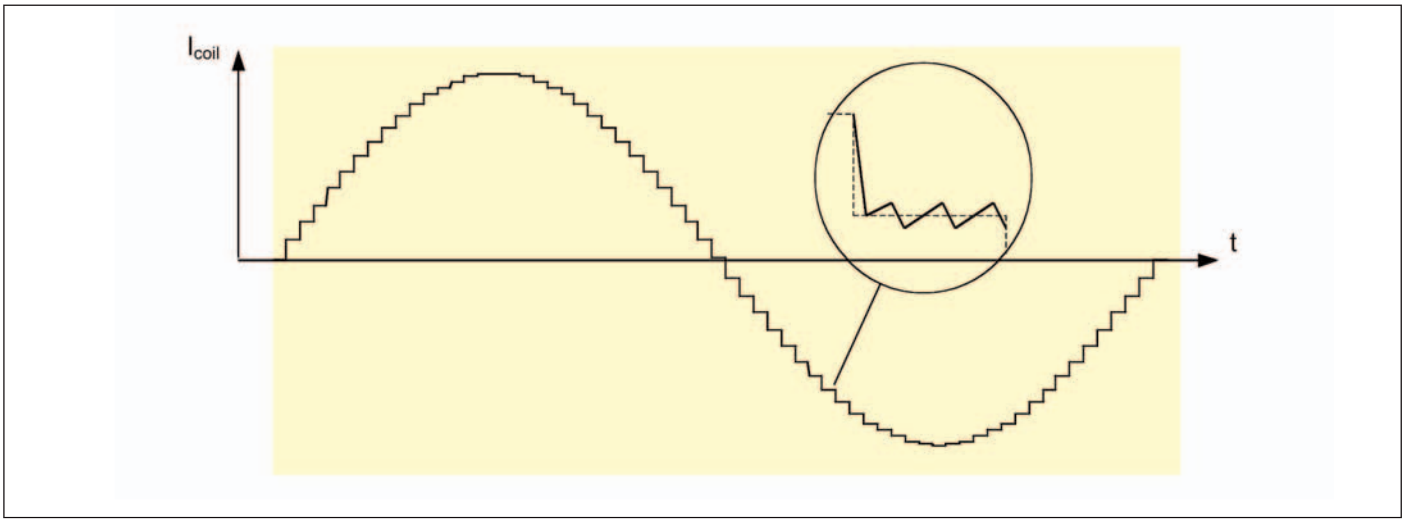
9.2.3 Motordriver

9.2.3.1 Current Waveforms in the Coils

The figure below illustrates the current fed to the motor coils by the motordriver in half-step mode.



Whereas the figure below shows the current fed to one coil in 1/16th microstepping (1 electrical period).



9.2.3.2 PWM Regulation

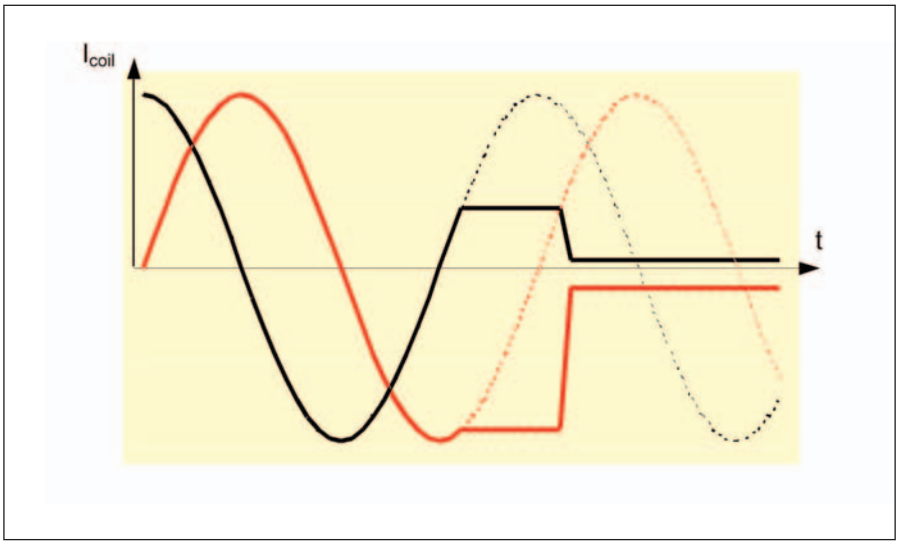
In order to force a given current (determined by **Irun** or **Ihold** and the current position of the rotor) through the motor coil while ensuring high energy transfer efficiency, a regulation based on PWM principle is used. The regulation loop performs a comparison of the sensed output current to an 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 above shows how the PWM circuit performs this regulation.

9.2.3.3 Motor Starting Phase

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

9.2.3.4 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/cos ratio between coils) during 1/4th of an electrical period at minimum velocity (thus 2 half-steps). 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.



9.2.3.5 Charge Pump Monitoring

If the charge pump voltage is not sufficient for driving the high side transistors (due to a 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.

9.2.3.6 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 allow to detect the following short circuits:

- External coil short circuit
- Short between one terminal of the coil and Vbat or Gnd
- One cannot detect internal short in the motor

Open circuits are detected by 100% PWM duty cycle value during a long time

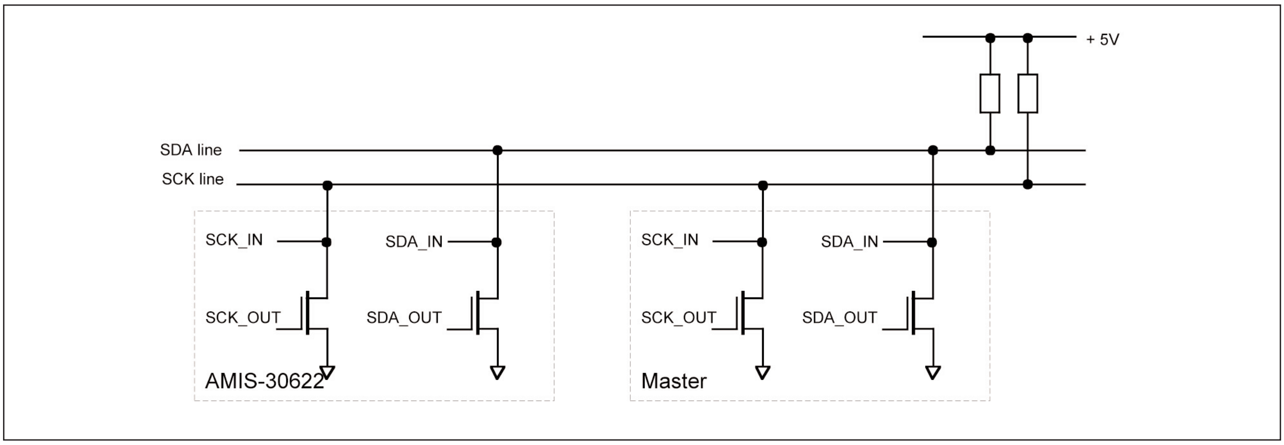
Pins	Fault Mode
Yi or Xi	Short circuit to GND
Yi or Xi	Short circuit to Vbat
Yi or Xi	Open
Y1 and Y2	Short circuited
X1 and X2	Short circuited
Xi and Yi	Short circuited

9.2.4 Inter-IC Control (I²C) Bus

The I²C interface enabled in the AMIS-30622 uses pins 1 and 2 as Data I/O and Serial Clock respectively.

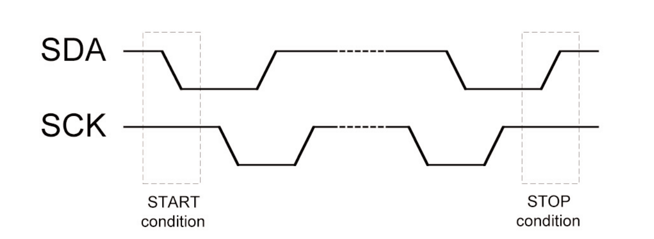
9.2.4.1 Physical Layer

Both SDA and SCK lines are connected to positive supply voltage via a current source or pull-up resistor. When there is no traffic on the bus both lines are high. Analog glitch filters are implemented to suppress spikes with a length up to 50 ns.

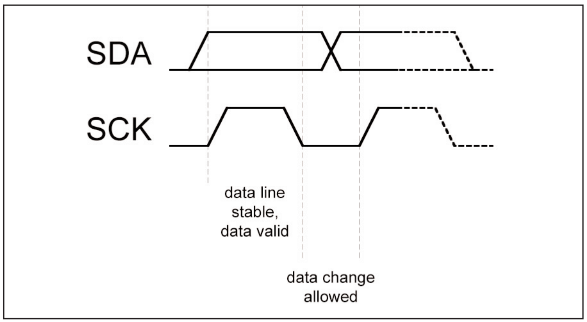


9.2.4.2 Communication on 2-wire Serial Bus Interface

Each communication starts with a Start condition and ends with a Stop condition. Both conditions are unique and cannot be confused with data. A high to low transition on the SDA line while SCK is high defines a Start condition. A low to high transition on the SDA line while SCK is high defines a Stop condition. (see figure "Start / Stop conditions" below). The master always generates the SCK clock. On every rising transition of SCK the data on SDA is valid. Data on SDA line is only allowed to change as long as SCK is low.

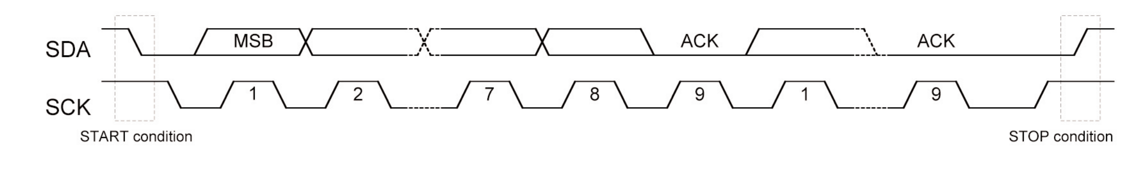


Start / Stop Conditions



Bit Transfer on 2-wire Serial Bus Interface

Every byte sent on SDA must be 8-bit, with the most significant bit (MSB) transferred first. The number of bytes that can be transmitted to the AMIS-30622 is restricted to 8 bytes. Each byte is followed by an acknowledge bit, which is issued by the receiving node (figure below).



9.2.4.3 Physical Address of the Circuit

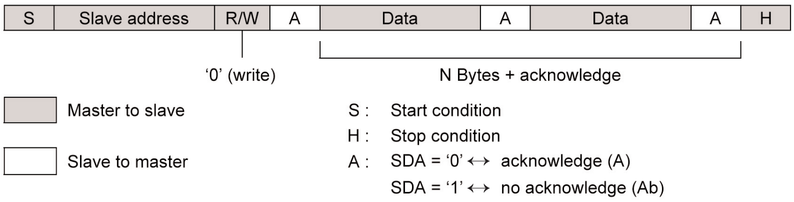
The circuit is provided with a physical address in order to discriminate this circuit from other ones on the I^{2}C bus. This address is coded on 7 bits (2 bits being internally hardwired to '1'), yielding the theoretical possibility of 32 different circuits on the same bus. It is a combination of four OTP memory bits (see § 9.2.2.13 OTP Memory Structure) and of the externally hardwired address bits (pin HW). HW must either be connected to Ground or to Vbat. When HW is not connected and left floating correct functionality of the positioner is not guaranteed. The motor will be driven to the programmed Secure Position. (see § 9.2.2.12 Application Parameters **`SecPos [10:0]`**)

The AMIS-30622 supports a "general call" address. Therefore the circuit is addressable with either the physical slave address or with address "000 0000".

AD6	AD5	AD4	AD3	AD2	AD1	AD0	Physical address
'1'	'1'	PA3	PA2	PA1	PA0		OTP memory
						HW	Hardwired bit (to Gnd or Vbat)

9.2.4.4 Write Data to AMIS-30622

A complete transmission consists of the followings: a Start condition, the slave address (7-bit), a read/write bit ('0' = write, '1' = read), and an acknowledge bit. Any further databytes 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-30622 pulls the SDA line to '0'. The AMIS-30622 reads the incoming data at SDA on every rising edge of the SCK signal. To finish the transmission the master has to transmit a Stop condition. Some commands for the AMIS-30622 are supporting 8 bytes of data, other commands are transmitting 2 bytes of data.



S	Slave address	R/W	A	Data	A	Data	A	H

'0' (write)

N Bytes + acknowledge

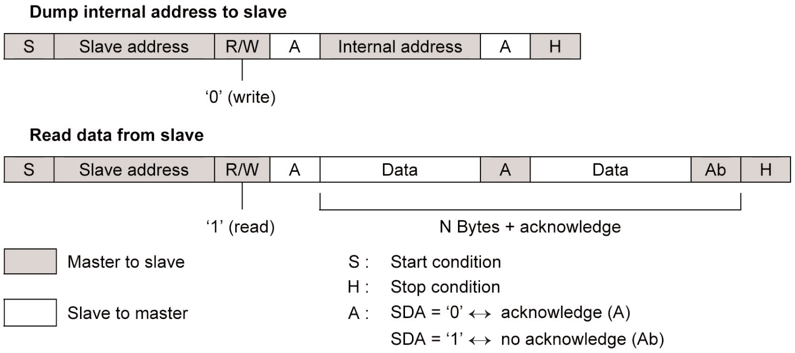
▨ Master to slave

□ Slave to master

S : Start condition
H : Stop condition
A : SDA = '0' ↔ acknowledge (A)
SDA = '1' ↔ no acknowledge (Ab)

9.2.4.5 Read Data from AMIS-30622

When reading data from a slave two transmissions are needed. 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-30622. The internal register address is stored in the circuit RAM. 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 to '0'. The last byte is not to acknowledge by the master and therefore the slave knows the end of transmission.



Dump internal address to slave

S	Slave address	R/W	A	Internal address	A	H

'0' (write)

Read data from slave

S	Slave address	R/W	A	Data	A	Data	Ab	H

'1' (read)

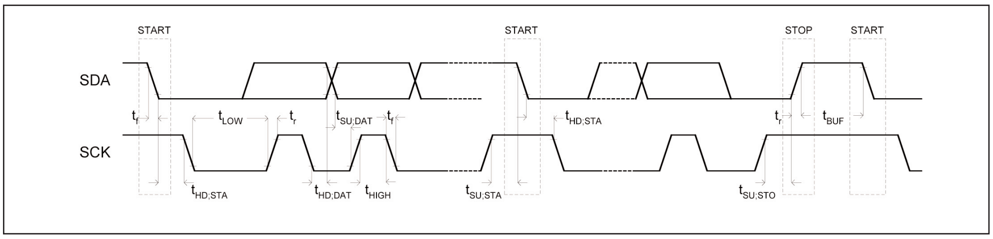
N Bytes + acknowledge

▨ Master to slave

□ Slave to master

S : Start condition
H : Stop condition
A : SDA = '0' ↔ acknowledge (A)
SDA = '1' ↔ no acknowledge (Ab)

9.2.4.6 Timing and Electrical Characteristics of the Serial Interface



See § 5 and § 6 for DC and AC parameter values.

9.2.4.7 Description of Application Commands

Communications between the AMIS-30622 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-30622

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

9.2.4.8 Command Overview

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**
RunInit	Reference search	**"1000 1000"**	**0x88**
SetMotorParam	Sets motor parameter	**"1000 1001"**	**0x89**
SetOTP	Zaps the OTP memory	**"1001 0000"**	**0x90**
SetPosition	Programmes a target and secure position	**"1000 1011"**	**0x8B**
SoftStop	Motor stopping with deceleration phase	**"1000 1111"**	**0x8F**

9.2.4.9 Commands Description

GetFullStatus1

This command is provided to the circuit by the Master to get a complete status of the circuit and of the stepper motor. The parameters sent via the 2-wire serial bus to the Master are:

- coil peak and hold currents value (**Irun** and **Ihold**)
- maximum and minimum velocities for the stepper motor (**Vmax** and **Vmin**)
- direction of movement clockwise / counter clockwise (**Shaft**)
- stepping mode (**StepMode**)
- acceleration (**deceleration**) for the stepper motor (**Acc**)
- acceleration shape (**AccShape**)
- status information (see further)
- motion status <**Motion [2:0]**>
- over current flags for coil #1 <**OVC1**> and coil #2 <**OVC2**>
- digital supply reset <**VddReset**>
- charge pump status <**CPFail**>
- external switch status <**ESW**>
- step loss <**StepLoss**>
- electrical defect <**ElDef**>
- under voltage <**UV2**>
- temperature information <**Tinfo**>
- temperature warning <**TW**>
- temperature shutdown <**TSD**>

Byte	Content	**GetFullStatus1** Command Structure							
		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	Slave Address	**1**	**1**	**OTP3**	**OTP2**	**OTP1**	**OTP0**	**HW**	**0**
1	**GetFullStatus1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**1**

Byte	Content	**GetFullStatus1** Command (Response) Structure							
		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	Slave Address	**1**	**1**	**OTP3**	**OTP2**	**OTP1**	**OTP0**	**HW**	**1**
1	Address	**1**	**1**	**1**	**1**	**OTP3**	**OTP2**	**OTP1**	**OTP0**
2	**Irun** & **Ihold**	**Irun [3:0]**				**Ihold [3:0]**			
3	**Vmax** & **Vmin**	**Vmax [3:0]**				**Vmin [3:0]**			
4	**Status 1**	**AccShape**	**StepMode[1:0]**		**Shaft**	**ACC[3:0]**			
5	**Status 2**	**VDDReset**	**StepLoss**	**ElDef**	**UV2**	**TSD**	**TW**	**Tinfo[1:0]**	
6	**Status 3**	**Motion[2:0]**			**ESW**	**OVC1**	**OVC2**	**1**	**CPFail**
7	N/A	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
8	N/A	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**

GetFullStatus2

This command is provided to the circuit by the Master to get the actual position of the stepper motor. The position is provided by the circuit in 16-bit format, with the three LSBs at '0' when in half stepping mode (**StepMode** = "00"). Furthermore programmed target position and secure position are also provided.

Notations:

- actual position of the stepper motor <**ActPos [15:0]**>
- target position of the stepper motor <**TagPos [15:0]**>
- secure position of the stepper motor <**SecPos[10:0])**>

GetFullStatus2 Command

Byte	Content				Structure				
		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	Slave Address	1	1	OTP3	OTP2	OTP1	OTP0	HW	0
1	**GetFullStatus2**	1	1	1	1	1	1	0	0

GetFullStatus2 Command (Response)

Byte	Content						Structure		
		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	Slave Address	1	1	OTP3	OTP2	OTP1	OTP0	HW	1
1	Address	1	1	1	1	OTP3	OTP2	OTP1	OTP0
2	**Actual Position 1**	ActPos[15:8]							
3	**Actual Position 2**	ActPos[7:0]							
4	**Target Position 1**	TagPos[15:0]							
5	**Target Position 2**	TagPos[7:0]							
6	**Secure Position**	SecPos[7:0]							
7	**Secure Position**	1	1	1	1	1	SecPos[10:8]		
8	N/A	1	1	1	1	1	1	1	1

GetOTPParam

This command is provided to the circuit by to read the content of an OTP Memory. For more information refer to see § 9.2.2.13 OTP Memory Structure.

GetOTPParam Command

Byte	Content				Structure				
		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	Slave Address	1	1	OTP3	OTP2	OTP1	OTP0	HW	0
1	**GetOTPParam**	1	0	0	0	0	0	1	0

GetOTPParam Command (Response)

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

GotoSecurePosition

The Master provides this command to one or all the Stepper-motors to move to the secure position **SecPos[10:0]**.

GotoSecurePosition Command

Byte	Content	Structure							
		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	Slave Address	1	1	OTP3	OTP2	OTP1	OTP0	HW	0
1	**GotoSecurePosition**	1	0	0	0	0	1	0	0

HardStop

This command is internally triggered when an electrical problem is detected in one or both coils, leading to switch off the H-bridges. If this problem is detected while the motor is moving, the <StepLoss> flag is raised allowing warning the Master that steps may have been lost at the next GetStatus command. The Master for some safety reasons can also issue a HardStop command.

HardStop Command

Byte	Content	Structure							
		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	Slave Address	1	1	OTP3	OTP2	OTP1	OTP0	HW	0
1	**HardStop**	1	0	0	0	0	1	0	1

ResetPosition

This command is provided to the circuit by the Master to reset **ActPos** and **TagPos** registers, in order to allow a positioning for an initialisation of the stepper motor position.

ResetPosition Command

Byte	Content	Structure							
		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	Slave Address	1	1	OTP3	OTP2	OTP1	OTP0	HW	0
1	**ResetPosition**	1	0	0	0	0	1	1	0

ResetToDefaults

The Master provides this command to the circuit in order to reset the whole Slave node into the initial state. ResetToDefaults will for instance **overload the RAM** with the Reset state of the Registers parameters. This is another way for the Master to initialise a slave node in case of emergency, or simply to refresh the RAM content.

ResetToDefaults Command

Byte	Content	Structure							
		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	Slave Address	1	1	OTP3	OTP2	OTP1	OTP0	HW	0
1	**ResetToDefaults**	1	0	0	0	0	1	1	1

RunInit

The Master provides this command to the circuit in order to initialize positioning of the motor by seeking the zero (or reference) position.

Once the **RunInit** command is started it cannot be interrupted by any other command, except on the occurrence of a condition leading to a motor shutdown (See § 9.2.2.7 Motor Shutdown Mode), or when a **HardStop** command is received. Furthermore the master has to check that the actual position of the stepper motor **does not** correspond to the target position of the first motion. This is very important otherwise the circuit goes into a deadlock state. Once the circuit is in deadlock state only a hardstop command followed by a **GetFullStatus1** command will cause the circuit to leave this state.

RunInit Command

Byte	Content				Structure				
		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	Slave Address	1	1	OTP3	OTP2	OTP1	OTP0	HW	0
1	**RunInit** command	1	0	0	0	1	0	0	0
2	N/A	1	1	1	1	1	1	1	1
3	N/A	1	1	1	1	1	1	1	1
4	**Vmax Vmin**	Vmax[3:0]				Vmin[3:0]			
5	Position2 byte 1	TagPos1[15:8]							
6	Position2 byte 2	TagPos1[7:0]							
7	Position1 byte 1	TagPos2[15:8]							
8	Position1 byte 2	TagPos2[7:0]							

SetMotorParam

This command is provided to the circuit by the Master to set the values for the Stepper motor parameters (listed below) in RAM.

- coil peak current value **(Irun)**
- coil hold current value **(Ihold)**
- maximum velocity for the stepper motor **(Vmax)**
- minimum velocity for the Stepper-motor (**Vmin**)
- acceleration shape (**AccShape**)
- stepping mode (**StepMode**)
- indicator of the stepper motor reference position (**Shaft**)
- acceleration (deceleration) for the stepper motor (**Acc**)
- secure position for the stepper motor (**SecPos**)

SetMotorParam Command

Byte	Content				Structure				
		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	Slave Address	1	1	OTP3	OTP2	OTP1	OTP0	HW	0
1	**SetMotorParam**	1	0	0	0	1	0	0	1
2	N/A	1	1	1	1	1	1	1	1
3	N/A	1	1	1	1	1	1	1	1
4	**Irun & I hold**	Irun[3:0]				Ihold[3:0]			
5	**Vmax & Vmin**	Vmax[3:0]				Vmin[3:0]			
6	**Status**	SecPos[10:8]		Shaft		Acc[3:0]			
7	**SecurePos**	SecPos[7:0]							
8	**StepMode**	1	1	1	AccShape	StepMode[1:0]		1	1

SetOTP

The Master provides this command to the circuit in order to zap the OTP memory.

SetOTP Command

Byte	Content				Structure				
		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	Slave Address	1	1	OTP3	OTP2	OTP1	OTP0	HW	0
1	ZapOTP	1	0	0	1	0	0	0	0
2	N/A	1	1	1	1	1	1	1	1
3	N/A	1	1	1	1	1	1	1	1
4	**OTP Address**	1	1	1	1	1	OTPA[2:0]		
5	**Pbit**	Pbit[7:0]							

SetPosition

This command is provided to the circuit by the Master to the motors to a given position relative to the zero position, defined in number of half or micro steps, according to StepMode[1:0] value.

SetPosition will not be performed if one of the following flags is set to one:

- temperature shutdown <**TSD**>
- under voltage <**UV2**>
- step loss <**StepLoss**>
- electrical defect <**ElDef**>

SetPosition Command									
Byte	Content	Structure							
		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	Slave Address	1	1	OTP3	OTP2	OTP1	OTP0	HW	0
1	**SetPosition**	1	0	0	0	1	0	1	1
2	N/A	1	1	1	1	1	1	1	1
3	N/A	1	1	1	1	1	1	1	1
4	**Position byte1**	TagPos[15:8]							
5	**Position byte2**	TagPos[7:0]							

SoftStop

If a **SoftStop** command occurs during a motion of the stepper motor, it provokes an immediate deceleration to **Vmin** followed by a stop, regardless of the position reached. This command occurs in the following cases:

- The chip temperature raises the thermal shutdown threshold.
- The Master requests a SoftStop.

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

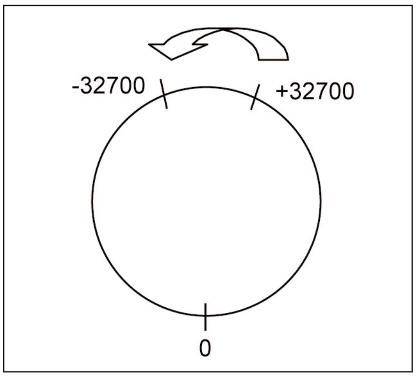
10.0 Features

10.1 Position Periodicity

Depending on the stepping mode the position can range between –4096 to +4095 in half-step mode to –32768 to +32767 in 1/16th microstepping mode (see § 8.5.1 Position Ranges) 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 minimum.

As an example in the figure below is illustrated the moving direction going from **ActPos** = +32700 to **SetPos** = –32700 is counter clockwise.

If a clockwise motion is required in this example, several consecutive **SetPosition** commands can be used.



11.0 Resistance to Electrical and Electromagnetic Disturbances

11.1 Electrostatic Discharges

See. § 1.1 Absolute Maximum Ratings

11.2 Schäffner Pulses

Shäffner pulses are applied to the power supply wires of the equipment implementing the AMIS-30622 (see application schematic), according to Renault 36-00-808/--E document.

Pulse	Amplitude	Rise Time	Pulse Duration	Rs	Operating Class
#1	-100V	≤ 1µs	2ms	10Ω	C
#2a	+100V	≤ 1µs	50µs	2Ω	B
#3a	-150V (from +13.5V)	5ns	100ns (burst)	50Ω	A
#3b	+100V (from +13.5V)	5ns	100ns (burst)	50Ω	A
#5b (load dump)	+21.5V (from +13.5V)	≤ 10ms	400ms	≤ 1Ω	C

11.3 EMC

Bulk current injection (BCI), according to Renault 36-00-808/--E document (p61).

Current	Opertaing Class
60mA	A
100mA	B
200mA	C

11.4 EMI

EMI requirement is given here as a target, since it is also PCB dependent. Any EMI issue will have to be solved on common basis with the customer.

Radiated disturbance electromagnetic quietness test, according to Renault 36-00-808/--E document:

- Permanent broadband limit (Renault 36-00-808/--E document diagram p98)
- Narrow band limit (Renault 36-00-808/--E document diagram p99)

11.5 Power Supply Micro-Interruptions

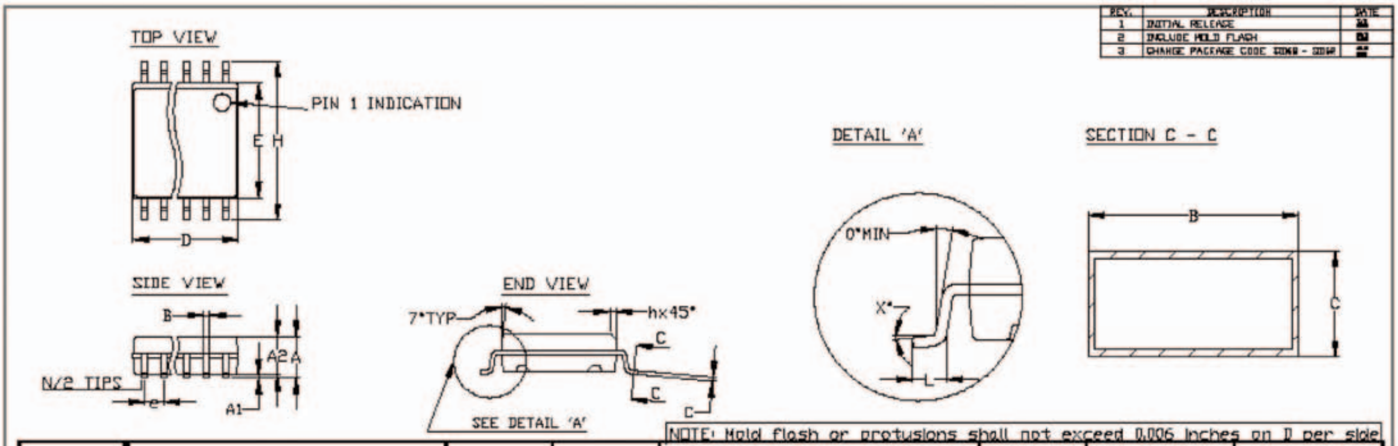
According to Renault 36-00-808/--E (p47 and followings).

Test	Operating Class
10µs micro-interruptions (1)	A
100µs micro-interruptions	B
5ms micro-interruptions	B
50ms micro-interruptions	C
300ms micro-interruptions	C

Note

(1) To achieve Class A a 100nF capacitor between Vbat and ground is needed in case HW is connected to Vbat. (see § 7 Typical Application).

12.0 Package Outlines



REV.	DESCRIPTION	DATE
1	INITIAL RELEASE	[illegible]
2	INCLUDE MOLD FLASH	[illegible]
3	CHANGE PACKAGE CODE [illegible]	[illegible]

NOTE: Mold flash or protusions shall not exceed 0,006 Inches on D per side

SYMBOL	INCHES			NOTE	NOTE	1			2	ALCATEL	ALCATEL
					VARI-		D		N	PACKAGE NR.	PACKAGE CODE
	MIN	NOM	MAX		ATIONS	MIN	NOM	MAX			
A	0,093	0,099	0,104		AA	0,398	0,405	0,412	16	87_0034	SO16B,SO16R
A1	0,004	0,008	0,012		AB	0,451	0,456	0,461	18	87_0035	SO18,SO18Q
A2	0,088	0,094	0,100		AC	0,496	0,503	0,510	20	87_0036	SO20,SO20Q
B	0,013	0,016	0,020		AD	0,599	0,606	0,613	24	87_0037	SO24,SO24Q
C	0,0090	0,0100	0,0125		AE	0,697	0,704	0,711	28	87_0038	SO28,SO28Q
D	SEE VARIATIONS			1	AF	0,810	0,818	0,822	32	87_0040	SO32,SO32Q
E	0,292	0,296	0,299								
e	0,050 BSC										
H	0,394	0,402	0,419								
h	0,010	0,015	0,019								
L	0,016	0,033	0,050								
N	SEE VARIATIONS			2							
X	0	5	8								

Note: See variations AC for dimensions D and N.

13.0 Conditioning

To be documented.