ST solutions for efficient and robust motion control

Version 1.0
3 phase motors drives introduction

Field Oriented Control with STM32
- Application fit
- Performance
- Development Tools
- Evaluation Boards

ST VDE approved IEC65335 libraries
Scalar drives of 3Φ motors for AC IM

- Work often without any feedback devices (open-loop control)

- **Low cost and easy-to-implement** solution (8-bit MCU)

- On the other side
  - Developed torque is not controlled directly (depends on load)
  - Transient response is not fast due to the predefined switching pattern of the inverter

- Adding a speed sensor (tachometer) and slightly increasing control scheme complexity, transients responses can be made faster and torque estimation possible
Scalar drives of 3Φ motors for PMSM

- Dislike AC IM, always requires speed/position info
  - Hall sensors
  - Drawn from electrical quantities (e.g. phase voltage) feedback (sensor-less)
- Two families of drives available
  - Six-step
    - Sensor-less solution is low cost (8 bit MCU): advanced ADC and timer peripherals are mandatory
    - Torque steadiness is not excellent → noisy compared to other methods
  - Sinusoidal
    - Sensor-ed can be handled by 8 bit MCUs → low cost MCU
    - Sensor-less solution for sinusoidal would require hard computation (not manageable by 8 bit MCUs) → scalar sensor-less wouldn’t be low cost;
    - Torque steadiness is better compared to six step → more quiet
- In both cases developed torque is not accurately controlled
Field Oriented Control drive (FOC)

- FOC drive is also called *vector control* drive as the algorithm is based on a vector representation of the stator current, voltage and magnetic flux.

- The method always requires rotor speed/position information:
  - Measured through real sensors: Hall sensors, quadrature encoder, tachometer, …
  - Computed indirectly from electrical quantities feedback (sensor-less)

- FOC scheme and rotor position estimation algorithm (where needed) must be executed at a rate comparable with PWM frequency:
  - Higher computational power required compared to scalar drives → higher cost vs scalar
  - 32bit MCU is optimal

- FOC drive ensures:
  - The torque steadiness typical of a sinusoidal control
  - Excellent performance in terms of **accurate static and dynamic speed regulation** and **rapid response** to sudden changes in load torque
  - Provide **torque control as an alternative to speed control**
Field Oriented Control with STM32 (3ph brushless): From block diagram to implementation

- 6-channel PWM Timer
- Power stage
- Fault signals
- 6xPWM
- Ia & Ib
- Vbus
- Tachometer/Encoder/Hall sensor
- No present for sensorless algorithm

Variables:
- Vα, Vβ, Va, Vb, Vc
- Iα, Iβ, Ia, Ib
- Vbus
- ω_m
- θ_s
- d, q
- α, β
- ADCs
- Speed/position Feedback TIMER
- FOC algorithm
- Hw peripherals
STM32 PMSM FOC SDK v3.x

• STM32 PMSM FOC SDK v3.x:
  is a Motor Control Software Development Kit
  for 3-phase Permanent Magnet Synchronous
  Motors (PMSM) based on Field Oriented
  Control (FOC)

• Key features:
  • Single/Dual simultaneous vector control
    (FOC)
  • Any combination of current reading topologies
    and/or speed/position sensors is supported
  • Wide range of STM32 microcontrollers
    families are supported
  • Full customization and real time
    communication through PC software ST MC
    Workbench
  • Wide range of motor control algorithms
    implemented for specific applications
  • Application example based on FreeRTOS
  • Increase code safety through
    • MISRA C rules 2004 compliancy
    • Strict ANSI C compliancy
    • New object oriented FW architecture
      (better code encapsulation, abstraction
      and modularity)
FOC block diagram and possible configurations

- Speed position feedback is mandatory
- Three speed/position sensors are supported by the STM32 FOC SDK library:
  - Quadrature encoder
    - Expensive sensor, usually only in robotics applications
  - Hall sensors
    - Cheaper sensors, usually for application requiring full torque at zero speed
  - Sensor-less
    - Use electrical quantities (mainly current feedback) to estimate rotor position
    - Used for many applications not requiring full torque at zero speed or very low speed operations (< 3-5% of nominal speed)
• Current feedback is mandatory

• Three current sensing HW topologies:
  • 1 shunt resistor placed on the DC link
    • ST patented algorithm
    • Only one op-amp /shunt resistor → lowest cost
    • Current reading algorithm may result in not accurate torque regulation
  • 3 shunt resistors placed in the three legs
    • Current reading accuracy: high
    • Best compromise cost / performances
  • 2 Isolated Current Sensors (ICS)
    • Not dissipative current sensing topology → mandatory when current exceed some tens Ampere
    • Expensive
  • Any possible configuration (2 motors x 3 current sensing x 3 speed sensors type) is supported by STM32 FOC SDK library
Features set, MCU support

STM32F103x HD/XL, STM32F2xx, STM32F4xx, STM32Fyy

STM32F103x LD/MD

STM32F100x, STM32F0xx

<table>
<thead>
<tr>
<th>Feature</th>
<th>STM32F103x LD/MD</th>
<th>STM32F103x HD/XL, STM32F2xx, STM32F4xx, STM32Fyy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1shunt</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flux Weakening</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IPMSM MTPA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feed Forward</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sensor-less (STO + PLL)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sensor-less (STO + Cordic)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Encoder</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hall sensors</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Debug &amp; Tuning</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ST MC Workbench support</td>
<td></td>
<td></td>
</tr>
<tr>
<td>USART based com protocol add-on</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3shunt</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FreeRTOS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max FOC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F103 ~25kHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F2xx T.B.D.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F4xx T.B.D.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dual FOC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max FOC dual</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F103 ~20kHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F2xx T.B.D.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F4xx T.B.D.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
FOC, cost optimized implementation
STM32F100x Value Line

• **Target applications:**
  - All those applications where:
    - Requirements for dynamic performances are moderate
    - Quietness of sinusoidal current control (vs six steps drive) is valuable
    - Extended speed range is required
  - Particularly suitable for **pumps, fans and compressors**

- More silent
- Lower torque ripple
- Extended speed range easier to be achieved
STM32F100 Value Line Block Diagram

- 32-bit ARM® Cortex™-M3 core
  - Up to 30 DMIPS at 24 MHz max
- 2.0 V to 3.6 V operation
- -40 to +105 °C
- Enhanced control
  - 16-bit 3-phase motor-control timer
  - 6x 16-bit PWM timers
- Advanced analog
  - Fast 12-bit 1.2 µs ADC
  - Dual-channel 12-bit DAC
- System integration
  - Internal 8 MHz RC oscillator
  - Built-in safe reset system

- STM32F100 FOC performance driving example - 3phase PMSM
  - 1shunt/sensorless @20kHz PWM, 10kHZ FOC
  - Motor Control code size is 15.82Kb
  - Motor Control RAM usage is 2.77Kb
  - FOC Total execution time is 65.22us (ADC ISR + TIM1 Update ISR)
  - FOC introduced CPU load is 65.2%
  - Total CPU load is ~70% (~60% at 8kHz FOC)
FOC single and dual motor drive - high performances
STM32F103x, STM32F2x, STM32F4x

• Target applications:
  • Wide range from home appliances to robotics, where:
  • Accurate and quick regulation of motor speed and/or torque is required (e.g. in
torque load transient or target speed abrupt variations)
  • CPU load granted to motor control must be low due to other duties
STM32F103 Performance Line Block Diagram

- 32-bit ARM® Cortex™-M3 core
  - Up to DMIPS at 72MHz
- 2V-3.6V Supply
- -40 to +105 °C
- From 16kB to 1MB flash memory
- Enhanced control
  - Up to 3x 16-bit Advanced timer
  - Up to 4x 16-bit PWM timers
- Advanced analog
  - Up to 3x fast 12-bit 1.2 μs ADC
- System integration
  - Internal 8 MHz RC oscillator
  - Built-in safe reset system

- STM32F103 FOC performance driving example - 3phase PMSM
  - 1shunt/sensorless @20kHz PWM, 16kHz FOC
  - Motor Control code size is 16.2Kb
  - Motor Control RAM usage is 2.5Kb
  - FOC Total execution time is 26.1 usec us (ADC ISR + TIM1 Update ISR)
  - FOC introduced CPU load is 26%
  - Total CPU load is ~30%
STM32F4 block diagram

- 168 MHz Cortex-M4 CPU
- Floating point unit (FPU)
- ART Accelerator™
- Multi-level AHB bus matrix
- 1-Mbyte Flash, 192-Kbyte SRAM
- 1.7 to 3.6 V supply
- RTC: <1 µA typ, sub second accuracy
- 2x full duplex I²S
- 3x 12-bit ADC 0.41 µs/2.4 MSPS
- 168 MHz timers

Notes:
1. HS requires an external PHY connected to the ULP interface
2. Crypto/hash processor on STM32F417 and STM32F415
STM32 FOC dual motor drive
Some performances figure examples

• STM32F103 HD, dual FOC
  • Motor 1, 3 shunt/sensorless @16kHz PWM/FOC
  • Motor 2, 3 shunt/sensorless @16kHz PWM, 16kHZ FOC.
  • Motor Control code size is 22.3Kb (below 1.5 times single motor case)
  • Motor Control RAM usage is 4.01Kb
  • FOCs introduced CPU load (including TIMx Update ISRs) is 80%
  • Total CPU load ~85%

• STM32F4xx HD, dual FOC
  • Motor 1, 3 shunt/sensorless @16kHz PWM/FOC
  • Motor 2, 3 shunt/sensorless @16kHz PWM, 16kHZ FOC.
  • Motor Control code size is 22.3Kb (below 1.5 times single motor case)
  • Motor Control RAM usage is 4.01Kb
  • FOCs introduced CPU load (including TIMx Update ISRs) is 37%
  • Total CPU load ~42%
**STM32 – 6 product series**

**Common core peripherals and architecture:**

<table>
<thead>
<tr>
<th>Communication peripherals:</th>
<th>USART, SPI, I2C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiple general-purpose timers</td>
<td></td>
</tr>
<tr>
<td>Integrated reset and brown-out warning</td>
<td></td>
</tr>
<tr>
<td>Multiple DMA</td>
<td></td>
</tr>
<tr>
<td>2x watchdogs</td>
<td></td>
</tr>
<tr>
<td>Real-time clock</td>
<td></td>
</tr>
<tr>
<td>Integrated regulator</td>
<td></td>
</tr>
<tr>
<td>PLL and clock circuit</td>
<td></td>
</tr>
<tr>
<td>External memory interface (FSMC)</td>
<td></td>
</tr>
<tr>
<td>Dual 12-bit DAC</td>
<td></td>
</tr>
<tr>
<td>Up to 3x 12-bit ADC (up to 0.41 μs)</td>
<td></td>
</tr>
<tr>
<td>Main oscillator and 32 kHz oscillator</td>
<td></td>
</tr>
<tr>
<td>Low-speed and high-speed internal RC oscillators</td>
<td></td>
</tr>
<tr>
<td>-40 to +85 °C and up to 105 °C operating temperature range</td>
<td></td>
</tr>
<tr>
<td>Low voltage 2.0 to 3.6 V or 1.65/1.7 to 3.6 V (depending on series)</td>
<td></td>
</tr>
<tr>
<td>5.0 V tolerant I/Os</td>
<td></td>
</tr>
<tr>
<td>Temperature sensor</td>
<td></td>
</tr>
</tbody>
</table>

**STM32 F4 series - High performance with DSP (STM32F405/415/407/417)**

- 168 MHz Cortex-M4 with DSP and FPU
- Up to 192-Kbyte SRAM
- Up to 1-Mbyte Flash
- 2x USB 2.0 OTG
- 3-phase MC timer
- 2x CAN 2.0B
- SDIO
- 2x PS audio
- Camera IF
- Ethernet IEEE 1588
- Crypto/hash processor and RNG


- 120 MHz Cortex-M3 CPU
- Up to 128-Kbyte SRAM
- Up to 1-Mbyte Flash
- 2x USB 2.0 OTG
- 3-phase MC timer
- 2x CAN 2.0B
- SDIO
- 2x PS audio
- Camera IF
- Ethernet IEEE 1588
- Crypto/hash processor and RNG

**STM32 F1 series - 5 product lines (STM32F100/101/102/103/105/107)**

- Up to 72 MHz Cortex-M3 CPU
- Up to 96-Kbyte SRAM
- Up to 1-Mbyte Flash
- USB 2.0 FS device
- OTG
- 3-phase MC timer
- Up to 2x CAN 2.0B
- Up to 2x PS audio
- Ethernet IEEE 1588

**STM32 F0 series - Entry-level (STM32F051)**

- 48 MHz Cortex-M0 CPU
- Up to 20-Kbyte SRAM
- Up to 128-Kbyte Flash
- 3-phase MC timer
- Comparator
- CEC

**STM32 L1 series - Ultra-low-power (STM32L151/152/162)**

- 32 MHz Cortex-M3 CPU
- Up to 48-Kbyte SRAM
- Up to 384-Kbyte Flash
- USB FS device
- Data EEPROM up to 12 Kbytes
- LCD 8x40 4x44
- Comparator
- BOR
- MSI
- VScal
- AES 128-bit

**STM32W series - Wireless (STM32W108)**

- 24 MHz Cortex-M3 CPU
- Up to 16-Kbyte SRAM
- Up to 256-Kbyte Flash
- 2.4 GHz IEEE 802.15.4 Transceiver
- Lower MAC Digital baseband
- AES 128-bit
STM32 – leading Cortex-M portfolio

Over 300 pin-to-pin compatible part numbers
MC Workbench

Quick setup of the library according customer needs

• ST Motor Control Workbench
  • PC software that reduces the design effort and time in the STM32 PMSM FOC firmware library configuration. The user through a graphical user interface (GUI) generate all parameter header files which configures the library according the application needs

Motor

Power Stage

Drive Management

Control Stage
Serial communication

- Real time communication
  - Using the ST MC workbench is possible to instantiate a “real time communication” to send start/stop commands or to set a speed ramp
  - Debug or fine tuning motor control variables (like speed PI parameters) can be assessed using the advanced tab
  - Plotting significant motor control variables (virtual oscilloscope) like target or measured motor speed

- RS232 (Available)
- SPI (T.B.I.)
- I2C (T.B.I.)
STM32 FOC SDK sources and docs

For further info about STM32 PMSM FOC SDK v3.x, please visit:

http://www.st.com/stm32

Downloads:

**STM32 PMSM FOC SDK v3.0:**

http://www.st.com/internet/com/SOFTWARE_RESOURCES/SW_COMPONENT/FIRMWARE/stm32_pmsm_foc_motorcontrol_fwlib.zip

**ST MC Workbenchv1.2.0:**

http://www.st.com/internet/com/SOFTWARE_RESOURCES/TOOL/CONFIGURATION_UTILITY/motorcontrol_workbench.zip

**TN0516:** Overview of the STM32F103xx/STM32F100xx PMSM single/dual FOC SDK V3.0

**UM1052:** STM32F103xx/ STM32F100xx/STM32F2xx/F4xx PMSM single/dual FOC SDK v3.2

**UM1053:** Advanced developers guide for STM32F100x/103x/2x/40x/41x MCUs PMSM single/dual FOC SDK v3.2

www.st.com
• **Main Features**
  - Driving Strategy: Vector Control
  - PMSM motor sensored and sensorless
  - Two (34-pin) dedicated motor control connectors
  - Encoder sensor input
  - Hall sensor input
  - Tachometer sensor input
  - Current sensing mode:
    - 3 shunt resistors
    - Single shunt

• **Key Component**
  - L6390D (Gate Drivers)
  - VIPer16LD (Power Supply down converter)
  - L7815ABV, L78M05CDT, LD1117S33TR (Voltage regulators)
  - STGP10NC60KD (IGBT)
  - TS391ILT, (Comparator)
  - M74HC14TTR (Logic)
Complementing MC starter kits
STM8/32 Evaluation boards

STM8 MC library v1.0
STM8/128-EVAL
Steuerung

STM32 PMSM FOC SDK v3.2
STM32F100x
STM32F103, F2xx, F4xx
STEVAL-IHM022V1

FOC
STM32100B-EVAL
STEVAL-IHM033V1

Dual FOC
STM32-EVAL

MC connector

Power Stages

Please visit http://www.st.com/evalboards or contact a local ST office

Thanks configurable tools it is possible to have diverse motor drive solutions
Complementing MC starter kits  
STM8/32 Evaluation boards

- STEVAL-IHM025V1: 1000W
  - 1 x IGBT SLLIMM™ STGIP14K60
  - 1 converter based on Viper16
  - 1 x IGBT STGP10NC60KD

- STEVAL-IHM027V1: 1000W
  - 1 x IGBT SLLIMM™ STGIPS10K60A
  - 1 converter based on Viper16
  - 1 x IGBT STGP10NC60KD

- STEVAL-IHM028V1: 2000W
  - 1 x IGBT SLLIMM™ STGIPS20K60
  - 1 x PWM SMPS VIPer26LD
  - 1 x IGBT STGW35NB60SD

- STEVAL-IHM035V1: 100W
  - 1 x IGBT SLLIMM™ STGIPN3H60
  - 1 x PWM SMPS VIPer16L

- STEVAL-IHM023V2: 1KW
  - 3 x PWM smart driver L6390
  - 1 converter based on Viper16
  - 7 x IGBT power switch STGP10NC60KD

- STEVAL-IHM021V2: 100W
  - 3 x PWM smart driver L6390
  - 1 converter based on Viper12
  - 6 x MOSFET power switch STD5N52U

- STEVAL-IHM032V1: 150W
  - 3 x PWM smart driver: 2xL6392D and 1x L6391D
  - 1 converter based on Viper12
  - 6 x IGBT power switch: STGD3HF60HD

SLLIMM™ (ST IPMs) based

Gate drivers & Power Transistors based

Please visit http://www.st.com/evalboards or contact a local ST office
Complementing MC starter kits
Low Voltage Power Stages

**STEVAL-IHM031V1**

120W

**Power stage up to 12/24V**
• 3 x dual PowerMOSFETs **STS8dnh3l**
• 2 x PWM smart driver **L6387E**
• 1x step down converter **L4976D**

**STEVAL-IEM003V1**

2000W

**Power stage up to 48V**
• 3 x PWM smart driver **L6388**
• 6x LV Power MOSFET **STV250N55F3**
• 1x step down converter **L4978D**

Please visit [http://www.st.com/evalboards](http://www.st.com/evalboards) or contact a local ST office
Complete 3ph motor drive solutions

Low voltage drives

- **STEVAL-IFN003V1**
  - PMSM FOC Motor Drive
  - 1 x 32bit Micro STM32F103C
  - 1 x Motor Drive Ic L6230PD
  - 45W

- **STEVAL-IFN004V1**
  - BLDC Six-Steps Motor Drive
  - 1 x 8bit-Micro STM8S
  - 1 x Motor Drive Ic L6230Q
  - 35W

High voltage drives

- **STEVAL-IHM036V1**
  - PMSM FOC Motor Drive
  - 1 x 32bit Micro STM32F100C6
  - 1 x IGBT SLLIMM™ STGIPN3H60
  - 100W

- **STEVAL-IHM034V1**
  - Dual motor drive + digital PFC
  - 1 x 32bit Micro STM32F103C8T6
  - 1 x IGBT SLLIMM™ STGIPS20K60
  - 1 converter based on Viper16L
  - 1300W

- **STEVAL-IHM038V1**
  - FAN Drive + PFC + IrDA
  - 1 x 32bit Micro STM32F100C6
  - 1 x IGBT SLLIMM™ STGIPN3H60
  - 1 PFC controller L6562A
  - 40W

Please visit [http://www.st.com/evalboards](http://www.st.com/evalboards) or contact a local ST office
IEC standard - Introduction

• IEC – International Electro-technical Commission
  • WW authority, provides standardization of electric & electronic devices

• IEC60335-1 – safety of household electronics appliances
  • Guarantee the security of the user for domestic appliances (and public places like shops, offices, not industry applications)
  • The appliance must remain safe in case of any component failure
    • Safety relies on electronics component?
      >> It must stay safe after two consecutive failures!
    • Safety relies on software?
      >> Class B or Class C required!
  • Definition of procedures, requirements and parts of MCU to be tested

• Certification Bodies
  • WW recognized test houses for software safety inspection (VDE, UL)
ST Class B software library focus

- IEC60335 - Annex Q defines three safety classes for software
  - Class A: Safety does not rely on SW
  - **Class B: SW prevents unsafe operation**
    - Class C: SW is intended to prevent special hazards (dual MCUs)

- IEC60335 - Annex T – MCU compliance aspects related to
  - **Micro specific HW**
    (fixed by design - dual robust watchdog, dual internal RC oscillators, high impedance I/Os at reset, Flash ECC, SRAM parity)
  - **Micro specific SW**
    (self diagnostic of the core, memories, clocks, execution)
  - **Application specific HW & SW**
    (analog I/O, digital I/O, interrupts, communication, spec. peripherals)

- **ST code & end user certification**
  - ST pre-certified software is integrated into user code
  - End-application is certified by any certification authority
Currently published FW packages

*Packages certified by VDE at May 2010*

- **STM8 family**
  - STM8S optimized package (Rev 1.0.2)*
  - STM8L10x optimized package (Rev 1.0.2)
  - STM8L15x optimized package (Rev 1.0.2)
  *) covering all members of STM8S family, package was updated (certification planned in Q3/2012)

- **STM32 family**
  - STM32 package (Rev 2.0.0)*
  *) based on standard peripheral FW library Rev 3.3.0, no support of connectivity and XL devices (certification is planned in Q3/2012 to be based on the FW library Rev 3.5.0 and covering connectivity & XL devices + all new incoming devices)
  *) STM32 package is available upon request thru local ST sales offices

- **Available documentation**
  - AN3181 for STM8
  - AN3307 for STM32
Stepper Motor Control

- Performance
- Evaluation Boards
STm
new stepper motor control

L6472  Dspin current mode
L6474  Easyspin
L6470  Dspin voltage mode
## New Spin...

<table>
<thead>
<tr>
<th>Product</th>
<th>Peculiar features</th>
<th>Operating range</th>
<th>Integrated mosfet</th>
<th>Common features</th>
</tr>
</thead>
<tbody>
<tr>
<td>L6470</td>
<td><strong>Up to 128 microsteps</strong></td>
<td></td>
<td><strong>3Arms</strong></td>
<td>• Programmable speed profile (*)</td>
</tr>
<tr>
<td></td>
<td><strong>Voltage mode operation</strong></td>
<td></td>
<td><strong>R&lt;sub_DS,ON&lt;/sub&gt; = 0.28 Ω</strong></td>
<td>• Programmable positioning (*)</td>
</tr>
<tr>
<td></td>
<td>Sensorless Stall Detection</td>
<td>8V – 45V</td>
<td>Integrated Current Sensing (no ext.shut)</td>
<td>8bit 5Mhz SPI interface (Daisy Chain compatible)</td>
</tr>
<tr>
<td>L6472</td>
<td><strong>Up to 16 microsteps</strong></td>
<td></td>
<td></td>
<td>• Integrated 16MHz oscillator</td>
</tr>
<tr>
<td></td>
<td><strong>Current mode</strong></td>
<td></td>
<td></td>
<td>• Integrated 5bit ADC</td>
</tr>
<tr>
<td></td>
<td><strong>Advanced</strong> phase current control</td>
<td></td>
<td></td>
<td>• Integrated 3V voltage regulator</td>
</tr>
<tr>
<td></td>
<td><strong>Accurate internal current sensing</strong></td>
<td></td>
<td></td>
<td>• Over Current, Over Temperature and Under Voltage protections</td>
</tr>
<tr>
<td>L6474</td>
<td><strong>Up to 16 microsteps</strong></td>
<td></td>
<td></td>
<td>• PowerSO (ES) and HTSSOP</td>
</tr>
<tr>
<td></td>
<td><strong>Current mode with adaptive decay</strong></td>
<td></td>
<td></td>
<td>(*) not available for L6474</td>
</tr>
<tr>
<td>L6480</td>
<td><strong>Up to 128 microsteps</strong></td>
<td>7.5V – 85V</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Voltage mode operation</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sensorless Stall Detection</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Integrated 15V/7.5V voltage regulator</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fully programmable gate driving</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Embedded miller clamp</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
L6472: the new State of the Art in µstepping Drivers

- 3V Volt. Reg.
- ADC
- Charge Pump
- Programmable speed profile
- Innovative Current mode driving
- Microstepping
- Comprehensive command set
- Protections
- 16MHz Oscillator
- SPI
- Thermal protection
- DAC & Comp
- Current sensing
- dSPIN

dSPIN

dSPIN
L6472  Monolithic Digital µstepping current mode Driver

- Supply voltage 8V – 45V
- 3Arms (7A peak)
- $R_{\text{DS,ON}} = 0.28$ Ω
- **Integrated current sensing** (no external shunt)
- Up to 16 microsteps
- **Innovative current control**
  - Avr phase current control
  - Adaptative decay
- **Programmable speed profile**
- **Programmable positioning**
- 8bit 5Mhz SPI interface
  (Daisy Chain compatible)
- Integrated 16MHz oscillator
- **Integrated 5bit ADC**
- **Integrated 3V voltage regulator**
- Over Current,
- Over Temperature
- Under Voltage protections
- PowerSO and HTSSOP
Intelligence integration

before dSPIN...

- MCU
- dedicated hi-performance MCU
- many digital + analog connections
- stepper driver
- +
- MCU
- dedicated hi-performance MCU
- many digital + analog connections
- stepper driver
- +
- MCU
- dedicated hi-performance MCU
- many digital + analog connections
- stepper driver
- +

system MCU
with dSPIN...

- System is heavily simplified
- No more dedicated MCU to perform speed profile and positioning calculations
- A lot less passive components

and... far better performances!
A complete digital interface to MCU

- The fast SPI interface with **daisy-chain** capability allows a single MCU to manage multiple devices.

- Programmable alarm **FLAG** open drain output for interrupt-based FW. In daisy-chain configuration, **FLAG** pins of different devices can be or-wired to save host controller GPIOs.

- **BUSY** open drain output allows the MCU to known when the last command has been performed. In daisy-chain configuration, **BUSY** pins of different devices can be or-wired to save host controller GPIOs.

- **BUSY** can be used to feedback the \( \mu \)step clock to the \( \mu \)C (programmable # of \( \mu \)steps).
Positioning and speed profiles: Leave them to \textit{dSPIN}!

MCU sends \textit{dSPIN} high level commands...

- Free-run $\rightarrow$ run at constant speed
- Positioning $\rightarrow$ reach the desired position

... and \textit{dSPIN} does the tricky job!
dSPIN

L6472 : Many Commands

GoTo(Target) command: reach the target position using shortest path

GoTo_DIR(Target, DIR) command: reach the target position moving the motor in the selected direction

GoUntil command moves the motor with a selected constant speed and stops the motor when the switch is closed; at that time one of the following actions can be taken:

And : GoHome, GoMark, Run, Move, SoftStop, HardStop, SoftHiz, ReleaseSW.....
What is a decay?

Inductors store the kinetic energy of moving electrons in the form of a magnetic field.

The total energy (or work) done in establishing the final current $I_2$ in the inductor from the starting current $I_1$ is:

$$W = L \int_{I_1}^{I_2} i \, di = \frac{1}{2} L (I_2 - I_1)^2$$

(assuming $i$ linear)

A decay is a way to remove the energy $W$ from the coil.
Why a decay: for stable current

Current is applied with a chopping technique

Energy must be removed in order to keep the current level stable \(\rightarrow\) decay is necessary
dSPIN

Why a decay: Falling steps

The energy must be removed from the inductance when you switch current level 1 to a lower current level 2 \( \rightarrow \) decay is necessary.

\[
W = L \int_{I_1}^{I_2} i \, di = \frac{1}{2} L (I_2 - I_1)^2
\]
dSPIN
Evolved current control

• **Automatic selection of the decay mode**
  Stable current control in microstepping

• **Slow decay and fast decay balancing**
  Reduced current ripple

• **Predictive current control**
  Average current control

• **Automatic OFF time adjustment**
  Fixed switching frequency
dSPIN

Challenges to perform the right decay

The quantity of energy to removed in decay1 and decay2 are different → must choose the right timing and speed decay
dSPIN

Challenges to perform the right decay

During the OFF state, both slow and fast decay can be performed

L6472 performs AUTO-ADJUSTED DECAY
dSPIN
Timing PWM to control current

<table>
<thead>
<tr>
<th>Address [Hex]</th>
<th>Register name</th>
<th>Register function</th>
</tr>
</thead>
<tbody>
<tr>
<td>h0F</td>
<td>TON_MIN</td>
<td>Minimum ON time</td>
</tr>
<tr>
<td>h10</td>
<td>TOFF_MIN</td>
<td>Minimum OFF time</td>
</tr>
<tr>
<td>h18</td>
<td>CONFIG</td>
<td>Bit10-14 : TSW</td>
</tr>
</tbody>
</table>

TON Must be > TON_MIN

In stable current, TON and TOFF are constant
dSPIN
Auto-adjusted Decay w/ one Fast Decay

Target Current level

<table>
<thead>
<tr>
<th>Address [Hex]</th>
<th>Register name</th>
<th>Register function</th>
</tr>
</thead>
<tbody>
<tr>
<td>h0E</td>
<td>T_FAST</td>
<td>Fast decay/fall step time</td>
</tr>
<tr>
<td>h0F</td>
<td>TON_MIN</td>
<td>Minimum ON time</td>
</tr>
<tr>
<td>h18</td>
<td>CONFIG</td>
<td>Bit10-14 : TSW</td>
</tr>
</tbody>
</table>

Fast decay for TOFF,FAST = T_FAST/8 in order to remove more energy than a slow decay

Slow decay for TOFF = TSW(*)

(*) No predictive control
dSPIN
Auto-adjusted Decay w/ multiple fast decay

Target Current level

- **T\textsubscript{ON1} < T\textsubscript{ON\_MIN}**
- **T\textsubscript{ON2} < T\textsubscript{ON\_MIN}**
- **T\textsubscript{ON3} > T\textsubscript{ON\_MIN}**

Fast decay for
- TOFF\textsubscript{1} = \(\frac{T\_\text{FAST}}{8}\)
- TOFF\textsubscript{2} = \(\frac{T\_\text{FAST}}{4}\)

Fast decay for TOFF\textsubscript{2} = \(\frac{T\_\text{FAST}}{4}\)

Mixed decay:
- **TOFF\textsubscript{3} = TSW (*)**
- TOFF Fast = TOFF\textsubscript{2} = \(\frac{T\_\text{FAST}}{4}\)
- TOFF Slow = TOFF\textsubscript{3} - TOFF Fast

(*) No predictive control

<table>
<thead>
<tr>
<th>Address [Hex]</th>
<th>Register name</th>
<th>Register function</th>
</tr>
</thead>
<tbody>
<tr>
<td>h0E</td>
<td>T_FAST</td>
<td>Fast decay/fall step time</td>
</tr>
<tr>
<td>h0F</td>
<td>T\textsubscript{ON_MIN}</td>
<td>Minimum ON time</td>
</tr>
<tr>
<td>h18</td>
<td>CONFIG</td>
<td>Bit10-14 : TSW</td>
</tr>
</tbody>
</table>
dSPIN

Fast Decay Mode during Falling Step

<table>
<thead>
<tr>
<th>Address [Hex]</th>
<th>Register name</th>
<th>Register function</th>
</tr>
</thead>
<tbody>
<tr>
<td>h0E</td>
<td>T_FAST</td>
<td>Fast decay/fall step time</td>
</tr>
<tr>
<td>h0F</td>
<td>TON_MIN</td>
<td>Minimum ON time</td>
</tr>
<tr>
<td>h18</td>
<td>CONFIG</td>
<td>Bit10-14 : TSW</td>
</tr>
</tbody>
</table>

Fast decay for TOFF1 = T_FALL_STEP/4

Fast decay for TOFF2 = T_FALL_STEP/2

TON1 < TON_MIN

TON2 > TON_MIN

Fast decay for TOFF3 = last T_FALL_STEP
In our case
TOFF 2 = T_FALL_STEP/2

Auto-adjusted Decay

Current Level Target
dSPIN
Predictive current control

The predictive current algorithm allows to control the average current.

The OFF time is regulated according to the TSW parameter.

\[
T_{\text{pred}}(n) = \frac{T_{\text{on}}(n) + T_{\text{on}}(n-1)}{2} \quad \text{and} \quad T_{\text{pred}}(n+1) = \frac{T_{\text{on}}(n+1) + T_{\text{on}}(n)}{2}
\]
dSPIN
Programmable output slew-rate

Four output slew-rate values can be selected via SPI in order to fit the application EMI / Power dissipation tradeoff.
dSPIN
Integrated 3V voltage regulator

Device logic supply management is also flexible!
1. Supply IC logic through the internal 3V regulator
2. Supply IC logic using an external 3V3 supply
3. Supply external components (e.g. a µC) through the internal voltage regulator
dSPIN

Daisy chaining

μC SPI signals

μC

DEV 1

DEV 2

DEV N
**easySPIN - L6474**
Flexible innovative microstepping motor driver

- Operating voltage: 8 – 45V
- 7.0 A output peak current (3.0 A r.m.s.)
- Low RDSon power MOSFETs
- Programmable power MOS slew-rate
- Up to 1/16 microstepping
- Current control with adaptive decay
- SPI interface
  - Low quiescent and standby currents
  - Non dissipative current sensing
- Full set of Protection
  - Programmable non dissipative over current (on all power MOS)
  - Two levels over temperature protection
  - UVLO
**easySPIN - L6474**

**Speed Profiles using STCK**

<table>
<thead>
<tr>
<th>Speed</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum speed</td>
<td>from 15.25 to 15610 step/s (15.25 step/s resolution)</td>
</tr>
<tr>
<td>Minimum speed</td>
<td>from 0 to 976 step/s (0.24 step/s resolution)</td>
</tr>
<tr>
<td>Acceleration &amp; Deceleration</td>
<td>from 14.55 to 59590 step/s² (14.55 step/s² resolution)</td>
</tr>
</tbody>
</table>

**L6472**: Acceleration and Deceleration are linear

**L6474**: Any shape can be performed with the STCK pin*

---

*STCK pin: Synchronous Timing Counter Kit
L6470: dSPIN

ST motor Drivers are moving the future
dSPIN
L6470 Monolithic Digital μ stepping voltage mode Driver

• Supply voltage 8V – 45V
• 3Arms (7A peak)
• $R_{DS,ON} = 0.28 \, \Omega$
• Integrated Current Sensing (no external shunt)
• Up to 128 microsteps
• Voltage mode operation
• Sensorless Stall Detection
• Programmable speed profile
• Programmable positioning
• 8bit 5Mhz SPI interface (Daisy Chain compatible)
• Integrated 16MHz oscillator
• Integrated 5bit ADC
• Integrated 3V voltage regulator
• Over Current, Over Temperature and Under Voltage protections
• HTSSOP and POWERSO packages
Voltage mode vs. Current mode

Current mode principle:
- System tries to impose phase current applying a switching voltage.
  It is a **closed-loop** approach.

Voltage mode principle:
- System applies a sinusoidal voltage to motor and phase. Phase current is not directly controlled.
  It is a **open-loop** approach.
dSPIN
Voltage mode vs. Current mode

1. Abrupt current changes cause strong mechanical vibrations. Current mode tries to follow even non idealities (reference voltage quantization and sampling)
   Noisy and jerky motion.

2. Peak current is controlled. Average current value is different from target one.
   Inaccurate positioning

3. Non constant switching freq.
   Torque ripple and EMI are difficult to control.
dSPIN

Voltage mode vs. Current mode

Smooth current transient reduces mechanical vibrations.
Motor movement is soft and silent!

Average current is controlled.
Accurate positioning.

Constant switching freq.
Torque ripple and EMI are under control.
dSPIN
Voltage mode vs. Current mode

• Current mode systems strain with several tricks (e.g. mixed decay) trying to find a solution to follow adequately the sinusoidal profile of the current
  • Results are generally quite poor, require fine tuning and are trade-off between adequate profile and current ripple

• Voltage mode intrinsically uses the best decay style
  • Current profile is very smooth
  • No compromise on current ripple. No mixed decays
  • No tuning of the decays
  Best decay is always used with each motor
dSPIN
Voltage mode: drawbacks and solutions

- Back-Electro Motive Force heavily influences voltage to current relation
- Windings applied voltages are perturbed by supply voltage fluctuations
- Phase resistances vary with temperature

- Effective and flexible BEMF compensation system
- Supply voltage compensation though integrated 5bit ADC
- Phase resistance compensation register
dSPI\N
BEMF compensation

Without BEMF compensation

With BEMF compensation
According to motor conditions (acc/deceleration, constant speed, hold) a different torque, and then current, could be needed.

*dSPIN* logic switches from different compensation parameters sets according to motor status.

- Acceleration
- Deceleration
- Const. speed
- Hold (in hold conditions BEMF comp. is disabled)

Motor speed

![BEMF Compensation Algorithm](image)

- Amplitude
- Speed

Sinewave Amplitude
dSPIN

BEMF compensation waveform
dSPIN

Supply voltage compensation (1)
dSPIN
Supply voltage compensation

- 5bit ADC measures actual motor supply voltage
- Compensation algorithm calculates correction coefficient
- Compensation coefficient is applied to sinewave amplitude
- L6470
- ADC
- COMP
- PWM + H-Bridge
- $V_S + n(t)$
- $V_{OUT}$
- Sinewave Amplitude
Motor phase resistance increases during operation causing a phase current reduction and a torque loss.

Resistance variation can be compensated by a programmable KThERM coefficient (1 to 1.47).
dSPIN

Sensorless stall detection

Using integrated current sensing and the adjustable STALL current threshold a cheap and easy stall detection can be implemented.
dSPIN

L6470 : Many Commands

- Maximum speed: from 15.25 to 15610 step/s (15.25 step/s resolution)
- Minimum speed: from 0 to 976 step/s (0.24 step/s resolution)
- Acceleration & Deceleration: from 14.55 to 59590 step/s² (14.55 step/s² resolution)

**GoTo(Target) command:** reach the target position using shortest path

**GoTo_DIR(Target, DIR) command:** reach the target position moving the motor in the selected direction

**GoUntil** command moves the motor with a selected constant speed and stops the motor when the switch is closed; at that time one of the following actions can be taken:

And: GoHome, GoMark, Run, Move, SoftStop, HardStop, SoftHiz, ReleaseSW.....

**Integrated position registers allows to map up to 32768 full steps** (@128 µstep operation)

Equivalent to about 164 rotations (200 step/rotation motor)
## dSPIN Register map

<table>
<thead>
<tr>
<th>Address [Hex]</th>
<th>Register name</th>
<th>Register function</th>
<th>Len. [bit]</th>
<th>Reset Hex</th>
<th>Reset Value</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>h01</td>
<td>ABS_POS</td>
<td>Current Position</td>
<td>22</td>
<td>00000000</td>
<td>0</td>
<td>R, WS</td>
</tr>
<tr>
<td>h02</td>
<td>EL_POS</td>
<td>Electrical Position</td>
<td></td>
<td></td>
<td>0</td>
<td>R, WS</td>
</tr>
<tr>
<td>h03</td>
<td>MARK</td>
<td>Mark Position</td>
<td></td>
<td></td>
<td>0</td>
<td>R, WS, WR</td>
</tr>
<tr>
<td>h04</td>
<td>SPEED</td>
<td>Current Speed</td>
<td></td>
<td></td>
<td>0 step/tick (0 step/s)</td>
<td>R</td>
</tr>
<tr>
<td>h05</td>
<td>ACC</td>
<td>Acceleration</td>
<td></td>
<td></td>
<td>125.5e-12 step/tick² (2008 step/s²)</td>
<td>R, WS</td>
</tr>
<tr>
<td>h06</td>
<td>DEC</td>
<td>Deceleration</td>
<td></td>
<td></td>
<td>125.5e-12 step/tick² (2008 step/s²)</td>
<td>R, WS</td>
</tr>
<tr>
<td>h07</td>
<td>MAX_SPEED</td>
<td>Maximum Speed</td>
<td></td>
<td></td>
<td>248e-6 step/tick (3 step/s)</td>
<td>R, WS, WR</td>
</tr>
<tr>
<td>h08</td>
<td>MIN_SPEED</td>
<td>Minimum Speed</td>
<td></td>
<td></td>
<td>0 step/tick (0 step/s)</td>
<td>R, WS</td>
</tr>
<tr>
<td>h15</td>
<td>FS_SPD</td>
<td>Full Step Speed</td>
<td>10</td>
<td>00000000</td>
<td>150.7e-6 step/tick² (1.7 step/s)</td>
<td>R, WS, WR</td>
</tr>
<tr>
<td>h09(2)</td>
<td>KVAL_HOLD</td>
<td>Holding $K_{VAL}$</td>
<td></td>
<td></td>
<td>0.16·$V_S$</td>
<td>R, WS, WR</td>
</tr>
<tr>
<td>h0A(2)</td>
<td>KVAL_RUN</td>
<td>Constant Speed $K_{VAL}$</td>
<td>8</td>
<td>29</td>
<td>0.16·$V_S$</td>
<td>R, WS, WR</td>
</tr>
<tr>
<td>h0B(2)</td>
<td>KVAL_ACC</td>
<td>Acceleration Starting $K_{VAL}$</td>
<td>8</td>
<td>29</td>
<td>0.16·$V_S$</td>
<td>R, WS, WR</td>
</tr>
<tr>
<td>h0C(2)</td>
<td>KVAL_DEC</td>
<td>Deceleration Starting $K_{VAL}$</td>
<td>8</td>
<td>29</td>
<td>0.16·$V_S$</td>
<td>R, WS, WR</td>
</tr>
</tbody>
</table>

**Absolute position register can be set**

**Motor electrical position (current microstep) can be set**

**Speed profile parameters**

**Torque control parameters**
dSPIN - *easy*SPIN  
Tools & Documentations

- **Sales Codes**
  - L6470H - Tray
  - L6470HTR - Tape&Reel
  - ES available on L6470PD

- **Product Page** [http://www.st.com/dspin](http://www.st.com/dspin)
  - Data Sheet
  - Application Note (AN3103)
  - d SPIN Evaluation Tool Software
  - Evaluation Board: [EVAL6470H](http://www.st.com/dspin)
  - Control boards [STEVAL-PCC009V2](http://www.st.com/dspin) (and –V1)
  - d SPIN Firmware Library
dSPIN - easySPIN
Tools & Documentations

• Sales Codes
  • L6472H - Tray
  • L6472HTR - Tape & Reel
  • ES available on L6472PD

• Product Page http://www.st.com/dspin
  • Data Sheet
  • Application Note
  • d SPIN Evaluation Tool Software
  • Evaluation Board: EVAL6472H
  • Control boards STEVAL-PCC009V2 (and –V1)
  • d SPIN Firmware Library
    • Available on http://www.st.com/dspin
dSPIN - easySPIN
Tools & Documentations

- FLAG LED (Red)
- Application reference area
- Power supply connector (8 V - 45 V)
- SYNC output
- Master SPI connector
- JP1: VDD supply from master SPI connector
- JP3: Daisy chain termination
- Slave SPI connector
- JP4: STICK to slave SPI connector
- JP5: DIR to slave SPI connector
- OSCIN/OSCOUT connector
- Phase A connector
- Phase B connector
- ADCIN input regulation
- JP2: VDD to VREG connection