



## 1.0 General Description

---

The AMIS-30624 is a single-chip micro-stepping motor driver with a position controller and control/diagnostic interface. It is ready to build intelligent peripheral systems where up to 32 drivers can be connected to one I<sup>2</sup>C master. This significantly reduces system complexity.

The chip receives positioning instructions through the bus and subsequently drives the stator coils so the two-phase stepper motor moves to the desired position. The on-chip position controller is configurable (OTP or RAM) for different motor types, positioning ranges and parameters for speed, acceleration and deceleration. Micro-stepping allows silent motor operation and increased positioning resolution. The advanced motion qualification mode enables verification of the complete mechanical system in function of the selected motion parameters. The AMIS-30624 can easily be connected to an I<sup>2</sup>C bus where the I<sup>2</sup>C master can fetch specific status information like actual position, error flags, etc. from each individual slave node.

An integrated sensorless stall detection stops the motor when running into stall. This enables silent, yet accurate position calibrations during a referencing run and allows semi-closed loop operation when approaching the mechanical end-stops.

The chip is implemented in I2T100 technology, enabling both high voltage analog circuitry and digital functionality on the same chip. The AMIS-30624 is fully compatible with the automotive voltage requirements.

## 2.0 Product Features

---

### Motor Driver

- Micro-stepping technology
- Sensorless stall detection
- Peak current up to 800mA
- Fixed frequency PWM current-control
- Selectable PWM frequency
- Automatic selection of fast and slow decay mode
- No external fly-back diodes required
- 14V/24V compliant
- Motion qualification mode

### Controller with RAM and OTP memory

- Position controller
- Configurable speeds and acceleration
- Input to connect optional motion switch

### I<sup>2</sup>C interface

- Bi-directional 2-wire bus for Inter IC Control
- Field programmable node addresses
- Full diagnostics and status information

### Protection

- Over-current protection
- Under-voltage management
- Open circuit detection
- High-temp warning and management
- Low-temp flag

### EMI compatibility

- High voltage outputs with slope control
- HV outputs with slope control

# AMIS-30624

## 3.0 Applications

---

The AMIS-30624 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.

## 4.0 Ordering Information

---

Table 1: Ordering Information

| Part Number      | Package            | Shipping Configuration | Temperature Range | Peak Current |
|------------------|--------------------|------------------------|-------------------|--------------|
| AMIS30624C6244G  | SOIC-20            | Tube/Tray              | -40°C.....125°C   | 800mA        |
| AMIS30624C6244RG | SOIC-20            | Tape & Reel            | -40°C.....125°C   | 800mA        |
| AMIS30624C6245G  | NQFP-32 (7 x 7 mm) | Tube/Tray              | -40°C.....125°C   | 800mA        |
| AMIS30624C6245RG | NQFP-32 (7 x 7 mm) | Tape & Reel            | -40°C.....125°C   | 800mA        |

## 5.0 Quick Reference Data

---

Table 2: Absolute Maximum Ratings

| Parameter           |   | Min. | Max.               | Unit |
|---------------------|---|------|--------------------|------|
| Vbb                 | Supply voltage                                | -0.3 | +40 <sup>(1)</sup> | V    |
| Tamb                | Ambient temperature under bias <sup>(2)</sup> | -50  | +150               | °C   |
| Tst                 | Storage temperature                           | -55  | +160               | °C   |
| Vesd <sup>(3)</sup> | Electrostatic discharge voltage on pins       | -2   | +2                 | kV   |

Notes:

(1) For limited time <0.5s

(2) The circuit functionality is not guaranteed.

(3) Human body model (100pF via 1.5 kΩ, according to JEDEC EIA-JESD22-A114-B)

Table 3: Operating Ranges

| Parameter |                             | Min.      | Max. | Unit |    |
|-----------|-----------------------------|-----------|------|------|----|
| Vbb       | Supply voltage              | +8        | +29  | V    |    |
| Top       | Operating temperature range | Vbb ≤ 18V | -40  | +125 | °C |
|           |                             | Vbb ≤ 29V | -40  | +85  | °C |

# AMIS-30624

## 6.0 Block Diagram

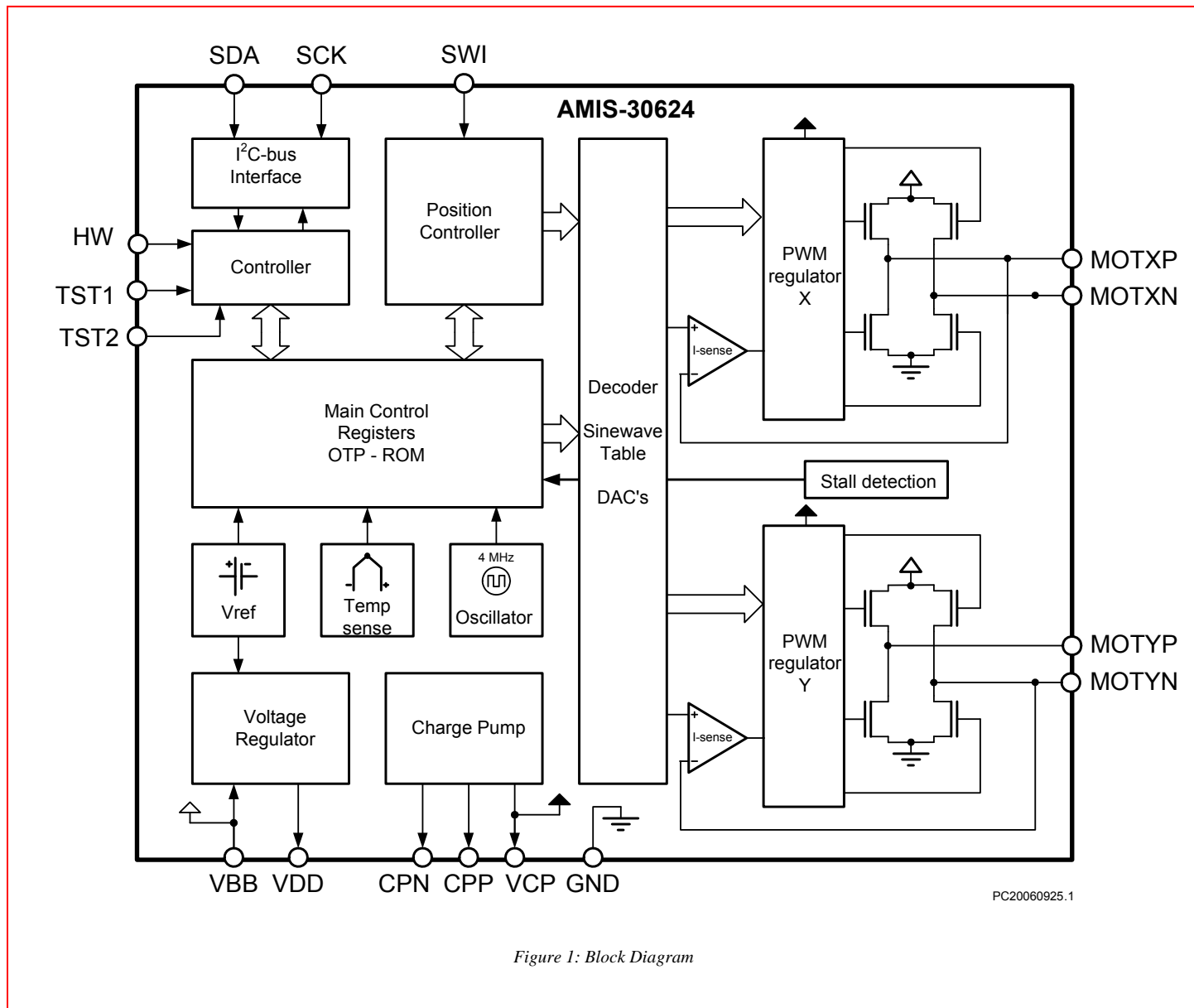


Figure 1: Block Diagram

# AMIS-30624

## 7.0 Pin Out

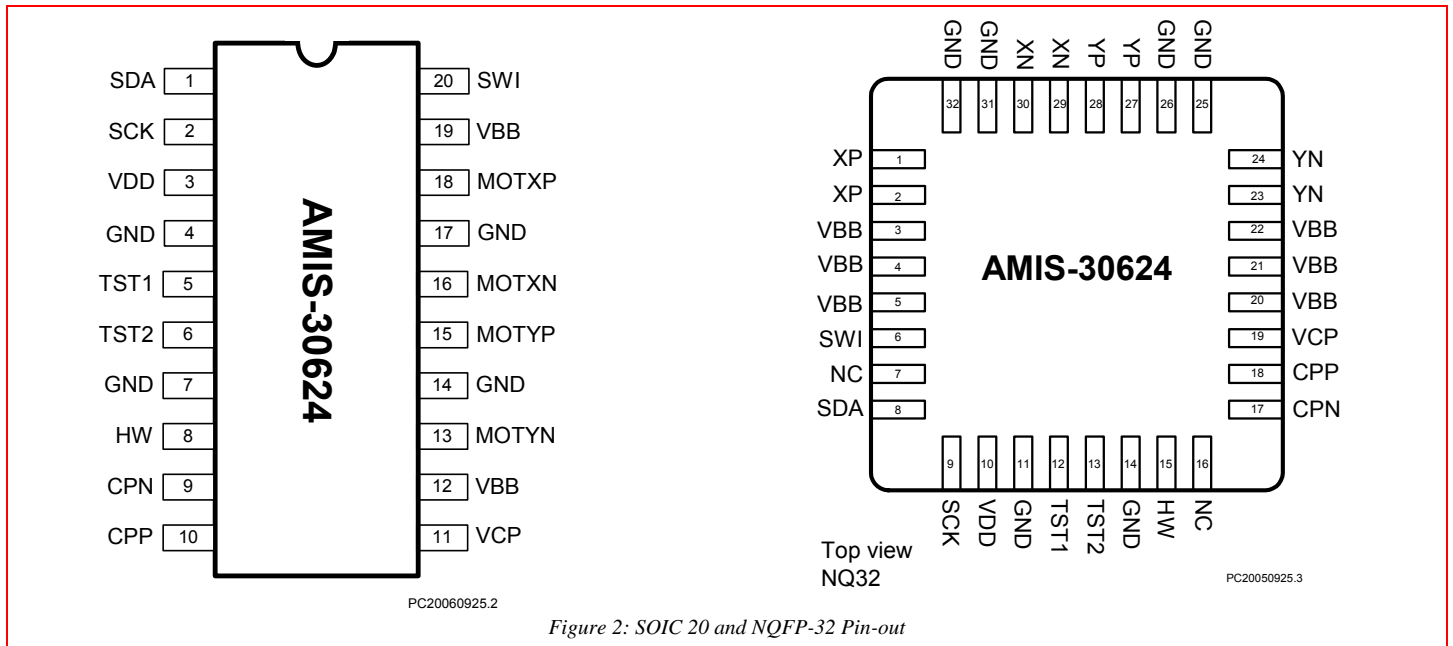


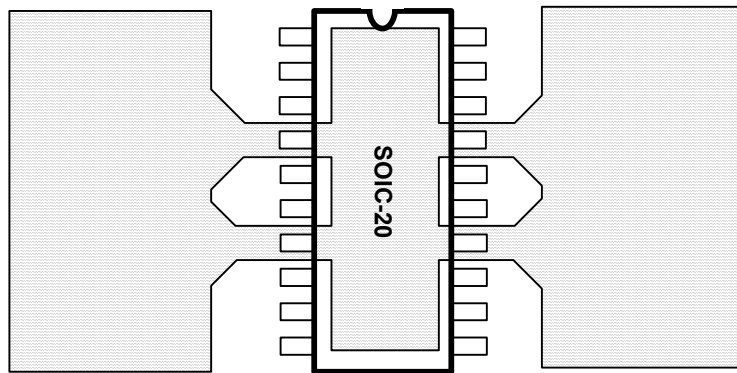
Table 4: Pin Description

| Pin Name | Pin Description  | SOIC-20   | NQFP-32                |
|----------|--|-----------|------------------------|
| SDA      | I <sup>2</sup> C serial data line                                    | 1         | 8                      |
| SCK      | I <sup>2</sup> C serial clock line                                   | 2         | 9                      |
| VDD      | 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                     |
| VBB      | 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                  |

## 8.0 Package Thermal Resistance

### 8.1 SOIC-20

To lower the junction-to-ambient thermal resistance, it is recommended to connect the ground leads to a printed circuit board (PCB) ground plane layout as illustrated in Figure 3. The junction-to-case thermal resistance is dependent on the copper area, copper thickness, PCB thickness and number of copper layers. Calculating with a total area of  $460 \text{ mm}^2$ ,  $35\mu\text{m}$  copper thickness,  $1.6\text{mm}$  PCB thickness and 1 layer, the thermal resistance is  $28^\circ\text{C/W}$ ; leading to a junction-ambient thermal resistance of  $63^\circ\text{C/W}$ .

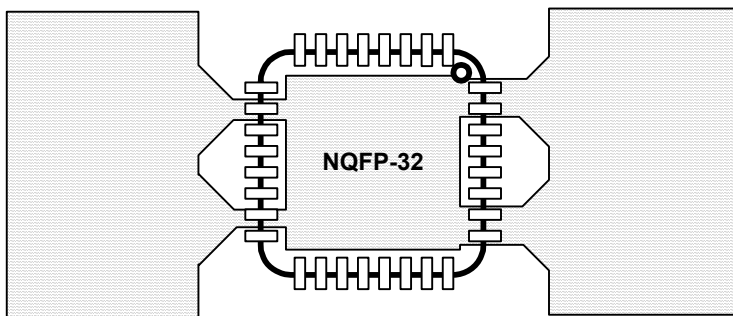


PC20041128.1

Figure 3: PCB Ground Plane Layout Condition

### 8.2 NQFP-32

The NQFP is designed to provide superior thermal performance, and using an exposed die pad on the bottom surface of the package partly contributes to this. In order to take full advantage of this thermal performance, the PCB must have features to conduct heat away from the package. A thermal grounded pad with thermal vias can achieve this. With a layout as shown in Figure 4, the thermal resistance junction – to – ambient can be brought down to a level of  $25^\circ\text{C/W}$ .



PC20041128.2

Figure 4: PCB Ground Plane Layout Condition

# AMIS-30624

## 9.0 DC Parameters

The DC parameters are given for V<sub>bb</sub> and temperature in their operating ranges. Currents flowing in the circuit are defined as positive.

Table 5: DC Parameters

| Symbol   | Pin(s)   | Parameter   | Test Conditions                               | Min.                  | Typ.                  | Max.                  | Unit |
|--|--|---|---|-----------------------|-----------------------|-----------------------|------|
| <b>Motor Driver</b>                            |  |   |   |                       |                       |                       |      |
| I <sub>MSmax,Peak</sub>                        | MOTXP<br>MOTXN<br>MOTYP<br>MOTYN               | Max. current through motor coil in normal operation                             |   |                       | 800                   |                       | mA   |
| I <sub>MSmax,RMS</sub>                         |  | Max. RMS current through coil in normal operation                               |   |                       | 570                   |                       | mA   |
| I <sub>MSabs</sub>                             |  | Absolute error on coil current  |   | -10                   |                       | 10                    | %    |
| I <sub>MSrel</sub>                             |  | Error on current ratio I <sub>coilx</sub> / I <sub>coily</sub>                  |   | -7                    |                       | 7                     | %    |
| R <sub>DSon</sub>                              |  | On resistance for each motor pin (including bond wire) at I <sub>MSmax</sub>    | V <sub>bb</sub> = 12V, T <sub>j</sub> = 50 °C |                       | 0.50                  | 1                     | Ω    |
|  |  |   | V <sub>bb</sub> = 8V, T <sub>j</sub> = 50 °C  |                       | 0.55                  | 1                     | Ω    |
|  | V <sub>bb</sub> = 12V, T <sub>j</sub> = 150 °C |   |   | 0.70                  | 1                     | Ω                     |      |
|  | V <sub>bb</sub> = 8V, T <sub>j</sub> = 150 °C  |   |   | 0.85                  | 1                     | Ω                     |      |
| I <sub>MSL</sub>                               | Pull down current                              | HiZ mode  |   | 2                     |                       | mA                    |      |
| <b>Thermal Warning &amp; Shutdown</b>          |  |   |   |                       |                       |                       |      |
| T <sub>tw</sub>                                |  | Thermal warning   |   | 138                   | 145                   | 152                   | °C   |
| T <sub>tsd</sub> <sup>(1)(2)</sup>             |  | Thermal shutdown  |   |                       | T <sub>tw</sub> + 10  |                       | °C   |
| T <sub>low</sub> <sup>(2)</sup>                |  | Low temperature warning   |   |                       | T <sub>tw</sub> - 155 |                       | °C   |
| <b>Supply and Voltage Regulator</b>            |  |   |   |                       |                       |                       |      |
| V <sub>bb</sub>                                | VBB  | Nominal operating supply range  | T <sub>amb</sub> ≤ 125 °C                     | 6.5                   |                       | 18                    | V    |
|  |  |   | T <sub>amb</sub> ≤ 85 °C                      | 6.5                   |                       | 29                    | V    |
| V <sub>bbOTP</sub>                             |  | Supply voltage for OTP zapping <sup>(3)</sup>                                   |   | 9.0                   |                       | 10.0                  | V    |
| I <sub>bat</sub>                               |  | Total current consumption   | Unloaded outputs                              |                       | 3.50                  | 10.0                  | mA   |
| I <sub>bat_s</sub>                             |  | Sleep mode current consumption  |   |                       | 50                    | 100                   | μA   |
| UV <sub>1</sub>                                |  | Stop voltage high threshold   |   | 7.8                   | 8.4                   | 8.9                   | V    |
| UV <sub>2</sub>                                |  | Stop voltage low threshold  |   | 7.1                   | 7.5                   | 8.0                   | V    |
| V <sub>dd</sub>                                | VDD  | Internal regulated output <sup>(4)</sup>  | 8V < V <sub>bb</sub> < 29V                    | 4.75                  | 5                     | 5.50                  | V    |
| I <sub>ddStop</sub>                            |  | Digital current consumption   | V <sub>bb</sub> < UV <sub>2</sub>             |                       | 2                     |                       | mA   |
| V <sub>ddReset</sub>                           |  | Digital supply reset level @ power down <sup>(5)</sup>                          |   |                       |                       | 4.5                   | V    |
| I <sub>ddLim</sub>                             |  | Current limitation  | Pin shorted to ground                         |                       |                       | 42                    | mA   |
| <b>Switch Input and Hardware Address Input</b> |  |   |   |                       |                       |                       |      |
| R <sub>t_OFF</sub>                             | SWI<br>HW                                      | Switch OFF resistance <sup>(6)</sup>  | Switch to Gnd or V <sub>bat</sub>             | 10                    |                       |                       | kΩ   |
| R <sub>t_ON</sub>                              |  | Switch ON resistance <sup>(6)</sup>   |   |                       |                       | 2                     | kΩ   |
| V <sub>bb_sw</sub>                             |  | V <sub>bb</sub> range for guaranteed operation of SWI and HW                    |   | 6                     |                       | 29                    | V    |
| V <sub>max_sw</sub>                            |  | Maximum voltage   | T < 1s  |                       |                       | 40V                   | V    |
| I <sub>lim_sw</sub>                            |  | Current limitation  | Short to Gnd or V <sub>bat</sub>              |                       | 30                    |                       | mA   |
| <b>I<sup>2</sup>C Serial Interface</b>         |  |   |   |                       |                       |                       |      |
| V <sub>IL</sub>                                | SDA<br>SCK                                     | Input level low <sup>(7)</sup>  |   | -0.5                  |                       | 0.3 * V <sub>dd</sub> | V    |
| V <sub>IH</sub>                                |  | Input level high <sup>(8)</sup>   |   | 0.7 * V <sub>dd</sub> |                       | V <sub>dd</sub> + 0.5 | V    |
| V <sub>nL</sub>                                |  | Noise margin at the LOW level for each connected device (including hysteresis)  |   | 0.1 * V <sub>dd</sub> |                       |                       | V    |
| V <sub>nH</sub>                                |  | Noise margin at the HIGH level for each connected device (including hysteresis) |   | 0.2 * V <sub>dd</sub> |                       |                       | V    |

# AMIS-30624

Table 5: DC Parameters (cont.)

| Charge Pump                      |         |   |                                       |                         |                           |                     |    |
|----------------------------------|---------|---|---------------------------------------|-------------------------|---------------------------|---------------------|----|
| V <sub>cp</sub>                  | VCP     | Output voltage                              | V <sub>bb</sub> > 15V                 | V <sub>bb</sub> +10     | V <sub>bb</sub> +12.5     | V <sub>bb</sub> +15 | V  |
|                                  |         |   | 8V < V <sub>bb</sub> < 15V            | 2 * V <sub>bb</sub> - 5 | 2 * V <sub>bb</sub> - 2.5 | 2 * V <sub>bb</sub> | V  |
| C <sub>buffer</sub>              |         | External buffer capacitor                   |                                       | 220                     |                           | 470                 | nF |
| C <sub>pump</sub>                | CPP CPN | External pump capacitor                     |                                       | 220                     |                           | 470                 | nF |
| Motion Qualification Mode Output |         |   |                                       |                         |                           |                     |    |
| V <sub>OUT</sub>                 | SWI     | Output voltage swing                        | TestBemf I <sup>2</sup> C command     |                         | 0 - 4,85                  |                     | V  |
| R <sub>OUT</sub>                 |         | Output impedance                            | Service mode I <sup>2</sup> C command |                         | 2                         |                     | kΩ |
| Av                               |         | Gain = V <sub>SWI</sub> / V <sub>BEMF</sub> | Service mode I <sup>2</sup> C command |                         | 0,50                      |                     |    |

Notes:

- (1) No more than 100 cumulated hours in life time above T<sub>tsd</sub>.
- (2) Thermal shutdown and a low temperature warning are derived from thermal warning.
- (3) A 10μF buffer capacitor of between VBB and GND is the minimum needed. Short connections to the power supply are recommended.
- (4) Pin VDD must not be used for any external supply
- (5) The RAM content will not be altered above this voltage.
- (6) External resistance value seen from pin SWI or HW, including 1kΩ series resistor.
- (7) If input voltages < - 0.3V, than a resistor between 22Ω to 100Ω needs to be put in series
- (8) If the I<sup>2</sup>C-bus is operated in Fast Mode V<sub>IHmin</sub> = 0.7 \* Vdd

# AMIS-30624

## 10.0 AC Parameters

The AC parameters are given for  $V_{bb}$  and temperature in their operating ranges. All timing values of the I<sup>2</sup>C transceiver are referred to  $V_{IHmax}$  and  $V_{ILmax}$  levels (see Figure 5).

Table 6: AC Parameters

| Symbol  | Pin(s)     | Parameter   | Test Conditions      | Min.               | Typ. | Max.                | Unit    |
|---|------------|---|----------------------|--------------------|------|---------------------|---------|
| <b>Power-up</b>                                   |            |   |                      |                    |      |                     |         |
| $T_{pu}$  |            | Power-up time   | Guaranteed by design |                    |      | 10                  | ms      |
| <b>Internal Oscillator</b>                        |            |   |                      |                    |      |                     |         |
| $f_{osc}$   |            | Frequency of internal oscillator  |                      | 3.6                | 4.0  | 4.4                 | MHz     |
| <b>I<sup>2</sup>C Transceiver (Generic)</b>       |            |   |                      |                    |      |                     |         |
| $C_B$   | SDA<br>SCK | Capacitive load of each bus line  |                      |                    |      | 400 <sup>(1)</sup>  | pF      |
| $C_I$   |            | Capacitance of SDA / SCK pin  |                      |                    |      | 10                  | pF      |
| $t_{SP}$  |            | Pulse width of spikes which must be suppressed by the input filter                          |                      | 50                 |      |                     | ns      |
| <b>I<sup>2</sup>C Transceiver (Standard Mode)</b> |            |   |                      |                    |      |                     |         |
| $f_{SCL}$   | SDA<br>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                |      |                     | $\mu$ s |
| $t_{LOW}$   |            | LOW period of the SCK clock   |                      | 4.7                |      |                     | $\mu$ s |
| $t_{HIGH}$  |            | HIGH period of the SCK clock  |                      | 4.0                |      |                     | $\mu$ s |
| $t_{SU,START}$                                    |            | Set-up time for a repeated START condition  |                      | 4.7                |      |                     | $\mu$ s |
| $t_{HD,DATA}$                                     |            | Data hold time for I <sup>2</sup> C bus devices   |                      | 0 <sup>(2)</sup>   |      | 3.45 <sup>(3)</sup> | $\mu$ s |
| $t_{SU,DATA}$                                     |            | Data set-up time  |                      | 250                |      |                     | ns      |
| $t_R$   |            | Rise time of SDA and SCK signals  |                      |                    |      | 1.0                 | $\mu$ s |
| $t_F$   |            | Fall time of SDA and SCK signals  |                      |                    |      | 0.3                 | $\mu$ s |
| $t_{SU,STOP}$                                     |            | Set-up time for STOP condition  |                      | 4.0                |      |                     | $\mu$ s |
| $t_{BUF}$   |            | Bus free time between STOP and START condition  |                      | 4.7                |      |                     | $\mu$ s |
| <b>I<sup>2</sup>C Transceiver (Fast Mode)</b>     |            |   |                      |                    |      |                     |         |
| $f_{SCL}$   | SDA<br>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                |      |                     | $\mu$ s |
| $t_{LOW}$   |            | LOW period of the SCK clock   |                      | 1.3                |      |                     | $\mu$ s |
| $t_{HIGH}$  |            | HIGH period of the SCK clock  |                      | 0.6                |      |                     | $\mu$ s |
| $t_{SU,START}$                                    |            | Set-up time for a repeated START condition  |                      | 0.6                |      |                     | $\mu$ s |
| $t_{HD,DATA}$                                     |            | Data hold time for I <sup>2</sup> C bus devices   |                      | 0 <sup>(2)</sup>   |      | 0.9 <sup>(3)</sup>  | $\mu$ s |
| $t_{SU,DATA}$                                     |            | Data set-up time  |                      | 100 <sup>(4)</sup> |      |                     | ns      |
| $t_R$   |            | Rise time of SDA and SCK signals  |                      | $20 + 0.1C_B$      |      | 300                 | ns      |
| $t_F$   |            | Fall time of SDA and SCK signals  |                      | $20 + 0.1C_B$      |      | 300                 | ns      |
| $t_{SU,STOP}$                                     |            | Set-up time for STOP condition  |                      | 0.6                |      |                     | $\mu$ s |
| $t_{BUF}$   |            | Bus free time between STOP and START condition  |                      | 1.3                |      |                     | $\mu$ s |



# AMIS-30624

Table 6: AC Parameters (cont.)

| Switch Input and Hardware Address Input |         |                                      |                            |      |      |      |         |
|---|---------|--------------------------------------|----------------------------|------|------|------|---------|
| $T_{sw}$                                | SWI HW2 | Scan pulse period <sup>(5)</sup>     |                            |      | 1024 |      | $\mu s$ |
| $T_{sw\_on}$                            |         | Scan pulse duration                  |                            |      | 128  |      | $\mu s$ |
| Motor driver                            |         |                                      |                            |      |      |      |         |
| $F_{pwm}$                               | MOTxx   | PWM frequency <sup>(5)</sup>         | PWMfreq = 0 <sup>(6)</sup> | 20.6 | 22.8 | 25.0 | kHz     |
|   |         |                                      | PWMfreq = 1 <sup>(6)</sup> | 41.2 | 45.6 | 50.0 | kHz     |
| $F_{jit\_depth}$                        |         | PWM jitter modulation depth          | PWMJen = 1 <sup>(6)</sup>  |      | 10   |      | %       |
| $T_{brise}$                             |         | Turn-on transient time               | Between 10% and 90%        |      | 170  |      | ns      |
| $T_{bfall}$                             |         | Turn-off transient time              |                            |      | 140  |      | ns      |
| $T_{stab}$                              |         | Run current stabilization time       |                            | 29   | 32   | 35   | ms      |
| Charge Pump                             |         |                                      |                            |      |      |      |         |
| $f_{CP}$                                | CPN CPP | Charge pump frequency <sup>(5)</sup> |                            |      | 250  |      | kHz     |

- Notes:
- (1) The maximum number of connected I<sup>2</sup>C 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.
  - (2) An I<sup>2</sup>C device must internally provide a hold time of at least 300ns for the SDA signal (referred to the  $V_{IHmin}$  of the SCL signal) to bridge the undefined region of the falling edge of SCL.
  - (3) 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.
  - (4) A Fast-mode I<sup>2</sup>C-bus device can be used in a standard-mode I<sup>2</sup>C bus system, but the requirement  $t_{SU,DATA} \geq 250ns$  must 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  $t_{rmax} + t_{SU,DATA} = 1000 + 250 = 1250ns$  (according to the standard-mode I<sup>2</sup>C-bus specification) before the SCL line is released.
  - (5) Derived from internal oscillator.
  - (6) See [SetMotorParam](#) and [PWM regulator](#).

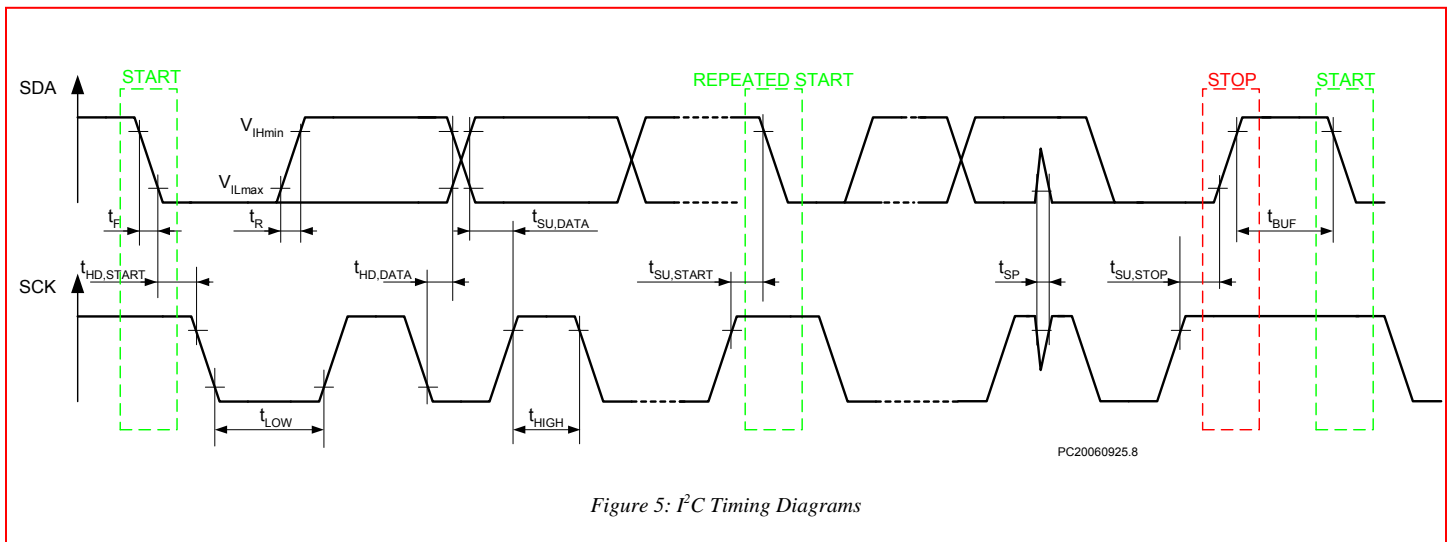


Figure 5: I<sup>2</sup>C Timing Diagrams

# AMIS-30624

## 11.0 Typical Application

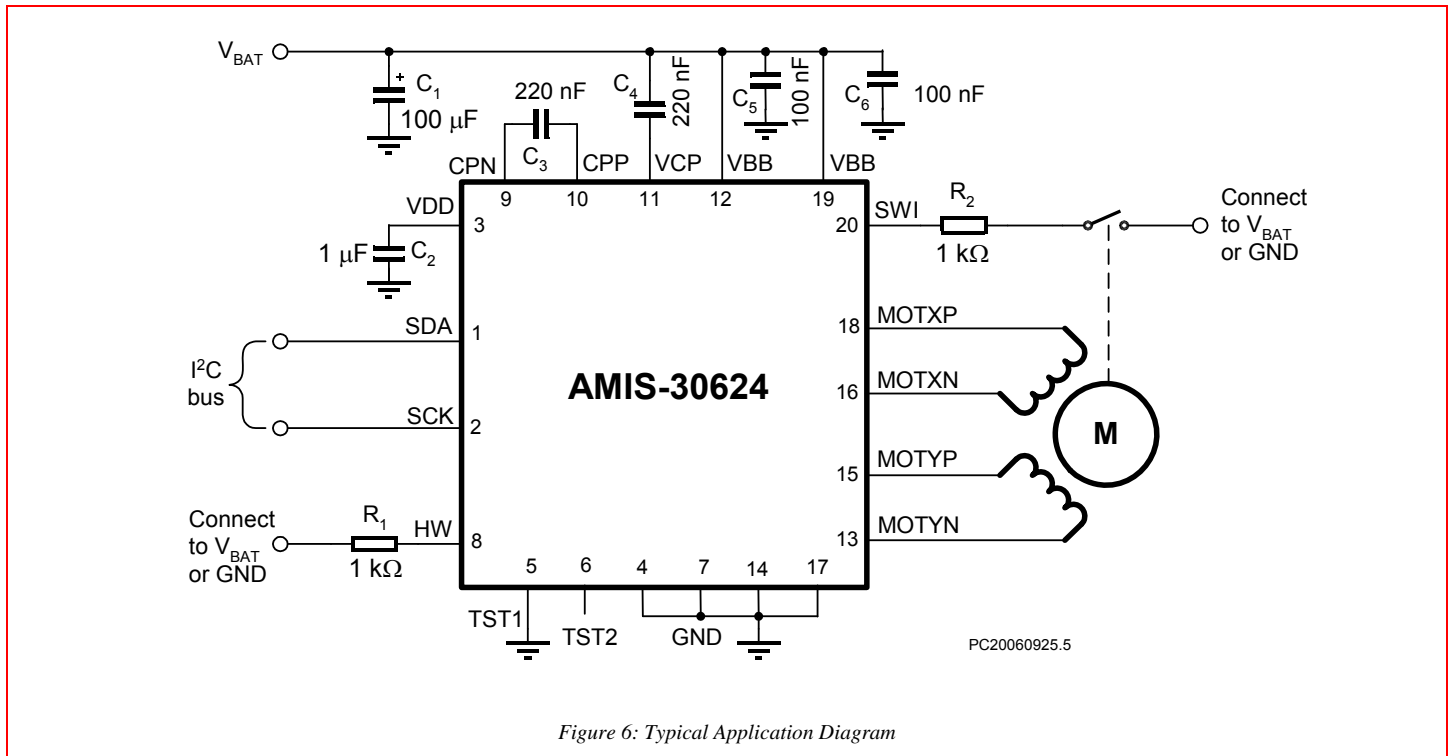


Figure 6: Typical Application Diagram

### Notes:

- (1) All resistors are  $\pm 5\%$ ,  $\frac{1}{4}$  W.
- (2) Depending on the application, the ESR value and working voltage of  $C_1$  must be carefully chosen.
- (3)  $C_2$  must be a ceramic capacitor to assure low ESR.
- (4)  $C_3$  and  $C_4$  must be as close as possible to pins CPN, CPP, VCP, and VBB to reduce EMC radiation.
- (5)  $C_5$  and  $C_6$  must be close to pins VBB and GND.

## 12.0 Positioning Parameters

### 12.1 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

# AMIS-30624

## 12.2 Maximum Velocity

For each stepping mode, the maximum velocity  $V_{max}$  can be programmed to 16 possible values given in Table 7.

The accuracy of  $V_{max}$  is derived from the internal oscillator. Under special circumstances it is possible to change the  $V_{max}$  parameter while a motion is ongoing. All 16 entries for the  $V_{max}$  parameter are divided into four groups. When changing  $V_{max}$  during a motion the application must take care that the new  $V_{max}$  parameter stays within the same group, otherwise steps might be lost.

Table 7: Maximum Velocity Selection Table

| Vmax Index |     |                    | Group | Stepping Mode               |   |   |  |
|------------|-----|--------------------|-------|-----------------------------|---|---|--|
| Hex        | Dec | Vmax (Full-step/s) |       | Half stepping (Half-step/s) | 1/4 <sup>th</sup> Micro-stepping (Micro-step/s) | 1/8 <sup>th</sup> Micro-stepping (Micro-step/s) | 1/16 <sup>th</sup> 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  |

## 12.3 Minimum Velocity

Once the maximum velocity is chosen, 16 possible values can be programmed for the minimum velocity  $V_{min}$ . Table 8 provides the obtainable values in full-step/s. The accuracy of  $V_{min}$  is derived from the internal oscillator.

Table 8: Obtainable Values in Full-step/s for the Minimum Velocity

| Vmin Index |     | Vmax Factor | Vmax (Full-step/s) |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
|------------|-----|-------------|--------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Hex        | Dec |             | A                  |     |     |     | B   |     |     |     | C   |     |     |     | D   |     |     |     |
|            |     |             | 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:
- (1) The Vmax factor is an approximation.
  - (2) In case of motion without acceleration (**AccShape = 1**) the length of the steps =  $1/V_{min}$ . In case of accelerated motion (**AccShape = 0**) the length of the first step is shorter than  $1/V_{min}$  depending of **Vmin**, **Vmax** and **Acc**.

# AMIS-30624

## 12.4 Acceleration and Deceleration

Sixteen possible values can be programmed for Acc (acceleration and deceleration between  $V_{min}$  and  $V_{max}$ ). Table 9 provides the obtainable values in full-step/s<sup>2</sup>. One observes restrictions for some combination 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 <sup>2</sup> ) |       |     |     |     |     |     |     |     |       |     |     |     |       |     |     |  |
| 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   | 19092                                    |       |     |     |     |     |     |     |     |       |     |     |     |       |     |     |  |
| A             | 10  | 14785                                    |       |     |     |     |     |     |     |     |       |     |     |     |       |     |     |  |
| B             | 11  |  | 21886 |     |     |     |     |     |     |     |       |     |     |     |       |     |     |  |
| C             | 12  |  | 24447 |     |     |     |     |     |     |     |       |     |     |     |       |     |     |  |
| D             | 13  |  | 27008 |     |     |     |     |     |     |     |       |     |     |     |       |     |     |  |
| E             | 14  |  | 29570 |     |     |     |     |     |     |     |       |     |     |     |       |     |     |  |
| F             | 15  |  | 29570 |     |     |     |     |     |     |     | 34925 |     |     |     | 40047 |     |     |  |

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

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

## 12.5 Positioning

The position is programmed in the command [SetPosition](#) and is given as a number of (micro)steps. According to the chosen stepping mode, the position word must be aligned as described in Table 10. When using command [GotoSecurePosition](#), data is automatically aligned.

Table 10: Position Word Alignment

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

- Notes:
- (1) LSB: Least significant bit
  - (2) S: Sign bit

# AMIS-30624

## 12.5.1. 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 Table 11.

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             |
|                  | 1/4 <sup>th</sup> micro-stepping  | -8192 to +8191   | 16384 micro-steps    | 14             |
|                  | 1/8 <sup>th</sup> micro-stepping  | -16384 to +16383 | 32768 micro-steps    | 15             |
|                  | 1/16 <sup>th</sup> micro-stepping | -32768 to +32767 | 65536 micro-steps    | 16             |
| SetPositionShort | Half-stepping                     | -1024 to +1023   | 2048 half-steps      | 11             |

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.

## 12.5.2. Secure Position

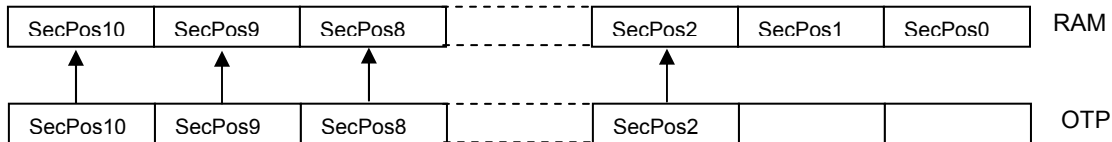
A secure position can be programmed. It is coded in 11-bits, thus having a lower resolution than normal positions, as shown in Table 12. See also the command [GotoSecurePosition](#).

Table 12: Secure Position

| Stepping Mode                     | Secure Position Resolution           |
|-----------------------------------|--------------------------------------|
| Half-stepping                     | 4 half-steps                         |
| 1/4 <sup>th</sup> micro-stepping  | 8 micro-steps (1/4 <sup>th</sup> )   |
| 1/8 <sup>th</sup> micro-stepping  | 16 micro-steps (1/8 <sup>th</sup> )  |
| 1/16 <sup>th</sup> micro-stepping | 32 micro-steps (1/16 <sup>th</sup> ) |

### Important Notes:

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



## 12.5.3. Shaft

A shaft bit which can be programmed in [OTP](#) or with command [SetMotorParam](#), defines whether a positive motion is a clockwise or counter-clockwise rotation (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.

## 13.0 Structural Description

---

See the block diagram in Figure 1.

### 13.1 Stepper Motor Driver

The motor driver 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 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 information to the control logic part. The same applies for the detection and reporting of an electrical problem that could occur on the coils or the charge pump.

### 13.2 Control Logic (Position Controller and Main Control)

The control logic block stores the information provided by the I<sup>2</sup>C 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 motor driver 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 motor driver part in order to manage possible problems and decide on internal actions and reporting to the I<sup>2</sup>C interface.

### 13.3 Motion Detection

Motion detection is based on the back emf, generated internally in the running motor. When the motor is blocked, for example when it hits the end-position, the velocity and as a result also the generated back emf, is disturbed. The AMIS-30624 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.

### 13.4 Miscellaneous

The AMIS-30624 also contains the following:

- An internal oscillator needed for the control logic and the PWM control of the motor driver
- 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

## 14.0 Functional Description

This chapter describes the following functional blocks in more detail:

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

The motion detection and I<sup>2</sup>C control are discussed in separate chapters.

### 14.1 Position Controller

#### 14.1.1. 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 ( $V_{min}$ ) to maximum velocity ( $V_{max}$ ), 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 configuration phase. The current in the coils is also programmable.

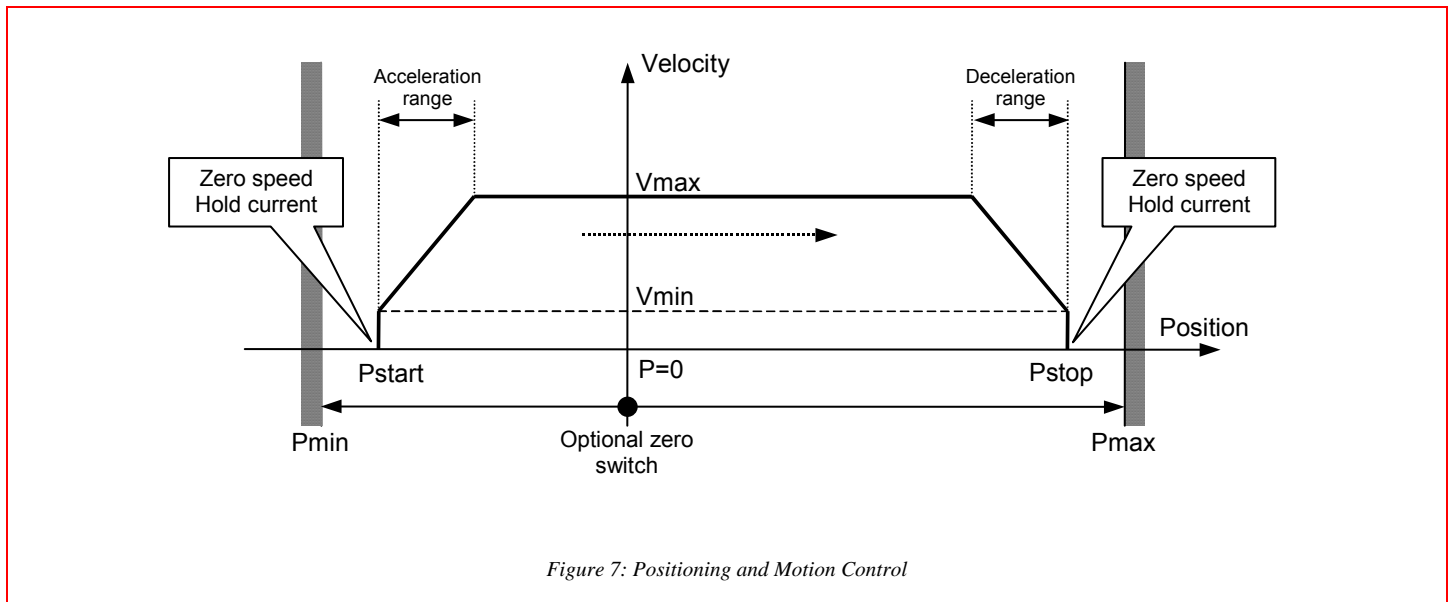


Figure 7: Positioning and Motion Control

Table 13: Position Related Parameters

| Parameter                     | Reference   |
|-------------------------------|---|
| $P_{max} - P_{min}$           | <a href="#">See Positioning</a>                   |
| Zero speed hold current       | <a href="#">See Ihold</a>                         |
| Maximum current               | <a href="#">See Irun</a>                          |
| Acceleration and deceleration | <a href="#">See Acceleration and deceleration</a> |
| $V_{min}$                     | <a href="#">See Minimum velocity</a>              |
| $V_{max}$                     | <a href="#">See Maximum velocity</a>              |

# AMIS-30624

Different positioning examples are shown in Table 14.

Table 14: Positioning Examples

| 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<br>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.   |  |



# AMIS-30624

## 14.1.2. Dual Positioning

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

A second relative motion to a position  $Pos1[15:0] + Pos2[15:0]$  is done at the specified  $V_{min}$  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.

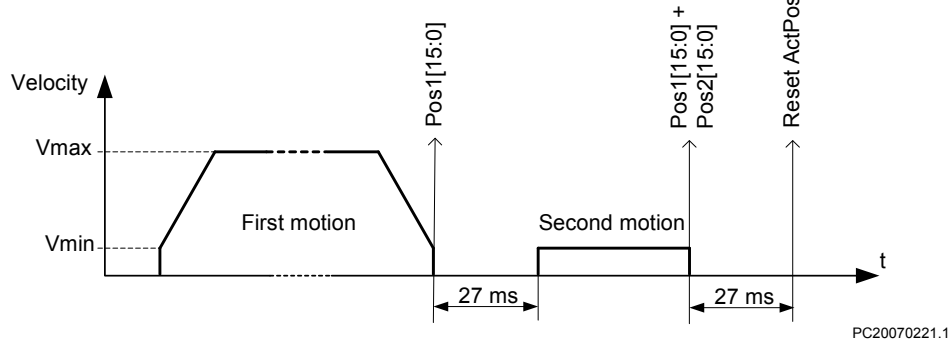


Figure 8: Dual Positioning

**Remark:** This operation cannot be interrupted or influenced by any further command unless the conditions exist to cause a [motor shutdown](#) or by a HardStop command. Sending a [SetDualPosition](#) command while a motion is already ongoing is not recommended.

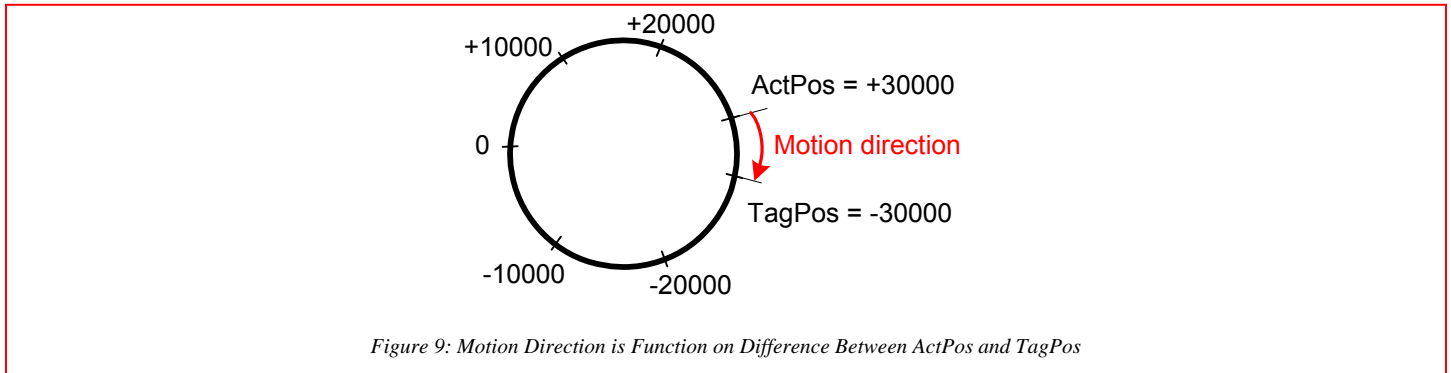
- Notes:**
- (0) The [priority encoder](#) describes the management of states and commands. All notes below are to be considered illustrative.
  - (1) The last [SetPosition](#) command issued during a DualPosition sequence will be kept in memory and executed afterwards. This also applies to the commands [SetMotorParam](#) and [GotoSecurePosition](#).
  - (2) Commands such as [GetFullStatus1](#) or [GetFullStatus2](#) will be executed while a Dual Positioning is running.
  - (3) A DualPosition sequence starts by setting TagPos register to SecPos value, provided secure position is enabled otherwise TagPos is reset to zero.
  - (4) The acceleration/deceleration value applied during a DualPosition sequence is the one stored in RAM before the [SetDualPosition](#) command is sent. The same applies for Shaft bit, but not for Irun, Ihold and StepMode, which can be changed during the Dual Positioning sequence.
  - (5) The Pos1, Pos2, Vmax and Vmin values programmed in a [SetDualPosition](#) command apply only for this sequence. All further positioning will use the parameters stored in RAM (programmed for instance by a former [SetMotorParam](#) command).
  - (6) Commands [ResetPosition](#), [SetDualPosition](#), and [SoftStop](#) will be ignored while a DualPosition sequence is ongoing, and will not be executed afterwards.
  - (7) A [SetMotorParam](#) command should not be sent during a [SetDualPosition](#) sequence.
  - (8) 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](#) command prior to send the [SetDualPosition](#) command.

# AMIS-30624

## 14.1.3. Position Periodicity

Depending on the stepping mode the position can range from  $-4096$  to  $+4095$  in half-step to  $-32768$  to  $+32767$  in  $1/16^{\text{th}}$  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 at a minimum.

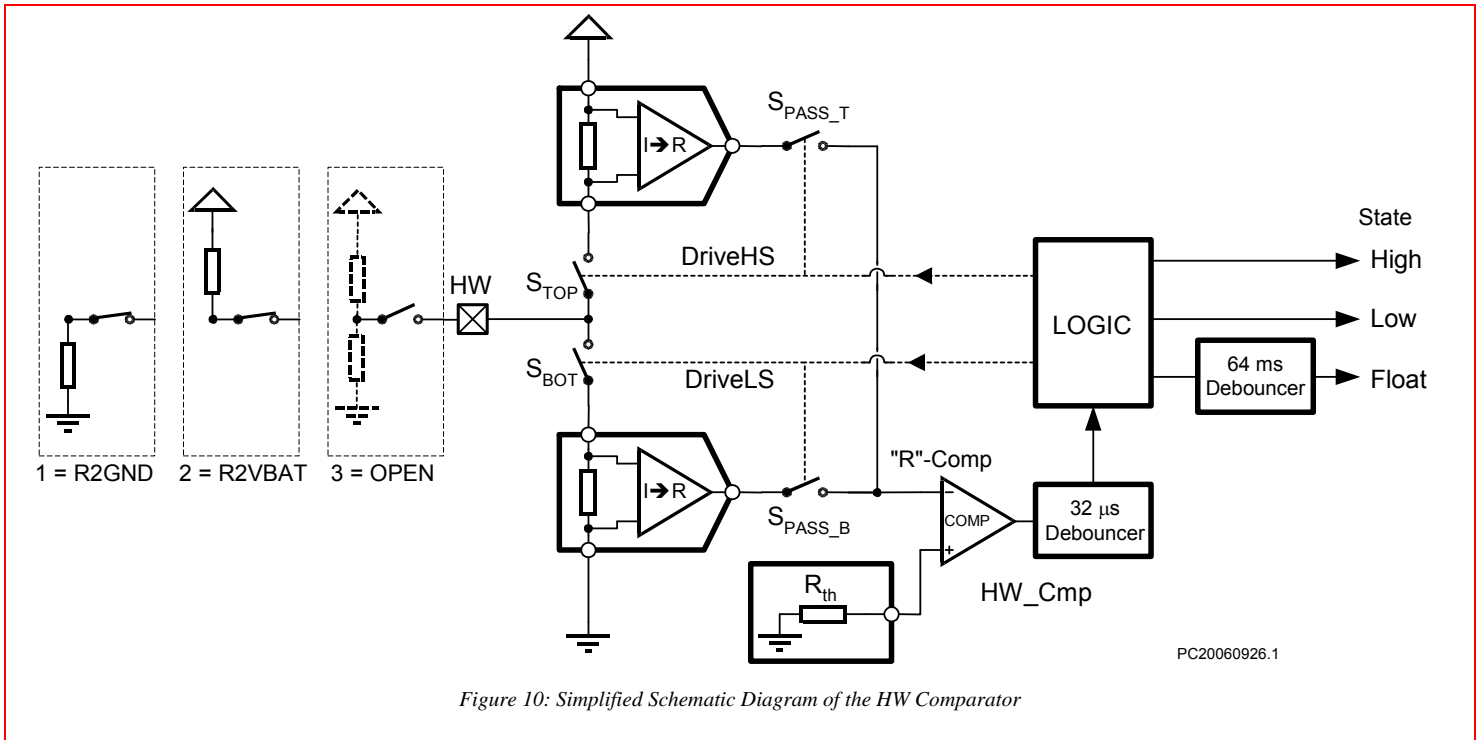
Figure 9 illustrates that the moving direction going from  $\text{ActPos} = +30000$  to  $\text{TagPos} = -30000$  is clockwise. If a counter clockwise motion is required in this example, several consecutive [SetPosition](#) commands can be used. For larger movements, one could also use the command [RunVelocity](#).



## 14.1.4. Hardwired Address HW

In Figure 10 a simplified schematic diagram is shown of the HW comparator circuit.

The HW pin is sensed via two switches  $S_{\text{TOP}}$  and  $S_{\text{BOT}}$ . The DriveHS and DriveLS control lines are alternatively closing  $S_{\text{TOP}}$  and  $S_{\text{BOT}}$ , connecting HW pin with a current to resistor converter. Closing  $S_{\text{TOP}}$  (DriveHS = 1) will sense a current to GND. In that case the top I  $\rightarrow$  R converter output is low, via the closed passing switch  $S_{\text{PASS}_T}$  this signal is fed to the "R" comparator which output HW\_Cmp is high. Closing bottom switch  $S_{\text{BOT}}$  (DriveLS = 1) will sense a current to VBAT. The corresponding I  $\rightarrow$  R converter output is low and via  $S_{\text{PASS}_B}$  fed to the comparator. The output HW\_Cmp will be high.



# AMIS-30624

Three cases can be distinguished (see also Figure 10):

- 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 32µs de-bounced 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 Table 15, the state is depending on the previous state, the condition of the two 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.

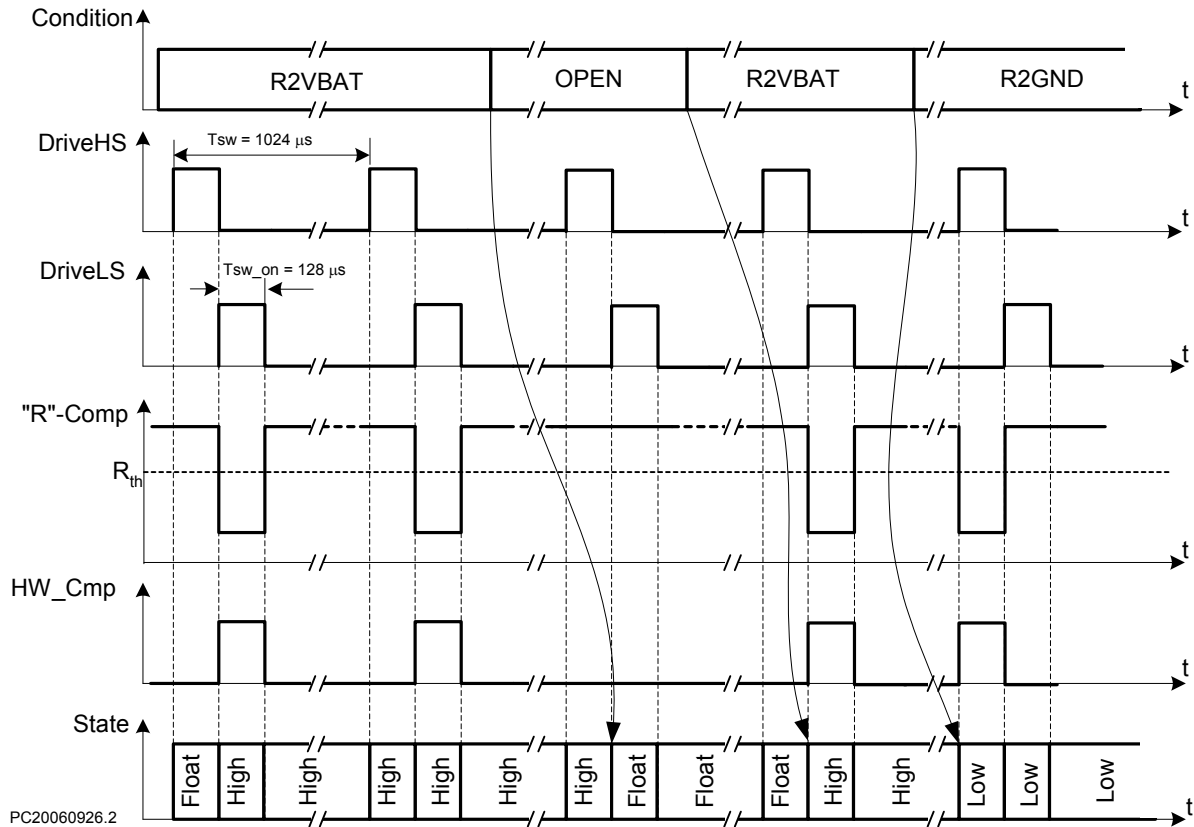


Figure 11: Timing Diagram Showing the Change in States for HW Comparator

# AMIS-30624

## R2VBAT

A resistor is connected between VBAT and HW. Every  $1024\mu\text{s}$   $S_{\text{BOT}}$  is closed for a period of  $128\mu\text{s}$  and a current is sensed. The output of the  $I \rightarrow R$  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  $S_{\text{BOT}}$  is closed the same condition is 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  $S_{\text{BOT}}$  is closed this will be sensed. There will be no current, the output of the corresponding  $I \rightarrow R$  converter is high and the HW\_Cmp will be low. The previous state was High. Based on 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. This prevents false triggering in case of micro interruptions of the power supply. See also [Electrical Transient Conduction Along Supply Lines](#).

## R2GND

If a resistor is connected between HW and the GND, a current is sensed every  $1024\mu\text{s}$  when  $S_{\text{TOP}}$  is closed. The output of the top  $I \rightarrow R$  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.

### 14.1.5. 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 recognised leading to respectively  $\text{ESW} = 0$  and  $\text{ESW} = 1$ .

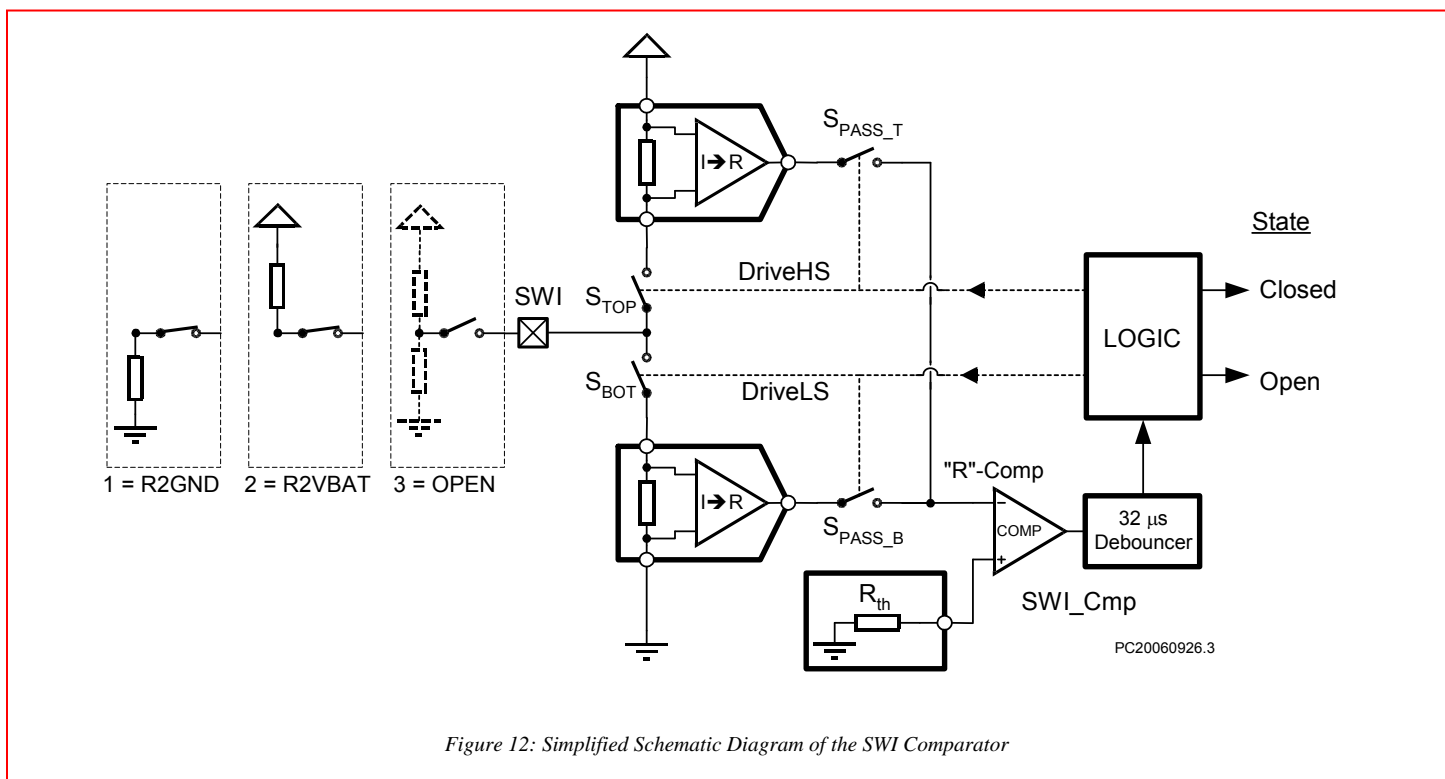


Figure 12: Simplified Schematic Diagram of the SWI Comparator

As illustrated in Figure 14, 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  $S_{\text{TOP}}$  or  $S_{\text{BOT}}$ ) the state of position switch together with the corresponding position is memorized.

Using the GetActualPos commands 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 Figure 13.

| 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] |          |            |        |

Figure 13: GetFullStatus1 FC Commando

**Important remark:** Every 512µs this information is refreshed.

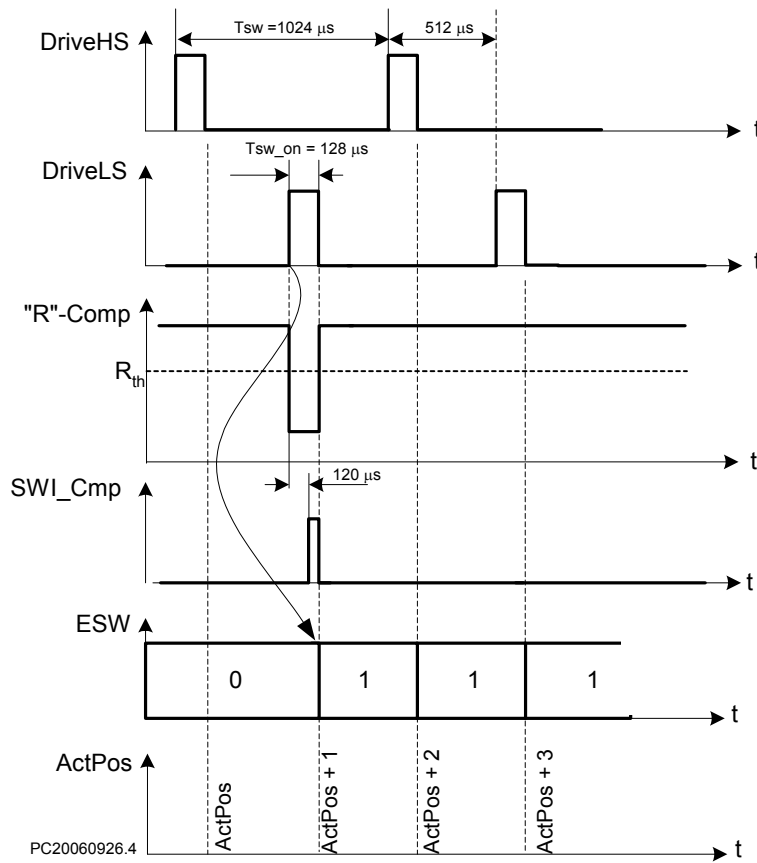


Figure 14: Timing Diagram Showing the Change in States for SWI Comparator

# AMIS-30624

## 14.2 Main Control and Register, OTP Memory + ROM

### 14.2.1. Power-up Phase

The power-up phase of the AMIS-30624 will not exceed 10ms. After this phase, the AMIS-30624 is in shutdown mode, ready to receive I<sup>2</sup>C messages and 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 Table 18).

### 14.2.2. Reset State

After power-up, or after a reset occurrence (e.g. a micro cut on pin VBB has made Vdd 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 18 and Table 19.

### 14.2.3. Soft Stop

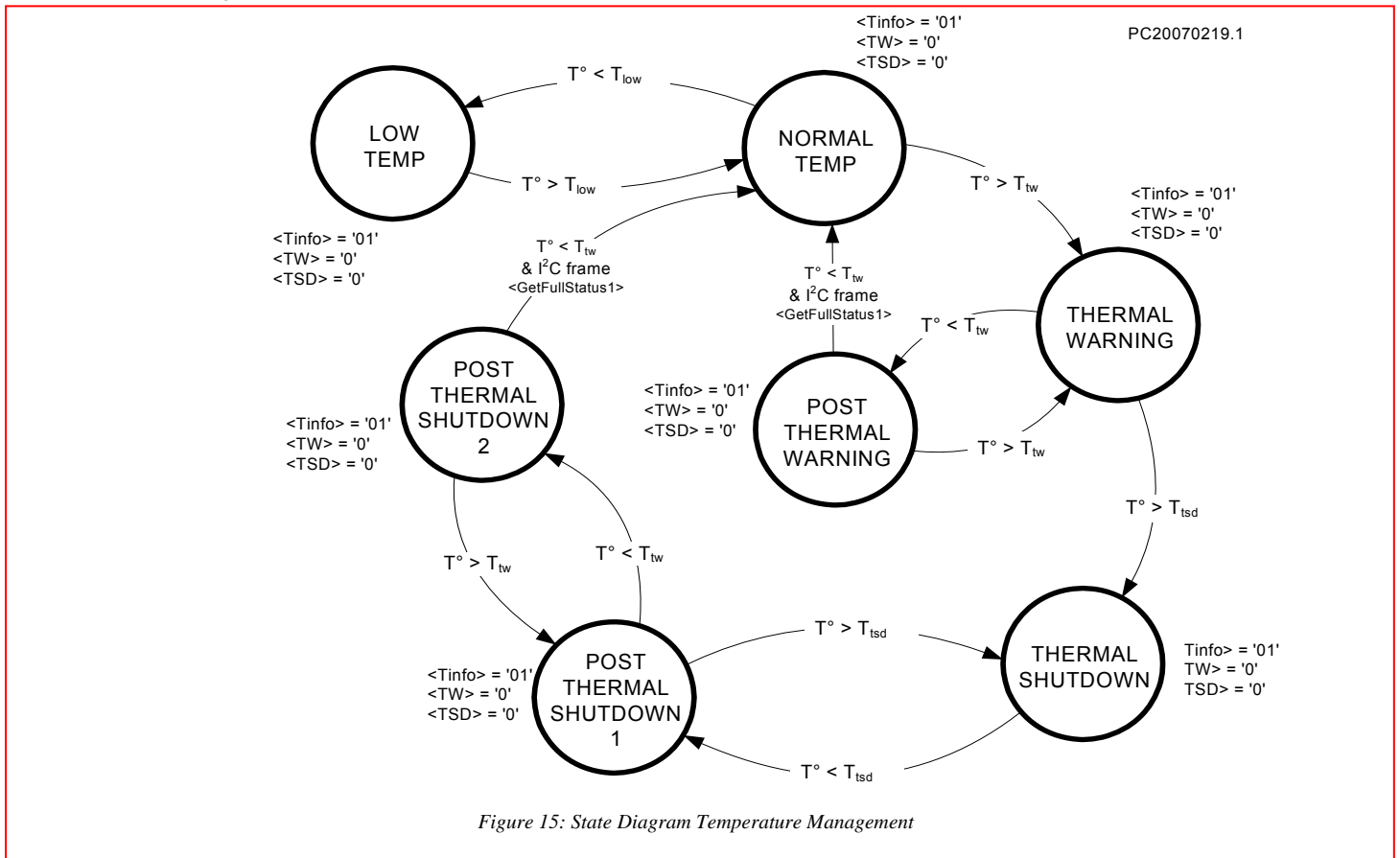
A soft stop is an immediate interruption of a motion, but with a deceleration phase. At the end of this action, the register TagPos is loaded with the value contained in register ActPos to avoid an attempt of the circuit to achieve the motion (see Table 18). The circuit is then ready to execute a new positioning command, provided thermal and electrical conditions allow for it.

### 14.2.4. Thermal Shutdown Mode

When thermal shutdown occurs, the circuit performs a SoftStop command and goes to motor shutdown mode (see Figure 15).

### 14.2.5. Temperature Management

The AMIS-30624 monitors temperature by means of two thresholds and one shutdown level, as illustrated in the Figure 15. The only condition necessary to reset flags <TW> and <TSD> (respectively thermal warning and thermal shutdown) is when the temperature is lower than T<sub>tw</sub> causing the occurrence of a [GetFullStatus1](#) I<sup>2</sup>C frame.



# AMIS-30624

## 14.2.6. Battery Under-voltage Management

The AMIS-30624 monitors the battery voltage by means of one threshold and one shutdown level, as illustrated in Figure 16. The only condition necessary to reset **flags** <UV2> and <StepLoss> is to recover a battery voltage higher than UV1 and to receive a [GetFullStatus1](#) command.

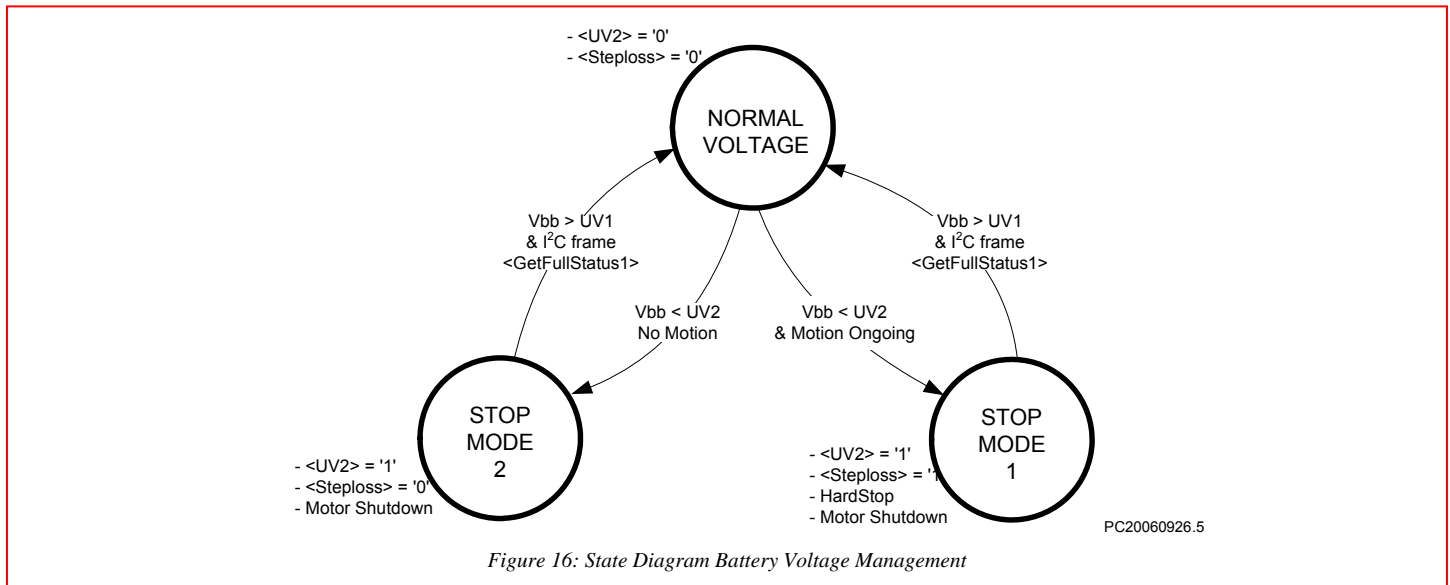


Figure 16: State Diagram Battery Voltage Management

## 14.2.7. OTP Register

### 14.2.7.1 OTP Memory Structure

Table 16 shows where the parameters to be stored in the OTP memory are located.

Table 16: 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    |          | 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 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 any OTP bit at '0' could still be zapped by using a [SetOTPparam](#) command.

Table 17: OTP Overwrite Protection

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

Note:  
Zapped bits will really be "active" after a [GetOTPparam](#) or a [ResetToDefault](#) command or after a power-up.

# AMIS-30624

The command used to load the application parameters via the I<sup>2</sup>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 verification of the correct byte zapping.

## 14.2.7.2 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 an I<sup>2</sup>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<sup>2</sup>C bus.

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

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

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

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



# AMIS-30624

**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                 |

**StepMode** Indicator of stepping mode to be used.

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

**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 HW connection is lost. If *SecPos* [10:0] = "100 0000 0000", this means that secure position is disabled, e.g. the stepper motor 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'. See also Table 10.

# AMIS-30624

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     |
| D     | 1 1 0 1 | 546                |       |
| 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 <sup>2</sup> ) |
|-------|---------|--|
| 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

# AMIS-30624

## 14.2.8. RAM Registers

Table 18: RAM Registers

| Register                           | Mnemonic       | Length (Bit) | Related Commands  | Comment   | Reset State     |       |
|------------------------------------|----------------|--------------|---|---|-----------------|-------|
| Actual position                    | ActPos         | 16           | <a href="#">GetFullStatus2</a><br><a href="#">GotoSecurePos</a><br><a href="#">ResetPosition</a>                                | 16-bit signed   | Note 1          |       |
| Last programmed position           | Pos/<br>TagPos | 16           | <a href="#">GetFullStatus2</a><br><a href="#">GotoSecurePos</a><br><a href="#">ResetPosition</a><br><a href="#">SetPosition</a> | 16-bit signed<br>(see <a href="#">Positioning</a> )   |                 |       |
| Acceleration shape                 | AccShape       | 1            | <a href="#">GetFullStatus1</a><br><a href="#">ResetToDefault1</a><br><a href="#">SetMotorParam</a>                              | '0' ⇒ normal acceleration from Vmin to Vmax<br>'1' ⇒ motion at Vmin without acceleration      | '0'             |       |
| Coil peak current                  | Irun           | 4            | <a href="#">GetFullStatus1</a><br><a href="#">ResetToDefault1</a><br><a href="#">SetMotorParam</a>                              | Operating current<br>See look-up table <a href="#">Irun</a>                                   | From OTP memory |       |
| Coil hold current                  | Ihold          | 4            | <a href="#">GetFullStatus1</a><br><a href="#">ResetToDefault1</a><br><a href="#">SetMotorParam</a>                              | Standstill current<br>See look-up table <a href="#">Ihold</a>                                 |                 |       |
| Minimum velocity                   | Vmin           | 4            | <a href="#">GetFullStatus1</a><br><a href="#">ResetToDefault1</a><br><a href="#">SetMotorParam</a>                              | See Section <a href="#">13.3 Minimum Velocity</a><br>See look-up table <a href="#">Vmin</a>   |                 |       |
| Maximum velocity                   | Vmax           | 4            | <a href="#">GetFullStatus1</a><br><a href="#">ResetToDefault1</a><br><a href="#">SetMotorParam</a>                              | See Section <a href="#">13.2 Maximum Velocity</a><br>See look-up table <a href="#">Vmax</a>   |                 |       |
| Shaft                              | Shaft          | 1            | <a href="#">GetFullStatus1</a><br><a href="#">ResetToDefault1</a><br><a href="#">SetMotorParam</a>                              | Direction of movement<br>for positive velocity  |                 |       |
| Acceleration/<br>deceleration      | Acc            | 4            | <a href="#">GetFullStatus1</a><br><a href="#">ResetToDefault1</a><br><a href="#">SetMotorParam</a>                              | See Section <a href="#">13.4 Acceleration</a><br>See look-up table <a href="#">Acc</a>        |                 |       |
| Secure position                    | SecPos         | 11           | <a href="#">GetFullStatus2</a><br><a href="#">ResetToDefault2</a><br><a href="#">SetMotorParam</a>                              | Target position when LIN connection fails; 11 MSBs of 16-bit position (LSBs fixed to '0')     |                 |       |
| Stepping mode                      | StepMode       | 2            | <a href="#">GetFullStatus1</a><br><a href="#">SetStallParam</a>   | See Section <a href="#">13.1 Stepping Modes</a><br>See look-up table <a href="#">StepMode</a> |                 |       |
| Stall detection absolute threshold | AbsThr         | 4            | <a href="#">GetFullStatus1</a><br><a href="#">SetStallParam</a>   | See Section 15.4 <a href="#">Motion detection</a>   |                 |       |
| Stall detection delta threshold    | DelThr         | 4            | <a href="#">GetFullStatus1</a><br><a href="#">SetStallParam</a>   | See Section 15.4 <a href="#">Motion detection</a>   |                 |       |
| Stall detection delay              | FS2StallEn     | 3            | <a href="#">GetFullStatus2</a><br><a href="#">SetStallParam</a>   | See Section 15.4 <a href="#">Motion detection</a>   |                 | '000' |
| Stall detection sampling           | MinSamples     | 3            | <a href="#">GetFullStatus2</a><br><a href="#">SetStallParam</a>   | See Section 15.4 <a href="#">Motion detection</a>   |                 | '000' |
| PWM jitter                         | PWMJEn         | 1            | <a href="#">GetFullStatus2</a><br><a href="#">SetStallParam</a>   | '1' means jitter is added   |                 | '0'   |
| 100% duty cycle stall disable      | DC100SDis      | 1            | <a href="#">GetFullStatus2</a><br><a href="#">SetStallParam</a>   | '1' means stall detection is disabled in case PWM regulator runs at $\delta = 100\%$          | '0'             |       |
| PWM frequency                      | PWMFreq        | 1            | <a href="#">SetMotorParam</a>   | '1' means 44 kHz is selected  | '0'             |       |

Note:

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.

# AMIS-30624

## 14.2.9. Flags Table

Table 19: Flags Table

| Flag                    | Mnemonic   | Length (Bit) | Related Commands               | Comment  | Reset State |
|-------------------------|------------|--------------|--------------------------------|--|-------------|
| Charge pump failure     | CPFail     | 1            | <a href="#">GetFullStatus1</a> | '0' = charge pump OK<br>'1' = charge pump failure<br>reset only after GetFullStatus1   | '0'         |
| Electrical defect       | ElDef      | 1            | <a href="#">GetFullStatus1</a> | <OVC1> or <OVC2> or <open circuit 1> or <open circuit 2> or <CPFail><br>resets only after GetFullStatus1   | '0'         |
| External switch status  | ESW        | 1            | <a href="#">GetFullStatus1</a> | '0' = open<br>'1' = close  | '0'         |
| Motion status           | Motion     | 3            | <a href="#">GetFullStatus1</a> | "x00" = Stop<br>"001" = inner motion acceleration<br>"010" = inner motion deceleration<br>"011" = inner motion max. speed<br>"101" = outer motion acceleration<br>"110" = outer motion deceleration<br>"111" = outer motion max. speed | "000"       |
| Over current in coil X  | OVC1       | 1            | <a href="#">GetFullStatus1</a> | '1' = over current<br>reset only after GetFullStatus1  | '0'         |
| Over current in coil Y  | OVC2       | 1            | <a href="#">GetFullStatus1</a> | '1' = over current<br>reset only after GetFullStatus1  | '0'         |
| Secure position enabled | SecEn      | 1            | <a href="#">Internal use</a>   | '0' if SecPos = "100 0000 0000"<br>'1' otherwise   | NA          |
| Step loss               | StepLoss   | 1            | <a href="#">GetFullStatus1</a> | '1' = step loss due to under voltage, over current or open circuit   | '1'         |
| Delta high stall        | DelStallHi | 1            | <a href="#">GetFullStatus2</a> | '1' = Vbemf > Average + DeltaThr   | '0'         |
| Delta low stall         | DelStallLo | 1            | <a href="#">GetFullStatus2</a> | '1' = Vbemf < Average – DeltaThr   | '0'         |
| Absolute stall          | AbsStall   | 1            | <a href="#">GetFullStatus2</a> | '1' = Vbemf > AbsThr   | '0'         |
| Stall                   | Stall      | 1            | <a href="#">GetFullStatus1</a> | Stall detected   | '0'         |
| Temperature info        | Tinfo      | 2            | <a href="#">GetFullStatus1</a> | "00" = normal temperature range<br>"01" = low temperature warning<br>"10" = high temperature warning<br>"11" = motor shutdown  | "00"        |
| Thermal shutdown        | TSD        | 1            | <a href="#">GetFullStatus1</a> | '1' = shutdown (> 155°C typ.)<br>reset only after GetFullStatus1 and if <Tinfo> = "00"   | '0'         |
| Thermal warning         | TW         | 1            | <a href="#">GetFullStatus1</a> | '1' = over temp (> 145°C)<br>reset only after GetFullStatus1 and if <Tinfo> = "00"   | '0'         |
| Battery stop voltage    | UV2        | 1            | <a href="#">GetFullStatus1</a> | '0' = Vbb > UV2<br>'1' = Vbb ≤ UV2<br>reset only after GetFullStatus1  | '0'         |
| Digital supply reset    | VddReset   | 1            | <a href="#">GetFullStatus1</a> | 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 GetFullStatus1 command.   | '1'         |

# AMIS-30624

## 14.2.10. Priority Encoder

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

Table 20: Priority Encoder

| State →  | Stopped   | GotoPos                                      | DualPosition                                     | SoftStop                                     | HardStop                                     | ShutDown   |
|--|---|--|--|--|--|--|
| Command ↓  | Motor Stopped, Hold in Coils                            | Motor Motion Ongoing                         | No Influence on RAM and TagPos                   | Motor Decelerating                           | Motor Forced to Stop                         | Motor Stopped, H-bridges in Hi-Z   |
| GetOTPparam  | OTP refresh; I <sup>2</sup> C slave response            | OTP refresh; I <sup>2</sup> C slave response | OTP refresh; I <sup>2</sup> C slave response     | OTP refresh; I <sup>2</sup> C slave response | OTP refresh; I <sup>2</sup> C slave response | OTP refresh; I <sup>2</sup> C slave response                                       |
| GetFullStatus1 [attempt to clear all flags] (note 1)           | I <sup>2</sup> C slave response                         | I <sup>2</sup> C slave response              | I <sup>2</sup> C slave response                  | I <sup>2</sup> C slave response              | I <sup>2</sup> C slave response              | I <sup>2</sup> C slave response; if (<TSD> or <ElFlag> = '0' then → <b>Stopped</b> |
| GetFullStatus2   | I <sup>2</sup> C slave response                         | I <sup>2</sup> C slave response              | I <sup>2</sup> C slave response                  | I <sup>2</sup> C slave response              | I <sup>2</sup> C slave response              | I <sup>2</sup> C 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 2) | 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 (note 5)                          | RAM update                                       | RAM update                                   | RAM update                                   | RAM update   |
| SetStallParam  | RAM update  | RAM update                                   | RAM update                                       | RAM update                                   | RAM update                                   | RAM update   |
| ResetPosition  | TagPos and ActPos reset                                 |  |  |  |  | TagPos and ActPos reset  |
| SetPosition  | TagPos updated; → <b>GotoPos</b>                        | TagPos updated                               | TagPos updated                                   |  |  |  |
| RunVelocity  | Continuous motion; → <b>GotoPos</b>                     |  |  |  |  |  |
| GotoSecPosition  | If <SecEn> = '1' then TagPos = SecPos; → <b>GotoPos</b> | If <SecEn> = '1' then TagPos = SecPos        | If <SecEn> = '1' then TagPos = SecPos            |  |  |  |
| DualPosition   | → <b>DualPosition</b>                                   |  |  |  |  |  |
| HardStop   |   | → <b>HardStop</b> ; <StepLoss> = '1'         | → <b>HardStop</b> ; <StepLoss> = '1'             | → <b>HardStop</b> ; <StepLoss> = '1'         |  |  |
| SoftStop   |   | → <b>SoftStop</b>                            |  |  |  |  |
| HardStop [↔ (<CPFail> or <UV2> or <ElDef>) = '1' ⇒ <HS> = '1'] | → <b>Shutdown</b>                                       | → <b>HardStop</b>                            | → <b>HardStop</b>                                | → <b>HardStop</b>                            |  |  |
| Thermal shutdown [<TSD> = '1']                                 | → <b>Shutdown</b>                                       | → <b>SoftStop</b>                            | → <b>SoftStop</b>                                |  |  |  |
| Motion finished  | NA  | → <b>Stopped</b>                             | → <b>Stopped</b>                                 | → <b>Stopped</b> ; TagPos =ActPos            | → <b>Stopped</b> ; TagPos =ActPos            | NA   |

With the following color code:

|  |  |
|--|--|
|  | Command ignored                                      |
|  | Transition to another state                          |
|  | Master is responsible for proper update (see Note 5) |

# AMIS-30624

- Notes:**
- 1)  $\langle \text{ElFlag} \rangle = \langle \text{CPFail} \rangle$  or  $\langle \text{UV2} \rangle$  or  $\langle \text{ElDef} \rangle$  or  $\langle \text{VDDreset} \rangle$
  - 2) 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  $\langle \text{VddReset} \rangle$  flag).
  - 3) 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.  $\text{AccShape}$  is forced to '1' during second motion even if a `ResetToDefault` command is issued during a DualPosition sequence, in which case  $\text{AccShape}$  at '0' will be taken into account after the DualPosition sequence. A [GetFullStatus1](#) command will return the default parameters for  $V_{\text{max}}$  and  $V_{\text{min}}$  stored in RAM.
  - 4) Shutdown state can be left only when  $\langle \text{TSD} \rangle$  and  $\langle \text{ElFlag} \rangle$  flags are reset.
  - 5) 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).
  - 6) A `SetMotorParam` command sent while a motion is ongoing (state `GotoPos`) should not attempt to modify  $\text{Acc}$  and  $V_{\text{min}}$  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.
  - 7)  $\langle \text{SecEn} \rangle = '1'$  when register `SecPos` is loaded with a value different from the most negative value (i.e. different from  $0x400 = "100\ 0000\ 0000"$ )
  - 8)  $\langle \text{Stop} \rangle$  flag allows user to distinguish whether state stopped was entered after HardStop/SoftStop or not.  $\langle \text{Stop} \rangle$  is set to '1' when leaving state `HardStop` or `SoftStop` and is reset during first clock edge occurring in state `Stopped`.
  - 9) While in state stopped, if `ActPos`  $\rightarrow$  `TagPos` there is a transition to state `GotoPos`. This transition has the lowest priority, meaning that  $\langle \text{Stop} \rangle$ ,  $\langle \text{TSD} \rangle$ , etc. are first evaluated for possible transitions.
  - 10) If  $\langle \text{StepLoss} \rangle$  is active, then [SetPosition](#) and `GotoSecurePosition` commands are ignored (they will not modify `TagPos` register whatever the state). Other command like `DualPosition` or `ResetPosition` will be executed if allowed by current state.  $\langle \text{StepLoss} \rangle$  can only be cleared by a [GetFullStatus1](#) command.

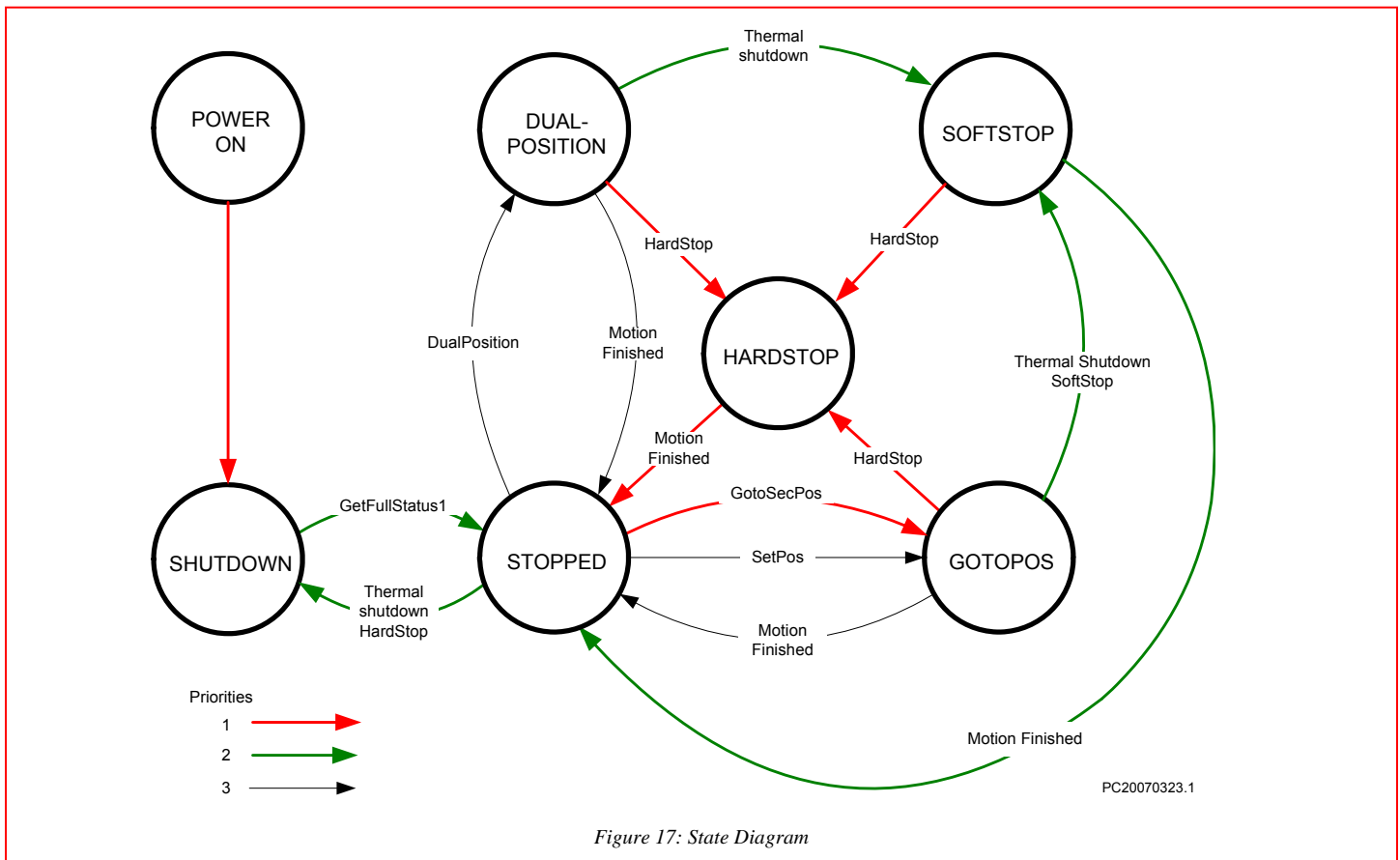


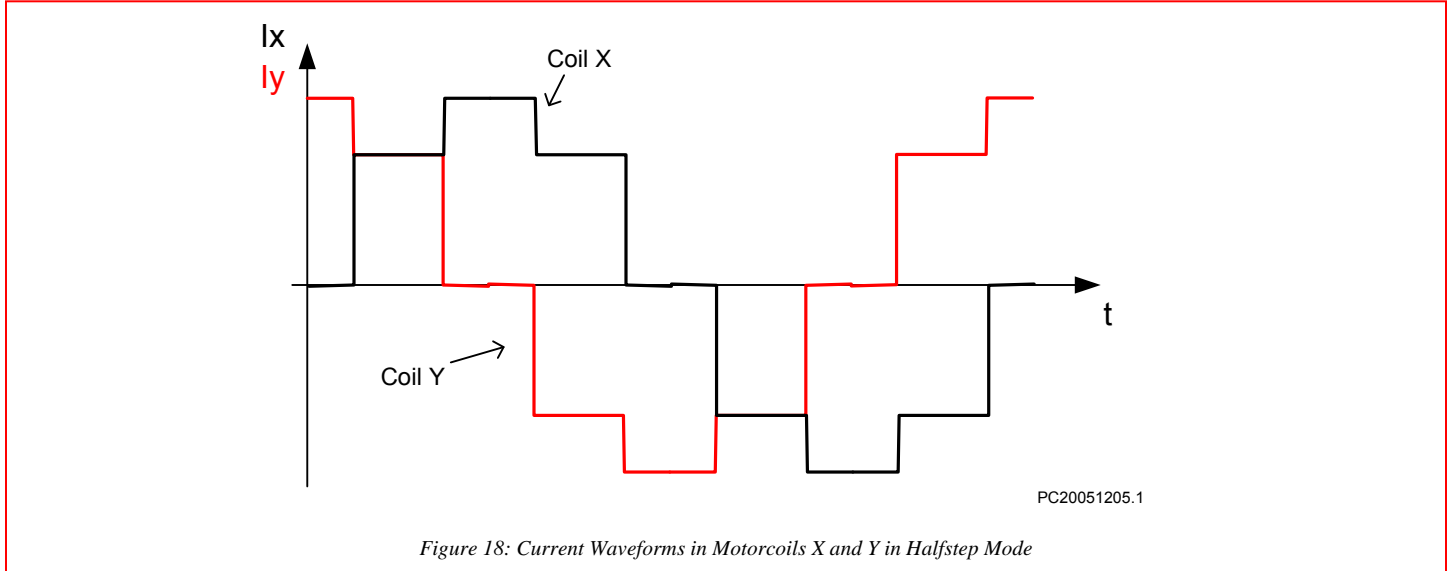
Figure 17: State Diagram

# AMIS-30624

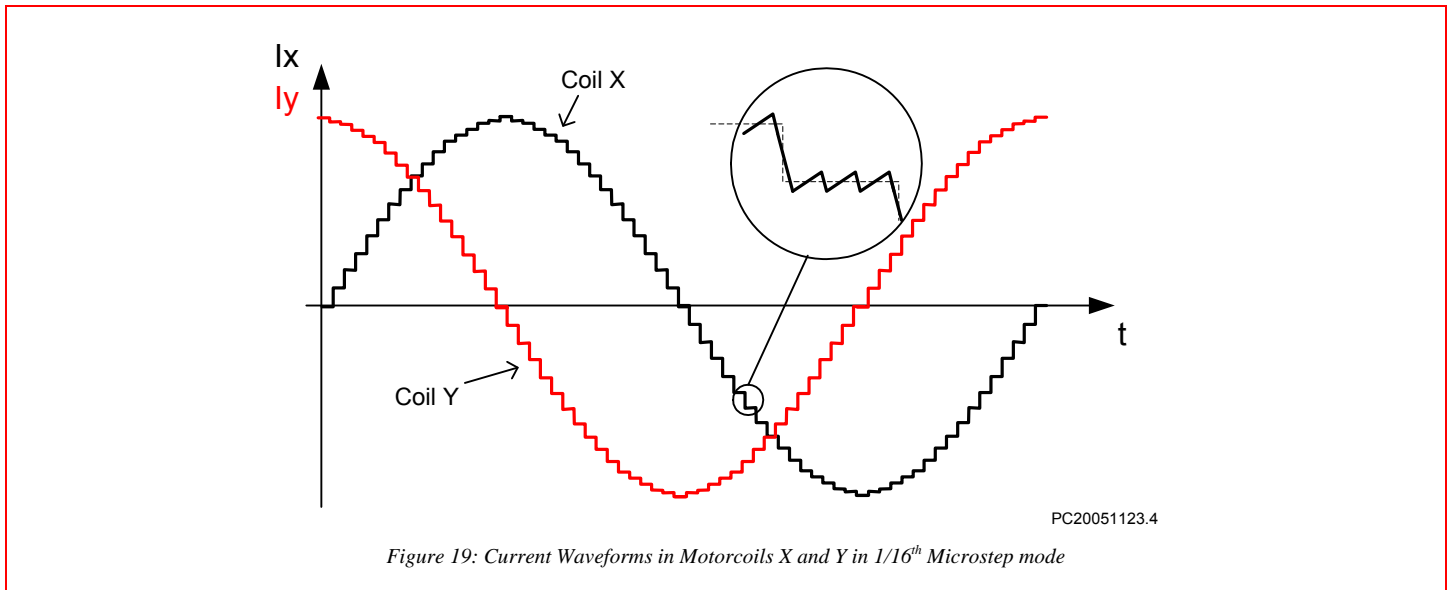
## 14.3 Motor Driver

### 14.3.1. Current Waveforms in the Coils

Figure 18 illustrates the current fed to the motor coils by the motor driver in half step mode.



Whereas Figure 19 below shows the current fed to one coil in 1/16<sup>th</sup> micro stepping (one electrical period).



### 14.3.2. PWM Regulation

In order to force a given current (determined by  $I_{run}$  or  $I_{hold}$  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

# AMIS-30624

zoom over one micro-step in Figure 19 shows how the PWM circuit performs this regulation. To reduce the current ripple, a higher PWM frequency should be selectable. The RAM register PWMfreq is used for this (Bit 6 in Data 7 of [SetMotorParam](#)).

Table 21: PWM Frequency Selection

| PWMfreq | Applied PWM Frequency |
|---------|-----------------------|
| 0       | 22.8 kHz              |
| 1       | 45.6 kHz              |

### 14.3.3. 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. (Bit 0 in Data 7 of [SetMotorParam](#) or [SetStallParam](#)). Readout with [GetFullStatus1](#).

Table 22: PWM Jitter Selection

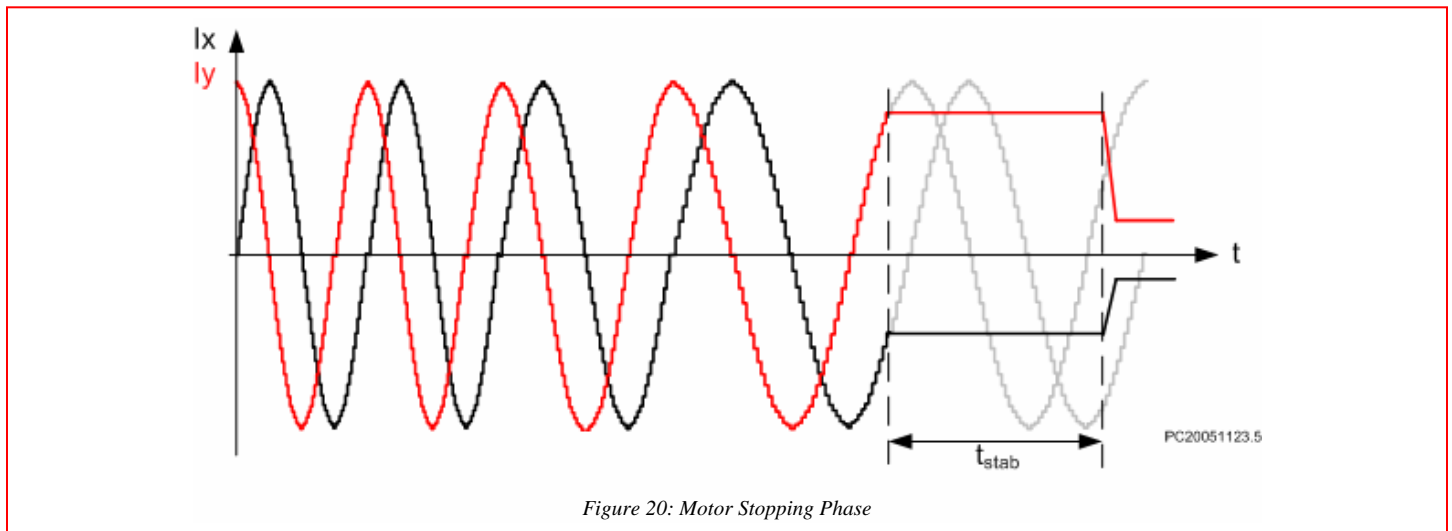
| PWMJEn | Status                        |
|--------|-------------------------------|
| 0      | Single PWM frequency          |
| 1      | Added jitter to PWM frequency |

### 14.3.4. Motor Starting Phase

At motion start, the currents in the coils are directly switched from  $I_{hold}$  to  $I_{run}$  with a new sine/cosine ratio corresponding to the first half (or micro) step of the motion.

### 14.3.5. 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 Table 6). The currents are then set to the hold values, respectively,  $I_{hold} \times \sin(TagPos)$  and  $I_{hold} \times \cos(TagPos)$  as illustrated below. A new positioning order can then be executed.



### 14.3.6. 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 the flag <CPFail> (available with command [GetFullStatus1](#)).

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



# AMIS-30624

## 14.3.7. 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 of the following short circuits:

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

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

Table 23: 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       |

**Remark:** One cannot detect an internal short in the motor.

## 14.3.8. Motor Shutdown Mode

A motor shutdown occurs when:

- The chip temperature rises above the thermal shutdown threshold Tsd (see [Thermal Shutdown Mode](#))
- The battery voltage goes below UV2 (see [Battery Voltage Management](#))
- Flag <ElDef> = '1', meaning an electrical problem is detected on one or both coils, e.g. a short circuit
- Flag <CPFail> = '1', meaning there is a charge pump failure

A motor shutdown leads to the following:

- 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<sup>2</sup>C interface remains active, being able to receive orders or send status.

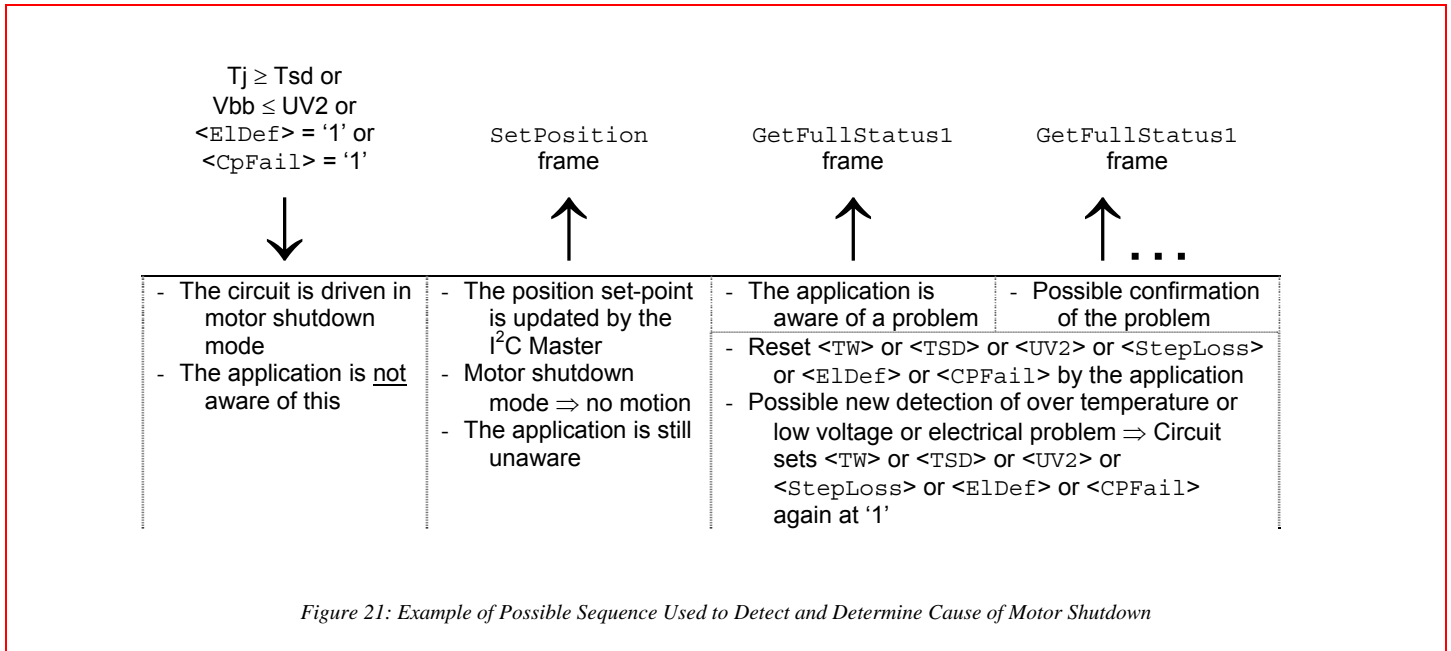
The conditions to get out of a motor shutdown mode are:

- Reception of a [GetFullStatus1](#) command **AND**
- The four causes above are no longer detected

This leads to H-bridges in lhold 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.

# AMIS-30624



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

**Warning:** The application should limit the number of consecutive [GetFullStatus1](#) commands to try to get the AMIS-30624 out of shutdown mode. When this proves to be unsuccessful, for example if there is a permanent defect, the reliability of the circuit could be altered since [GetFullStatus1](#) attempts to disable the protection of the H-bridges.

## 14.4 Motion Detection

Motion detection is based on the back emf generated internally in the running motor. When the motor is blocked, for example when it hits the end-position, the velocity and as a result also the generated back emf, is disturbed. The AMIS-30624 senses the back emf, calculates a moving average and compares the value with two independent threshold levels: Absolute threshold ([AbsThrf3:0](#)) and Delta threshold ([DelThrf3:0](#)). Instructions for the correct use of these two levels in combination with three additional parameters (MinSamples, FS2StallEn and DC100SDis) are outside the scope of this datasheet. Detailed information is available in a dedicated white paper "Robust Motion Control with AMIS-3062x Stepper Motor Drivers", available on <http://www.amis.com/>.

If the motor is accelerated by a pulling or propelling force and the resulting back emf increases above the Delta threshold (+ ΔTHR), then  $\langle DelStallHi \rangle$  is set. When the motor is slowing down and the resulting back emf decreases below the Delta threshold (- ΔTHR), then  $\langle DelStallLo \rangle$  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,  $\langle AbsStall \rangle$  is set. The  $\langle Stall \rangle$  flag is the OR function of  $\langle DelStallLo \rangle$  OR  $\langle DelStallHi \rangle$  OR  $\langle AbsStall \rangle$ .

# AMIS-30624

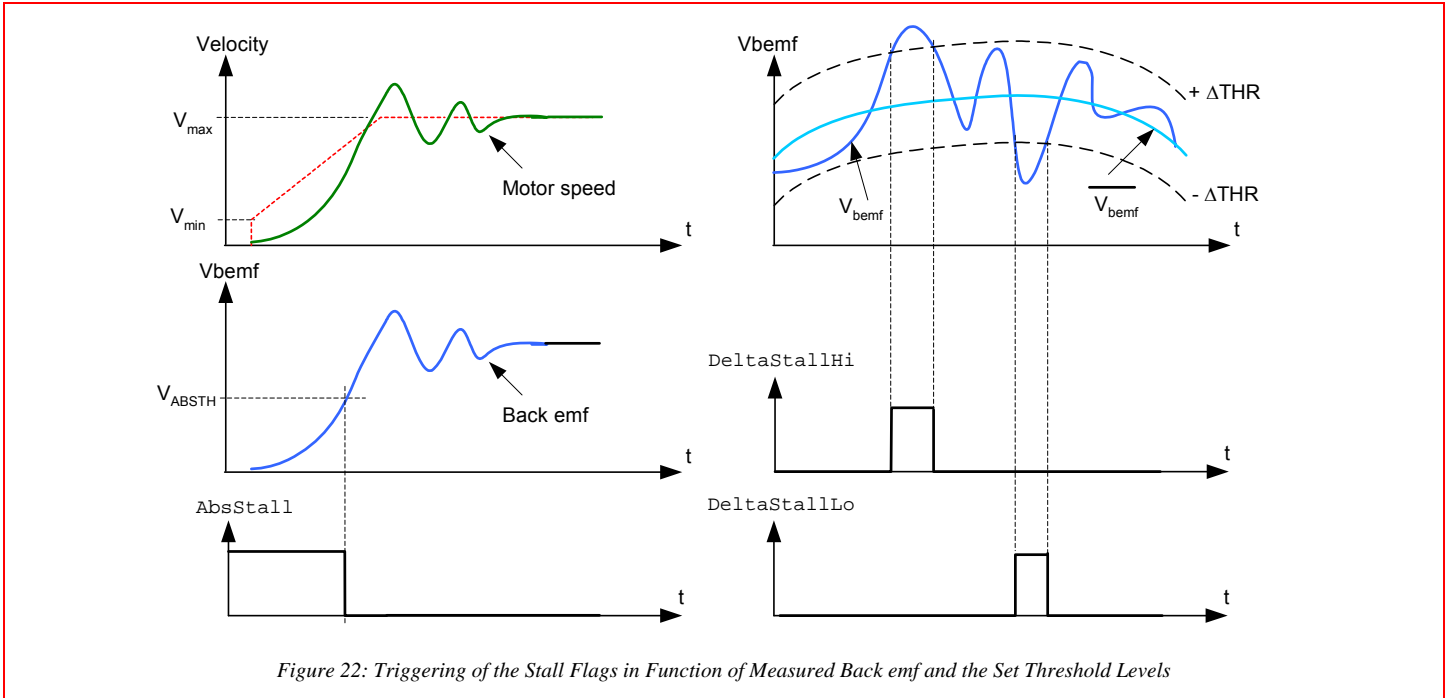


Figure 22: Triggering of the Stall Flags in Function of Measured Back emf and the Set Threshold Levels

Table 24: 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       |

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

If during positioning a mechanical obstacle is detected (stall), an (internal) hardstop is generated. The motor will stop immediately and as a consequence the <StepLoss> and <Stall> flags are set. The position in the internal counter will be copied to the ActPos register. All flags can be read out with the [GetFullStatus1](#).

If Stall appears during DualPosition then the first phase is cancelled (via internal Hardstop) and after timeout (26.6ms) the second phase at  $V_{min}$  starts.

### Important Remark:

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 empty or zero. Both levels can be programmed using the I<sup>2</sup>C command [SetStallParam](#) in the registers AbsThr[3:0] and DelThr[3:0]. Also in the OTP register [AbsThr\[3:0\]](#) and [DelThr\[3:0\]](#) can be set using the I<sup>2</sup>C command [SetOTPParam](#). These values are copied in the RAM registers during power on reset.

# AMIS-30624

Value Table:

Table 25: Absolute Threshold Settings

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

Table 26: Delta Threshold Settings

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

## MinSamples

MinSamples[2:0] is a Bemf sampling delay time expressed in number of PWM cycles, for more information please refer to the white paper “Robust Motion Control with AMIS-3062x Stepper Motor Drivers”.

Table 27: Back emf Sample Delay Time

| Index | MinSamples[2:0] | t <sub>DELAY</sub> (μs) |             |
|-------|-----------------|-------------------------|-------------|
|       |                 | PWMfreq = 0             | PWMfreq = 1 |
| 0     | 000             | 87                      | 43          |
| 1     | 001             | 130                     | 65          |
| 2     | 010             | 174                     | 87          |
| 3     | 011             | 217                     | 109         |
| 4     | 100             | 261                     | 130         |
| 5     | 101             | 304                     | 152         |
| 6     | 110             | 348                     | 174         |
| 7     | 111             | 391                     | 196         |

## 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 28: 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                  |

For more information please refer to the white paper “Robust Motion Control with AMIS-3062x Stepper Motor Drivers”.

## DC100StEn

When a motor with large back – e.m.f. is operated at high velocity and low supply voltage, then the PWM duty cycle can be as high as 100 percent. 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” (Bit 1 in Data 7 of [SetStallParam](#)) enables the function where stall detection is switched off when PWM duty cycle equals 100 percent. For more information the white paper “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<sup>2</sup>C command [TestBemf](#). The SWI pin will be converted into an analog output on which the Bemf 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. More information is available in the white paper “Robust Motion Control with AMIS-3062x Stepper Motor Drivers”.

# AMIS-30624

## 15.0 I<sup>2</sup>C Bus Description

### 15.1 General Description

AMIS-30624 uses a simple bi-directional 2-wire bus for efficient inter-ic control. This bus is called the Inter IC or I<sup>2</sup>C-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 kbit/s.
- On-chip filtering rejects spikes on the bus data line to preserve data integrity.
- No need to design bus interfaces because I<sup>2</sup>C-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.

### 15.2 Concept

The I<sup>2</sup>C-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 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 is a slave device. See Table 30.

Table 29: Definition of I<sup>2</sup>C –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                       |

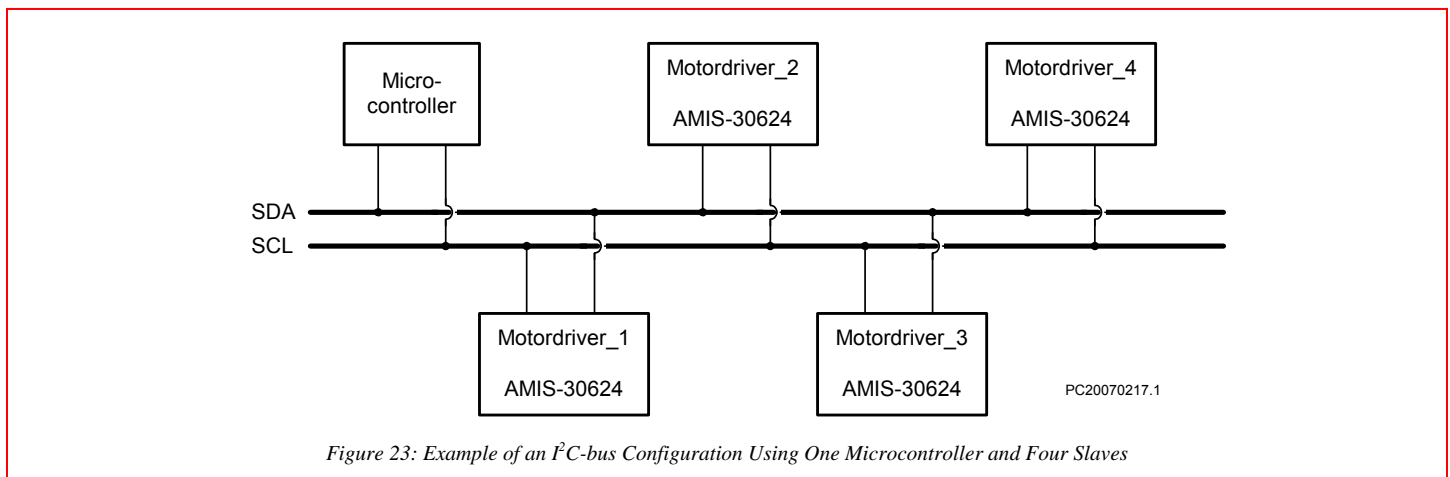


Figure 23: Example of an I<sup>2</sup>C-bus Configuration Using One Microcontroller and Four Slaves

Figure 23 highlights the master-slave and receiver-transmitter relationships to be found on the I<sup>2</sup>C-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

# AMIS-30624

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 I<sup>2</sup>C-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.

## 15.3 General Characteristics

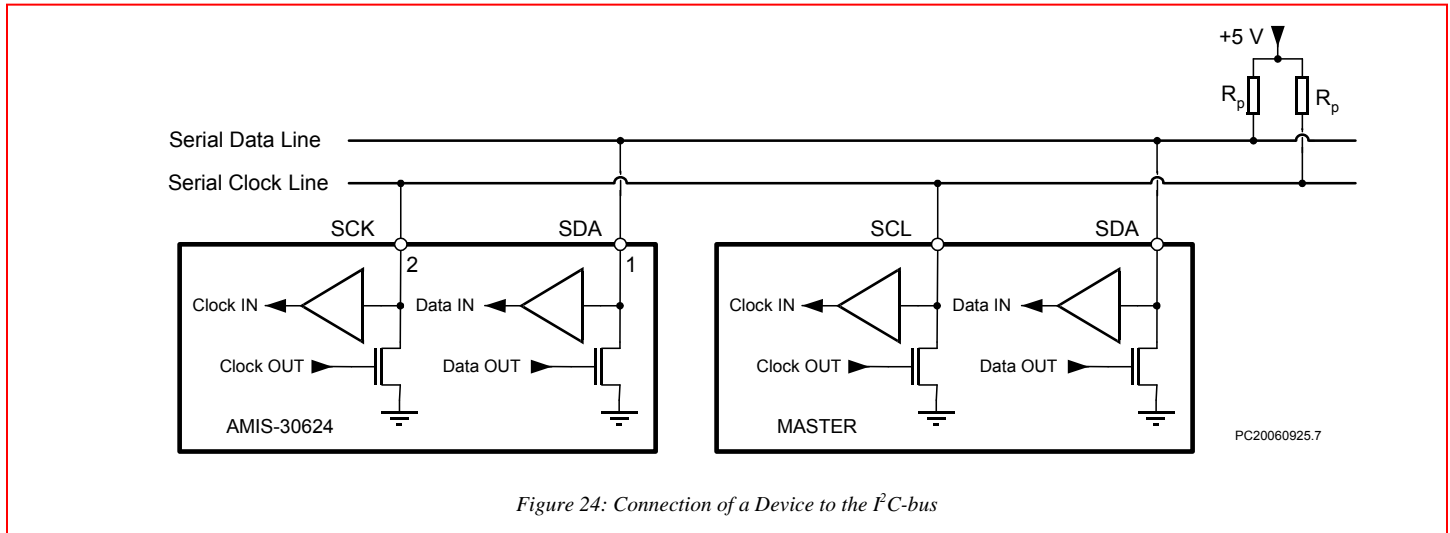


Figure 24: Connection of a Device to the I<sup>2</sup>C-bus

Both SDA and SCK are bi-directional lines connected to a positive supply voltage via a pull-up resistor (see Figure 24). 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 I<sup>2</sup>C-bus can be transferred up to 400kb/s in fast mode. The number of interfaces connected to the bus is dependent on the maximum bus capacitance limit (See C<sub>B</sub> in Table 6) and the available number of addresses.

## 15.4 Bit Transfer

The levels for logic '0' (LOW) and '1' (HIGH) are not fixed in the I<sup>2</sup>C standard but dependent on the used VDD level. Using AMIS-30624, the levels are specified in Table 5. One clock pulse is generated for each data bit transferred.

# AMIS-30624

## 15.4.1. 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 25).

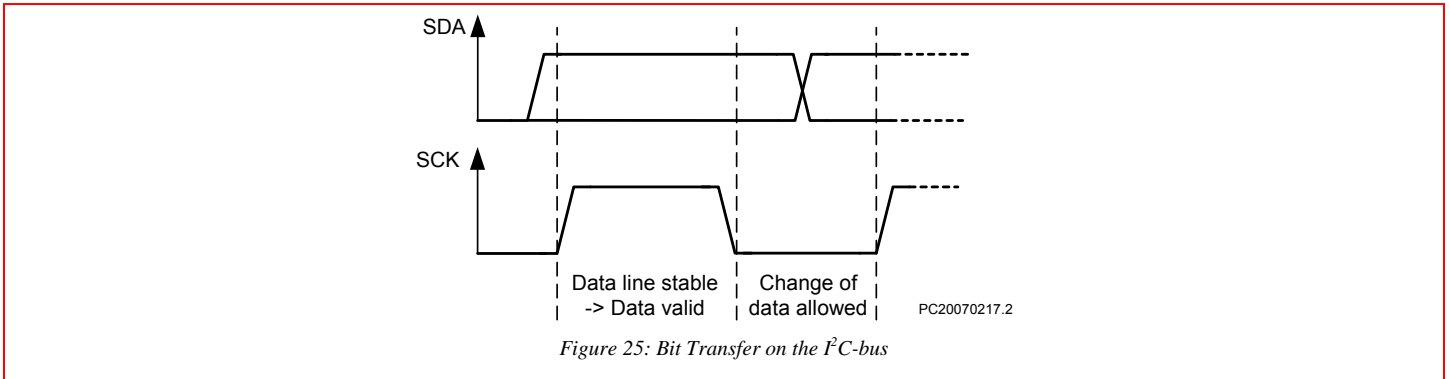


Figure 25: Bit Transfer on the I<sup>2</sup>C-bus

## 15.4.2. START and STOP Conditions

Within the procedure of the I<sup>2</sup>C-bus, unique situations arise, which are defined as START (S) and STOP (P) conditions (See Figure 26). 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 27). The symbol S will be used to represent START and repeated START, unless otherwise noted.

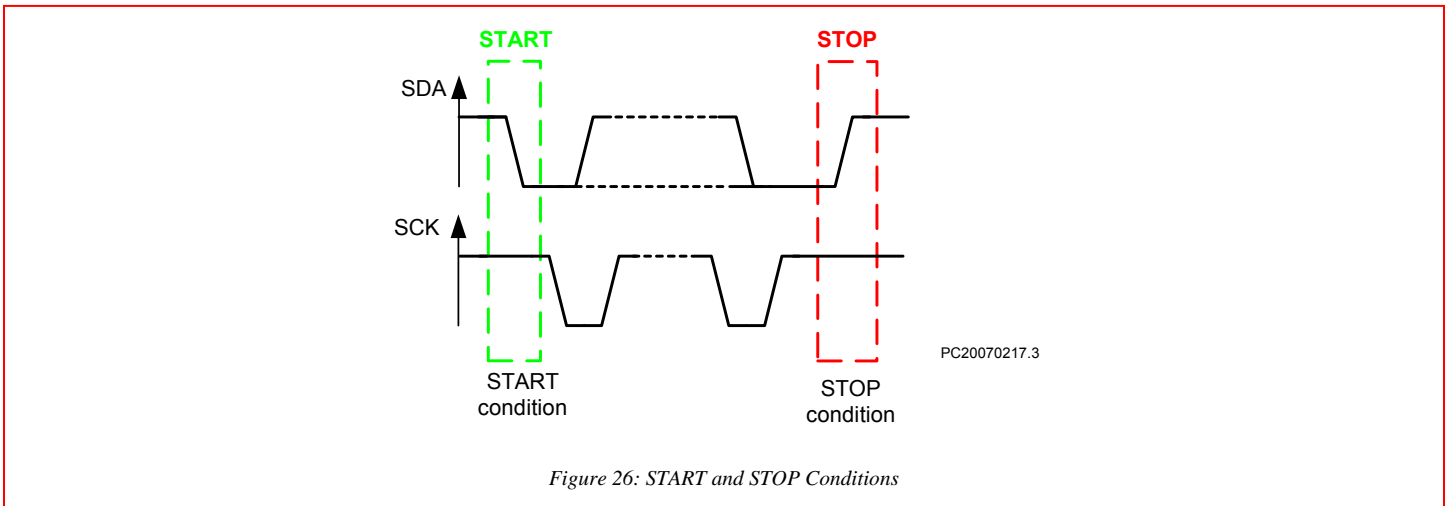


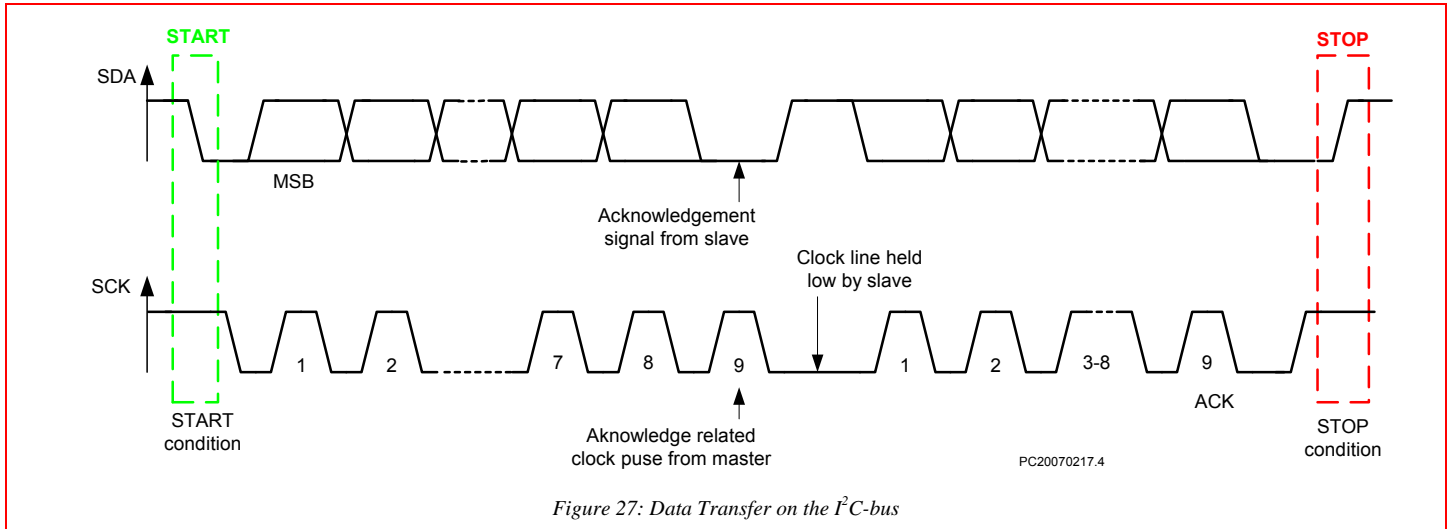
Figure 26: START and STOP Conditions

# AMIS-30624

## 15.5 Transferring Data

### 15.5.1. 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 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 27). 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.

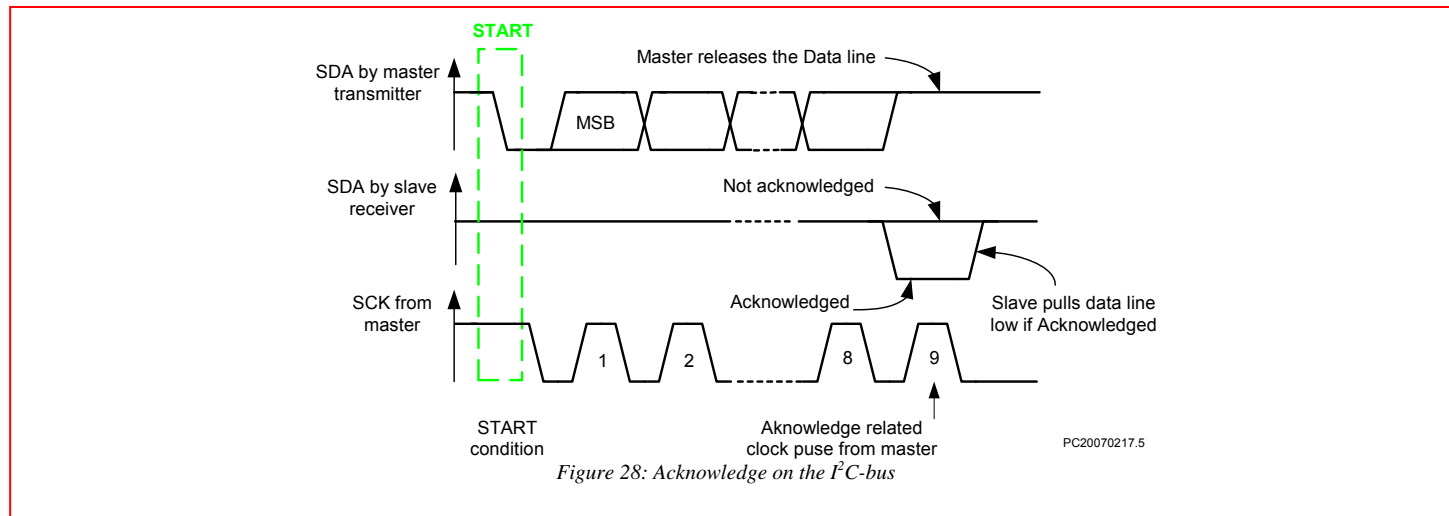


### 15.5.2. Acknowledge

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 28). Of course, set-up and hold times must also be taken into account (see Table 6). When AMIS-30624 doesn't acknowledge the slave address, the data line will be left HIGH. The master can then generate either a STOP condition to abort the transfer, or a repeated START condition to start a new transfer.

If AMIS-30624 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 then 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 as slave-transmitter shall release the data line to allow the master to generate STOP or repeated START condition.





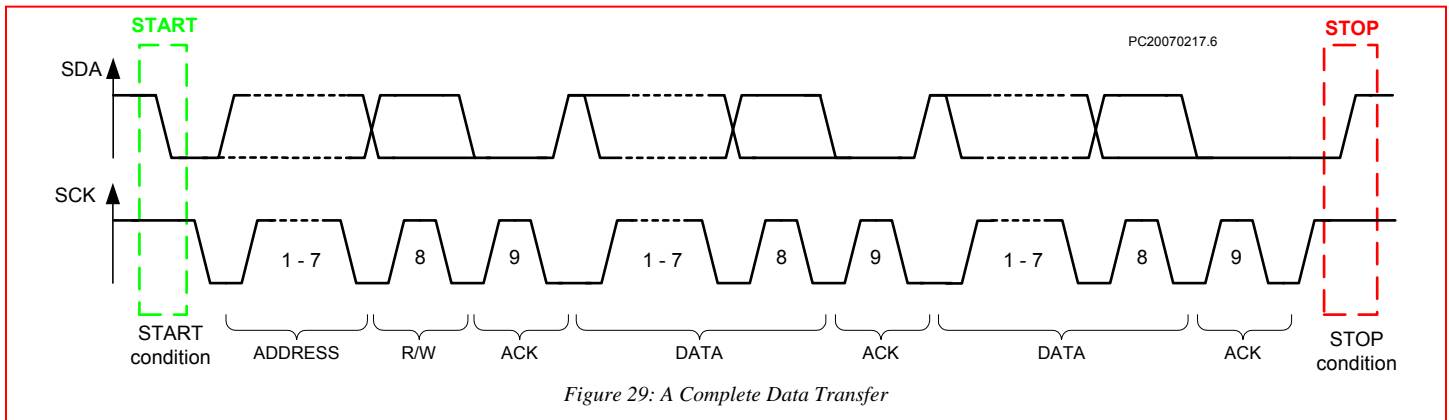
# AMIS-30624

## 15.5.3. Clock Generation

The master generates the clock on the SCK line to transfer messages on the I<sup>2</sup>C-bus. Data is only valid during the HIGH period of the clock.

## 15.6 Data Formats with 7-bit Addresses

Data transfers follow the format shown in Figure 29. 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.



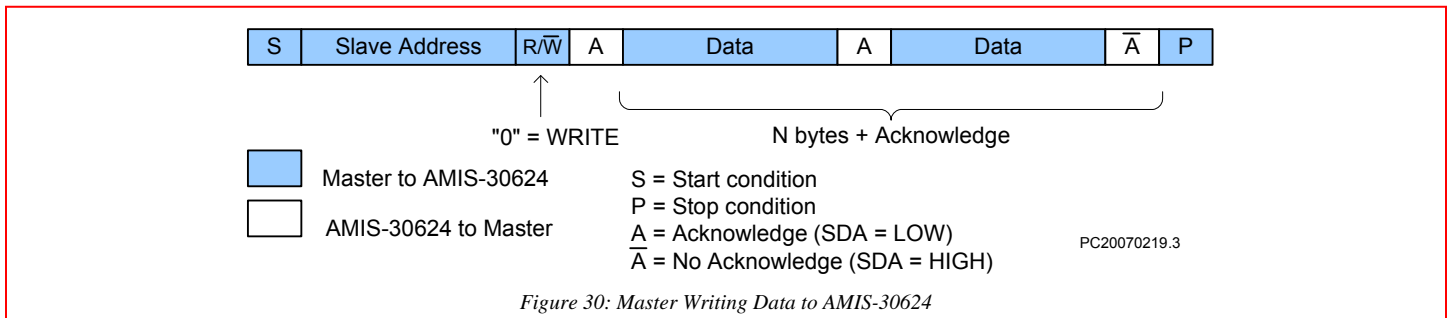
However, if a master still wishes to communicate on the bus, it can generate a repeated START (Sr) and address another slave without first generating a STOP condition. Various combinations of read/write formats are then possible within such a transfer.

### 15.6.1. Data Transfer Formats

#### 15.6.1.1 Writing Data to AMIS-30624

When writing to AMIS-30624, 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 pulls the SDA line to '0'. The AMIS-30624 reads the incoming data at SDA on every rising edge of the SCK signal
- Stop condition to finish the transmission



Some commands for the AMIS-30624 are supporting eight bytes of data, other commands are transmitting two bytes of data. See Table 30.

# AMIS-30624

## 15.6.1.2 Reading Data to AMIS-30624

When reading data from AMIS-30624 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. This internal register address is stored in the circuit RAM.

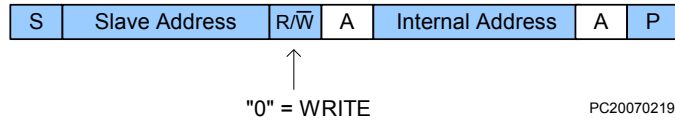


Figure 31: Master Reading Data from AMIS-30624: First Transmission is Addressing

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.

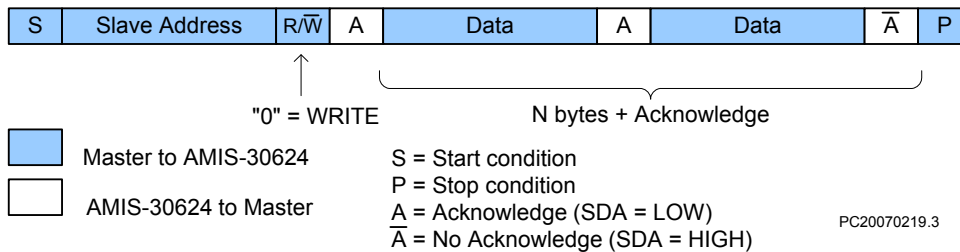


Figure 32: Master Reading Data from AMIS-30624: Second Transmission is Reading Data

- Notes:
- (1) Each byte is followed by an acknowledgment bit as indicated by the A or  $\bar{A}$  in the sequence.
  - (2) I<sup>2</sup>C-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.

## 15.7 7-bit Addressing

The addressing procedure for the I<sup>2</sup>C-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.

### 15.7.1. 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/  $\bar{W}$  bit.

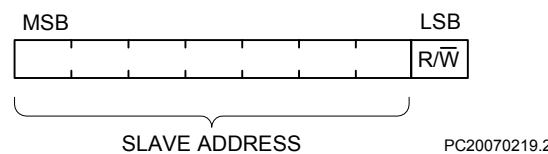
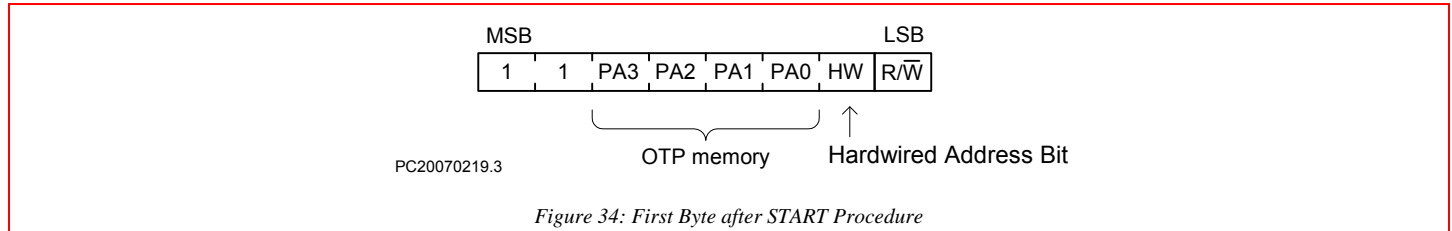


Figure 33: First Byte after START Procedure

# AMIS-30624

AMIS-30624 is provided with a physical address in order to discriminate this circuit from other circuits on the I<sup>2</sup>C 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](#)) 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 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](#)).



## 15.7.2. General Call Address

The AMIS-30624 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.

# AMIS-30624

## 16.0 I<sup>2</sup>C Application Commands

### 16.1 Introduction

Communications between the AMIS-30624 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

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

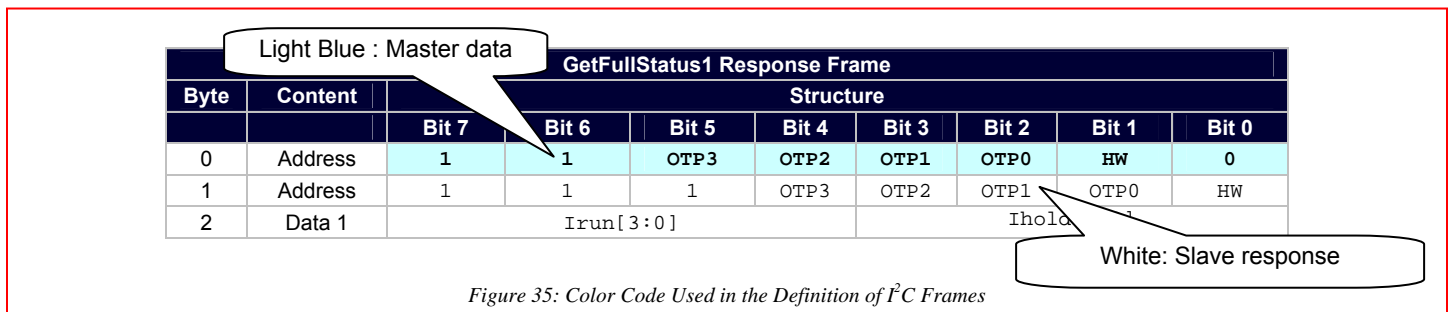
The I<sup>2</sup>C-bus master will have to use commands to manage the different application tasks the AMIS-30624 can feature. The commands summary is given in Table 30.

### 16.2 Commands Table

Table 30: I<sup>2</sup>C 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 I<sup>2</sup>C 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.



# AMIS-30624

## 16.3 Application Commands

### 16.3.1. 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 Table 18 and Table 19 to see the meaning of the parameters sent back to the I<sup>2</sup>C 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<sup>2</sup>C command frame:

| 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     |

| 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               |

# AMIS-30624

## 16.3.2. 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 Table 18 and Table 19 to see the meaning of the parameters sent back to the I<sup>2</sup>C master.

GetFullStatus2 corresponds to the following I<sup>2</sup>C command frame:

| 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     |

| 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

# AMIS-30624

## 16.3.3. GetOTPParam

This command is provided to the circuit by the I<sup>2</sup>C master to read the content of the OTP memory. More information can be found in [OTP Memory Structure](#).

GetOTPParam corresponds to the following I<sup>2</sup>C command frame:

| 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     |

| 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 |       |       |       |       |       |       |       |

## 16.3.4. GotoSecurePosition

This command is provided by the I<sup>2</sup>C master to one or all the stepper motors to move to the secure position SecPos [10:0]. See the [priority encoder](#) description for more details. The priority encoder table also acknowledges the cases where a GotoSecurePosition command will be ignored.

GotoSecurePosition corresponds to the following I<sup>2</sup>C command frame:

| 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     |

## 16.3.5. 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 I<sup>2</sup>C 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 I<sup>2</sup>C master for some safety reasons can also issue a HardStop command.

HardStop corresponds to the following I<sup>2</sup>C command frame:

| 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     |

## 16.3.6. ResetPosition

This command is provided to the circuit by the I<sup>2</sup>C master to reset ActPos and TagPos registers to zero. This can be helpful to prepare for instance a relative positioning.

# AMIS-30624

ResetPosition corresponds to the following I<sup>2</sup>C command frame:

| 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     |

## 16.3.7. ResetToDefault

This command is provided to the circuit by the I<sup>2</sup>C 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 18). This is another way for the I<sup>2</sup>C 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 I<sup>2</sup>C command frame:

| 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     |

## 16.3.8. RunVelocity

This command is provided to the circuit by the I<sup>2</sup>C master in order to put the motor in continuous motion state.

RunVelocity corresponds to the following I<sup>2</sup>C command frame:

| 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     |

## 16.3.9. SetDualPosition

This command is provided to the circuit by the I<sup>2</sup>C master in order to perform a positioning of the motor using two different velocities. See Section [Dual Positioning](#).

**Note1:** 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](#) 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 corresponds to the following I<sup>2</sup>C command frame:



# AMIS-30624

| 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

## 16.3.10. 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](#) for the meaning of these parameters.

SetStallParam corresponds to the following I<sup>2</sup>C command frame:

| 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]       |       |       |          |               |       |           |        |
| 6                           | Data 5  | MinSamples[2:0] |       |       | Shaft    | Acc[3:0]      |       |           |        |
| 7                           | Data 6  | AbsThr[3:0]     |       |       |          |               |       |           |        |
| 8                           | Data 7  | FS2StallEn[2:0] |       |       | AccShape | StepMode[1:0] |       | DC100StEn | PWMJEn |

## 16.3.11. SetMotorParam

This command is provided to the circuit by the I<sup>2</sup>C master to set the values for the stepper motor parameters (listed below) in RAM. Refer to Table 18 to see the meaning of the parameters sent by the I<sup>2</sup>C master.

**Important:** If a SetMotorParam occurs while a motion is ongoing, it will modify at once the motion parameters (see [Position Controller](#)). Therefore the application should not change parameters other than Vmax and Vmin while a motion is running, otherwise correct positioning cannot be guaranteed.

SetMotorParam corresponds to the following I<sup>2</sup>C command frame:

| 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]    |         |       |          |               |       |       |        |
| 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 |

# AMIS-30624

## 16.3.12. SetOTPPParam

This command is provided to the circuit by the I<sup>2</sup>C 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 Vbb voltage value. See parameter VbbOTP in Table 5. This is a mandatory condition to ensure reliable zapping.

SetOTPPParam corresponds to the following I<sup>2</sup>C command frame:

| SetOTPPParam 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

## 16.3.13. SetPosition

This command is provided to the circuit by the I<sup>2</sup>C master to drive the motor to a given absolute position. See [Positioning](#) for more details. The priority encoder table ([see Priority Encoder](#)) acknowledges the cases where a SetPosition command will be ignored.

SetPosition corresponds to the following I<sup>2</sup>C command frame:

| 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.

## 16.3.14. SoftStop

This command will be internally triggered when the chip temperature rises above the thermal shutdown threshold (see Table 5 and Section 14.2.5). It provokes an immediate deceleration to V<sub>min</sub> (see [Minimum Velocity](#)) 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 I<sup>2</sup>C Master for some safety reasons can also issue a SoftStop command.

SoftStop corresponds to the following I<sup>2</sup>C command frame:

| 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     |

## 16.3.15. TestBemf

This command is provided to the circuit by the I<sup>2</sup>C 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.

# AMIS-30624

TestBemf corresponds to the following I<sup>2</sup>C command frame:

| 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     |

# AMIS-30624

## 17.0 Resistance to Electrical and Electromagnetic Disturbances

### 17.1 Electrostatic Discharges

Table 31: Absolute Maximum Ratings

| Parameter           | Min. | Max. | Unit |
|---------------------|------|------|------|
| Vesd <sup>(1)</sup> | -2   | +2   | kV   |

Notes:

(1) Human body model (100pF via 1.5 kΩ, according to JEDEC EIA-JESD22-A114-B.)

### 17.2 Electrical Transient Conduction Along Supply Lines

Test pulses are applied to the power supply wires of the equipment implementing the AMIS-30624 (see application schematic), according to ISO 7637-1 document. Operating Classes are defined in ISO 7637-2.

Table 32: Test Pulses and Test Levels According to ISO 7637-1

| 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               |

### 17.3 EMC

Bulk current injection (BCI), according to ISO 11452-4. Operating Classes are defined in ISO 7637-2.

Table 33: Bulk Current Injection Operating Classes

| Current        | Operating Class |
|----------------|-----------------|
| 60mA envelope  | A               |
| 100mA envelope | B               |
| 200mA envelope | C               |

### 17.4 Power Supply Micro-interruptions

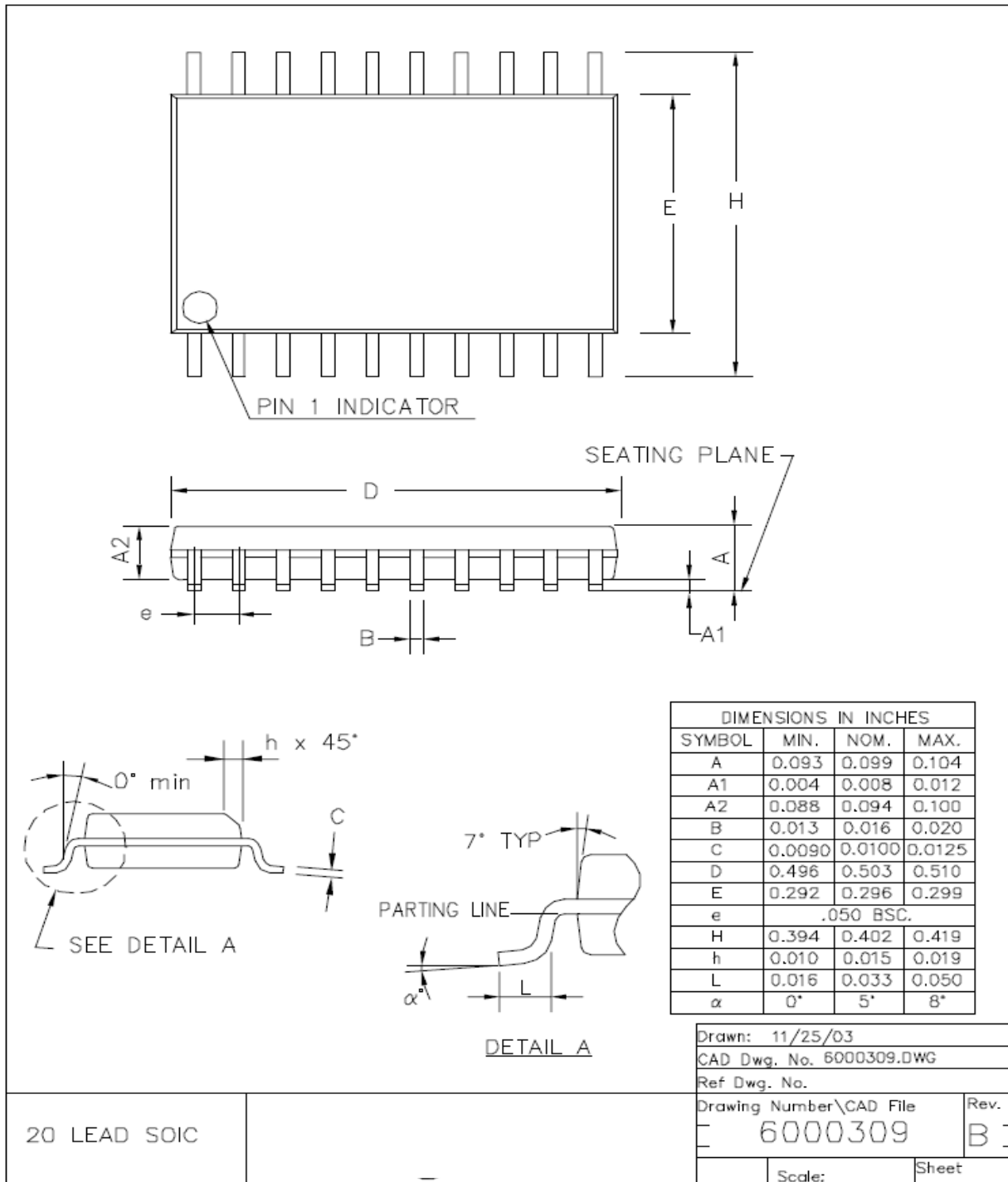
According to ISO 16750-2

Table 34: Immunity to Power Supply Micro-interruptions

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

# AMIS-30624

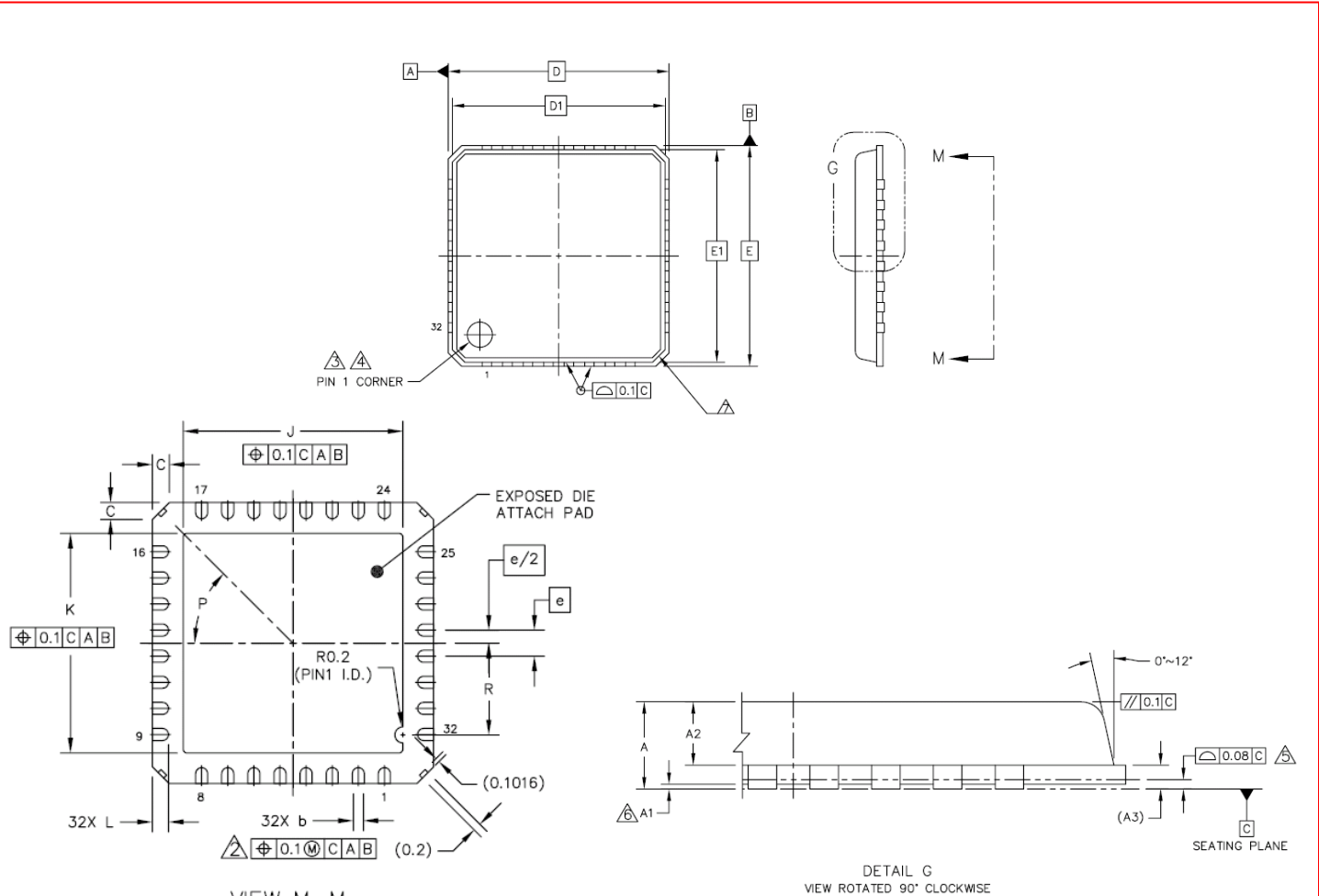
## 18.0 Package Outline



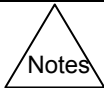
P010045.B

Figure 36: SOIC-20: Plastic Small Outline; 20 leads; Body Width 300mil. AMIS reference: SOIC300 20 300G

# AMIS-30624



| Dimensions: |       |       |       |        |
|-------------|-------|-------|-------|--------|
| Dim         | Min   | Nom   | Max   | Unit   |
| A           | 0.8   |       | 0.9   | mm     |
| A1          | 0     | 0.02  | 0.05  | mm     |
| A2          | 0.576 | 0.615 | 0.654 | mm     |
| A3          |       | 0.203 |       | mm     |
| b           | 0.25  | 0.3   | 0.35  | mm     |
| C           | 0.24  | 0.42  | 0.6   | mm     |
| D           |       | 7     |       | mm     |
| D1          |       | 6.75  |       | mm     |
| E           |       | 7     |       | mm     |
| E1          |       | 6.75  |       | mm     |
| e           |       | 0.65  |       | mm     |
| J           | 5.37  | 5.47  | 5.57  | mm     |
| K           | 5.37  | 5.47  | 5.57  | mm     |
| L           | 0.35  | 0.4   | 0.45  | mm     |
| P           |       | 45    |       | Degree |
| R           | 2.185 |       | 2.385 | mm     |



- 2) Dimensions apply to plated terminal and are measured between 0.2 and 0.25 mm from terminal tip.
- 3) The pin #1 indication 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

7x7 NQFP

Figure 37: NQFP-32: No lead Quad Flat Pack; 32 pins; body size 7 x 7 mm. AMIS reference: NQFP-32

## 19.0 Soldering

### 19.1 Introduction to Soldering Surface Mount Packages

This text gives a very brief insight to a complex technology. A more in-depth account of soldering ICs can be found in the ON Semiconductor “Data Handbook IC26; Integrated Circuit Packages” (document order number 9398 652 90011). There is no soldering method that is ideal for all surface mount IC packages. Wave soldering is not always suitable for surface mount ICs, or for printed-circuit boards with high population densities. In these situations re-flow soldering is often used.

### 19.2 Re-flow Soldering

Re-flow soldering requires solder paste (a suspension of fine solder particles, flux and binding agent) to be applied to the PCB by screen printing, stencilling or pressure-syringe dispensing before package placement. Several methods exist for reflowing; for example, infrared/convection heating in a conveyor type oven.

Throughput times (preheating, soldering and cooling) vary between 100 and 200 seconds depending on the heating method. Typical re-flow peak temperatures range from 215 to 260°C. The top-surface temperature of the packages should preferably be kept below 230°C.

### 19.3 Wave Soldering

Conventional single wave soldering is not recommended for surface mount devices (SMDs) or PCBs with a high component density, as solder bridging and non-wetting can present major problems. To overcome these problems, the double-wave soldering method was specifically developed.

If wave soldering is used the following conditions must be observed for optimal results:

- Use a double-wave soldering method comprising a turbulent wave with high upward pressure followed by a smooth laminar wave.
- For packages with leads on two sides and a pitch (e):
  - Larger than or equal to 1.27mm, the footprint longitudinal axis is preferred to be parallel to the transport direction of the PCB;
  - Smaller than 1.27mm, the footprint longitudinal axis must be parallel to the transport direction of the PCB. The footprint must incorporate solder thieves at the downstream end.
- For packages with leads on four sides, the footprint must be placed at a 45° angle to the transport direction of the PCB. The footprint must incorporate solder thieves downstream and at the side corners.

During placement and before soldering, the package must be fixed with a droplet of adhesive. The adhesive can be applied by screen printing, pin transfer or syringe dispensing. The package can be soldered after the adhesive is cured. Typical dwell time is four seconds at 250°C. A mildly-activated flux will eliminate the need for removal of corrosive residues in most applications.

### 19.4 Manual Soldering

Fix the component by first soldering two diagonally-opposite end leads. Use a low voltage (24V or less) soldering iron applied to the flat part of the lead. Contact time must be limited to 10 seconds at up to 300°C.

When using a dedicated tool, all other leads can be soldered in one operation within two to five seconds between 270 and 320°C.

Table 35: Soldering Process

| Package                         | Soldering Method                  |                        |
|---------------------------------|-----------------------------------|------------------------|
|                                 | Wave                              | Re-flow <sup>(1)</sup> |
| BGA, SQFP                       | Not suitable                      | Suitable               |
| HLQFP, HSQFP, HSOP, HTSSOP, SMS | Not suitable <sup>(2)</sup>       | Suitable               |
| PLCC (3) , SO, SOJ              | Suitable                          | Suitable               |
| LQFP, QFP, TQFP                 | Not recommended <sup>(3)(4)</sup> | Suitable               |
| SSOP, TSSOP, VSO                | Not recommended <sup>(5)</sup>    | Suitable               |

- Notes:
- (1) All surface mount (SMD) packages are moisture sensitive. Depending upon the moisture content, the maximum temperature (with respect to time) and body size of the package, there is a risk that internal or external package cracks may occur due to vaporization of the moisture in them (the so called popcorn effect). For details, refer to the drypack information in the “Data Handbook IC26; Integrated Circuit Packages; Section: Packing Methods.”
  - (2) These packages are not suitable for wave soldering as a solder joint between the printed-circuit board and heatsink (at bottom version) can not be achieved, and as solder may stick to the heatsink (on top version).
  - (3) If wave soldering is considered, then the package must be placed at a 45° angle to the solder wave direction. The package footprint must incorporate solder thieves downstream and at the side corners.
  - (4) Wave soldering is only suitable for LQFP, TQFP and QFP packages with a pitch (e) equal to or larger than 0.8mm; it is definitely not suitable for packages with a pitch (e) equal to or smaller than 0.65mm.
  - (5) Wave soldering is only suitable for SSOP and TSSOP packages with a pitch (e) equal to or larger than 0.65mm; it is definitely not suitable for packages with a pitch (e) equal to or smaller than 0.5mm.

# AMIS-30624

## 20.0 Revision History


---

| Revision | Date                            | Modifications   |
|----------|---------------------------------|---|
| 1.0      | July 16, 2002                   | First non-preliminary issue                                   |
| 2.1      | December 5 <sup>th</sup> , 2005 | Complete review   |
| 3.0      | February 21, 2007               | Public release  |
| 3.1      | March 23, 2007                  | Update <sup>2</sup> C commands, adding links                  |
| 4.0      | June 17, 2008                   | Move content into ON Semiconductor template; update OPN table |

## 21.0 Company or Product Inquiries

---

For more information about ON Semiconductor's products or services visit our Web site at <http://www.amis.com>.

ON Semiconductor and  are registered trademarks of Semiconductor Components Industries, LLC (SCILLC). SCILLC reserves the right to make changes without further notice to any products herein. SCILLC makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does SCILLC assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation special, consequential or incidental damages. "Typical" parameters which may be provided in SCILLC data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. SCILLC does not convey any license under its patent rights nor the rights of others. SCILLC products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the SCILLC product could create a situation where personal injury or death may occur. Should Buyer purchase or use SCILLC products for any such unintended or unauthorized application, Buyer shall indemnify and hold SCILLC and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that SCILLC was negligent regarding the design or manufacture of the part. SCILLC is an Equal Opportunity/Affirmative Action Employer. This literature is subject to all applicable copyright laws and is not for resale in any manner.

### PUBLICATION ORDERING INFORMATION

**LITERATURE FULFILLMENT:**

Literature Distribution Center for ON Semiconductor  
P.O. Box 5163, Denver, Colorado 80217 USA  
**Phone:** 303-675-2175 or 800-344-3860 Toll Free USA/Canada  
**Fax:** 303-675-2176 or 800-344-3867 Toll Free USA/Canada  
**Email:** [orderlit@onsemi.com](mailto:orderlit@onsemi.com)

**N. American Technical Support:** 800-282-9855  
Toll Free USA/Canada  
**Europe, Middle East and Africa Technical Support:**  
Phone: 421 33 790 2910  
**Japan Customer Focus Center**  
Phone: 81-3-5773-3850

ON Semiconductor Website: [www.onsemi.com](http://www.onsemi.com)

Order Literature: <http://www.onsemi.com/orderlit>

For additional information, please contact your local Sales Representative



## Данный компонент на территории Российской Федерации

### Вы можете приобрести в компании MosChip.

Для оперативного оформления запроса Вам необходимо перейти по данной ссылке:

<http://moschip.ru/get-element>

Вы можете разместить у нас заказ для любого Вашего проекта, будь то серийное производство или разработка единичного прибора.

В нашем ассортименте представлены ведущие мировые производители активных и пассивных электронных компонентов.

Нашей специализацией является поставка электронной компонентной базы двойного назначения, продукции таких производителей как XILINX, Intel (ex.ALTERA), Vicor, Microchip, Texas Instruments, Analog Devices, Mini-Circuits, Amphenol, Glenair.

Сотрудничество с глобальными дистрибьюторами электронных компонентов, предоставляет возможность заказывать и получать с международных складов практически любой перечень компонентов в оптимальные для Вас сроки.

На всех этапах разработки и производства наши партнеры могут получить квалифицированную поддержку опытных инженеров.

Система менеджмента качества компании отвечает требованиям в соответствии с ГОСТ Р ИСО 9001, ГОСТ РВ 0015-002 и ЭС РД 009

### Офис по работе с юридическими лицами:

105318, г.Москва, ул.Щербаковская д.3, офис 1107, 1118, ДЦ «Щербаковский»

Телефон: +7 495 668-12-70 (многоканальный)

Факс: +7 495 668-12-70 (доб.304)

E-mail: [info@moschip.ru](mailto:info@moschip.ru)

Skype отдела продаж:

moschip.ru

moschip.ru\_4

moschip.ru\_6

moschip.ru\_9