



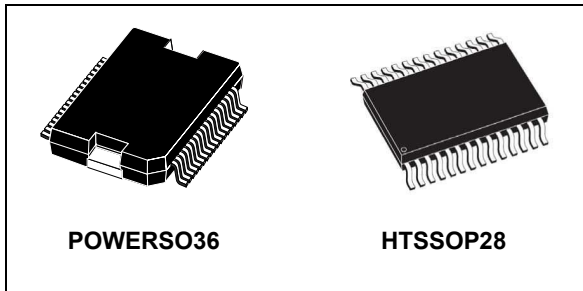
HESTORE.HU
elektronikai alkatrész áruház

EN: This Datasheet is presented by the manufacturer.

Please visit our website for pricing and availability at www.hestore.hu.

Fully integrated microstepping motor driver with motion engine and SPI

Datasheet - production data



Applications

- Bipolar stepper motors

Description

The L6470 device, realized in analog mixed signal technology, is an advanced fully integrated solution suitable for driving two-phase bipolar stepper motors with microstepping. It integrates a dual low $R_{DS(on)}$ DMOS full bridge with all of the power switches equipped with an accurate on-chip current sensing circuitry suitable for non-dissipative current control and overcurrent protection. Thanks to a unique control system, a true 1/128 steps resolution is achieved. The digital control core can generate user defined motion profiles with acceleration, deceleration, speed or target position, easily programmed through a dedicated registers set. All commands and data registers, including those used to set analogue values (i.e. current control value, current protection trip point, deadtime, PWM frequency, etc.) are sent through a standard 5-Mbit/s SPI. A very rich set of protections (thermal, low bus voltage, overcurrent, motor stall) allows the design of a fully protected application, as required by the most demanding motor control applications.

Features

- Operating voltage: 8 - 45 V
- 7.0 A out peak current (3.0 A_{r.m.s.})
- Low $R_{DS(on)}$ Power MOSFETs
- Programmable speed profile and positioning
- Programmable power MOS slew rate
- Up to 1/128 microstepping
- Sensorless stall detection
- SPI interface
- Low quiescent and standby currents
- Programmable non-dissipative overcurrent protection on high and low-side
- Two-levels of overtemperature protection

Table 1. Device summary

| Order codes | Package | Packaging |
|-------------|-----------|---------------|
| L6470H | HTSSOP28 | Tube |
| L6470HTR | HTSSOP28 | Tape and reel |
| L6470PD | POWERSO36 | Tube |
| L6470PDTR | POWERSO36 | Tape and reel |

Contents

- 1 Block diagram 8**
- 2 Electrical data 9**
 - 2.1 Absolute maximum ratings 9
 - 2.2 Recommended operating conditions 10
 - 2.3 Thermal data 10
- 3 Electrical characteristics 11**
- 4 Pin connection 16**
 - Pin list 17
- 5 Typical applications 19**
- 6 Functional description 21**
 - 6.1 Device power-up 21
 - 6.2 Logic I/O 21
 - 6.3 Charge pump 21
 - 6.4 Microstepping 22
 - Automatic full-step mode 23
 - 6.5 Absolute position counter 23
 - 6.6 Programmable speed profiles 23
 - 6.7 Motor control commands 24
 - 6.7.1 Constant speed commands 24
 - 6.7.2 Positioning commands 24
 - 6.7.3 Motion commands 25
 - 6.7.4 Stop commands 25
 - 6.7.5 Step-clock mode 26
 - 6.7.6 GoUntil and ReleaseSW commands 27
 - 6.8 Internal oscillator and oscillator driver 27
 - 6.8.1 Internal oscillator 28
 - 6.8.2 External clock source 28
 - 6.9 Overcurrent detection 29

| | | |
|----------|---|-----------|
| 6.10 | Undervoltage lockout (UVLO) | 30 |
| 6.11 | Thermal warning and thermal shutdown | 30 |
| 6.12 | Reset and standby | 30 |
| 6.13 | External switch (SW pin) | 31 |
| 6.14 | Programmable DMOS slew rate, deadtime and blanking time | 31 |
| 6.15 | Integrated analog-to-digital converter | 32 |
| 6.16 | Internal voltage regulator | 32 |
| 6.17 | BUSY\SYNC pin | 32 |
| 6.17.1 | BUSY operation mode | 33 |
| 6.17.2 | SYNC operation mode | 33 |
| 6.18 | FLAG pin | 33 |
| 7 | Phase current control | 34 |
| 7.1 | PWM sinewave generators | 34 |
| 7.2 | Sensorless stall detection | 35 |
| 7.3 | Low speed optimization | 35 |
| 7.4 | BEMF compensation | 36 |
| 7.5 | Motor supply voltage compensation | 36 |
| 7.6 | Winding resistance thermal drift compensation | 37 |
| 8 | Serial interface | 38 |
| 9 | Programming manual | 40 |
| 9.1 | Registers and flags description | 40 |
| 9.1.1 | ABS_POS | 41 |
| 9.1.2 | EL_POS | 41 |
| 9.1.3 | MARK | 42 |
| 9.1.4 | SPEED | 42 |
| 9.1.5 | ACC | 42 |
| 9.1.6 | DEC | 43 |
| 9.1.7 | MAX_SPEED | 43 |
| 9.1.8 | MIN_SPEED | 43 |
| 9.1.9 | FS_SPD | 44 |
| 9.1.10 | KVAL_HOLD, KVAL_RUN, KVAL_ACC and KVAL_DEC | 44 |
| 9.1.11 | INT_SPEED | 45 |
| 9.1.12 | ST_SLP | 45 |

| | | |
|------------|-------------------------------|-----------|
| 9.1.13 | FN_SLP_ACC | 45 |
| 9.1.14 | FN_SLP_DEC | 45 |
| 9.1.15 | K_THERM | 46 |
| 9.1.16 | ADC_OUT | 46 |
| 9.1.17 | OCD_TH | 47 |
| 9.1.18 | STALL_TH | 47 |
| 9.1.19 | STEP_MODE | 47 |
| 9.1.20 | ALARM_EN | 49 |
| 9.1.21 | CONFIG | 49 |
| 9.1.22 | STATUS | 55 |
| 9.2 | Application commands | 56 |
| 9.2.1 | Command management | 58 |
| 9.2.2 | Nop | 58 |
| 9.2.3 | SetParam (PARAM, VALUE) | 59 |
| 9.2.4 | GetParam (PARAM) | 59 |
| 9.2.5 | Run (DIR, SPD) | 60 |
| 9.2.6 | StepClock (DIR) | 60 |
| 9.2.7 | Move (DIR, N_STEP) | 61 |
| 9.2.8 | GoTo (ABS_POS) | 61 |
| 9.2.9 | GoTo_DIR (DIR, ABS_POS) | 62 |
| 9.2.10 | GoUntil (ACT, DIR, SPD) | 62 |
| 9.2.11 | ReleaseSW (ACT, DIR) | 63 |
| 9.2.12 | GoHome | 63 |
| 9.2.13 | GoMark | 63 |
| 9.2.14 | ResetPos | 64 |
| 9.2.15 | ResetDevice | 64 |
| 9.2.16 | SoftStop | 64 |
| 9.2.17 | HardStop | 65 |
| 9.2.18 | SoftHiZ | 65 |
| 9.2.19 | HardHiZ | 65 |
| 9.2.20 | GetStatus | 66 |
| 10 | Package information | 67 |
| 10.1 | HTSSOP28 package information | 68 |
| 10.2 | POWERSO36 package information | 70 |
| 11 | Revision history | 72 |

List of tables

| | | |
|-----------|--|----|
| Table 1. | Device summary | 1 |
| Table 2. | Absolute maximum ratings | 9 |
| Table 3. | Recommended operating conditions | 10 |
| Table 4. | Thermal data | 10 |
| Table 5. | Electrical characteristics | 11 |
| Table 6. | Pin description | 17 |
| Table 7. | Typical application values | 19 |
| Table 8. | CL values according to external oscillator frequency | 28 |
| Table 9. | Register map | 40 |
| Table 10. | EL_POS register | 41 |
| Table 11. | MIN_SPEED register | 43 |
| Table 12. | Voltage amplitude regulation registers | 44 |
| Table 13. | Winding resistance thermal drift compensation coefficient | 46 |
| Table 14. | ADC_OUT value and motor supply voltage compensation feature | 46 |
| Table 15. | Overcurrent detection threshold | 47 |
| Table 16. | Stall detection threshold | 47 |
| Table 17. | STEP_MODE register | 47 |
| Table 18. | Step mode selection | 48 |
| Table 19. | SYNC output frequency | 48 |
| Table 20. | SYNC signal source | 49 |
| Table 21. | ALARM_EN register | 49 |
| Table 22. | CONFIG register | 49 |
| Table 23. | Oscillator management | 50 |
| Table 24. | External switch hard stop interrupt mode | 51 |
| Table 25. | Overcurrent event | 51 |
| Table 26. | Programmable power bridge output slew rate values | 51 |
| Table 27. | Motor supply voltage compensation enable | 51 |
| Table 28. | PWM frequency: integer division factor | 52 |
| Table 29. | PWM frequency: multiplication factor | 52 |
| Table 30. | Available PWM frequencies [kHz]: 8-MHz oscillator frequency | 53 |
| Table 31. | Available PWM frequencies [kHz]: 16-MHz oscillator frequency | 53 |
| Table 32. | Available PWM frequencies [kHz]: 24-MHz oscillator frequency | 54 |
| Table 33. | Available PWM frequencies [kHz]: 32-MHz oscillator frequency | 54 |
| Table 34. | STATUS register | 55 |
| Table 35. | STATUS register DIR bit | 55 |
| Table 36. | STATUS register MOT_STATUS bits | 56 |
| Table 37. | Application commands | 56 |
| Table 38. | Nop command structure | 58 |
| Table 39. | SetParam command structure | 59 |
| Table 40. | GetParam command structure | 59 |
| Table 41. | Run command structure | 60 |
| Table 42. | Stepclock command structure | 60 |
| Table 43. | Move command structure | 61 |
| Table 44. | GoTo command structure | 61 |
| Table 45. | GoTo_DIR command structure | 62 |
| Table 46. | GoUntil command structure | 62 |
| Table 47. | ReleaseSW command structure | 63 |
| Table 48. | GoHome command structure | 63 |

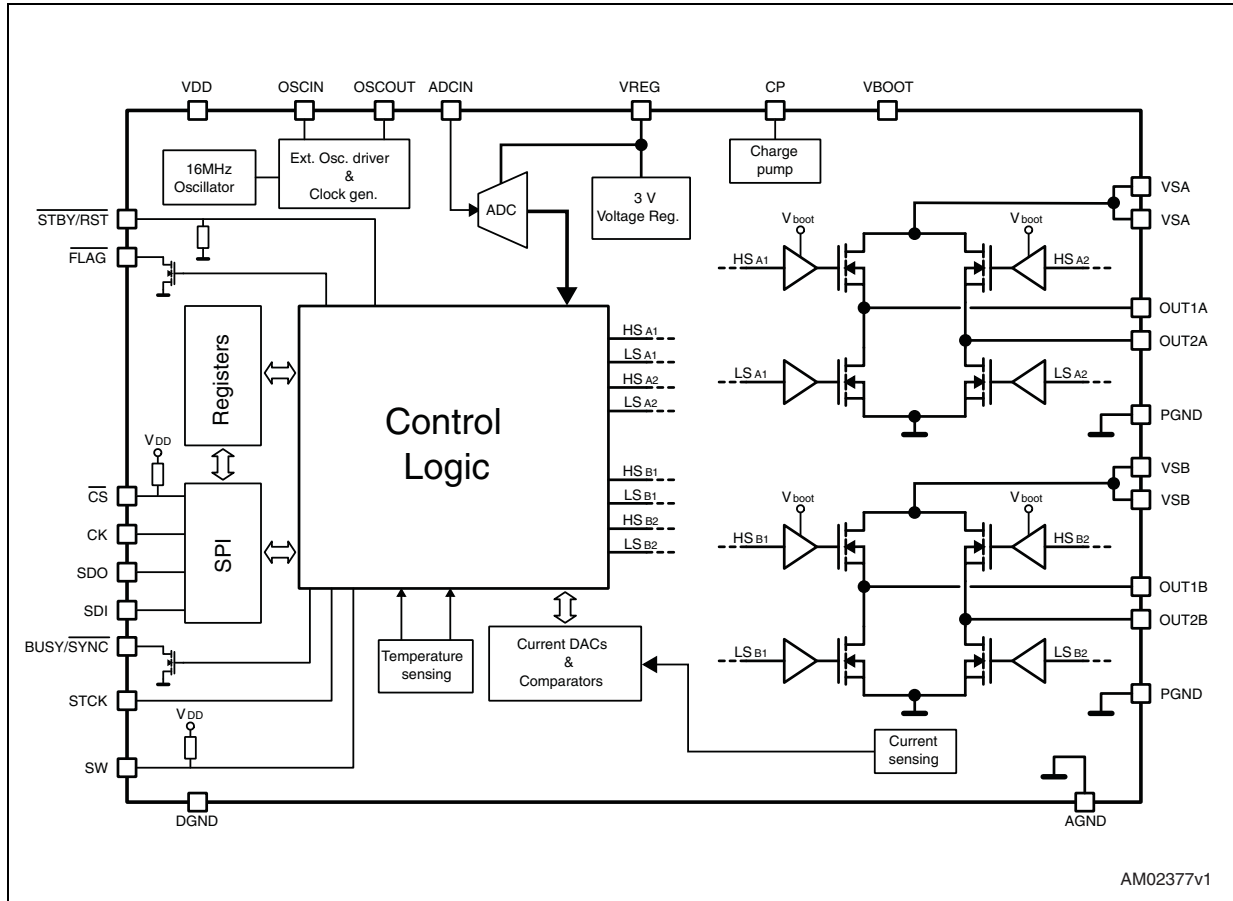
| | | |
|-----------|---|----|
| Table 49. | GoMark command structure | 63 |
| Table 50. | ResetPos command structure | 64 |
| Table 51. | ResetDevice command structure | 64 |
| Table 52. | SoftStop command structure | 64 |
| Table 53. | HardStop command structure | 65 |
| Table 54. | SoftHiZ command structure | 65 |
| Table 55. | HardHiZ command structure | 65 |
| Table 56. | GetStatus command structure | 66 |
| Table 57. | HTSSOP28 package mechanical data | 69 |
| Table 58. | POWERSO36 package mechanical data | 71 |
| Table 59. | Revision history | 72 |

List of figures

| | | |
|------------|---|----|
| Figure 1. | Block diagram | 8 |
| Figure 2. | HTSSOP28 pin connection (top view) | 16 |
| Figure 3. | POWERSO36 pin connection (top view) | 16 |
| Figure 4. | Bipolar stepper motor control application using L6470 | 20 |
| Figure 5. | Charge pump circuitry | 22 |
| Figure 6. | Normal mode and microstepping (128 microsteps) | 22 |
| Figure 7. | Automatic full-step switching | 23 |
| Figure 8. | Constant speed command examples | 24 |
| Figure 9. | Positioning command examples | 25 |
| Figure 10. | Motion command examples | 25 |
| Figure 11. | OSCIN and OSCOUT pin configuration | 29 |
| Figure 12. | External switch connection | 31 |
| Figure 13. | Internal 3 V linear regulator | 32 |
| Figure 14. | Current distortion and compensation | 35 |
| Figure 15. | BEMF compensation curve | 36 |
| Figure 16. | Motor supply voltage compensation circuit | 37 |
| Figure 17. | SPI timings diagram | 38 |
| Figure 18. | Daisy chain configuration | 39 |
| Figure 19. | Command with 3-byte argument | 58 |
| Figure 20. | Command with 3-byte response | 58 |
| Figure 21. | Command response aborted | 58 |
| Figure 22. | HTSSOP28 package outline | 68 |
| Figure 23. | POWERSO36 package outline | 70 |

1 Block diagram

Figure 1. Block diagram



2 Electrical data

2.1 Absolute maximum ratings

Table 2. Absolute maximum ratings

| Symbol | Parameter | Test condition | Value | Unit |
|-----------------------|---|----------------------------|--------------|------|
| V_{DD} | Logic interface supply voltage | | 5.5 | V |
| V_S | Motor supply voltage | $V_{SA} = V_{SB} = V_S$ | 48 | V |
| $V_{GND, diff}$ | Differential voltage between AGND, PGND and DGND | | ± 0.3 | V |
| V_{boot} | Bootstrap peak voltage | | 55 | V |
| V_{REG} | Internal voltage regulator output pin and logic supply voltage | | 3.6 | V |
| V_{ADCIN} | Integrated ADC input voltage range (ADCIN pin) | | -0.3 to +3.6 | V |
| V_{OSC} | OSCIN and OSCOUT pin voltage range | | -0.3 to +3.6 | V |
| $V_{out, diff}$ | Differential voltage between V_{SA} , OUT1 _A , OUT2 _A , PGND and V_{SB} , OUT1 _B , OUT2 _B , PGND pins | $V_{SA} = V_{SB} = V_S$ | 48 | V |
| V_{LOGIC} | Logic inputs voltage range | | -0.3 to +5.5 | V |
| $I_{out}^{(1)}$ | R.m.s. output current | | 3 | A |
| $I_{out, peak}^{(1)}$ | Pulsed output current | $T_{PULSE} < 1 \text{ ms}$ | 7 | A |
| T_{OP} | Operating junction temperature | | -40 to 150 | °C |
| T_s | Storage temperature range | | -55 to 150 | °C |
| P_{tot} | Total power dissipation ($T_A = 25 \text{ °C}$) ⁽²⁾ | | 5 | W |

1. Maximum output current limit is related to metal connection and bonding characteristics. Actual limit must satisfy maximum thermal dissipation constraints.

2. HTSSOP28 mounted on EVAL6470H.

2.2 Recommended operating conditions

Table 3. Recommended operating conditions

| Symbol | Parameter | Test condition | Value | | | Unit |
|-----------------|---|--|-------|-----|-----------|------|
| V_{DD} | Logic interface supply voltage | 3.3 V logic outputs | | 3.3 | | V |
| | | 5 V logic outputs | | 5 | | |
| V_S | Motor supply voltage | $V_{SA} = V_{SB} = V_S$ | 8 | | 45 | V |
| V_{out_diff} | Differential voltage between V_{SA} , OUT1 _A , OUT2 _A , PGND and V_{SB} , OUT1 _B , OUT2 _B , PGND pins | $V_{SA} = V_{SB} = V_S$ | | | 45 | V |
| $V_{REG,in}$ | Logic supply voltage | V_{REG} voltage imposed by external source | 3.2 | 3.3 | | V |
| V_{ADC} | Integrated ADC input voltage (ADCIN pin) | | 0 | | V_{REG} | V |

2.3 Thermal data

Table 4. Thermal data

| Symbol | Parameter | Package | Typ. | Unit |
|------------|-------------------------------------|--------------------------|------|------|
| R_{thJA} | Thermal resistance junction ambient | HTSSOP28 ⁽¹⁾ | 22 | °C/W |
| | | POWERSO36 ⁽²⁾ | 12 | |

1. HTSSOP28 mounted on the EVAL6470H rev 1.0 board: a four-layer FR4 PCB with a dissipating copper surface of about 40 cm² on each layer and 15 via holes below the IC.
2. POWERSO36 mounted on the EVAL6470PD rev 1.0 board: a four-layer FR4 PCB with a dissipating copper surface of about 40 cm² on each layer and 22 via holes below the IC.

3 Electrical characteristics

$V_{SA} = V_{SB} = 36\text{ V}$; $V_{DD} = 3.3\text{ V}$; internal 3 V regulator; $T_J = 25\text{ °C}$, unless otherwise specified.

Table 5. Electrical characteristics

| Symbol | Parameter | Test condition | Min. | Typ. | Max. | Unit |
|-------------------------------|---|---|------|------|------|------|
| General | | | | | | |
| V_{SthOn} | V_S UVLO turn-on threshold | | 7.5 | 8.2 | 8.9 | V |
| V_{SthOff} | V_S UVLO turn-off threshold | | 6.6 | 7.2 | 7.8 | V |
| $V_{SthHyst}$ | V_S UVLO threshold hysteresis | | 0.7 | 1 | 1.3 | V |
| I_q | Quiescent motor supply current | Internal oscillator selected; VREG = 3.3 V ext.; CP floating | | 0.5 | 0.65 | mA |
| $T_{j(WRN)}$ | Thermal warning temperature | | | 130 | | °C |
| $T_{j(SD)}$ | Thermal shutdown temperature | | | 160 | | °C |
| Charge pump | | | | | | |
| V_{pump} | Voltage swing for charge pump oscillator | | | 10 | | V |
| $f_{pump,min}$ | Minimum charge pump oscillator frequency ⁽¹⁾ | | | 660 | | kHz |
| $f_{pump,max}$ | Maximum charge pump oscillator frequency ⁽¹⁾ | | | 800 | | kHz |
| I_{boot} | Average boot current | $f_{sw,A} = f_{sw,B} = 15.6\text{ kHz}$ POW_SR = '10' | | 1.1 | 1.4 | mA |
| Output DMOS transistor | | | | | | |
| $R_{DS(on)}$ | High-side switch on-resistance | $T_J = 25\text{ °C}$, $I_{out} = 3\text{ A}$ | | 0.37 | | Ω |
| | | $T_J = 125\text{ °C}$, ⁽²⁾ $I_{out} = 3\text{ A}$ | | 0.51 | | |
| $R_{DS(on)}$ | Low-side switch on-resistance | $T_J = 25\text{ °C}$, $I_{out} = 3\text{ A}$ | | 0.18 | | |
| | | $T_J = 125\text{ °C}$, ⁽²⁾ $I_{out} = 3\text{ A}$ | | 0.23 | | |
| I_{DSS} | Leakage current | OUT = V_S | | | 3.1 | mA |
| | | OUT = GND | -0.3 | | | |
| t_r | Rise time ⁽³⁾ | POW_SR = '00', $I_{out} = +1\text{ A}$ | | 100 | | ns |
| | | POW_SR = '00', $I_{out} = -1\text{ A}$ | | 80 | | |
| | | POW_SR = '11', $I_{out} = \pm 1\text{ A}$ | | 100 | | |
| | | POW_SR = '10', $I_{out} = \pm 1\text{ A}$ | | 200 | | |
| | | POW_SR = '01', $I_{out} = \pm 1\text{ A}$ | | 300 | | |

Table 5. Electrical characteristics (continued)

| Symbol | Parameter | Test condition | Min. | Typ. | Max. | Unit |
|---------------------------------|---|--|------|------|------|------|
| t _f | Fall time ⁽³⁾ | POW_SR = '00'; I _{out} = +1 A | | 90 | | ns |
| | | POW_SR = '00'; I _{out} = -1 A | | 110 | | |
| | | POW_SR = '11', I _{out} = ±1 A | | 110 | | |
| | | POW_SR = '10', I _{out} = ±1 A | | 260 | | |
| | | POW_SR = '01', I _{load} = ±1 A | | 375 | | |
| SR _{out_r} | Output rising slew rate | POW_SR = '00', I _{out} = +1 A | | 285 | | V/μs |
| | | POW_SR = '00', I _{out} = -1 A | | 360 | | |
| | | POW_SR = '11', I _{out} = ±1 A | | 285 | | |
| | | POW_SR = '10', I _{out} = ±1 A | | 150 | | |
| | | POW_SR = '01', I _{out} = ±1 A | | 95 | | |
| SR _{out_f} | Output falling slew rate | POW_SR = '00', I _{out} = +1 A | | 320 | | V/μs |
| | | POW_SR = '00', I _{out} = -1 A | | 260 | | |
| | | POW_SR = '11', I _{out} = ±1 A | | 260 | | |
| | | POW_SR = '10', I _{out} = ±1 A | | 110 | | |
| | | POW_SR = '01', I _{out} = ±1 A | | 75 | | |
| Deadtime and blanking | | | | | | |
| t _{DT} | Deadtime ⁽¹⁾ | POW_SR = '00' | | 250 | | ns |
| | | POW_SR = '11', f _{OSC} = 16 MHz | | 375 | | |
| | | POW_SR = '10', f _{OSC} = 16 MHz | | 625 | | |
| | | POW_SR = '01', f _{OSC} = 16 MHz | | 875 | | |
| t _{blank} | Blanking time ⁽¹⁾ | POW_SR = '00' | | 250 | | ns |
| | | POW_SR = '11', f _{OSC} = 16 MHz | | 375 | | |
| | | POW_SR = '10', f _{OSC} = 16 MHz | | 625 | | |
| | | POW_SR = '01', f _{OSC} = 16 MHz | | 875 | | |
| Source-drain diodes | | | | | | |
| V _{SD,HS} | High-side diode forward ON voltage | I _{out} = 1 A | | 1 | 1.1 | V |
| V _{SD,LS} | Low-side diode forward ON voltage | I _{out} = 1 A | | 1 | 1.1 | V |
| t _{rrHS} | High-side diode reverse recovery time | I _{out} = 1 A | | 30 | | ns |
| t _{rrLS} | Low-side diode reverse recovery time | I _{out} = 1 A | | 100 | | ns |
| Logic inputs and outputs | | | | | | |
| V _{IL} | Low logic level input voltage | | | | 0.8 | V |
| V _{IH} | High logic level input voltage | | 2 | | | V |
| I _{IH} | High logic level input current ⁽⁴⁾ | V _{IN} = 5 V | | | 1 | μA |
| I _{IL} | Low logic level input current ⁽⁵⁾ | V _{IN} = 0 V | -1 | | | μA |

Table 5. Electrical characteristics (continued)

| Symbol | Parameter | Test condition | Min. | Typ. | Max. | Unit |
|---|---|---|------|------|------|---------------|
| V_{OL} | Low logic level output voltage ⁽⁶⁾ | $V_{DD} = 3.3\text{ V}$, $I_{OL} = 4\text{ mA}$ | | | 0.3 | V |
| | | $V_{DD} = 5\text{ V}$, $I_{OL} = 4\text{ mA}$ | | | 0.3 | |
| V_{OH} | High logic level output voltage | $V_{DD} = 3.3\text{ V}$, $I_{OH} = 4\text{ mA}$ | 2.4 | | | V |
| | | $V_{DD} = 5\text{ V}$, $I_{OH} = 4\text{ mA}$ | 4.7 | | | |
| R_{PU} R_{PD} | CS pull-up and STBY pull-down resistors | $\overline{CS} = \text{GND}$; $\overline{STBY/RST} = 5\text{ V}$ | 335 | 430 | 565 | k Ω |
| I_{logic} | Internal logic supply current | 3.3 V V_{REG} externally supplied, internal oscillator | | 3.7 | 4.3 | mA |
| $I_{logic,STBY}$ | Standby mode internal logic supply current | 3.3 V V_{REG} externally supplied | | 2 | 2.5 | μA |
| f_{STCK} | Step-clock input frequency | | | | 2 | MHz |
| Internal oscillator and external oscillator driver | | | | | | |
| $f_{osc,i}$ | Internal oscillator frequency | $T_j = 25\text{ }^\circ\text{C}$, $V_{REG} = 3.3\text{ V}$ | -3% | 16 | +3% | MHz |
| $f_{osc,e}$ | Programmable external oscillator frequency | | 8 | | 32 | MHz |
| $V_{OSCOUTH}$ | OSCOUT clock source high level voltage | Internal oscillator 3.3 V V_{REG} externally supplied; $I_{OSCOUT} = 4\text{ mA}$ | 2.4 | | | V |
| $V_{OSCOU TL}$ | OSCOUT clock source low level voltage | Internal oscillator 3.3 V V_{REG} externally supplied; $I_{OSCOUT} = 4\text{ mA}$ | | | 0.3 | V |
| $t_{rOSCOUT}$ $t_{fOSCOUT}$ | OSCOUT clock source rise and fall time | Internal oscillator | | | 20 | ns |
| t_{extosc} | Internal to external oscillator switching delay | | | 3 | | ms |
| t_{intosc} | External to internal oscillator switching delay | | | 1.5 | | μs |
| SPI | | | | | | |
| $f_{CK,MAX}$ | Maximum SPI clock frequency ⁽⁷⁾ | | 5 | | | MHz |
| t_{rCK} t_{fCK} | SPI clock rise and fall time ⁽⁷⁾ | $C_L = 30\text{ pF}$ | | | 25 | ns |
| t_{hCK} t_{lCK} | SPI clock high and low time ⁽⁷⁾ | | 75 | | | ns |
| t_{setCS} | Chip select setup time ⁽⁷⁾ | | 350 | | | ns |
| t_{holCS} | Chip select hold time ⁽⁷⁾ | | 10 | | | ns |
| t_{disCS} | Deselect time ⁽⁷⁾ | | 800 | | | ns |
| t_{setSDI} | Data input setup time ⁽⁷⁾ | | 25 | | | ns |
| t_{holSDI} | Data input hold time ⁽⁷⁾ | | 20 | | | ns |

Table 5. Electrical characteristics (continued)

| Symbol | Parameter | Test condition | Min. | Typ. | Max. | Unit |
|-----------------------------------|---|---|------|-----------|------|------------|
| t_{enSDO} | Data output enable time ⁽⁷⁾ | | | | 38 | ns |
| t_{disSDO} | Data output disable time ⁽⁷⁾ | | | | 47 | ns |
| t_{vSDO} | Data output valid time ⁽⁷⁾ | | | | 57 | ns |
| t_{holSDO} | Data output hold time ⁽⁷⁾ | | 37 | | | ns |
| Switch input (SW) | | | | | | |
| R_{PUSW} | SW input pull-up resistance | SW = GND | 60 | 85 | 110 | k Ω |
| PWM modulators | | | | | | |
| f_{PWM} | Programmable PWM frequency ⁽¹⁾ | $f_{osc} = 16$ MHz | 2.8 | | 62.5 | kHz |
| | | $f_{osc} = 32$ MHz | 5.6 | | 125 | |
| N_{PWM} | PWM resolution | | | 8 | | bit |
| Stall detection | | | | | | |
| $I_{STALL,MAX}$ | Maximum programmable stall threshold | STALL_TH = '1111111' | | 4 | | A |
| $I_{STALL,MIN}$ | Minimum programmable stall threshold | STALL_TH = '0000000' | | 31.2 5 | | mA |
| $I_{STALL,RES}$ | Programmable stall threshold resolution | | | 31.2 5 | | mA |
| Overcurrent protection | | | | | | |
| $I_{OCD,MAX}$ | Maximum programmable overcurrent detection threshold | OCD_TH = '1111' | | 6 | | A |
| $I_{OCD,MIN}$ | Minimum programmable overcurrent detection threshold | OCD_TH = '0000' | | 0.37 5 | | A |
| $I_{OCD,RES}$ | Programmable overcurrent detection threshold resolution | | | 0.37 5 | | A |
| $t_{OCD,Flag}$ | OCD to flag signal delay time | $dl_{out}/dt = 350$ A/ μ s | | 650 | 1000 | ns |
| $t_{OCD,SD}$ | OCD to shutdown delay time | $dl_{out}/dt = 350$ A/ μ s POW_SR = '10' | | 600 | | ns |
| Standby | | | | | | |
| I_{qSTBY} | Quiescent motor supply current in standby conditions | $V_S = 8$ V | | 26 | 34 | μ A |
| | | $V_S = 36$ V | | 30 | 36 | |
| $t_{STBY,min}$ | Minimum standby time | | | 10 | | μ s |
| $t_{logicwu}$ | Logic power-on and wake-up time | | | 38 | 45 | μ s |
| t_{cpwu} | Charge pump power-on and wake-up time | Power bridges disabled, $C_p = 10$ nF, $C_{boot} = 220$ nF | | 650 | | μ s |
| Internal voltage regulator | | | | | | |
| V_{REG} | Voltage regulator output voltage | | 2.9 | 3 | 3.2 | V |
| I_{REG} | Voltage regulator output current | | | | 40 | mA |

Table 5. Electrical characteristics (continued)

| Symbol | Parameter | Test condition | Min. | Typ. | Max. | Unit |
|---|--|---------------------------|------|-----------|------|------|
| $V_{REG, drop}$ | Voltage regulator output voltage drop | $I_{REG} = 40 \text{ mA}$ | | 50 | | mV |
| $I_{REG, STBY}$ | Voltage regulator standby output current | | | | 10 | mA |
| Integrated analog-to-digital converter | | | | | | |
| N_{ADC} | Analog-to-digital converter resolution | | | 5 | | bit |
| $V_{ADC, ref}$ | Analog-to-digital converter reference voltage | | | V_{REG} | | V |
| f_S | Analog-to-digital converter sampling frequency | | | f_{PWM} | | kHz |

1. Accuracy depends on oscillator frequency accuracy.
2. Tested at 25 °C in a restricted range and guaranteed by characterization.
3. Rise and fall time depends on motor supply voltage value. Refer to SR_{out} values in order to evaluate the actual rise and fall time.
4. Not valid for $\overline{STBY/RST}$ pin which has internal pull-down resistor.
5. Not valid for SW and CS pins which have internal pull-up resistors.
6. \overline{FLAG} , \overline{BUSY} and SYNC open drain outputs included.
7. See [Figure 17 on page 38](#) – SPI timings diagram for details.

4 Pin connection

Figure 2. HTSSOP28 pin connection (top view)

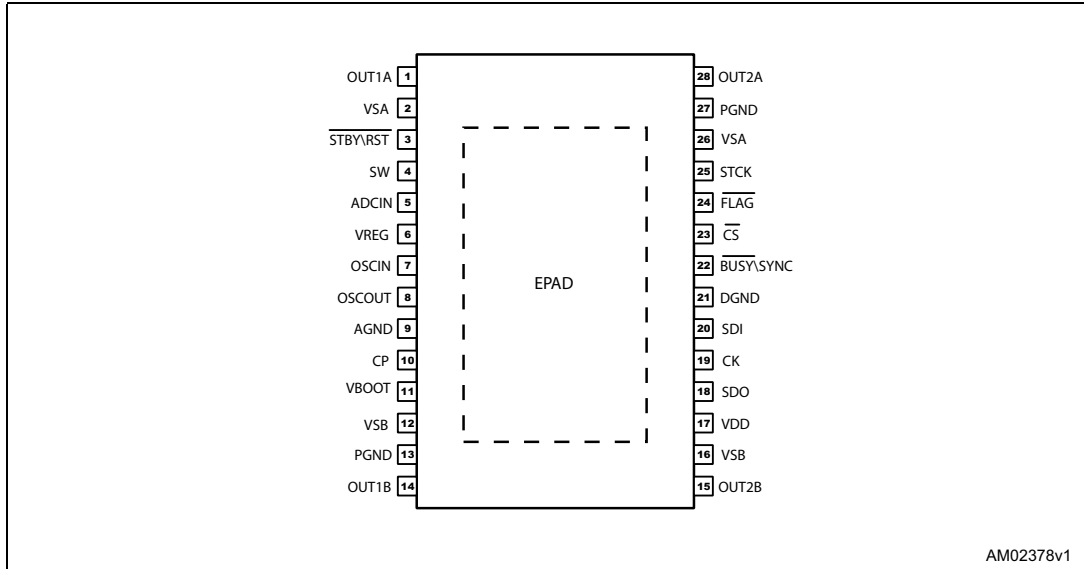
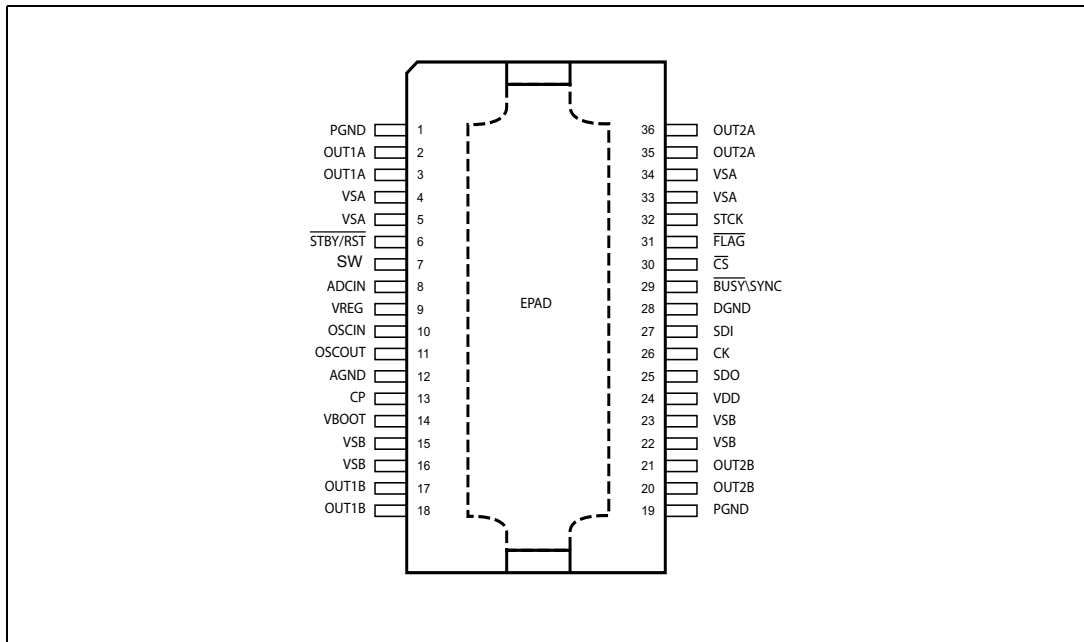


Figure 3. POWERSO36 pin connection (top view)



Pin list

Table 6. Pin description

| No. | | Name | Type | Function |
|--------|----------------|--------------------------------------|-------------------|--|
| HTSSOP | POWERSO | | | |
| 17 | 24 | VDD | Power | Logic outputs supply voltage (pull-up reference) |
| 6 | 9 | VREG | Power | Internal 3 V voltage regulator output and 3.3 V external logic supply |
| 7 | 10 | OSCIN | Analog input | Oscillator pin 1. To connect an external oscillator or clock source. If this pin is unused, it should be left floating. |
| 8 | 11 | OSCOU | Analog output | Oscillator pin 2. To connect an external oscillator. When the internal oscillator is used this pin can supply 2/4/8/16 MHz. If this pin is unused, it should be left floating. |
| 10 | 13 | CP | Output | Charge pump oscillator output |
| 11 | 14 | VBOOT | Supply voltage | Bootstrap voltage needed for driving the high-side power DMOS of both bridges (A and B) |
| 5 | 8 | ADCIN | Analog input | Internal analog-to-digital converter input |
| 2, 26 | 4, 5, 33, 34 | VSA | Power supply | Full bridge A power supply pin. It must be connected to VSB. |
| 12, 16 | 15, 16, 22, 23 | VSB | Power supply | Full bridge B power supply pin. It must be connected to VSA. |
| 27, 13 | 1, 19 | PGND | Ground | Power ground pin |
| 1 | 2, 3 | OUT1A | Power output | Full bridge A output 1 |
| 28 | 35, 36 | OUT2A | Power output | Full bridge A output 2 |
| 14 | 17, 18 | OUT1B | Power output | Full bridge B output 1 |
| 15 | 20, 21 | OUT2B | Power output | Full bridge B output 2 |
| 9 | 12 | AGND | Ground | Analog ground. |
| 4 | 7 | SW | Logical input | External switch input pin. If not used the pin should be connected to VDD. |
| 21 | 28 | DGND | Ground | Digital ground |
| 22 | 29 | $\overline{\text{BUSY}}/\text{SYNC}$ | Open drain output | By default, this BUSY pin is forced low when the device is performing a command. Otherwise the pin can be configured to generate a synchronization signal. |
| 18 | 25 | SDO | Logic output | Data output pin for serial interface |
| 20 | 27 | SDI | Logic input | Data input pin for serial interface |
| 19 | 26 | CK | Logic input | Serial interface clock |
| 23 | 30 | $\overline{\text{CS}}$ | Logic input | Chip select input pin for serial interface |

Table 6. Pin description (continued)

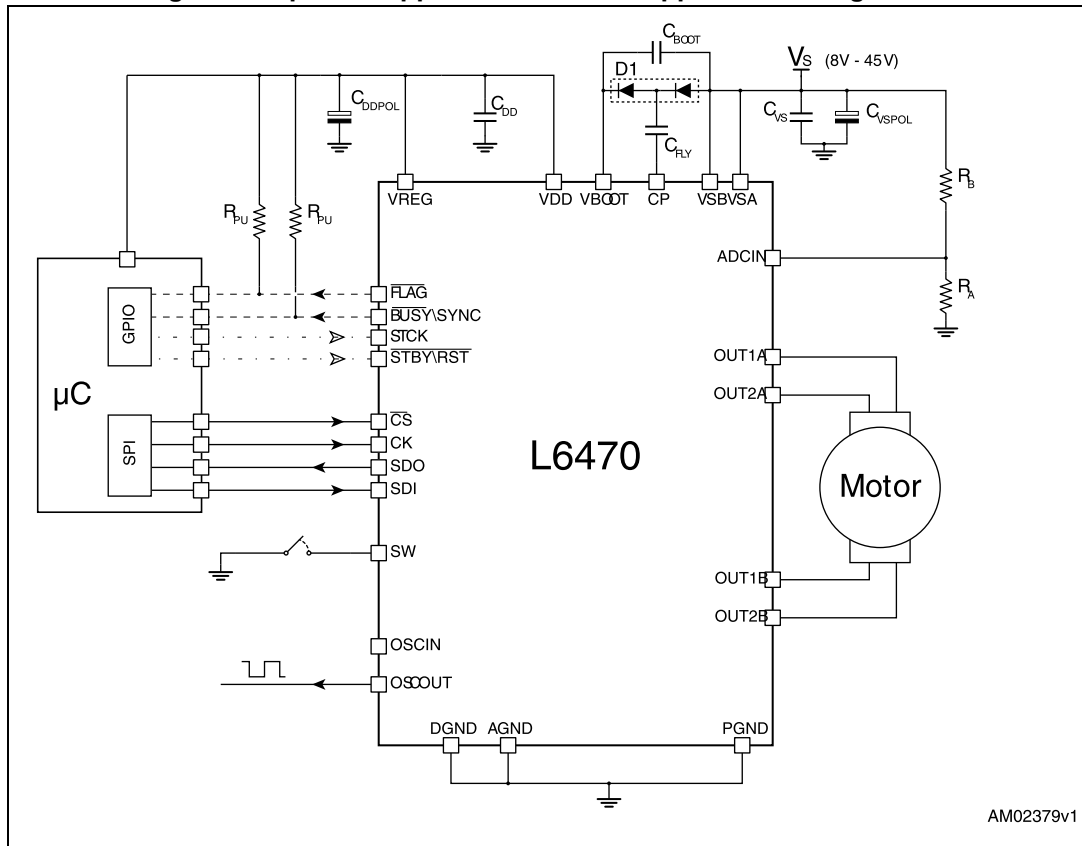
| No. | | Name | Type | Function |
|--------|---------|-----------------------------|-------------------|---|
| HTSSOP | POWERSO | | | |
| 24 | 31 | $\overline{\text{FLAG}}$ | Open drain output | Status flag pin. An internal open drain transistor can pull the pin to GND when a programmed alarm condition occurs (step loss, OCD, thermal pre-warning or shutdown, UVLO, wrong command, non-performable command) |
| 3 | 6 | $\overline{\text{STBYRST}}$ | Logic input | Standby and reset pin. LOW logic level resets the logic and puts the device into Standby mode. If not used, it should be connected to VDD. |
| 25 | 32 | STCK | Logic input | Step-clock input |
| EPAD | EPAD | Exposed pad | Ground | Internally connected to PGND, AGND and DGND pins |

5 Typical applications

Table 7. Typical application values

| Name | Value |
|--------------|--------------------------------|
| C_{VS} | 220 nF |
| C_{VSPOL} | 100 μ F |
| C_{REG} | 100 nF |
| C_{REGPOL} | 47 μ F |
| C_{DD} | 100 nF |
| C_{DDPOL} | 10 μ F |
| D1 | Charge pump diodes |
| C_{BOOT} | 220 nF |
| C_{FLY} | 10 nF |
| R_{PU} | 39 k Ω |
| R_{SW} | 100 Ω |
| C_{SW} | 10 nF |
| R_A | 2.7 k Ω ($V_S = 36$ V) |
| R_B | 62 k Ω ($V_S = 36$ V) |

Figure 4. Bipolar stepper motor control application using L6470



AM02379v1

6 Functional description

6.1 Device power-up

At power-up end, the device state is the following:

- Registers are set to default
- Internal logic is driven by internal oscillator and a 2 MHz clock is provided by the OSCOUT pin
- Bridges are disabled (High Z)
- UVLO bit in the STATUS register is forced low (fail condition)
- FLAG output is forced low.

During power-up, the device is under reset (all logic IOs disabled and power bridges in high impedance state) until the following conditions are satisfied:

- V_S is greater than V_{SthOn}
- V_{REG} is greater than $V_{REGth} = 2.8$ V typical
- Internal oscillator is operative.

Any motion command makes the device exit from High Z state (HardStop and SoftStop included).

6.2 Logic I/O

Pins \overline{CS} , CK, SDI, STCK, SW and $\overline{STBY}\overline{RST}$ are TTL/CMOS 3.3 V - 5 V compatible logic inputs.

Pin SDO is a TTL/CMOS compatible logic output. VDD pin voltage sets the logic output pin voltage range; when it is connected to VREG or 3.3 V external supply voltage, the output is 3.3 V compatible. When VDD is connected to a 5 V supply voltage, SDO is 5 V compatible.

VDD is not internally connected to V_{REG} , an external connection is always needed.

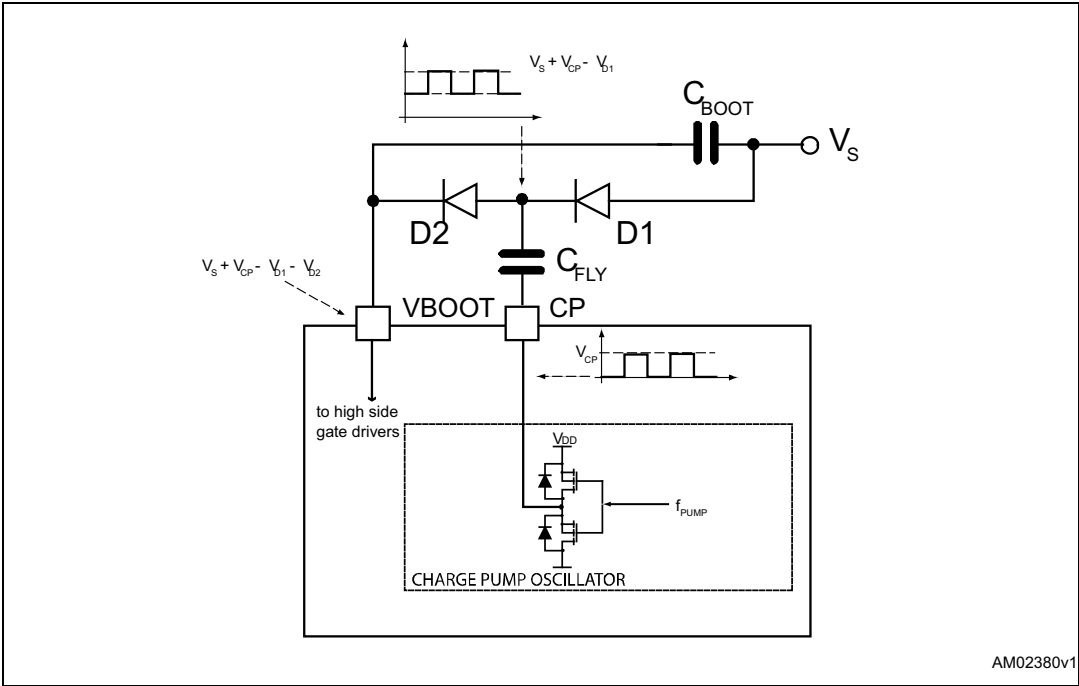
A 10 μ F capacitor should be connected to the VDD pin in order to obtain a proper operation.

Pins \overline{FLAG} and $\overline{BUSY}\overline{SYNC}$ are open drain outputs.

6.3 Charge pump

To ensure the correct driving of the high-side integrated MOSFETs, a voltage higher than the motor power supply voltage needs to be applied to the VBOOT pin. The high-side gate driver supply voltage, V_{boot} , is obtained through an oscillator and a few external components realizing a charge pump (see [Figure 5](#)).

Figure 5. Charge pump circuitry

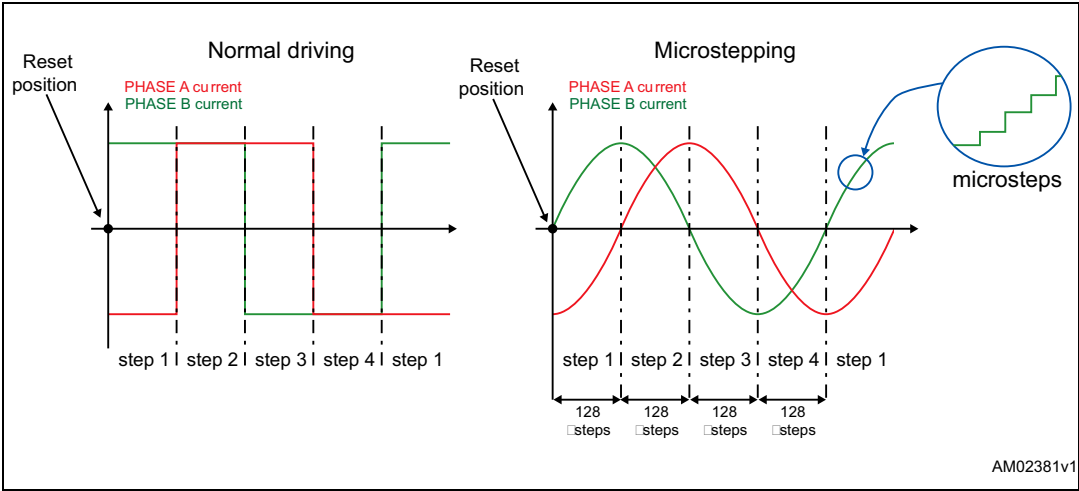


6.4 Microstepping

The driver is able to divide the single step into up to 128 microsteps. Stepping mode can be programmed by the STEP_SEL parameter in the STEP_MODE register (see Table 18 on page 48).

Step mode can only be changed when bridges are disabled. Every time the step mode is changed the electrical position (i.e. the point of microstepping sinewave that is generated) is reset to zero and the absolute position counter value (see Section 6.5) becomes meaningless.

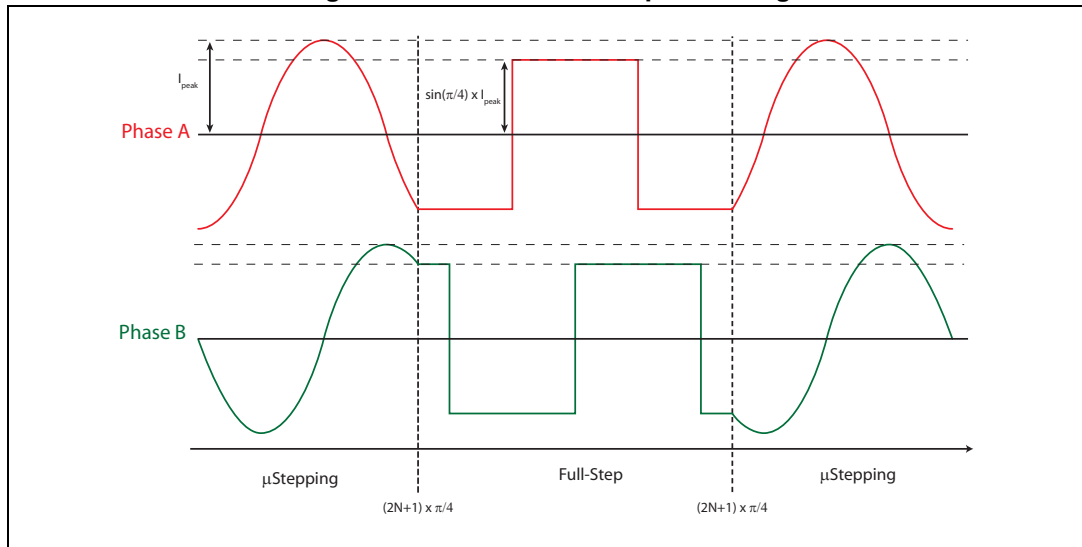
Figure 6. Normal mode and microstepping (128 microsteps)



Automatic full-step mode

When motor speed is greater than a programmable full-step speed threshold, the L6470 device switches automatically to Full-step mode (see [Figure 7](#)); the driving mode returns to microstepping when motor speed decreases below the full-step speed threshold. The full-step speed threshold is set through the FS_SPD register (see [Section 9.1.9 on page 44](#)).

Figure 7. Automatic full-step switching



6.5 Absolute position counter

An internal 22-bit register (ABS_POS) records the motor motion according to the selected step mode; the stored value unit is equal to the selected step mode (full, half, quarter, etc.). The position range is from -2^{21} to $+2^{21}-1$ (μ steps (see [Section 9.1.1 on page 41](#))).

6.6 Programmable speed profiles

The user can easily program a customized speed profile defining independently acceleration, deceleration, maximum and minimum speed values by the ACC, DEC, MAX_SPEED and MIN_SPEED registers respectively (see [Section 9.1.5 on page 42](#), [Section 9.1.6 on page 43](#), [Section 9.1.7 on page 43](#) and [Section 9.1.8 on page 43](#)).

When a command is sent to the device, the integrated logic generates the microstep frequency profile that performs a motor motion compliant to speed profile boundaries.

All acceleration parameters are expressed in $\text{step}/\text{tick}^2$ and all speed parameters are expressed in step/tick ; the unit of measurement does not depend on the selected step mode. Acceleration and deceleration parameters range from 2^{-40} to $(2^{12}-2) \cdot 2^{-40}$ $\text{step}/\text{tick}^2$ (equivalent to 14.55 to 59590 step/s^2).

The minimum speed parameter ranges from 0 to $(2^{12}-1) \cdot 2^{-24}$ step/tick (equivalent to 0 to 976.3 step/s).

The maximum speed parameter ranges from 2^{-18} to $(2^{10}-1) \cdot 2^{-18}$ step/tick (equivalent to 15.25 to 15610 step/s).

6.7 Motor control commands

The L6470 device can accept different types of commands:

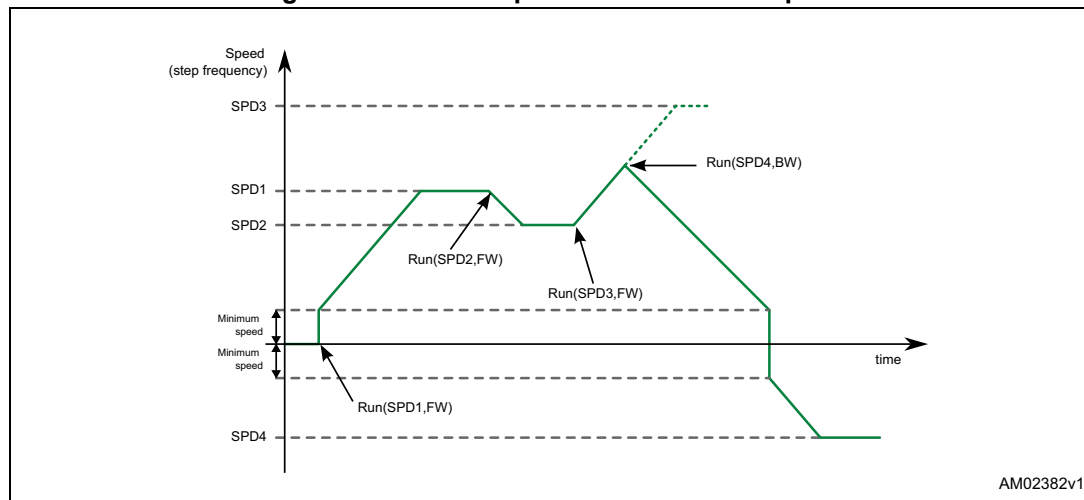
- constant speed commands (Run, GoUntil, ReleaseSW)
- absolute positioning commands (GoTo, GoTo_DIR, GoHome, GoMark)
- motion commands (Move)
- stop commands (SoftStop, HardStop, SoftHiz, HardHiz).

For detailed command descriptions refer to [Section 9.2 on page 56](#).

6.7.1 Constant speed commands

A constant speed command produces a motion in order to reach and maintain a user-defined target speed starting from the programmed minimum speed (set in the MIN_SPEED register) and with the programmed acceleration/deceleration value (set in the ACC and DEC registers). A new constant speed command can be requested anytime.

Figure 8. Constant speed command examples



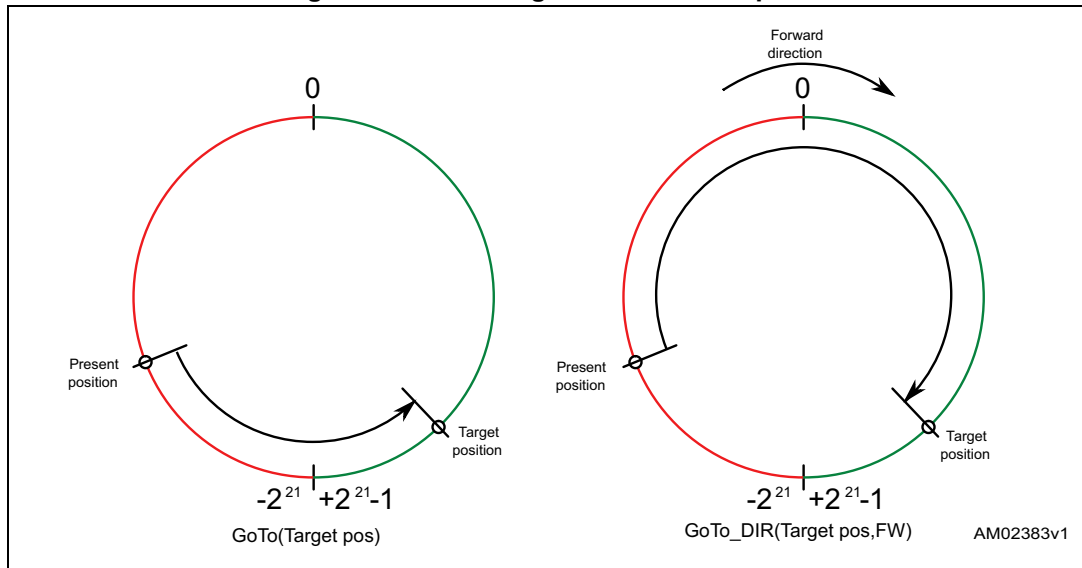
6.7.2 Positioning commands

An absolute positioning command produces a motion in order to reach a user-defined position that is sent to the device together with the command. The position can be reached performing the minimum path (minimum physical distance) or forcing a direction (see [Figure 9](#)).

The performed motor motion is compliant to programmed speed profile boundaries (acceleration, deceleration, minimum and maximum speed).

Note that with some speed profiles or positioning commands, the deceleration phase can start before the maximum speed is reached.

Figure 9. Positioning command examples



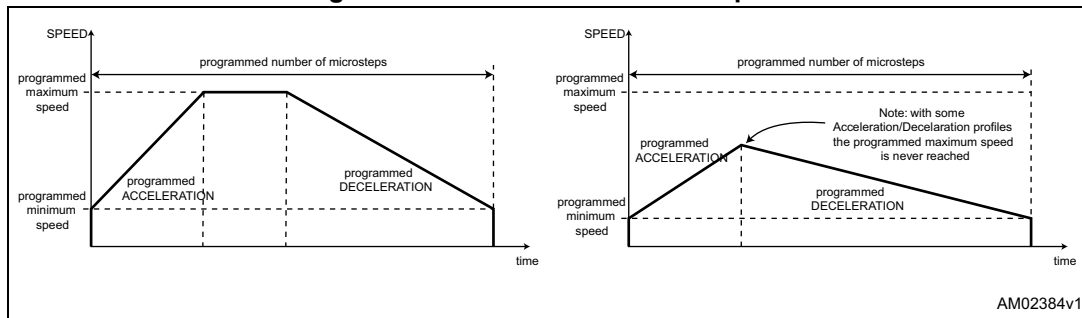
6.7.3 Motion commands

Motion commands produce a motion in order to perform a user-defined number of microsteps in a user-defined direction that are sent to the device together with the command (see [Figure 10](#)).

The performed motor motion is compliant to programmed speed profile boundaries (acceleration, deceleration, minimum and maximum speed).

Note that with some speed profiles or motion commands, the deceleration phase can start before the maximum speed is reached.

Figure 10. Motion command examples



6.7.4 Stop commands

A stop command forces the motor to stop. Stop commands can be sent anytime.

The SoftStop command causes the motor to decelerate with programmed deceleration value until the MIN_SPEED value is reached and then stops the motor keeping the rotor position (a holding torque is applied).

The HardStop command stops the motor instantly, ignoring deceleration constraints and keeping the rotor position (a holding torque is applied).

The SoftHiZ command causes the motor to decelerate with programmed deceleration value until the MIN_SPEED value is reached and then forces the bridges in high impedance state (no holding torque is present).

The HardHiZ command instantly forces the bridges into high impedance state (no holding torque is present).

6.7.5 Step-clock mode

In Step-clock mode the motor motion is defined by the step-clock signal applied to the STCK pin. At each step-clock rising edge, the motor is moved one microstep in the programmed direction and the absolute position is consequently updated.

When the system is in Step-clock mode, the SCK_MOD flag in the STATUS register is raised, the SPEED register is set to zero and motor status is considered stopped whatever the STCK signal frequency (MOT_STATUS parameter in STATUS register equal to "00").

6.7.6 GoUntil and ReleaseSW commands

In most applications the power-up position of the stepper motor is undefined, so an initialization algorithm driving the motor to a known position is necessary.

The GoUntil and ReleaseSW commands can be used in combination with external switch input (see [Section 6.13 on page 31](#)) to easily initialize the motor position.

The GoUntil command makes the motor run at the constant target speed until the SW input is forced low (falling edge). When this event occurs, one of the following actions can be performed:

- The ABS_POS register is set to zero (home position) and the motor decelerates to zero speed (as a SoftStop command)
- The ABS_POS register value is stored in the MARK register and the motor decelerates to zero speed (as a SoftStop command).

If the SW_MODE bit of the CONFIG register is set to '0', the motor does not decelerate but it immediately stops (as a HardStop command).

The ReleaseSW command makes the motor run at the programmed minimum speed until the SW input is forced high (rising edge). When this event occurs, one of the following actions can be performed:

- The ABS_POS register is set to zero (home position) and the motor immediately stops (as a HardStop command)
- The ABS_POS register value is stored in the MARK register and the motor immediately stops (as a HardStop command).

If the programmed minimum speed is less than 5 step/s, the motor is driven at 5 step/s.

6.8 Internal oscillator and oscillator driver

The control logic clock can be supplied by the internal 16-MHz oscillator, an external oscillator (crystal or ceramic resonator) or a direct clock signal.

These working modes can be selected by EXT_CLK and OSC_SEL parameters in the CONFIG register (see [Table 23 on page 50](#)).

At power-up the device starts using the internal oscillator and provides a 2-MHz clock signal on the OSCOUT pin.

Attention: In any case, before changing clock source configuration, a hardware reset is mandatory. Switching to different clock configurations during operation may cause unexpected behavior.

6.8.1 Internal oscillator

In this mode the internal oscillator is activated and OSCIN is unused. If the OSCOUT clock source is enabled, the OSCOUT pin provides a 2, 4, 8 or 16-MHz clock signal (according to OSC_SEL value); it is otherwise unused (see [Figure 11](#)).

6.8.2 External clock source

Two types of external clock source can be selected: crystal/ceramic resonator or direct clock source. Four programmable clock frequencies are available for each external clock source: 8, 16, 24 and 32 MHz.

When an external crystal/resonator is selected, the OSCIN and OSCOUT pins are used to drive the crystal/resonator (see [Figure 11](#)). The crystal/resonator and load capacitors (C_L) must be placed as close as possible to the pins. Refer to [Table 8](#) for the choice of load capacitor values according to the external oscillator frequency.

Table 8. CL values according to external oscillator frequency

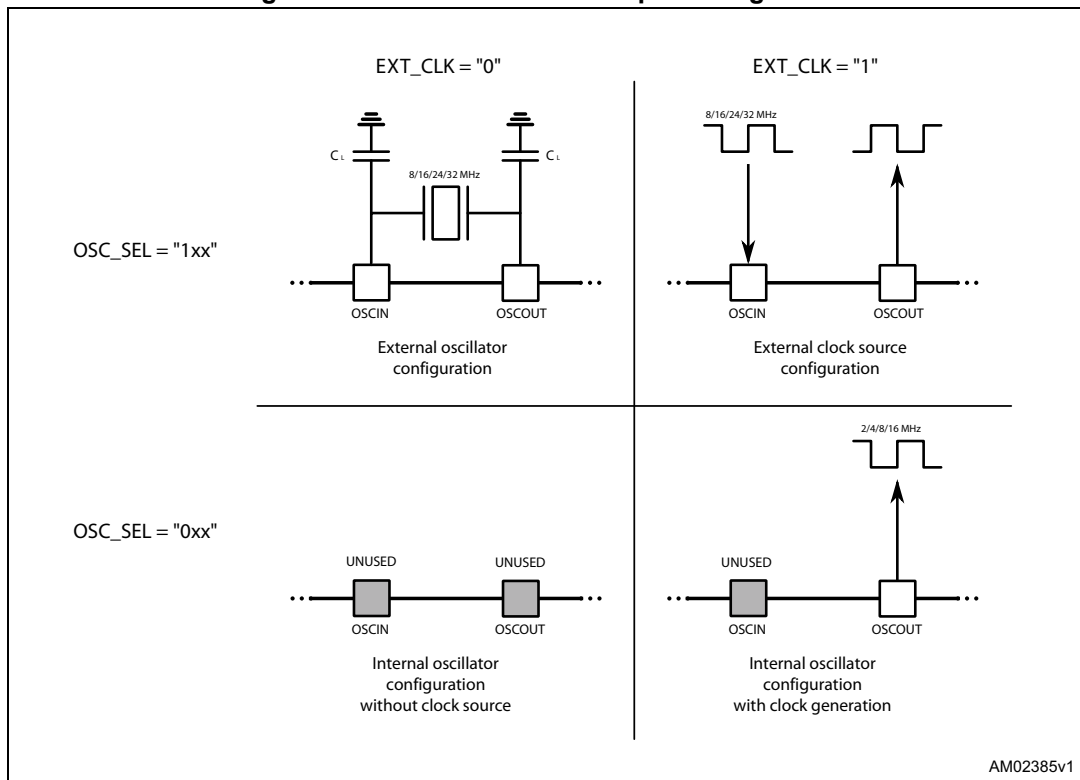
| Crystal/resonator freq. ⁽¹⁾ | C_L ⁽²⁾ |
|--|-----------------------------------|
| 8 MHz | 25 pF ($ESR_{max} = 80 \Omega$) |
| 16 MHz | 18 pF ($ESR_{max} = 50 \Omega$) |
| 24 MHz | 15 pF ($ESR_{max} = 40 \Omega$) |
| 32 MHz | 10 pF ($ESR_{max} = 40 \Omega$) |

1. First harmonic resonance frequency.

2. Lower ESR value allows the driving of greater load capacitors.

If a direct clock source is used, it must be connected to the OSCIN pin and the OSCOUT pin supplies the inverted OSCIN signal (see [Figure 11](#)).

Figure 11. OSCIN and OSCOUT pin configuration



AM02385v1

Note: When OSCIN is UNUSED, it should be left floating.
When OSCOUT is UNUSED, it should be left floating.

6.9 Overcurrent detection

When the current in any of the Power MOSFETs exceeds a programmed overcurrent threshold, the STATUS register OCD flag is forced low until the overcurrent event has expired and a GetStatus command is sent to the IC (see [Section 9.1.22 on page 55](#) and [Section 9.1.17 on page 47](#)). The overcurrent event expires when all the Power MOSFET currents fall below the programmed overcurrent threshold.

The overcurrent threshold can be programmed through the OCD_TH register in one of 16 available values ranging from 375 mA to 6 A with steps of 375 mA (see [Table 9 on page 40](#), [Section 9.1.17](#)).

It is possible to set whether an overcurrent event causes or not the MOSFET turn-off (bridges in high impedance status) acting on the OC_SD bit in the CONFIG register (see [Section 9.1.21](#)). The OCD flag in the STATUS register is raised anyway (see [Table 34 on page 55](#), [Section 9.1.22](#)).

When the IC outputs are turned off by an OCD event, they cannot be turned on until the OCD flag is released by a GetStatus command.

Attention: The overcurrent shutdown is a critical protection feature. It is not recommended to disable it.

6.10 Undervoltage lockout (UVLO)

The L6470 device provides a motor supply UVLO protection. When the motor supply voltage falls below the V_{StHOff} threshold voltage, the STATUS register UVLO flag is forced low. When a GetStatus command is sent to the IC, and the undervoltage condition has expired, the UVLO flag is released (see [Section 9.1.22 on page 55](#) and [9.2.20 on page 66](#)). The undervoltage condition expires when the motor supply voltage goes over the V_{StHOn} threshold voltage. When the device is in undervoltage condition, no motion command can be performed. The UVLO flag is forced low by logic reset (power-up included) even if no UVLO condition is present.

6.11 Thermal warning and thermal shutdown

An internal sensor allows the L6470 to detect when the device internal temperature exceeds a thermal warning or an overtemperature threshold.

When the thermal warning threshold ($T_{j(WRN)}$) is reached, the TH_WRN bit in the STATUS register is forced low (see [Section 9.1.22](#)) until the temperature decreases below $T_{j(WRN)}$ and a GetStatus command is sent to the IC (see [Section 9.1.22](#) and [9.2.20](#)).

When the thermal shutdown threshold ($T_{j(OFF)}$) is reached, the device goes into thermal shutdown condition: the TH_SD bit in the STATUS register is forced low, the power bridges are disabled bridges in high impedance state and the HiZ bit in the STATUS register is raised (see [Section 9.1.22](#)).

The thermal shutdown condition only expires when the temperature goes below the thermal warning threshold ($T_{j(WRN)}$).

On exiting thermal shutdown condition, the bridges are still disabled (HiZ flag high); any motion command makes the device exit from High Z state (HardStop and SoftStop included).

6.12 Reset and standby

The device can be reset and put into Standby mode through a dedicated pin. When the $\overline{STBY\overline{RST}}$ pin is driven low, the bridges are left open (High Z state), the internal charge pump is stopped, the SPI interface and control logic are disabled and the internal 3 V voltage regulator maximum output current is reduced to $I_{REG,STBY}$; as a result, the L6470 heavily reduces the power consumption. At the same time the register values are reset to default and all protection functions are disabled. $\overline{STBY\overline{RST}}$ input must be forced low at least for $t_{STBY,min}$ in order to ensure the complete switch to Standby mode.

On exiting Standby mode, as well as for IC power-up, a delay of up to $t_{logicwu}$ must be given before applying a new command to allow proper oscillator and logic startup and a delay of up to t_{cpwu} must be given to allow the charge pump startup.

On exiting Standby mode, the bridges are disabled (HiZ flag high) and any motion command makes the device exit High Z state (HardStop and SoftStop included).

Attention: It is not recommended to reset the device when outputs are active. The device should be switched to high impedance state before being reset.

6.13 External switch (SW pin)

The SW input is internally pulled up to V_{DD} and detects if the pin is open or connected to ground (see [Figure 12](#)).

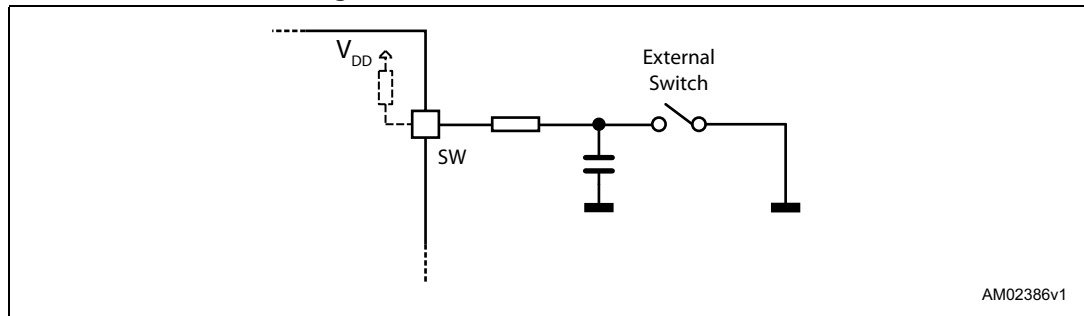
The SW_F bit of the STATUS register indicates if the switch is open ('0') or closed ('1') (see [Section 9.1.22 on page 55](#)); the bit value is refreshed at every system clock cycle (125 ns). The SW_EVN flag of the STATUS register is raised when a switch turn-on event (SW input falling edge) is detected (see [Section 9.1.22](#)). A GetStatus command releases the SW_EVN flag (see [Section 9.2.20 on page 66](#)).

By default, a switch turn-on event causes a HardStop interrupt (SW_MODE bit of CONFIG register set to '0'). Otherwise (SW_MODE bit of CONFIG register set to '1'), switch input events do not cause interrupts and the switch status information is at the user's disposal (see [Table 34 on page 55](#), [Section 9.1.22](#)).

The switch input may be used by the GoUntil and ReleaseSW commands as described in [Section 9.2.10 on page 62](#) and [9.2.11 on page 63](#).

If the SW input is not used, it should be connected to VDD.

Figure 12. External switch connection



6.14 Programmable DMOS slew rate, deadtime and blanking time

Using the POW_SR parameter in the CONFIG register, it is possible to set the commutation speed of the power bridges output (see [Table 26 on page 51](#), [Section 9.1.21 on page 49](#)).

6.15 Integrated analog-to-digital converter

The L6470 device integrates an N_{ADC} bit ramp-compare analog-to-digital converter with a reference voltage equal to V_{REG} . The analog-to-digital converter input is available through the ADCIN pin and the conversion result is available in the ADC_OUT register (see [Section 9.1.16 on page 46](#)).

Sampling frequency is equal to the programmed PWM frequency.

The ADC_OUT value can be used for motor supply voltage compensation or can be at the user's disposal.

6.16 Internal voltage regulator

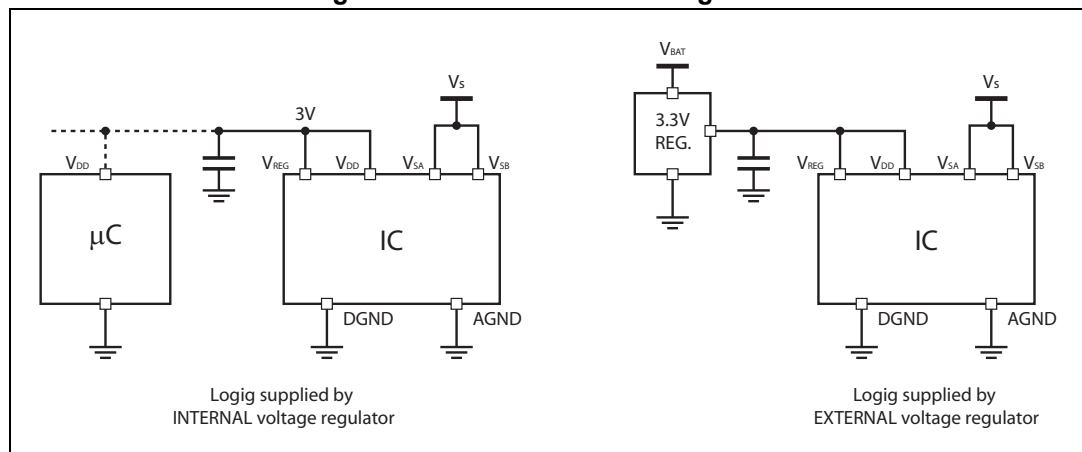
The L6470 integrates a voltage regulator which generates a 3 V voltage starting from the motor power supply (V_{SA} and V_{SB}). In order to make the voltage regulator stable, at least 22 μF should be connected between the V_{REG} pin and ground (suggested value is 47 μF).

The internal voltage regulator can be used to supply the V_{DD} pin in order to make the device digital output range 3.3 V compatible ([Figure 13](#)). A digital output range, 5 V compatible, may be obtained connecting the V_{DD} pin to an external 5 V voltage source. In both cases, a 10 μF capacitance should be connected to the V_{DD} pin in order to obtain a correct operation.

The internal voltage regulator is able to supply a current up to $I_{REG,MAX}$, internal logic consumption included (I_{logic}). When the device is in Standby mode, the maximum current that can be supplied is $I_{REG,STBY}$, internal consumption included ($I_{logic,STBY}$).

If an external 3.3 V regulated voltage is available, it can be applied to the V_{REG} pin in order to supply all the internal logic and to avoid power dissipation of the internal 3 V voltage regulator ([Figure 13](#)). The external voltage regulator should never sink current from the V_{REG} pin.

Figure 13. Internal 3 V linear regulator



6.17 BUSY\SYNC pin

This pin is an open drain output which can be used as the busy flag or synchronization signal according to the SYNC_EN bit value (STEP_MODE register).

6.17.1 BUSY operation mode

The pin works as busy signal when the SYNC_EN bit is set low (default condition). In this mode the output is forced low while a constant speed, absolute positioning or motion command is under execution. The $\overline{\text{BUSY}}$ pin is released when the command has been executed (target speed or target position reached). The STATUS register includes a BUSY flag that is the BUSY pin mirror (see [Section 9.1.22 on page 55](#)).

In the case of daisy chain configuration, BUSY pins of different ICs can be hard-wired to save host controller GPIOs.

6.17.2 SYNC operation mode

The pin works as synchronization signal when the SYNC_EN bit is set high. In this mode a step-clock signal is provided on the output according to a SYNC_SEL and STEP_SEL parameter combination (see [Section 9.1.19 on page 47](#)).

6.18 FLAG pin

By default, an internal open drain transistor pulls the $\overline{\text{FLAG}}$ pin to ground when at least one of the following conditions occurs:

- Power-up or standby/reset exit
- Stall detection on A bridge
- Stall detection on B bridge
- Overcurrent detection
- Thermal warning
- Thermal shutdown
- UVLO
- Switch turn-on event
- Wrong command
- Non-performable command.

It is possible to mask one or more alarm conditions by programming the ALARM_EN register (see [Section 9.1.20 on page 49](#), [Table 21 on page 49](#)). If the corresponding bit of the ALARM_EN register is low, the alarm condition is masked and it does not cause a FLAG pin transition; all other actions imposed by alarm conditions are performed anyway. In the case of daisy chain configuration, FLAG pins of different ICs can be or-wired to save host controller GPIOs.

7 Phase current control

The L6470 controls the phase current applying a sinusoidal voltage to motor windings. Phase current amplitude is not directly controlled but depends on phase voltage amplitude, load torque, motor electrical characteristics and rotation speed. Sinewave amplitude is proportional to the motor supply voltage multiplied by a coefficient (K_{VAL}). K_{VAL} ranges from 0 to 100% and the sinewave amplitude can be obtained through the following formula:

Equation 1

$$V_{OUT} = V_S \cdot K_{VAL}$$

Different K_{VAL} values can be programmed for acceleration, deceleration and constant speed phases and when the motor is stopped (HOLD phase) through the $KVAL_ACC$, $KVAL_DEC$, $KVAL_RUN$ and $KVAL_HOLD$ registers (see [Section 9.1.10 on page 44](#)). $KVAL$ value is calculated according to the following formula:

Equation 2

$$K_{VAL} = [(K_{VAL_X} + BEMF_COMP) \times VSCOMP \times K_THERM] \times \text{microstep}$$

where K_{VAL_X} is the starting K_{VAL} value programmed for present motion phase ($KVAL_ACC$, $KVAL_DEC$, $KVAL_RUN$ or $KVAL_HOLD$), $BEMF_COMP$ is the BEMF compensation curve value, $VSCOMP$ and K_THERM are the motor supply voltage and winding resistance compensation factors and microstep is the current microstep value (fraction of target peak current).

The L6470 device offers various methods to guarantee a stable current value, allowing the compensation of:

- low speed optimization ([Section 7.3](#))
- back electromotive force value ([Section 7.4](#))
- motor supply voltage variation ([Section 7.5](#))
- windings resistance variation ([Section 7.6](#)).

7.1 PWM sinewave generators

The two voltage sinewaves applied to the stepper motor phases are generated by two PWM modulators.

The PWM frequency (f_{PWM}) is proportional to the oscillator frequency (f_{OSC}) and can be obtained through the following formula:

Equation 3

$$f_{PWM} = \frac{f_{OSC}}{512 \cdot N} \cdot m$$

'N' is the integer division factor and 'm' is the multiplication factor. 'N' and 'm' values can be programmed by the F_PWM_INT and F_PWM_DEC parameters in the CONFIG register (see [Table 28 on page 52](#) and [Table 29 on page 52](#), [Section 9.1.21 on page 49](#)).

Available PWM frequencies are listed in [Section 9.1.21](#) from [Table 30 on page 53](#) to [Table 33 on page 54](#).

7.2 Sensorless stall detection

Depending on motor speed and load angle characteristics, the L6470 offers a motor stall condition detection using a programmable current comparator.

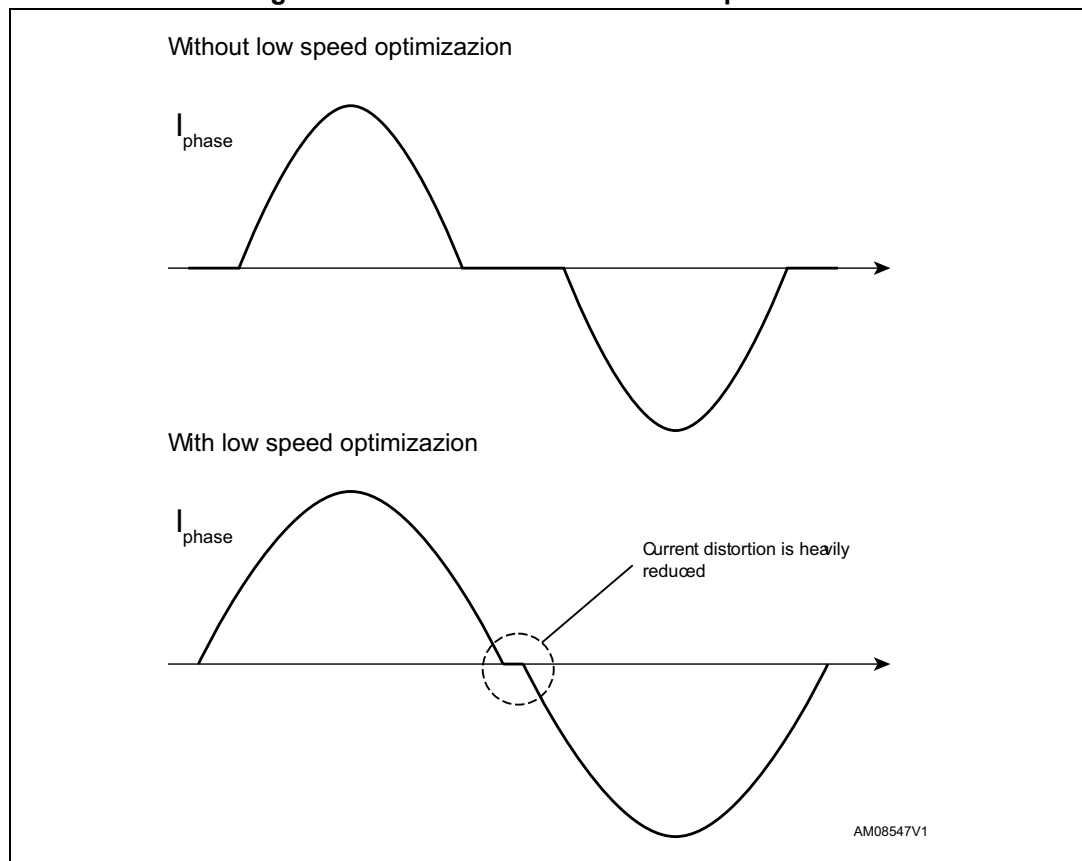
When a stall event occurs, the respective flag (STEP_LOSS_A or STEP_LOSS_B) is forced low until a GetStaus command or a system reset occurs (see [Section 9.2.20 on page 66](#)).

7.3 Low speed optimization

When the motor is driven at a very low speed using a small driving voltage, the resulting phase current can be distorted. As a consequence, the motor position is different from the ideal one (see [Figure 14](#)).

The L6470 device implements a low speed optimization in order to remove this effect.

Figure 14. Current distortion and compensation



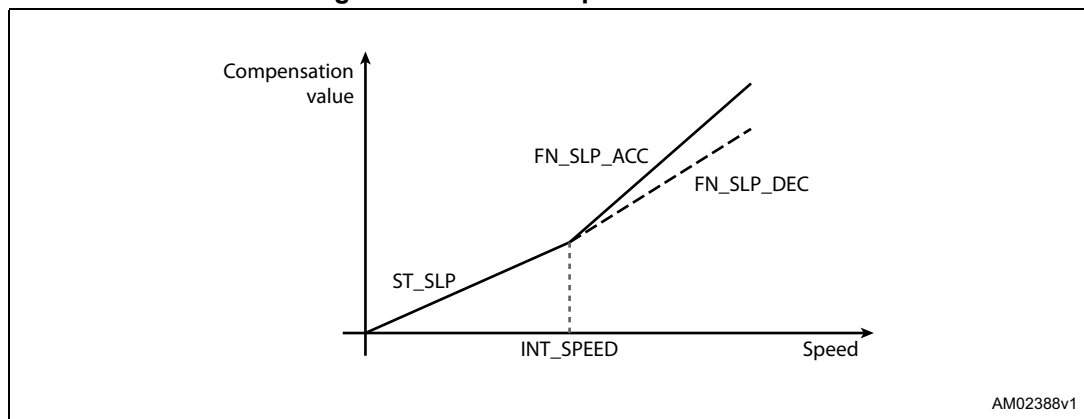
The optimization can be enabled setting high the LSPD_OPT bit in the MIN_SPEED register (see [Section 9.1.8 on page 43](#)) and is active in a speed range from zero to MIN_SPEED. When low speed optimization is enabled, speed profile minimum speed is forced to zero.

7.4 BEMF compensation

Using the speed information, a compensation curve is added to the amplitude of the voltage waveform applied to the motor winding in order to compensate the BEMF variations during acceleration and deceleration (see [Figure 15](#)).

The compensation curve is approximated by a stacked line with a starting slope (ST_SLP) when speed is lower than a programmable threshold speed (INT_SPEED) and a fine slope (FN_SLP_ACC and FN_SLP_DEC) when speed is greater than the threshold speed (see [Section 9.1.11 on page 45](#), [9.1.12 on page 45](#), [9.1.13 on page 45](#) and [Section 9.1.14 on page 45](#)).

Figure 15. BEMF compensation curve



To obtain different current values during acceleration and deceleration phases, two different final slope values, and consequently two different compensation curves, can be programmed.

The acceleration compensation curve is applied when the motor runs. No BEMF compensation is applied when the motor is stopped.

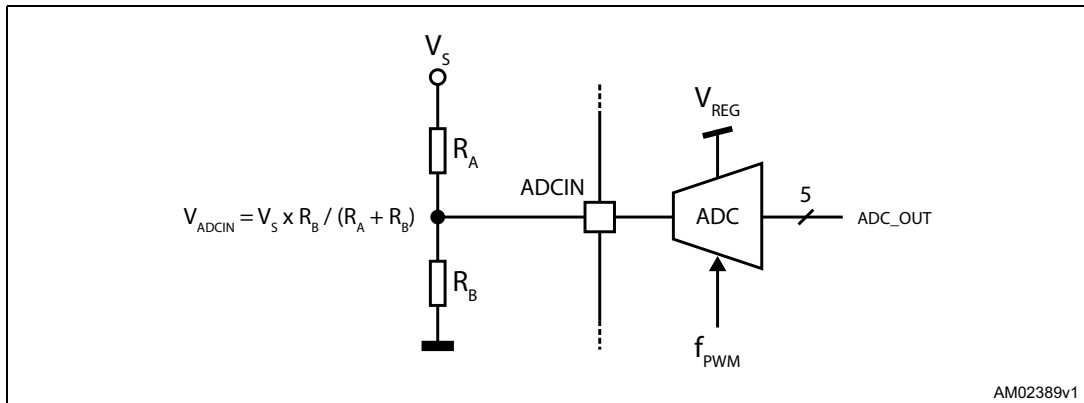
7.5 Motor supply voltage compensation

The sinewave amplitude generated by the PWM modulators is directly proportional to the motor supply voltage (V_S). When the motor supply voltage is different from its nominal value, the motor phases are driven with an incorrect voltage. The L6470 device can compensate motor supply voltage variations in order to avoid this effect.

The motor supply voltage should be connected to the integrated ADC input through a resistor divider in order to obtain $V_{REG}/2$ voltage at the ADCIN pin when V_S is at its nominal value (see [Figure 16](#)).

The ADC input is sampled at f_S frequency, which is equal to PWM frequency.

Figure 16. Motor supply voltage compensation circuit



Motor supply voltage compensation can be enabled setting high the EN_VSCOMP bit of the CONFIG register (see [Table 22 on page 49](#), [Section 9.1.21 on page 49](#)). If the EN_VSCOMP bit is low, the compensation is disabled and the internal analog-to-digital converter is at the user's disposal; sampling rate is always equal to PWM frequency.

7.6 Winding resistance thermal drift compensation

The higher the winding resistance, the greater the voltage to be applied in order to obtain the same phase current.

The L6470 integrates a register (K_THERM) which can be used to compensate phase resistance increment due to temperature rising.

The value in the K_THERM register (see [Section 9.1.15 on page 46](#)) multiplies the duty cycle value allowing a higher phase resistance value to be faced.

The compensation algorithm and the eventual motor temperature measurement should be implemented by microcontroller firmware.

8 Serial interface

The integrated 8-bit serial peripheral interface (SPI) is used for a synchronous serial communication between the host microprocessor (always master) and the L6470 (always slave).

The SPI uses chip select (\overline{CS}), serial clock (CK), serial data input (SDI) and serial data output (SDO) pins. When \overline{CS} is high, the device is unselected and the SDO line is inactive (high-impedance).

The communication starts when \overline{CS} is forced low. The CK line is used for synchronization of data communication.

All commands and data bytes are shifted into the device through the SDI input, most significant bit first. The SDI is sampled on the rising edges of the CK.

All output data bytes are shifted out of the device through the SDO output, most significant bit first. The SDO is latched on the falling edges of the CK. When a return value from the device is not available, an all zero byte is sent.

After each byte transmission the \overline{CS} input must be raised and be kept high for at least t_{disCS} in order to allow the device to decode the received command and put the return value into the SHIFT register.

All timing requirements are shown in [Figure 17](#) (see [Section 3: Electrical characteristics on page 11](#) for values).

Multiple devices can be connected in daisy chain configuration, as shown in [Figure 18](#).

Figure 17. SPI timings diagram

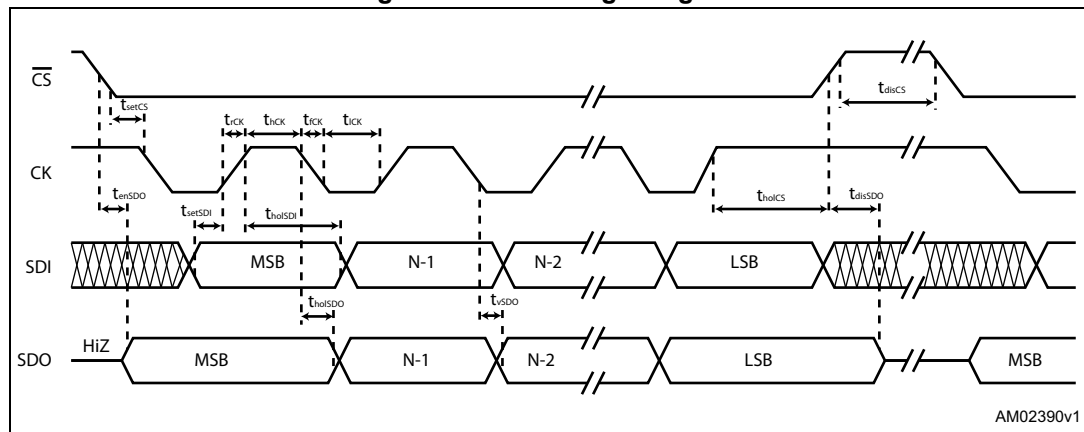
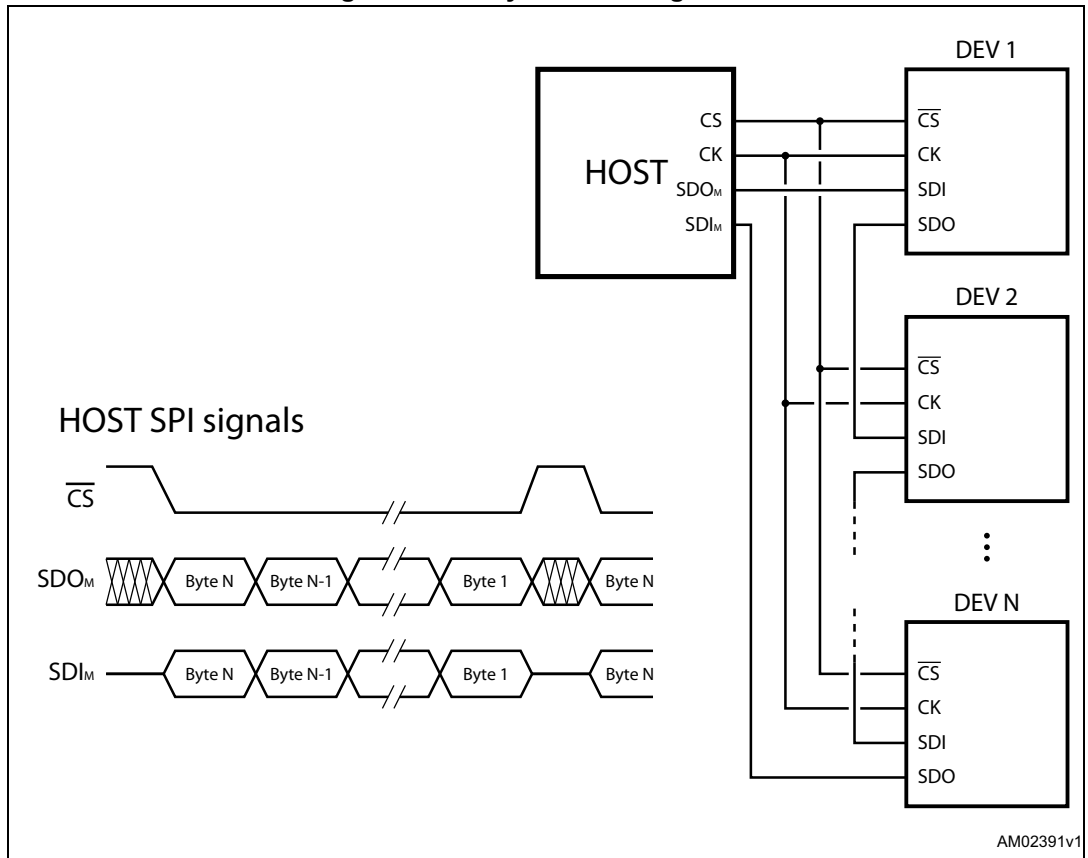


Figure 18. Daisy chain configuration



9 Programming manual

9.1 Registers and flags description

[Table 9](#) is a map of the user registers available (detailed description in respective paragraphs):

Table 9. Register map

| Address [Hex] | Register name | Register function | Len. [bit] | Reset Hex | Reset value | Remarks ⁽¹⁾ |
|---------------|---------------|--|------------|-------------------|--|------------------------|
| h01 | ABS_POS | Current position | 22 | 000000 | 0 | R, WS |
| h02 | EL_POS | Electrical position | 9 | 000 | 0 | R, WS |
| h03 | MARK | Mark position | 22 | 000000 | 0 | R, WR |
| h04 | SPEED | Current speed | 20 | 00000 | 0 step/tick (0 step/s) | R |
| h05 | ACC | Acceleration | 12 | 08A | 125.5e-12 step/tick ² (2008 step/s ²) | R, WS |
| h06 | DEC | Deceleration | 12 | 08A | 125.5e-12 step/tick ² (2008 step/s ²) | R, WS |
| h07 | MAX_SPEED | Maximum speed | 10 | 041 | 248e-6 step/tick (991.8 step/s) | R, WR |
| h08 | MIN_SPEED | Minimum speed | 13 | 000 | 0 step/tick (0 step/s) | R, WS |
| h15 | FS_SPD | Full-step speed | 10 | 027 | 150.7e-6 step/tick (602.7 step/s) | R, WR |
| h09 | KVAL_HOLD | Holding K _{VAL} | 8 | 29 | 0.16·VS | R, WR |
| h0A | KVAL_RUN | Constant speed K _{VAL} | 8 | 29 | 0.16·VS | R, WR |
| h0B | KVAL_ACC | Acceleration starting K _{VAL} | 8 | 29 | 0.16·VS | R, WR |
| h0C | KVAL_DEC | Deceleration starting K _{VAL} | 8 | 29 | 0.16·VS | R, WR |
| h0D | INT_SPEED | Intersect speed | 14 | 0408 | 15.4e-6 step/tick (61.5 step/s) | R, WH |
| h0E | ST_SLP | Start slope | 8 | 19 | 0.038% s/step | R, WH |
| h0F | FN_SLP_ACC | Acceleration final slope | 8 | 29 | 0.063% s/step | R, WH |
| h10 | FN_SLP_DEC | Deceleration final slope | 8 | 29 | 0.063% s/step | R, WH |
| h11 | K_THERM | Thermal compensation factor | 4 | 0 | 1.0 | R, WR |
| h12 | ADC_OUT | ADC output | 5 | XX ⁽²⁾ | | R |
| h13 | OCD_TH | OCD threshold | 4 | 8 | 3.38A | R, WR |
| h14 | STALL_TH | STALL threshold | 7 | 40 | 2.03A | R, WR |
| h16 | STEP_MODE | Step mode | 8 | 7 | 128 microsteps | R, WH |
| h17 | ALARM_EN | Alarm enable | 8 | FF | All alarms enabled | R, WS |

Table 9. Register map (continued)

| Address [Hex] | Register name | Register function | Len. [bit] | Reset Hex | Reset value | Remarks ⁽¹⁾ |
|---------------|---------------|-------------------|------------|---------------------|---|------------------------|
| h18 | CONFIG | IC configuration | 16 | 2E88 | Internal oscillator, 2 MHz OSCOUT clock, supply voltage compensation disabled, overcurrent shutdown enabled, slew rate = 290 V/μs PWM frequency = 15.6 kHz. | R, WH |
| h19 | STATUS | Status | 16 | XXXX ⁽²⁾ | High impedance state, UVLO/Reset flag set. | R |
| h1A | RESERVED | Reserved address | | | | |
| h1B | RESERVED | Reserved address | | | | |

1. R: readable, WH: writable only when outputs are in high impedance, WS: writable only when motor is stopped, WR: always writable.
2. According to startup conditions.

9.1.1 ABS_POS

The ABS_POS register contains the current motor absolute position in agreement with the selected step mode; the stored value unit is equal to the selected step mode (full, half, quarter, etc.). The value is in 2's complement format and it ranges from -2^{21} to $+2^{21}-1$.

At power-on the register is initialized to "0" (HOME position).

Any attempt to write the register when the motor is running causes the command to be ignored and the NOTPERF_CMD flag to rise (see [Section 9.1.22 on page 55](#)).

9.1.2 EL_POS

The EL_POS register contains the current electrical position of the motor. The two MSbits indicate the current step and the other bits indicate the current microstep (expressed in step/128) within the step.

Table 10. EL_POS register

| Bit 8 | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------|-------|-----------|-------|-------|-------|-------|-------|-------|
| STEP | | MICROSTEP | | | | | | |

When the EL_POS register is written by the user, the new electrical position is instantly imposed. When the EL_POS register is written, its value must be masked in order to match with the step mode selected in the STEP_MODE register in order to avoid a wrong microstep value generation (see [Section 9.1.19 on page 47](#)); otherwise the resulting microstep sequence is incorrect.

Any attempt to write the register when the motor is running causes the command to be ignored and the NOTPERF_CMD flag to rise (see [Section 9.1.22](#)).

9.1.3 MARK

The MARK register contains an absolute position called MARK according to the selected step mode; the stored value unit is equal to the selected step mode (full, half, quarter, etc.). It is in 2's complement format and it ranges from -2^{21} to $+2^{21}-1$.

9.1.4 SPEED

The SPEED register contains the current motor speed, expressed in step/tick (format unsigned fixed point 0.28).

In order to convert the SPEED value in step/s, the following formula can be used:

Equation 4

$$[\text{step/s}] = \frac{\text{SPEED} \cdot 2^{-28}}{\text{tick}}$$

where *SPEED* is the integer number stored in the register and tick is 250 ns.

The available range is from 0 to 15625 step/s with a resolution of 0.015 step/s.

Note: The range effectively available to the user is limited by the *MAX_SPEED* parameter.

Any attempt to write the register causes the command to be ignored and the NOTPERF_CMD flag to rise (see [Section 9.1.22 on page 55](#)).

9.1.5 ACC

The ACC register contains the speed profile acceleration expressed in step/tick² (format unsigned fixed point 0.40).

In order to convert ACC value in step/s², the following formula can be used:

Equation 5

$$[\text{step/s}^2] = \frac{\text{ACC} \cdot 2^{-40}}{\text{tick}^2}$$

where *ACC* is the integer number stored in the register and tick is 250 ns.

The available range is from 14.55 to 59590 step/s² with a resolution of 14.55 step/s².

The 0xFFFF value of the register is reserved and it should never be used.

Any attempt to write to the register when the motor is running causes the command to be ignored and the NOTPERF_CMD flag to rise (see [Section 9.1.22](#)).

9.1.6 DEC

The DEC register contains the speed profile deceleration expressed in step/tick² (format unsigned fixed point 0.40).

In order to convert DEC value in step/s², the following formula can be used:

Equation 6

$$[\text{step/s}^2] = \frac{\text{DEC} \cdot 2^{-40}}{\text{tick}^2}$$

where *DEC* is the integer number stored in the register and tick is 250 ns.

The available range is from 14.55 to 59590 step/s² with a resolution of 14.55 step/s².

Any attempt to write the register when the motor is running causes the command to be ignored and the NOTPERF_CMD flag to rise (see [Section 9.1.22 on page 55](#)).

9.1.7 MAX_SPEED

The MAX_SPEED register contains the speed profile maximum speed expressed in step/tick (format unsigned fixed point 0.18).

In order to convert it in step/s, the following formula can be used:

Equation 7

$$[\text{step/s}] = \frac{\text{MAX_SPEED} \cdot 2^{-18}}{\text{tick}}$$

where *MAX_SPEED* is the integer number stored in the register and tick is 250 ns.

The available range is from 15.25 to 15610 step/s with a resolution of 15.25 step/s.

9.1.8 MIN_SPEED

The MIN_SPEED register contains the following parameters:

Table 11. MIN_SPEED register

| Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|----------|-----------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| LSPD_OPT | MIN_SPEED | | | | | | | | | | | |

The MIN_SPEED parameter contains the speed profile minimum speed. Its value is expressed in step/tick and to convert it in step/s, the following formula can be used:

Equation 8

$$[\text{step/s}] = \frac{\text{MIN_SPEED} \cdot 2^{-24}}{\text{tick}}$$

where *MIN_SPEED* is the integer number stored in the register and tick is the ramp 250 ns.

The available range is from 0 to 976.3 step/s with a resolution of 0.238 step/s.

When the LSPD_OPT bit is set high, the low speed optimization feature is enabled and the MIN_SPEED value indicates the speed threshold below which the compensation works. In this case the minimum speed of the speed profile is set to zero.

An attempt to write the register when the motor is running causes the NOTPERF_CMD flag to rise.

9.1.9 FS_SPD

The FS_SPD register contains the threshold speed. When the actual speed exceeds this value, the step mode is automatically switched to full-step two-phase on. Its value is expressed in step/tick (format unsigned fixed point 0.18) and to convert it in step/s, the following formula can be used.

Equation 9

$$[\text{step/s}] = \frac{(\text{FS_SPD} + 0.5) \cdot 2^{-18}}{\text{tick}}$$

If the FS_SPD value is set to h3FF (max.) the system always works in microstepping mode (SPEED must go beyond the threshold to switch to Full-step mode). Setting FS_SPD to zero does not have the same effect as setting Step mode to full-step two-phase on: the zero FS_SPD value is equivalent to a speed threshold of about 7.63 step/s.

The available range is from 7.63 to 15625 step/s with a resolution of 15.25 step/s.

9.1.10 KVAL_HOLD, KVAL_RUN, KVAL_ACC and KVAL_DEC

The KVAL_HOLD register contains the K_{VAL} value that is assigned to the PWM modulators when the motor is stopped (compensation excluded).

The KVAL_RUN register contains the K_{VAL} value that is assigned to the PWM modulators when the motor is running at constant speed (compensation excluded).

The KVAL_ACC register contains the starting K_{VAL} value that can be assigned to the PWM modulators during acceleration (compensation excluded).

The KVAL_DEC register contains the starting K_{VAL} value that can be assigned to the PWM modulators during deceleration (compensation excluded).

The available range is from 0 to $0.996 \times V_S$ with a resolution of $0.004 \times V_S$, as shown in [Table 12](#).

Table 12. Voltage amplitude regulation registers

| KVAL_X [7 ... 0] | | | | | | | | Output voltage |
|------------------|---|---|---|---|---|---|---|------------------------|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | $V_S \times (1/256)$ |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | $V_S \times (254/256)$ |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | $V_S \times (255/256)$ |

9.1.11 INT_SPEED

The INT_SPEED register contains the speed value at which the BEMF compensation curve changes slope (see [Section 7.4 on page 36](#)). Its value is expressed in step/tick and to convert it in step/s, the following formula can be used:

Equation 10

$$[\text{step/s}] = \frac{\text{INT_SPEED} \cdot 2^{-26}}{\text{tick}}$$

where *INT_SPEED* is the integer number stored in the register and tick is 250 ns.

The available range is from 0 to 976.5 step/s with a resolution of 0.0596 step/s.

Any attempt to write the register when the motor is running causes the command to be ignored and the NOTPERF_CMD flag to rise (see [Section 9.1.22 on page 55](#)).

9.1.12 ST_SLP

The ST_SLP register contains the BEMF compensation curve slope that is used when the speed is lower than the intersect speed (see [Section 7.4](#)). Its value is expressed in s/step and the available range is from 0 to 0.004 with a resolution of 0.000015.

When ST_SLP, FN_SLP_ACC and FN_SLP_DEC parameters are set to zero, no BEMF compensation is performed.

Any attempt to write the register when the motor is running causes the command to be ignored and the NOTPERF_CMD flag to rise (see [Section 9.1.22](#)).

9.1.13 FN_SLP_ACC

The FN_SLP_ACC register contains the BEMF compensation curve slope that is used when the speed is greater than the intersect speed during acceleration (see [Section 7.4](#)). Its value is expressed in s/step and the available range is from 0 to 0.004 with a resolution of 0.000015.

When ST_SLP, FN_SLP_ACC and FN_SLP_DEC parameters are set to zero, no BEMF compensation is performed.

Any attempt to write the register when the motor is running causes the command to be ignored and the NOTPERF_CMD flag to rise (see [Section 9.1.22](#)).

9.1.14 FN_SLP_DEC

The FN_SLP_DEC register contains the BEMF compensation curve slope that is used when the speed is greater than the intersect speed during deceleration (see [Section 7.4](#)). Its value is expressed in s/step and the available range is from 0 to 0.004 with a resolution of 0.000015.

When ST_SLP, FN_SLP_ACC and FN_SLP_DEC parameters are set to zero, no BEMF compensation is performed.

Any attempt to write the register when the motor is running causes the command to be ignored and the NOTPERF_CMD flag to rise (see [Section 9.1.22](#)).

9.1.15 K_THERM

The K_THERM register contains the value used by the winding resistance thermal drift compensation system (see [Section 7.6 on page 37](#)).

The available range is from 1 to 1.46875 with a resolution of 0.03125, as shown in [Table 13](#).

Table 13. Winding resistance thermal drift compensation coefficient

| K_THERM [3 ... 0] | | | | Compensation coefficient |
|-------------------|---|---|---|--------------------------|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1.03125 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 1 | 1 | 1 | 0 | 1.4375 |
| 1 | 1 | 1 | 1 | 1.46875 |

9.1.16 ADC_OUT

The ADC_OUT register contains the result of the analog-to-digital conversion of the ADCIN pin voltage; the result is available even if the supply voltage compensation is disabled.

Any attempt to write to the register causes the command to be ignored and the NOTPERF_CMD flag to rise (see [Section 9.1.22 on page 55](#)).

Table 14. ADC_OUT value and motor supply voltage compensation feature

| V _S | V _{ADCIN} /V _{REG} | ADC_OUT [4 ... 0] | | | | | Compensation coefficient |
|---------------------------------------|--------------------------------------|-------------------|---|---|---|---|--------------------------|
| Greater than V _{S,nom} + 50% | > 24/32 | 1 | 1 | X | X | X | 0.65625 |
| V _{S,nom} + 50% | 24/32 | 1 | 1 | 0 | 0 | 0 | 0.65625 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| V _{S,nom} | 16/32 | 1 | 0 | 0 | 0 | 0 | 1 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| V _{S,nom} - 50% | 8/32 | 0 | 1 | 0 | 0 | 0 | 1.968875 |
| Lower than V _{S,nom} - 50% | < 8/32 | 0 | 0 | X | X | X | 1.968875 |

9.1.17 OCD_TH

The OCD_TH register contains the overcurrent threshold value (see [Section 6.9 on page 29](#)). The available range is from 375 mA to 6 A, in steps of 375 mA, as shown in [Table 15](#).

Table 15. Overcurrent detection threshold

| OCD_TH [3 ... 0] | | | | Overcurrent detection threshold |
|------------------|-----|-----|-----|---------------------------------|
| 0 | 0 | 0 | 0 | 375 mA |
| 0 | 0 | 0 | 1 | 750 mA |
| ... | ... | ... | ... | ... |
| 1 | 1 | 1 | 0 | 5.625 A |
| 1 | 1 | 1 | 1 | 6 A |

9.1.18 STALL_TH

The STALL_TH register contains the stall detection threshold value (see [Section 7.2 on page 35](#)). The available range is from 31.25 mA to 4 A with a resolution of 31.25 mA.

Table 16. Stall detection threshold

| STALL_th [6 ... 0] | | | | | | | Stall detection threshold |
|--------------------|-----|-----|-----|-----|-----|-----|---------------------------|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 31.25 mA |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 62.5 mA |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 3.969 A |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 4 A |

9.1.19 STEP_MODE

The STEP_MODE register has the following structure:

Table 17. STEP_MODE register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---------|----------|-------|-------|------------------|----------|-------|-------|
| SYNC_EN | SYNC_SEL | | | 0 ⁽¹⁾ | STEP_SEL | | |

1. When the register is written, this bit should be set to 0.

The STEP_SEL parameter selects one of eight possible stepping modes:

Table 18. Step mode selection

| STEP_SEL[2 0] | | | Step mode |
|--------------------|---|---|-----------------|
| 0 | 0 | 0 | Full-step |
| 0 | 0 | 1 | Half-step |
| 0 | 1 | 0 | 1/4 microstep |
| 0 | 1 | 1 | 1/8 microstep |
| 1 | 0 | 0 | 1/16 microstep |
| 1 | 0 | 1 | 1/32 microstep |
| 1 | 1 | 0 | 1/64 microstep |
| 1 | 1 | 1 | 1/128 microstep |

Every time the step mode is changed, the electrical position (i.e. the point of microstepping sinewave that is generated) is reset to the first microstep.

Warning: Every time STEP_SEL is changed, the value in the ABS_POS register loses meaning and should be reset.

Any attempt to write the register when the motor is running causes the command to be ignored and the NOTPERF_CMD flag to rise (see [Section 9.1.22 on page 55](#)).

When the SYNC_EN bit is set low, $\overline{\text{BUSY/SYNC}}$ output is forced low during command execution, otherwise, when the SYNC_EN bit is set high, $\overline{\text{BUSY/SYNC}}$ output provides a clock signal according to the SYNC_SEL parameter.

Table 19. SYNC output frequency

| | | STEP_SEL (f_{FS} is the full-step frequency) | | | | | | | |
|----------|-----|---|--------------|------------------|------------------|------------------|-------------------|-------------------|-------------------|
| | | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| SYNC_SEL | 000 | $f_{FS} / 2$ | $f_{FS} / 2$ | $f_{FS} / 2$ | $f_{FS} / 2$ | $f_{FS} / 2$ | $f_{FS} / 2$ | $f_{FS} / 2$ | $f_{FS} / 2$ |
| | 001 | NA | f_{FS} | f_{FS} | f_{FS} | f_{FS} | f_{FS} | f_{FS} | f_{FS} |
| | 010 | NA | NA | $2 \cdot f_{FS}$ | $2 \cdot f_{FS}$ | $2 \cdot f_{FS}$ | $2 \cdot f_{FS}$ | $2 \cdot f_{FS}$ | $2 \cdot f_{FS}$ |
| | 011 | NA | NA | NA | $4 \cdot f_{FS}$ | $4 \cdot f_{FS}$ | $4 \cdot f_{FS}$ | $4 \cdot f_{FS}$ | $4 \cdot f_{FS}$ |
| | 100 | NA | NA | NA | NA | $8 \cdot f_{FS}$ | $8 \cdot f_{FS}$ | $8 \cdot f_{FS}$ | $8 \cdot f_{FS}$ |
| | 101 | NA | NA | NA | NA | NA | $16 \cdot f_{FS}$ | $16 \cdot f_{FS}$ | $16 \cdot f_{FS}$ |
| | 110 | NA | NA | NA | NA | NA | NA | $32 \cdot f_{FS}$ | $32 \cdot f_{FS}$ |
| | 111 | NA | NA | NA | NA | NA | NA | NA | $64 \cdot f_{FS}$ |

The synchronization signal is obtained starting from electrical position information (EL_POS register) according to [Table 20](#):

Table 20. SYNC signal source

| SYNC_SEL[2 0] | | | Source |
|--------------------|---|---|-----------|
| 0 | 0 | 0 | EL_POS[7] |
| 0 | 0 | 1 | EL_POS[6] |
| 0 | 1 | 0 | EL_POS[5] |
| 0 | 1 | 1 | EL_POS[4] |
| 1 | 0 | 0 | EL_POS[3] |
| 1 | 0 | 1 | EL_POS[2] |
| 1 | 1 | 0 | EL_POS[1] |
| 1 | 1 | 1 | EL_POS[0] |

9.1.20 ALARM_EN

The ALARM_EN register allows the selection of which alarm signals are used to generate the FLAG output. If the respective bit of the ALARM_EN register is set high, the alarm condition forces the FLAG pin output down.

Table 21. ALARM_EN register

| ALARM_EN bit | Alarm condition |
|--------------|----------------------------------|
| 0 (LSB) | Overcurrent |
| 1 | Thermal shutdown |
| 2 | Thermal warning |
| 3 | Undervoltage |
| 4 | Stall detection (Bridge A) |
| 5 | Stall detection (Bridge B) |
| 6 | Switch turn-on event |
| 7 (MSB) | Wrong or non-performable command |

9.1.21 CONFIG

The CONFIG register has the following structure:

Table 22. CONFIG register

| Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
|-----------|----------|-----------|-----------|---------|---------|--------|-------|
| F_PWM_INT | | | F_PWM_DEC | | | POW_SR | |
| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| OC_SD | RESERVED | EN_VSCOMP | SW_MODE | EXT_CLK | OSC_SEL | | |

The OSC_SEL and EXT_CLK bits set the system clock source:

Table 23. Oscillator management

| EXT_CLK | OSC_SEL[2 0] | | | Clock source | OSCIN | OSCOUT |
|---------|-------------------|---|---|--|---------------------------|--------------------------------|
| 0 | 0 | 0 | 0 | Internal oscillator: 16 MHz | Unused | Unused |
| 0 | 0 | 0 | 1 | | | |
| 0 | 0 | 1 | 0 | | | |
| 0 | 0 | 1 | 1 | | | |
| 1 | 0 | 0 | 0 | Internal oscillator: 16 MHz | Unused | Supplies a 2-MHz clock |
| 1 | 0 | 0 | 1 | Internal oscillator: 16 MHz | Unused | Supplies a 4-MHz clock |
| 1 | 0 | 1 | 0 | Internal oscillator: 16 MHz | Unused | Supplies an 8-MHz clock |
| 1 | 0 | 1 | 1 | Internal oscillator: 16 MHz | Unused | Supplies a 16-MHz clock |
| 0 | 1 | 0 | 0 | External crystal or resonator: 8 MHz | Crystal/resonator driving | Crystal/resonator driving |
| 0 | 1 | 0 | 1 | External crystal or resonator: 16 MHz | Crystal/resonator driving | Crystal/resonator driving |
| 0 | 1 | 1 | 0 | External crystal or resonator: 24 MHz | Crystal/resonator driving | Crystal/resonator driving |
| 0 | 1 | 1 | 1 | External crystal or resonator: 32 MHz | Crystal/resonator driving | Crystal/resonator driving |
| 1 | 1 | 0 | 0 | Ext clock source: 8 MHz (Crystal/resonator driver disabled) | Clock source | Supplies inverted OSCIN signal |
| 1 | 1 | 0 | 1 | Ext clock source: 16 MHz (Crystal/resonator driver disabled) | Clock source | Supplies inverted OSCIN signal |
| 1 | 1 | 1 | 0 | Ext clock source: 24 MHz (Crystal/resonator driver disabled) | Clock source | Supplies inverted OSCIN signal |
| 1 | 1 | 1 | 1 | Ext clock source: 32 MHz (Crystal/resonator driver disabled) | Clock source | Supplies inverted OSCIN signal |

The SW_MODE bit sets the external switch to act as HardStop interrupt or not:

Table 24. External switch hard stop interrupt mode

| SW_MODE | Switch mode |
|---------|--------------------|
| 0 | HardStop interrupt |
| 1 | User disposal |

The OC_SD bit sets whether an overcurrent event causes or not the bridges to turn off; the OCD flag in the STATUS register is forced low anyway:

Table 25. Overcurrent event

| OC_SD | Overcurrent event |
|-------|--------------------------|
| 1 | Bridges shut down |
| 0 | Bridges do not shut down |

The POW_SR bits set the slew rate value of power bridge output:

Table 26. Programmable power bridge output slew rate values

| POW_SR [1 ... 0] | | Output slew rate ⁽¹⁾ [V/μs] |
|---------------------|---|---|
| 0 | 0 | 320 |
| 0 | 1 | 75 |
| 1 | 0 | 110 |
| 1 | 1 | 260 |

1. See S_{Rout_r} and S_{Rout_f} parameters in [Table 5 on page 11](#) for details.

The EN_VSCOMP bit sets whether the motor supply voltage compensation is enabled or not.

Table 27. Motor supply voltage compensation enable

| EN_VSCOMP | Motor supply voltage compensation |
|-----------|-----------------------------------|
| 0 | Disabled |
| 1 | Enabled |

The F_PWM_INT bits set the integer division factor of PWM frequency generation.

Table 28. PWM frequency: integer division factor

| F_PWM_INT [2 0] | | | Integer division factor |
|-------------------------|---|---|-------------------------|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 2 |
| 0 | 1 | 0 | 3 |
| 0 | 1 | 1 | 4 |
| 1 | 0 | 0 | 5 |
| 1 | 0 | 1 | 6 |
| 1 | 1 | 0 | 7 |
| 1 | 1 | 1 | |

The F_PWM_DEC bits set the multiplication factor of PWM frequency generation.

Table 29. PWM frequency: multiplication factor

| F_PWM_DEC [2 0] | | | Multiplication factor |
|----------------------|---|---|-----------------------|
| 0 | 0 | 0 | 0.625 |
| 0 | 0 | 1 | 0.75 |
| 0 | 1 | 0 | 0.875 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1.25 |
| 1 | 0 | 1 | 1.5 |
| 1 | 1 | 0 | 1.75 |
| 1 | 1 | 1 | 2 |

In the following tables all available PWM frequencies are listed according to oscillator frequency, F_PWM_INT and F_PWM_DEC values (CONFIG register OSC_SEL parameter must be correctly programmed).

Table 30. Available PWM frequencies [kHz]: 8-MHz oscillator frequency

| F_PWM_INT | F_PWM_DEC | | | | | | | |
|-----------|-----------|------|------|------|------|------|------|------|
| | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| 000 | 9.8 | 11.7 | 13.7 | 15.6 | 19.5 | 23.4 | 27.3 | 31.3 |
| 001 | 4.9 | 5.9 | 6.8 | 7.8 | 9.8 | 11.7 | 13.7 | 15.6 |
| 010 | 3.3 | 3.9 | 4.6 | 5.2 | 6.5 | 7.8 | 9.1 | 10.4 |
| 011 | 2.4 | 2.9 | 3.4 | 3.9 | 4.9 | 5.9 | 6.8 | 7.8 |
| 100 | 2.0 | 2.3 | 2.7 | 3.1 | 3.9 | 4.7 | 5.5 | 6.3 |
| 101 | 1.6 | 2.0 | 2.3 | 2.6 | 3.3 | 3.9 | 4.6 | 5.2 |
| 110 | 1.4 | 1.7 | 2.0 | 2.2 | 2.8 | 3.3 | 3.9 | 4.5 |

Table 31. Available PWM frequencies [kHz]: 16-MHz oscillator frequency

| F_PWM_INT | F_PWM_DEC | | | | | | | |
|-----------|-----------|------|------|------|------|------|------|------|
| | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| 000 | 19.5 | 23.4 | 27.3 | 31.3 | 39.1 | 46.9 | 54.7 | 62.5 |
| 001 | 9.8 | 11.7 | 13.7 | 15.6 | 19.5 | 23.4 | 27.3 | 31.3 |
| 010 | 6.5 | 7.8 | 9.1 | 10.4 | 13.0 | 15.6 | 18.2 | 20.8 |
| 011 | 4.9 | 5.9 | 6.8 | 7.8 | 9.8 | 11.7 | 13.7 | 15.6 |
| 100 | 3.9 | 4.7 | 5.5 | 6.3 | 7.8 | 9.4 | 10.9 | 12.5 |
| 101 | 3.3 | 3.9 | 4.6 | 5.2 | 6.5 | 7.8 | 9.1 | 10.4 |
| 110 | 2.8 | 3.3 | 3.9 | 4.5 | 5.6 | 6.7 | 7.8 | 8.9 |

Table 32. Available PWM frequencies [kHz]: 24-MHz oscillator frequency

| F_PWM_INT | F_PWM_DEC | | | | | | | |
|-----------|-----------|------|------|------|------|------|------|------|
| | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| 000 | 29.3 | 35.2 | 41.0 | 46.9 | 58.6 | 70.3 | 82.0 | 93.8 |
| 001 | 14.6 | 17.6 | 20.5 | 23.4 | 29.3 | 35.2 | 41.0 | 46.9 |
| 010 | 9.8 | 11.7 | 13.7 | 15.6 | 19.5 | 23.4 | 27.3 | 31.3 |
| 011 | 7.3 | 8.8 | 10.3 | 11.7 | 14.6 | 17.6 | 20.5 | 23.4 |
| 100 | 5.9 | 7.0 | 8.2 | 9.4 | 11.7 | 14.1 | 16.4 | 18.8 |
| 101 | 4.9 | 5.9 | 6.8 | 7.8 | 9.8 | 11.7 | 13.7 | 15.6 |
| 110 | 4.2 | 5.0 | 5.9 | 6.7 | 8.4 | 10.0 | 11.7 | 13.4 |

Table 33. Available PWM frequencies [kHz]: 32-MHz oscillator frequency

| F_PWM_INT | F_PWM_DEC | | | | | | | |
|-----------|-----------|------|------|------|------|------|-------|-------|
| | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| 000 | 39.1 | 46.9 | 54.7 | 62.5 | 78.1 | 93.8 | 109.4 | 125.0 |
| 001 | 19.5 | 23.4 | 27.3 | 31.3 | 39.1 | 46.9 | 54.7 | 62.5 |
| 010 | 13.0 | 15.6 | 18.2 | 20.8 | 26.0 | 31.3 | 36.5 | 41.7 |
| 011 | 9.8 | 11.7 | 13.7 | 15.6 | 19.5 | 23.4 | 27.3 | 31.3 |
| 100 | 7.8 | 9.4 | 10.9 | 12.5 | 15.6 | 18.8 | 21.9 | 25.0 |
| 101 | 6.5 | 7.8 | 9.1 | 10.4 | 13.0 | 15.6 | 18.2 | 20.8 |
| 110 | 5.6 | 6.7 | 7.8 | 8.9 | 11.2 | 13.4 | 15.6 | 17.9 |

Any attempt to write the CONFIG register when the motor is running causes the command to be ignored and the NOTPERF_CMD flag to rise (see [Section 9.1.22](#)).

9.1.22 STATUS

Table 34. STATUS register

| Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
|-------------|-------------|-------------|--------|--------|--------|-------|-----------|
| SCK_MOD | STEP_LOSS_B | STEP_LOSS_A | OCD | TH_SD | TH_WRN | UVLO | WRONG_CMD |
| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| NOTPERF_CMD | MOT_STATUS | | DIR | SW_EVN | SW_F | BUSY | HiZ |

When the HiZ flag is high, it indicates that the bridges are in high impedance state. Any motion command makes the device exit from High Z state (HardStop and SoftStop included), unless error flags forcing a High Z state are active.

The UVLO flag is active low and is set by an undervoltage lockout or reset events (power-up included).

The TH_WRN, TH_SD, OCD flags are active low and indicate, respectively, thermal warning, thermal shutdown and overcurrent detection events.

STEP_LOSS_A and STEP_LOSS_B flags are forced low when a stall is detected on bridge A or bridge B respectively.

The NOTPERF_CMD and WRONG_CMD flags are active high and indicate, respectively, that the command received by SPI cannot be performed or does not exist at all.

The SW_F flag reports the SW input status (low for open and high for closed).

The SW_EVN flag is active high and indicates a switch turn-on event (SW input falling edge).

The UVLO, TH_WRN, TH_SD, OCD, STEP_LOSS_A, STEP_LOSS_B, NOTPERF_CMD, WRONG_CMD and SW_EVN flags are latched: when the respective conditions make them active (low or high), they remain in that state until a GetStatus command is sent to the IC.

The BUSY bit reflects the $\overline{\text{BUSY}}$ pin status. The BUSY flag is low when a constant speed, positioning or motion command is under execution and is released (high) after the command has been completed.

The SCK_MOD bit is an active high flag indicating that the device is working in Step-clock mode. In this case the step-clock signal should be provided through the STCK input pin. The DIR bit indicates the current motor direction:

Table 35. STATUS register DIR bit

| DIR | Motor direction |
|-----|-----------------|
| 1 | Forward |
| 0 | Reverse |

MOT_STATUS indicates the current motor status:

Table 36. STATUS register MOT_STATUS bits

| MOT_STATUS | | Motor status |
|------------|---|----------------|
| 0 | 0 | Stopped |
| 0 | 1 | Acceleration |
| 1 | 0 | Deceleration |
| 1 | 1 | Constant speed |

Any attempt to write to the register causes the command to be ignored and the NOTPERF_CMD flag to rise.

9.2 Application commands

The command summary is given in [Table 37](#).

Table 37. Application commands

| Command mnemonic | Command binary code | | | | | Action |
|-----------------------|---------------------|---------|-----|------------|-----|--|
| | [7 5] | [4] | [3] | [2 1] | [0] | |
| NOP | 000 | 0 | 0 | 00 | 0 | Nothing |
| SetParam(PARAM,VALUE) | 000 | [PARAM] | | | | Writes VALUE in PARAM register |
| GetParam(PARAM) | 001 | [PARAM] | | | | Returns the stored value in PARAM register |
| Run(DIR,SPD) | 010 | 1 | 0 | 00 | DIR | Sets the target speed and the motor direction |
| StepClock(DIR) | 010 | 1 | 1 | 00 | DIR | Puts the device into Step-clock mode and imposes DIR direction |
| Move(DIR,N_STEP) | 010 | 0 | 0 | 00 | DIR | Makes N_STEP (micro)steps in DIR direction (Not performable when motor is running) |
| GoTo(ABS_POS) | 011 | 0 | 0 | 00 | 0 | Brings motor into ABS_POS position (minimum path) |
| GoTo_DIR(DIR,ABS_POS) | 011 | 0 | 1 | 00 | DIR | Brings motor into ABS_POS position forcing DIR direction |
| GoUntil(ACT,DIR,SPD) | 100 | 0 | ACT | 01 | DIR | Performs a motion in DIR direction with speed SPD until SW is closed, the ACT action is executed then a SoftStop takes place. |
| ReleseSW(ACT, DIR) | 100 | 1 | ACT | 01 | DIR | Performs a motion in DIR direction at minimum speed until the SW is released (open), the ACT action is executed then a HardStop takes place. |
| GoHome | 011 | 1 | 0 | 00 | 0 | Brings the motor into HOME position |
| GoMark | 011 | 1 | 1 | 00 | 0 | Brings the motor into MARK position |
| ResetPos | 110 | 1 | 1 | 00 | 0 | Resets the ABS_POS register (set HOME position) |

Table 37. Application commands (continued)

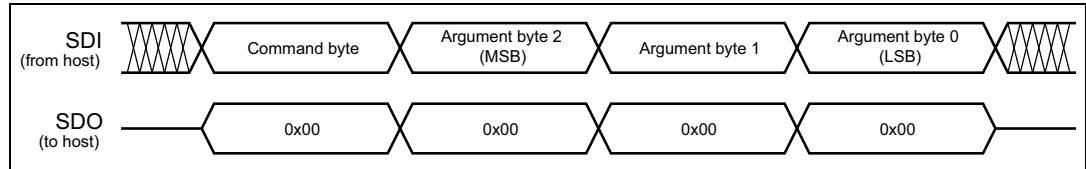
| Command mnemonic | Command binary code | | | | | Action |
|------------------|---------------------|-----|-----|------------|-----|--|
| | [7 5] | [4] | [3] | [2 1] | [0] | |
| ResetDevice | 110 | 0 | 0 | 00 | 0 | Device is reset to power-up conditions. |
| SoftStop | 101 | 1 | 0 | 00 | 0 | Stops motor with a deceleration phase |
| HardStop | 101 | 1 | 1 | 00 | 0 | Stops motor immediately |
| SoftHiZ | 101 | 0 | 0 | 00 | 0 | Puts the bridges into high impedance status after a deceleration phase |
| HardHiZ | 101 | 0 | 1 | 00 | 0 | Puts the bridges into high impedance status immediately |
| GetStatus | 110 | 1 | 0 | 00 | 0 | Returns the STATUS register value |
| RESERVED | 111 | 0 | 1 | 01 | 1 | RESERVED COMMAND |
| RESERVED | 111 | 1 | 1 | 00 | 0 | RESERVED COMMAND |

9.2.1 Command management

The host microcontroller can control motor motion and configure the L6470 device through a complete set of commands.

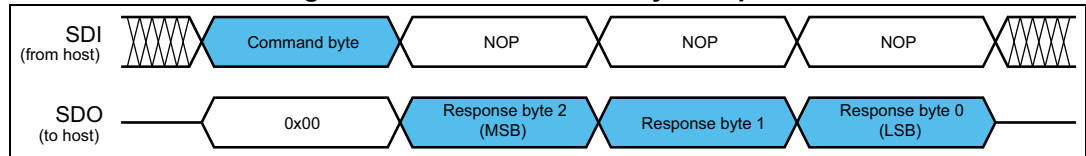
All commands are composed by a single byte. After the command byte, some bytes of arguments should be needed (see [Figure 19](#)). Argument length can vary from 1 to 3 bytes.

Figure 19. Command with 3-byte argument



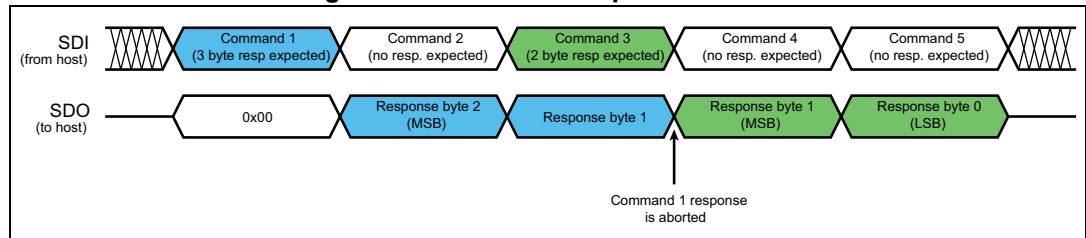
By default, the device returns an all zero response for any received byte, the only exceptions are the GetParam and GetStatus commands. When one of these commands is received, the following response bytes represent the related register value (see [Figure 20](#)). Response length can vary from 1 to 3 bytes.

Figure 20. Command with 3-byte response



During response transmission, new commands can be sent. If a command requiring a response is sent before the previous response is completed, the response transmission is aborted and the new response is loaded into the output communication buffer (see [Figure 21](#)).

Figure 21. Command response aborted



When a byte that does not correspond to a command is sent to the IC, it is ignored and the WRONG_CMD flag in the STATUS register is raised (see [Section 9.1.22](#)).

9.2.2 Nop

Table 38. Nop command structure

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|-------|-------|-------|-------|-------|-------|-------|-------|-----------|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | From host |

Nothing is performed.

9.2.3 SetParam (PARAM, VALUE)

Table 39. SetParam command structure

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|--------------------------|-------|-------|-------|-------|-------|-------|-------|-----------|
| 0 | 0 | 0 | PARAM | | | | | From host |
| VALUE Byte 2 (if needed) | | | | | | | | |
| VALUE Byte 1 (if needed) | | | | | | | | |
| VALUE Byte 0 | | | | | | | | |

The SetParam command sets the PARAM register value equal to VALUE; PARAM is the respective register address listed in [Table 12 on page 44](#).

The command should be followed by the new register VALUE (most significant byte first). The number of bytes making up the VALUE argument depends on the length of the target register (see [Table 12](#)).

Some registers cannot be written (see [Table 12](#)); any attempt to write one of those registers causes the command to be ignored and the WRONG_CMD flag to rise at the end of the command byte as if an unknown command code were sent (see [Section 9.1.22 on page 55](#)).

Some registers can only be written in particular conditions (see [Table 12](#)); any attempt to write one of those registers when the conditions are not satisfied causes the command to be ignored and the NOTPERF_CMD flag to rise at the end of the last argument byte (see [Section 9.1.22](#)).

Any attempt to set an inexistent register (wrong address value) causes the command to be ignored and the WRONG_CMD flag to rise at the end of the command byte as if an unknown command code were sent.

9.2.4 GetParam (PARAM)

Table 40. GetParam command structure

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|------------------------|-------|-------|-------|-------|-------|-------|---------|-----------|
| 0 | 0 | 1 | PARAM | | | | | From host |
| ANS Byte 2 (if needed) | | | | | | | To host | |
| ANS Byte 1 (if needed) | | | | | | | To host | |
| ANS Byte 0 | | | | | | | To host | |

This command reads the current PARAM register value; PARAM is the respective register address listed in [Table 12](#).

The command response is the current value of the register (most significant byte first). The number of bytes making up the command response depends on the length of the target register (see [Table 12](#)).

The returned value is the register one at the moment of GetParam command decoding. If register values change after this moment, the response is not accordingly updated.

All registers can be read anytime.

Any attempt to read an inexistent register (wrong address value) causes the command to be ignored and the WRONG_CMD flag to rise at the end of the command byte as if an unknown command code were sent.

9.2.5 Run (DIR, SPD)

Table 41. Run command structure

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|--------------|-------|-------|-------|--------------|-------|-------|-------|-----------|
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | DIR | From host |
| X | X | X | X | SPD (Byte 2) | | | | From host |
| SPD (Byte 1) | | | | | | | | From host |
| SPD (Byte 0) | | | | | | | | From host |

The Run command produces a motion at SPD speed; the direction is selected by the DIR bit: '1' forward or '0' reverse. The SPD value is expressed in step/tick (format unsigned fixed point 0.28) that is the same format as the SPEED register (see [Section 9.1.4 on page 42](#)).

Note: The SPD value should be lower than MAX_SPEED and greater than MIN_SPEED otherwise the Run command is executed at MAX_SPEED or MIN_SPEED respectively.

This command keeps the BUSY flag low until the target speed is reached.

This command can be given anytime and is immediately executed.

9.2.6 StepClock (DIR)

Table 42. Stepclock command structure

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|-------|-------|-------|-------|-------|-------|-------|-------|-----------|
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | DIR | From host |

The StepClock command switches the device in Step-clock mode (see [Section 6.7.5 on page 26](#)) and imposes the forward (DIR = '1') or reverse (DIR = '0') direction.

When the device is in Step-clock mode, the SCK_MOD flag in the STATUS register is raised and the motor is always considered stopped (see [Section 6.7.5](#) and [Section 9.1.22 on page 55](#)).

The device exits from Step-clock mode when a constant speed, absolute positioning or motion command is sent through SPI. Motion direction is imposed by the respective StepClock command argument and can be changed by a new StepClock command without exiting Step-clock mode.

Events that cause bridges to be forced into high impedance state (overtemperature, overcurrent, etc.) do not cause the device to leave Step-clock mode.

The StepClock command does not force the BUSY flag low. This command can only be given when the motor is stopped. If a motion is in progress, the motor should be stopped and it is then possible to send a StepClock command.

Any attempt to perform a StepClock command when the motor is running causes the command to be ignored and the NOTPERF_CMD flag to rise (see [Section 9.1.22](#)).

9.2.7 Move (DIR, N_STEP)

Table 43. Move command structure

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|-----------------|-------|-----------------|-------|-------|-------|-------|-------|-----------|
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | DIR | From host |
| X | X | N_STEP (Byte 2) | | | | | | From host |
| N_STEP (Byte 1) | | | | | | | | From host |
| N_STEP (Byte 0) | | | | | | | | From host |

The Move command produces a motion of N_STEP microsteps; the direction is selected by the DIR bit ('1' forward or '0' reverse).

The N_STEP value is always in agreement with the selected step mode; the parameter value unit is equal to the selected step mode (full, half, quarter, etc.).

This command keeps the BUSY flag low until the target number of steps is performed. This command can only be performed when the motor is stopped. If a motion is in progress, the motor must be stopped and it is then possible to perform a Move command.

Any attempt to perform a Move command when the motor is running causes the command to be ignored and the NOTPERF_CMD flag to rise (see [Section 9.1.22 on page 55](#)).

9.2.8 GoTo (ABS_POS)

Table 44. GoTo command structure

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|------------------|-------|------------------|-------|-------|-------|-------|-------|-----------|
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | From host |
| X | X | ABS_POS (Byte 2) | | | | | | From host |
| ABS_POS (Byte 1) | | | | | | | | From host |
| ABS_POS (Byte 0) | | | | | | | | From host |

The GoTo command produces a motion to ABS_POS absolute position through the shortest path. The ABS_POS value is always in agreement with the selected step mode; the parameter value unit is equal to the selected step mode (full, half, quarter, etc.).

The GoTo command keeps the BUSY flag low until the target position is reached.

This command can be given only when the previous motion command has been completed (BUSY flag released).

Any attempt to perform a GoTo command when a previous command is under execution (BUSY low) causes the command to be ignored and the NOTPERF_CMD flag to rise (see [Section 9.1.22](#)).

9.2.9 GoTo_DIR (DIR, ABS_POS)

Table 45. GoTo_DIR command structure

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | | |
|------------------|-------|------------------|-------|-------|-------|-------|-------|-----------|-----------|
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | DIR | From host | |
| X | X | ABS_POS (Byte 2) | | | | | | | From host |
| ABS_POS (Byte 1) | | | | | | | | From host | |
| ABS_POS (Byte 0) | | | | | | | | From host | |

The GoTo_DIR command produces a motion to ABS_POS absolute position imposing a forward (DIR = '1') or a reverse (DIR = '0') rotation. The ABS_POS value is always in agreement with the selected step mode; the parameter value unit is equal to the selected step mode (full, half, quarter, etc.).

The GoTo_DIR command keeps the BUSY flag low until the target speed is reached. This command can be given only when the previous motion command has been completed (BUSY flag released).

Any attempt to perform a GoTo_DIR command when a previous command is under execution (BUSY low) causes the command to be ignored and the NOTPERF_CMD flag to rise (see [Section 9.1.22 on page 55](#)).

9.2.10 GoUntil (ACT, DIR, SPD)

Table 46. GoUntil command structure

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | | |
|--------------|-------|-------|-------|--------------|-------|-------|-------|-----------|-----------|
| 1 | 0 | 0 | 0 | ACT | 0 | 1 | DIR | From host | |
| X | X | X | X | SPD (Byte 2) | | | | | From host |
| SPD (Byte 1) | | | | | | | | From host | |
| SPD (Byte 0) | | | | | | | | From host | |

The GoUntil command produces a motion at SPD speed imposing a forward (DIR = '1') or a reverse (DIR = '0') direction. When an external switch turn-on event occurs (see [Section 6.13 on page 31](#)), the ABS_POS register is reset (if ACT = '0') or the ABS_POS register value is copied into the MARK register (if ACT = '1'); then the system performs a SoftStop command.

The SPD value is expressed in step/tick (format unsigned fixed point 0.28) that is the same format as the SPEED register (see [Section 9.1.4 on page 42](#)).

The SPD value should be lower than MAX_SPEED and greater than MIN_SPEED, otherwise the target speed is imposed at MAX_SPEED or MIN_SPEED respectively.

If the SW_MODE bit of the CONFIG register is set low, the external switch turn-on event causes a HardStop interrupt instead of the SoftStop one (see [Section 6.13](#) and [Section 9.1.21 on page 49](#)).

This command keeps the BUSY flag low until the switch turn-on event occurs and the motor is stopped. This command can be given anytime and is immediately executed.

9.2.11 ReleaseSW (ACT, DIR)

Table 47. ReleaseSW command structure

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|-------|-------|-------|-------|-------|-------|-------|-------|-----------|
| 1 | 0 | 0 | 1 | ACT | 0 | 1 | DIR | From host |

The ReleaseSW command produces a motion at minimum speed imposing a forward (DIR = '1') or reverse (DIR = '0') rotation. When SW is released (opened), the ABS_POS register is reset (ACT = '0') or the ABS_POS register value is copied into the MARK register (ACT = '1'); the system then performs a HardStop command.

Note that resetting the ABS_POS register is equivalent to setting the HOME position.

If the minimum speed value is less than 5 step/s or low speed optimization is enabled, the motion is performed at 5 step/s.

The ReleaseSW command keeps the BUSY flag low until the switch input is released and the motor is stopped.

9.2.12 GoHome

Table 48. GoHome command structure

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|-------|-------|-------|-------|-------|-------|-------|-------|-----------|
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | From host |

The GoHome command produces a motion to the HOME position (zero position) via the shortest path.

Note that this command is equivalent to the “GoTo(0...0)” command. If a motor direction is mandatory, the GoTo_DIR command must be used (see [Section 9.2.9](#)).

The GoHome command keeps the BUSY flag low until the home position is reached. This command can be given only when the previous motion command has been completed. Any attempt to perform a GoHome command when a previous command is under execution (BUSY low) causes the command to be ignored and the NOTPERF_CMD to rise (see [Section 9.1.22 on page 55](#)).

9.2.13 GoMark

Table 49. GoMark command structure

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|-------|-------|-------|-------|-------|-------|-------|-------|-----------|
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | From host |

The GoMark command produces a motion to the MARK position performing the minimum path.

Note that this command is equivalent to the “GoTo (MARK)” command. If a motor direction is mandatory, the GoTo_DIR command must be used.

The GoMark command keeps the BUSY flag low until the MARK position is reached. This command can be given only when the previous motion command has been completed (BUSY flag released).

Any attempt to perform a GoMark command when a previous command is under execution (BUSY low) causes the command to be ignored and the NOTPERF_CMD flag to rise (see [Section 9.1.22 on page 55](#)).

9.2.14 ResetPos

Table 50. ResetPos command structure

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|-------|-------|-------|-------|-------|-------|-------|-------|-----------|
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | From host |

The ResetPos command resets the ABS_POS register to zero. The zero position is also defined as HOME position (see [Section 6.5 on page 23](#)).

9.2.15 ResetDevice

Table 51. ResetDevice command structure

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|-------|-------|-------|-------|-------|-------|-------|-------|-----------|
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | From host |

The ResetDevice command resets the device to power-up conditions (see [Section 6.1 on page 21](#)).

Note: At power-up the power bridges are disabled.

9.2.16 SoftStop

Table 52. SoftStop command structure

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|-------|-------|-------|-------|-------|-------|-------|-------|-----------|
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | From host |

The SoftStop command causes an immediate deceleration to zero speed and a consequent motor stop; the deceleration value used is the one stored in the DEC register (see [Section 9.1.6 on page 43](#)).

When the motor is in high impedance state, a SoftStop command forces the bridges to exit from high impedance state; no motion is performed.

This command can be given anytime and is immediately executed. This command keeps the BUSY flag low until the motor is stopped.

9.2.17 HardStop

Table 53. HardStop command structure

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|-------|-------|-------|-------|-------|-------|-------|-------|-----------|
| 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | From host |

The HardStop command causes an immediate motor stop with infinite deceleration.

When the motor is in high impedance state, a HardStop command forces the bridges to exit from high impedance state; no motion is performed.

This command can be given anytime and is immediately executed. This command keeps the BUSY flag low until the motor is stopped.

9.2.18 SoftHiZ

Table 54. SoftHiZ command structure

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|-------|-------|-------|-------|-------|-------|-------|-------|-----------|
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | From host |

The SoftHiZ command disables the power bridges (high impedance state) after a deceleration to zero; the deceleration value used is the one stored in the DEC register (see [Section 9.1.6 on page 43](#)). When bridges are disabled, the HiZ flag is raised.

When the motor is stopped, a SoftHiZ command forces the bridges to enter into high impedance state.

This command can be given anytime and is immediately executed. This command keeps the BUSY flag low until the motor is stopped.

9.2.19 HardHiZ

Table 55. HardHiZ command structure

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|-------|-------|-------|-------|-------|-------|-------|-------|-----------|
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | From host |

The HardHiZ command immediately disables the power bridges (high impedance state) and raises the HiZ flag.

When the motor is stopped, a HardHiZ command forces the bridges to enter into high impedance state.

This command can be given anytime and is immediately executed. This command keeps the BUSY flag low until the motor is stopped.

9.2.20 GetStatus

Table 56. GetStatus command structure

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|---------------|-------|-------|-------|-------|-------|-------|-------|-----------|
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | From host |
| | | | | | | | | |
| STATUS MSByte | | | | | | | | To host |
| STATUS LSByte | | | | | | | | To host |

The GetStatus command returns the STATUS register value.

The GetStatus command resets the STATUS register warning flags. The command forces the system to exit from any error state. The GetStatus command DOES NOT reset the HiZ flag.

10 Package information

In order to meet environmental requirements, ST offers these devices in different grades of ECOPACK® packages, depending on their level of environmental compliance. ECOPACK specifications, grade definitions and product status are available at: www.st.com. ECOPACK is an ST trademark.

10.1 HTSSOP28 package information

Figure 22. HTSSOP28 package outline

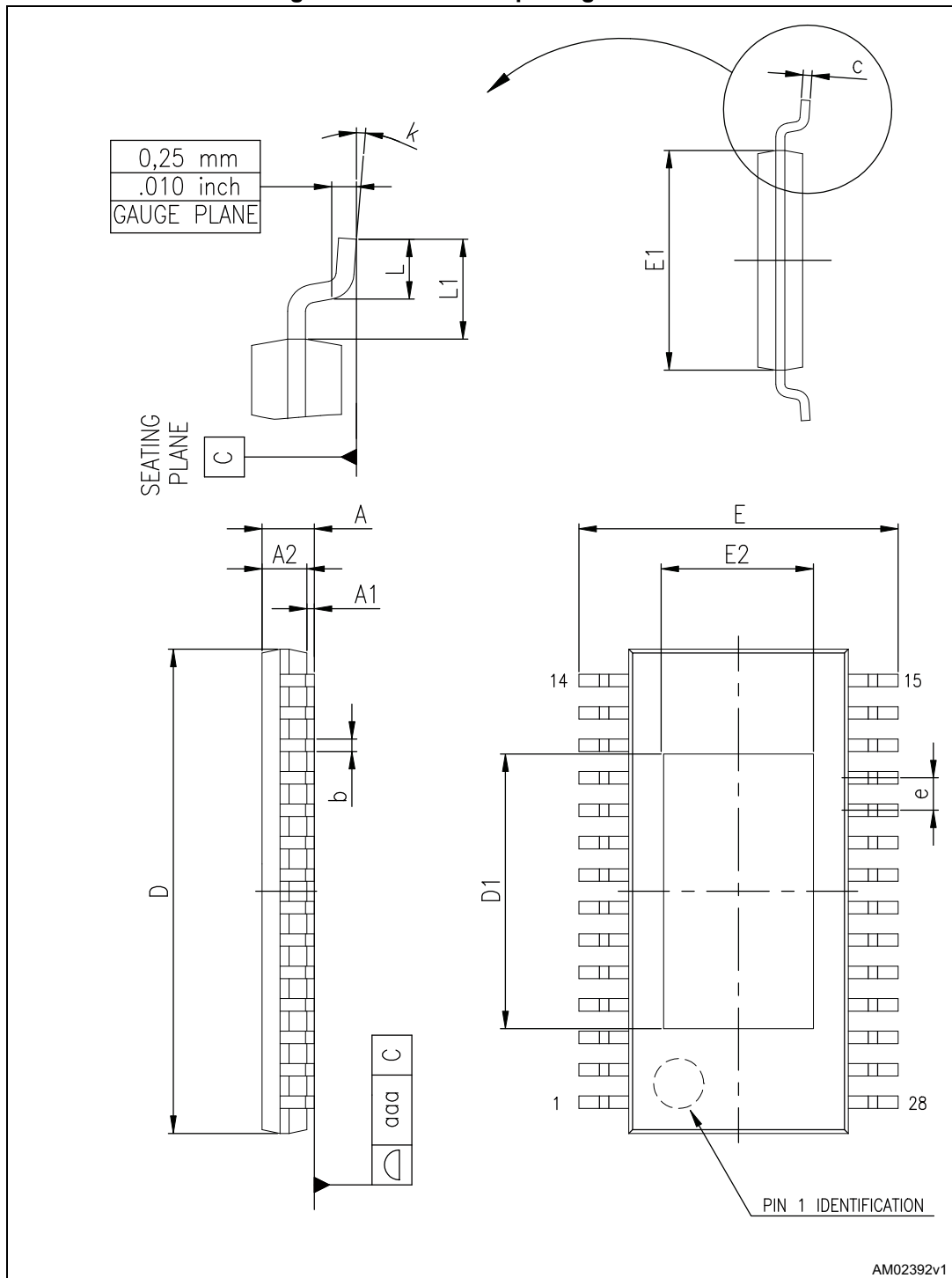


Table 57. HTSSOP28 package mechanical data

| Symbol | Dimensions (mm) | | |
|-------------------|-----------------|------|------|
| | Min. | Typ. | Max. |
| A | | | 1.2 |
| A1 | | | 0.15 |
| A2 | 0.8 | 1.0 | 1.05 |
| b | 0.19 | | 0.3 |
| c | 0.09 | | 0.2 |
| D ⁽¹⁾ | 9.6 | 9.7 | 9.8 |
| D1 | | 5.5 | |
| E | 6.2 | 6.4 | 6.6 |
| E1 ⁽²⁾ | 4.3 | 4.4 | 4.5 |
| E2 | | 2.8 | |
| e | | 0.65 | |
| L | 0.45 | 0.6 | 0.75 |
| L1 | | 1.0 | |
| K | 0° | | 8° |
| aaa | | 0.1 | |

1. Dimension "D" does not include mold flash, protrusions or gate burrs. Mold flash, protrusions or gate burrs must not exceed 0.15 mm per side.
2. Dimension "E1" does not include interlead flash or protrusions. Interlead flash or protrusions must not exceed 0.25 mm per side.

10.2 POWERSO36 package information

Figure 23. POWERSO36 package outline

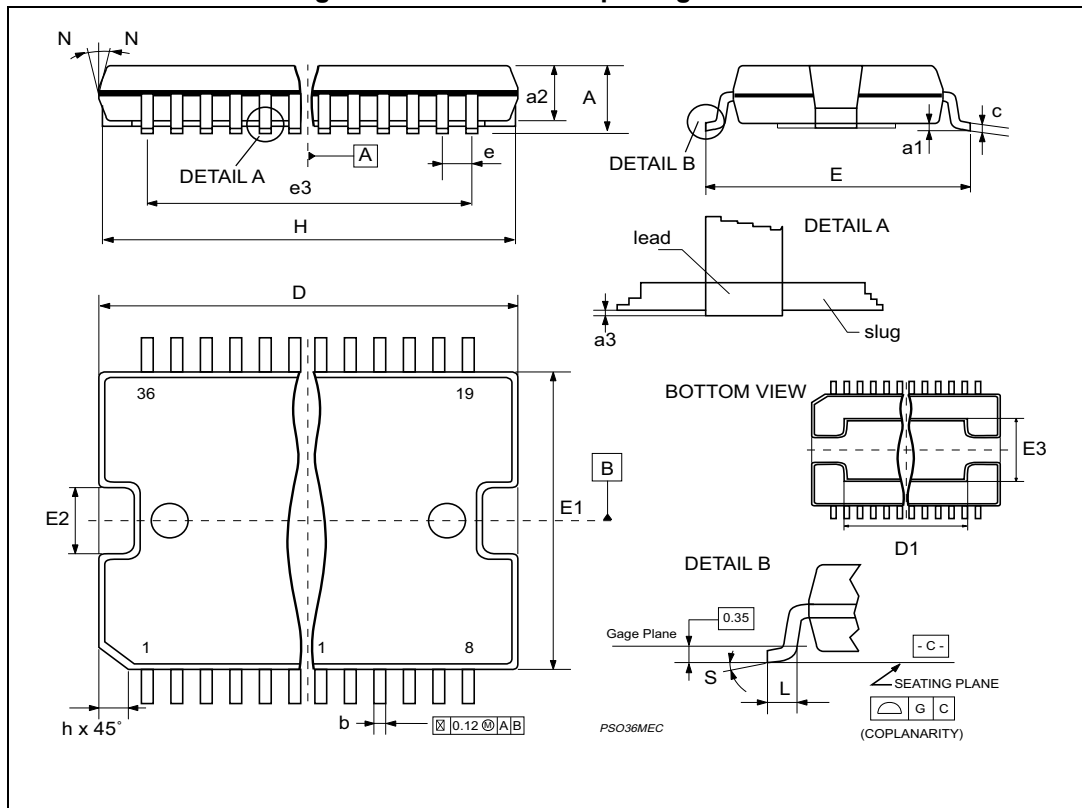


Table 58. POWERSO36 package mechanical data

| Symbol | Dimensions (mm) | | |
|-------------------|-----------------|-------|-------|
| | Min. | Typ. | Max. |
| A | | | 3.60 |
| a1 | 0.10 | | 0.30 |
| a2 | | | 3.30 |
| a3 | 0 | | 0.10 |
| b | 0.22 | | 0.38 |
| c | 0.23 | | 0.32 |
| D ⁽¹⁾ | 15.80 | | 16.00 |
| D1 | 9.40 | | 9.80 |
| E | 13.90 | | 14.50 |
| E1 ⁽¹⁾ | 10.90 | | 11.10 |
| E2 | | | 2.90 |
| E3 | 5.8 | | 6.2 |
| e | | 0.65 | |
| e3 | | 11.05 | |
| G | 0 | | 0.10 |
| H | 15.50 | | 15.90 |
| h | | | 1.10 |
| L | 0.80 | | 1.10 |
| N | | | 10° |
| S | 0° | | 8° |

1. Dimension "D/E1" does not include mold flash, protrusions or gate burrs. Mold flash, protrusions or gate burrs must not exceed 0.15 mm per side.

11 Revision history

Table 59. Revision history

| Date | Revision | Changes |
|-------------|----------|---|
| 06-Nov-2009 | 1 | Initial release |
| 05-Nov-2010 | 2 | Document status promoted from preliminary data to datasheet |
| 18-May-2011 | 3 | Updated: Table 4 , Table 5 Added: Section 6.7.6 , Section |
| 19-Jun-2012 | 4 | Added device in POWERSO36 and Figure 3 Updated: Table 2 , Table 3 , Table 4 , Table 5 , Table 6 , Table 9 and Section 9.1.11 . Minor text changes. |
| 20-Dec-2012 | 5 | Changed the title. Changed T _{OP} value in Table 2 Removed T _j value in Table 3 . Updated HTSSOP28 mechanical data. |
| 19-May-2014 | 6 | Updated Figure 3 on page 16 (replaced “DIR” pin by “SW” pin). Updated Section 6.4 on page 22 (replaced “the first microstep” by “zero”). Removed Section “Infinite acceleration/deceleration mode” from page 23. Updated Section 9.1.5 on page 42 (replaced “When the ACC value is set to 0xFFFF, the device works in infinite acceleration mode.” by “The 0xFFFF value of the register is reserved and it should never be used.”). Updated Section 9.1.6 on page 43 (removed “When the device is working in infinite acceleration mode, this value is ignored.”). Updated title of Table 36 on page 56 (replaced “MOT_STATE” by “MOT_STATUS”). Updated Section 10: Package information on page 67 (updated titles, reversed order of Figure 22 and Table 57 , Figure 23 and Table 58 , added note 1 below Table 58). Updated cross-references throughout document. Minor modifications throughout document. |
| 12-Mar-2015 | 7 | Removed “dSPIN™” from the main title on page 1 . Minor modifications throughout document. |

IMPORTANT NOTICE – PLEASE READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2015 STMicroelectronics – All rights reserved