



life.augmented

STSPIN32G4 Integrated Motor Controller

Ryan Satterlee

Motor Controls Application Engineer

STMicroelectronics



Agenda

1

Introduction on Motors and Controls

2

Controls and Sensing

3

STSPIN32 Hardware

4

ST's Motor Control Software Dev Kit



Motors

Quick overview



Brushed DC

- Mechanical contact via brushes
- Very easy to drive with DC
- Long term wear/reliability concerns
- Full bridge circuit typically used



Stepper

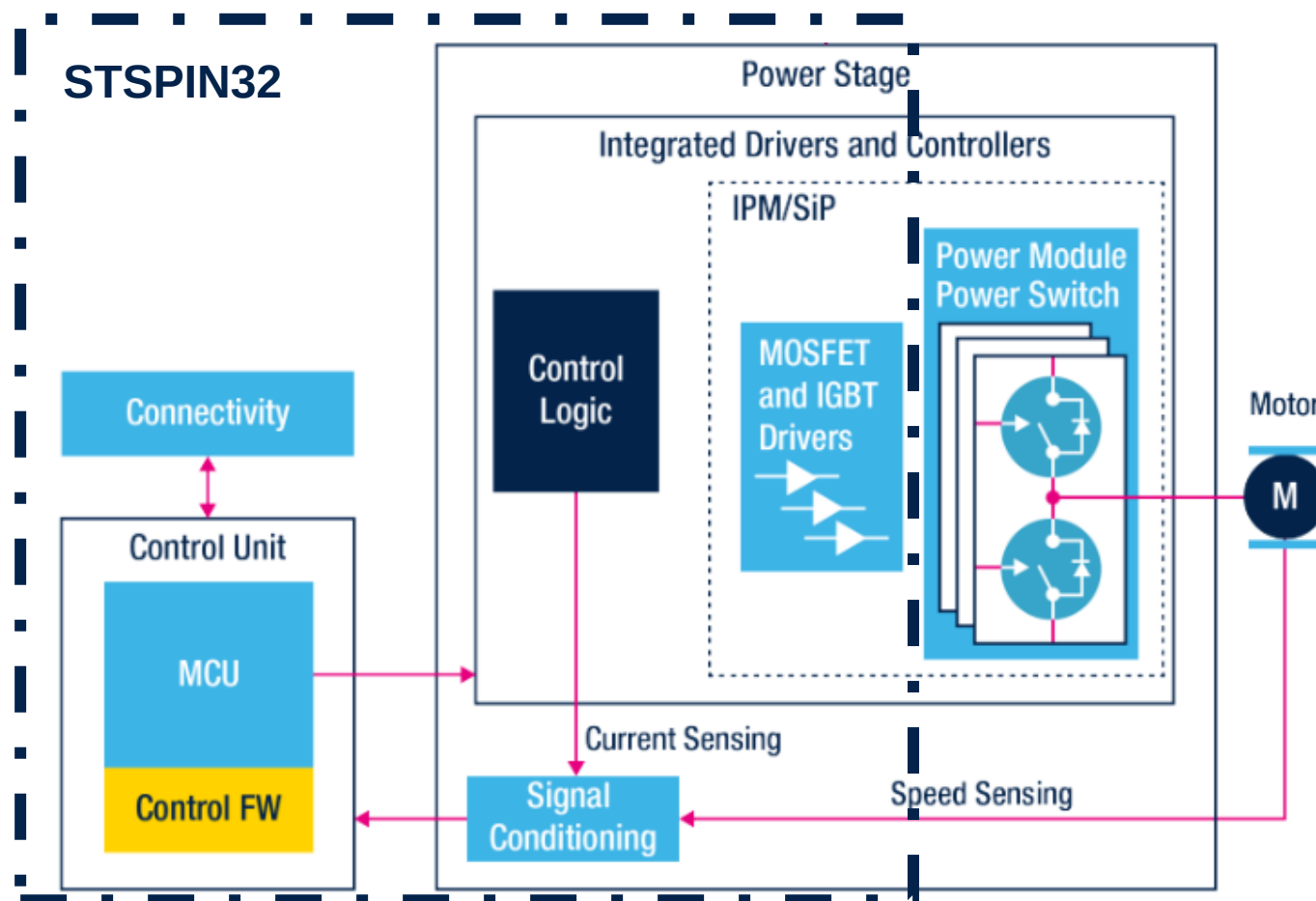
- 2 phase typical, lower speed
- Used in positioning applications
- Heavier and lower torque
- Dual full bridge driver



BLDC

- 3-phase coil stator and PM rotor
- Highest power/weight ratio
- Triple half bridge driver
- Higher processing demand on controller

High Level Motor Controls Overview for BLDC or PMSM



Motor Controls and Sensing

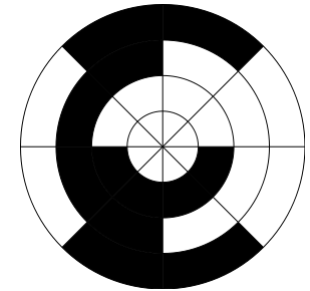
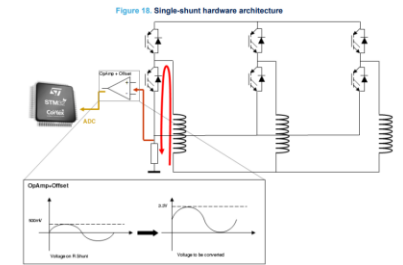


Motor Sensing

Typical ways to sense motor speed/position

Sensorless - via current sensor(s) and/or BEMF

- Requires a signal from the motor. Works very well for mid to high speed
- ST's FOC include 1-shunt, 3-shunt, or 2 Isolated Current Sensors (ICS)
- 6-step control typically uses BEMF voltage sensing from the phase lines

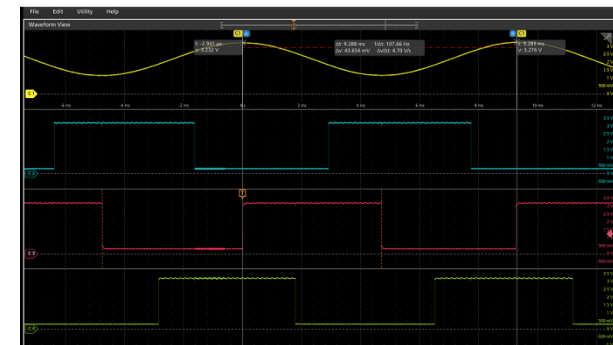


Encoders – via separate device external to the motor

- These give the highest resolution for position/angular control
- Priciest option and durability concerns with some setups

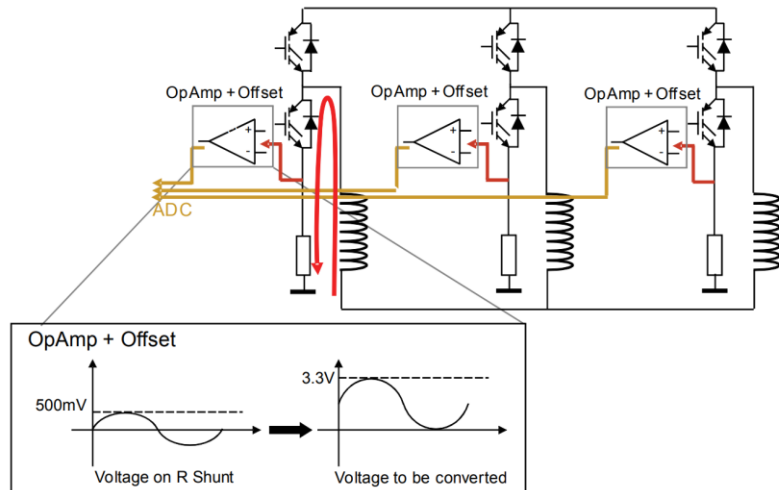
Hall-Effect - via sensors in motor for additional feedback

- Typically used for lower speed control alongside above sensorless method
- Reasonably cost effective, limits in resolution at very low speeds.



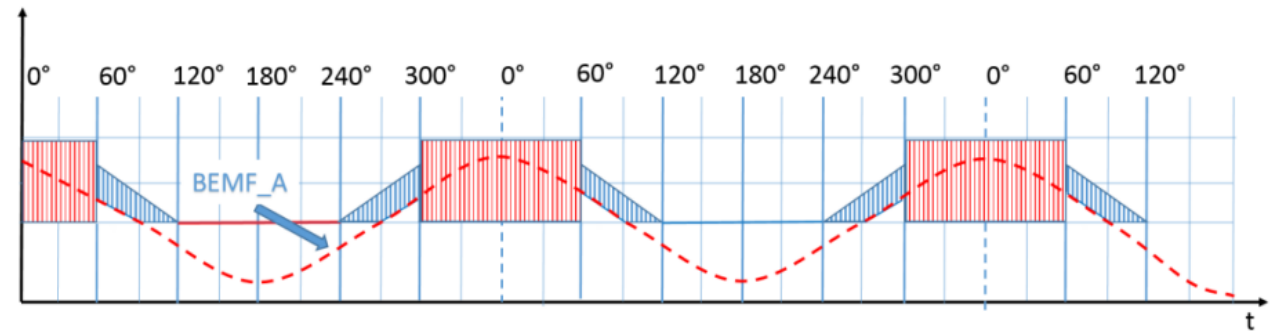
FOC Topologies

- 1-shunt, 3-shunt or ICS
- Hall or Sensorless
- Encoder
- Sinusoidal Commutation



6-Step Topologies

- Hall Sensing
- Sensorless Back EMF via Phase voltage divider
- Block/Trapezoidal Commutation

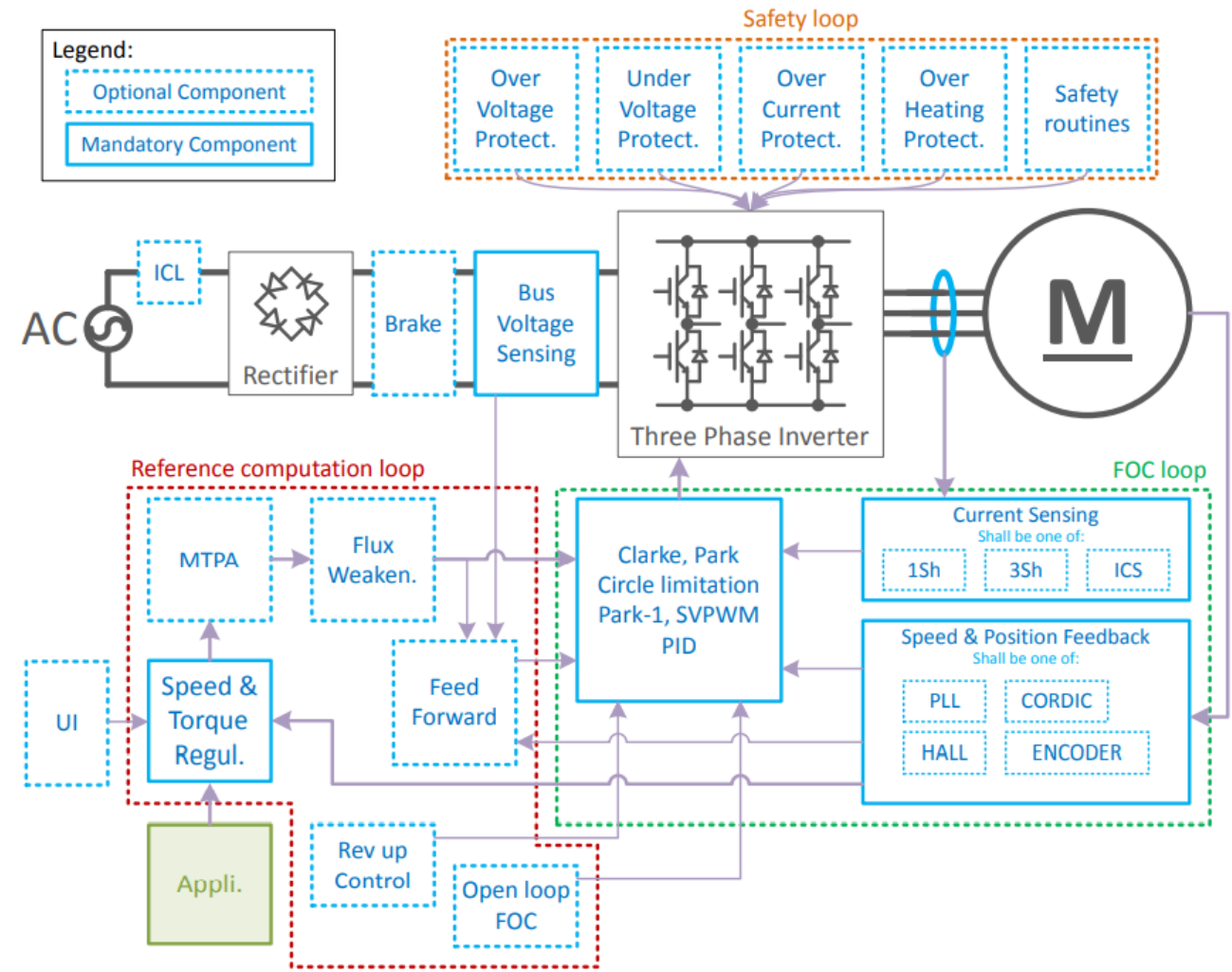
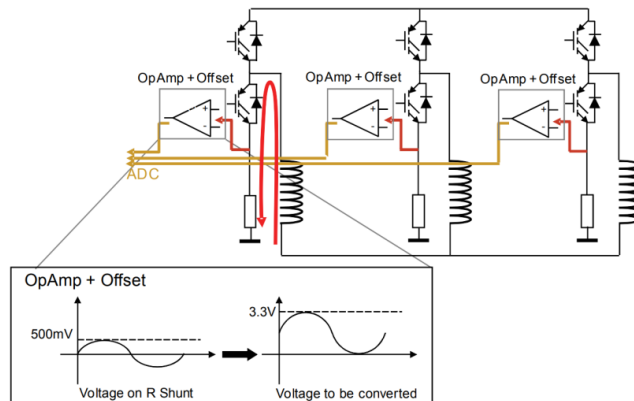




FOC System Overview

FOC Features

- Sensorless or Sensored Control Options
 - 1-shunt, 3-shunt or ICS
 - Hall Effect Sensing
 - Encoder
- High efficiency operation
- Quite drive capability
- Reduced torque ripple
- In-depth HW/SW protection
- Simplified hardware arch
- Dynamic system compensation
- Typically works best w/Sinusoidal BEMF
- Leverage STM32 Peripherals

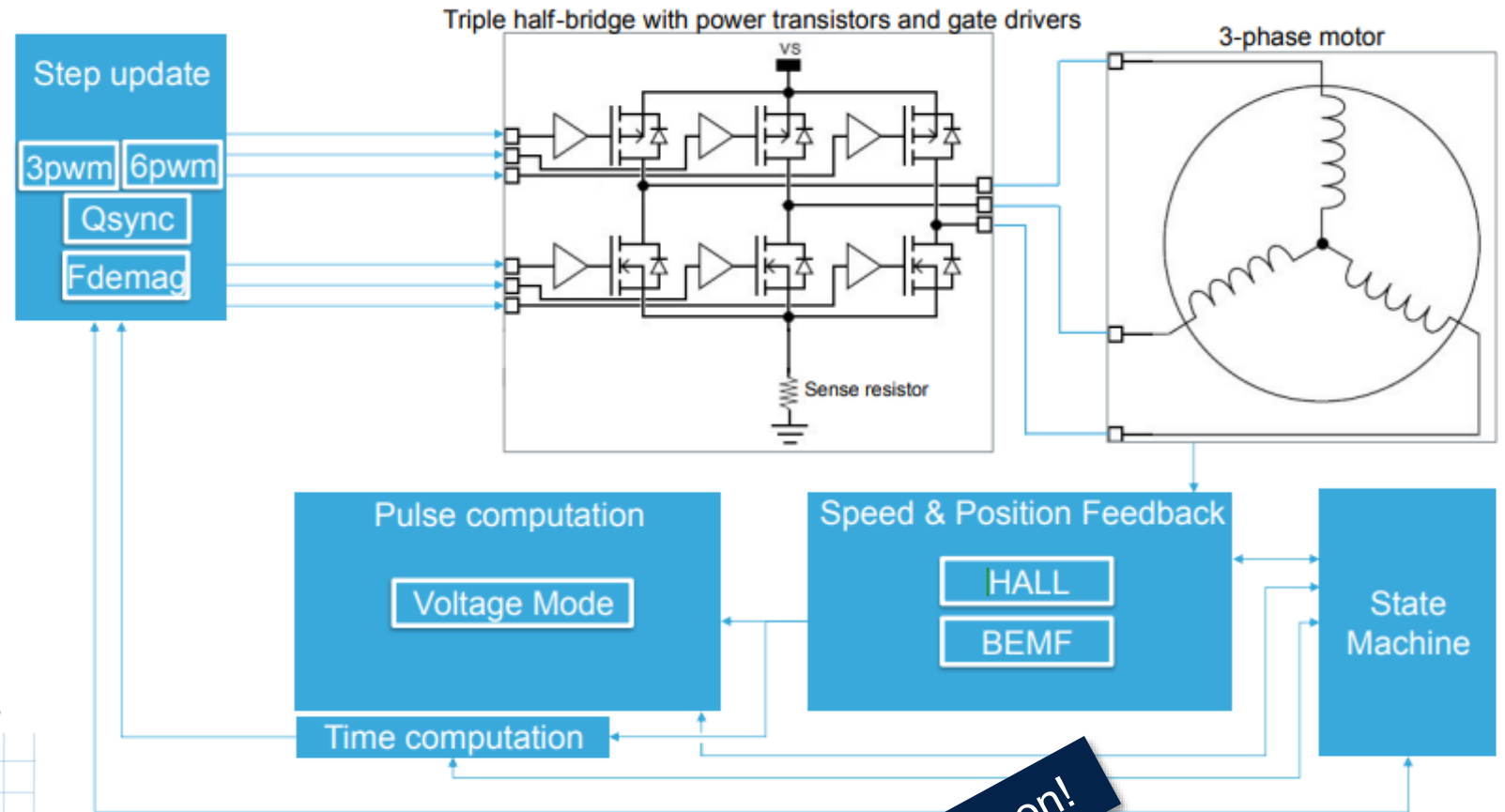




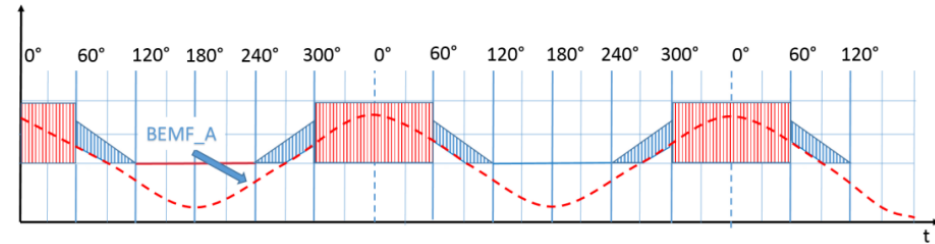
6-Step System Overview

6-Step Features

- Sensorless or Sensored Control Options
 - Hall Effect Sensing
 - Sensorless Back EMF
- Efficient operation
- Quite drive capability
- Lower Memory footprint
- Lower CPU Utilization
- HW/SW protection
- Simplified hardware architecture
- Typically works best w/Trapezoid BEMF
- Leverage STM32 Peripherals



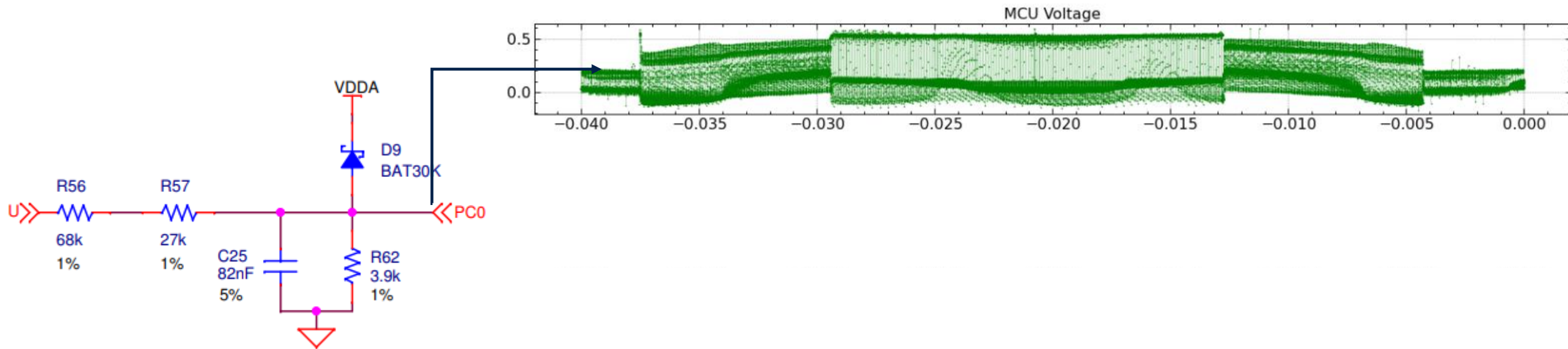
Coming soon!





BEMF Sensing

Sensorless 6-Step Back EMF Sensing for commutation control



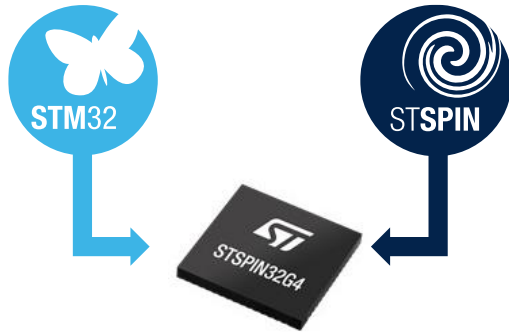


STPIN32G4



STSPIN32 Series Overview

3-Phase BLDC Drivers with Embedded STM32 MCU



Key Applications

Industrial Controls
Robotics
Home Appliances
Power Tools
Drones
Fans and Pumps
Vacuum Cleaners

High Voltage

STSPIN32F0251/2 up to 250 V
STM32 Cortex M0 + 250 V 3-phase Driver
STSPIN32F0601/2 up to 600 V
STM32 Cortex M0 + 600 V 3-phase Driver

Higher computational power

STSPIN32G4 up to 75V
STM32 Cortex M4 + 1A 3-phase Driver

Low Voltage

STSPIN32F0/A/B- up to 45 V
STM32 Cortex M0 + 45 V 3-phase Driver



STSPIN32G4 vs STSPIN32F0 Family

Controls Comparison

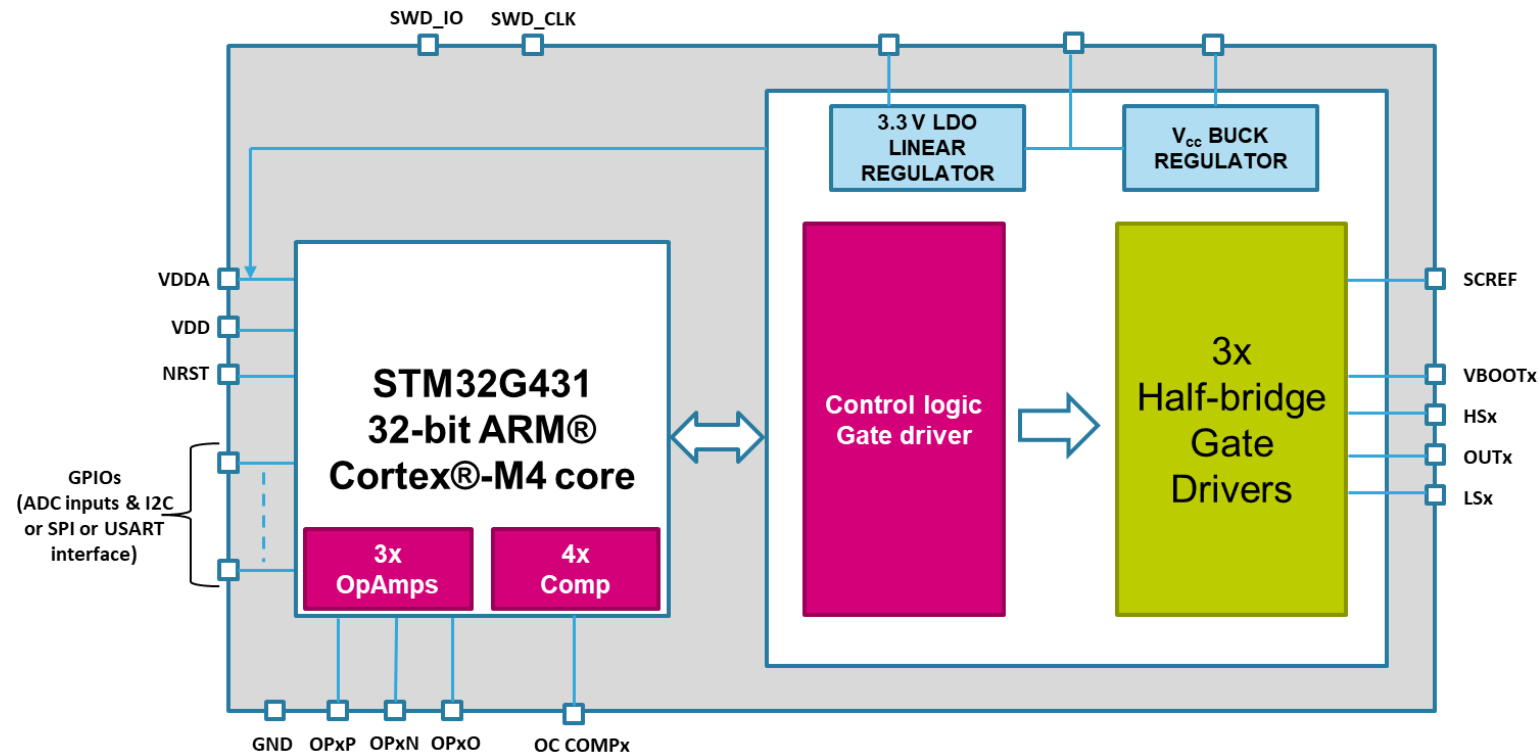
	STSPIN32F0A	STSPIN32F0B	STSPIN32G4
Microcontroller	STM32F031	STM32F031	STM32G431
ARM Core	Cortex M0	Cortex M0	Cortex M4 FPU + ART+ Math Coprocessor
Proc Speed	48 MHz	48 MHz	170 MHz
RAM	4 Kb	4 Kb	32 Kb
Flash	32 Kb	32 Kb	128 Kb
GPIO	16 ⁽¹⁾	20 ⁽¹⁾	40 ⁽¹⁾
ADC	1 x	1 x	2 x
OPAMPs	3 x	1 x	3 x
Comparator	1 x	1 x	4 x

⁽¹⁾ Motor control driving signals not included



STSPIN32G4 Overview

Advanced 75V 3-phase BLDC Driver with embedded Cortex M4 STM32

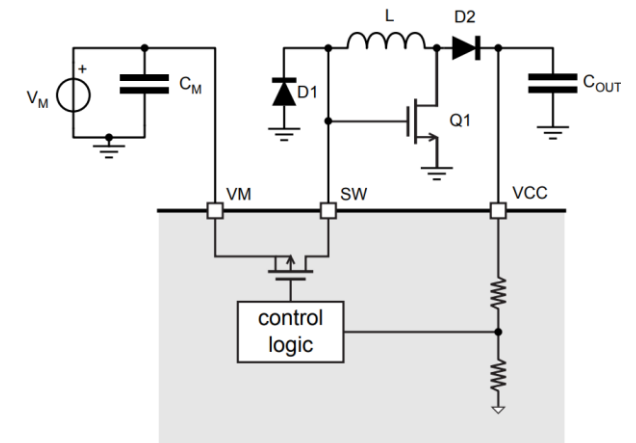
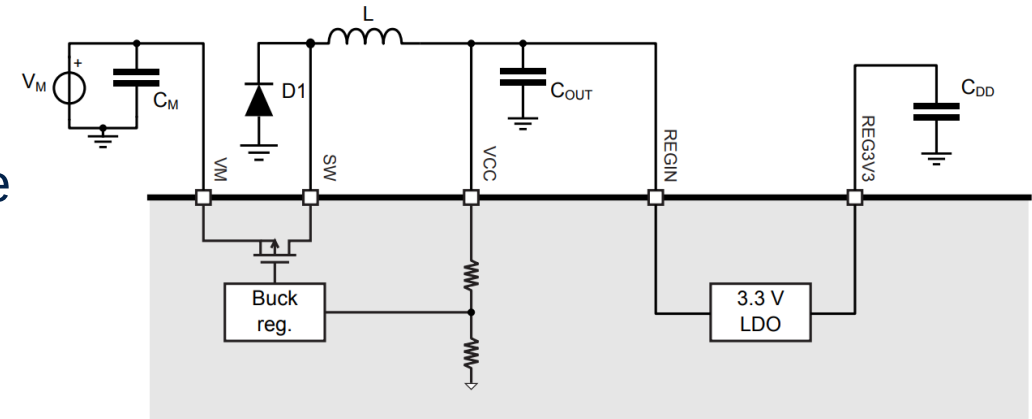




Power Management

75V Buck/Boost → Gate Driver voltage → Logic Voltage

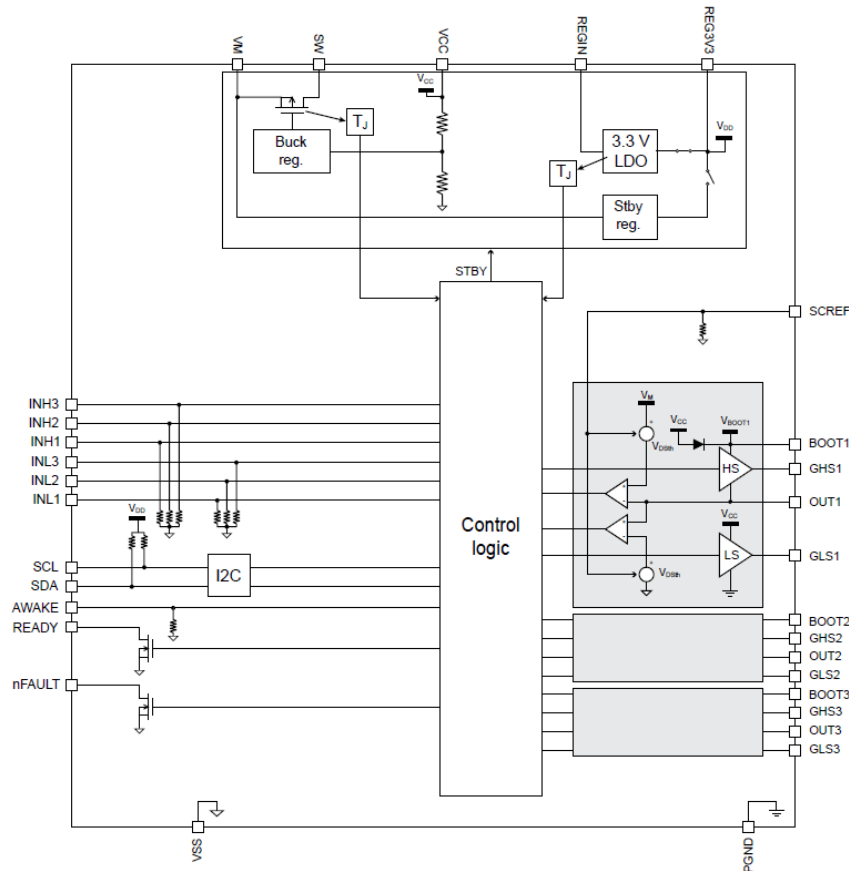
- Up to **75V** can be fed to V_m (**System Bus**) pin
- An integrated Buck converter will automatically regulate the V_{cc} with supplied Inductor/Cap/Diode
- V_{cc} (**Gate Driver Bus**) can be programmed to be regulated from **8→15V**
- That V_{cc} voltage can be fed back into an integrated LDO for **3.3V (Logic Bus)**
- With an additional Diode/MOSFET, Boost is possible as well for low voltage applications





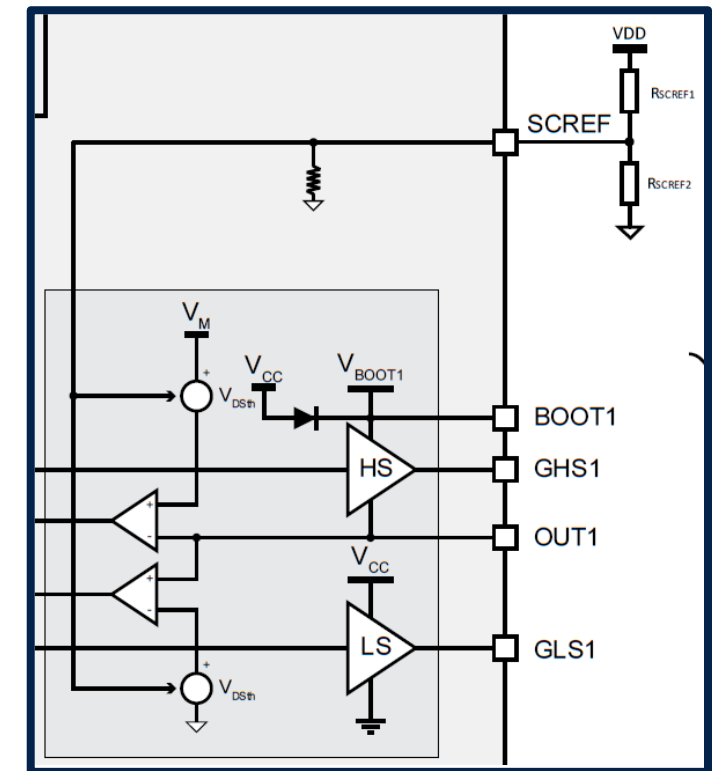
STSPIN32G4 Gate Drivers

3 phase Gate Driver Characteristics



Triple half-bridge driver:

- I_{GATE} sink/source **1A**
- Bootstrap Diodes embedded
- Cross Conduction prevention: interlocking feature
- UVLO & OTP with Fault feedback
- In-Out prop delay= 120ns
- **V_{DS} monitoring :**
Reference Voltage (0.2V-2.5V) for embedded monitoring of external MOS V_{DS} : $V_{DSth} = V_{SCREF}$
- I²C config and status monitoring
- Fault, Wake and Ready pins





Programmable Gate Driver

Register Configurations via I²C

Register	Bit Name	Description
POWMNG	REG3V3_DIS	Enable 3.3V linear regulator
	VCC_DIS	Enable V _{cc} buck regulator
	STBY_REG_EN	Enable standby linear regulator
	VCC_VAL	V _{cc} output: 8V, 10V, 12V, 15V

Power Management

Gate Driver

Ready/Signaling

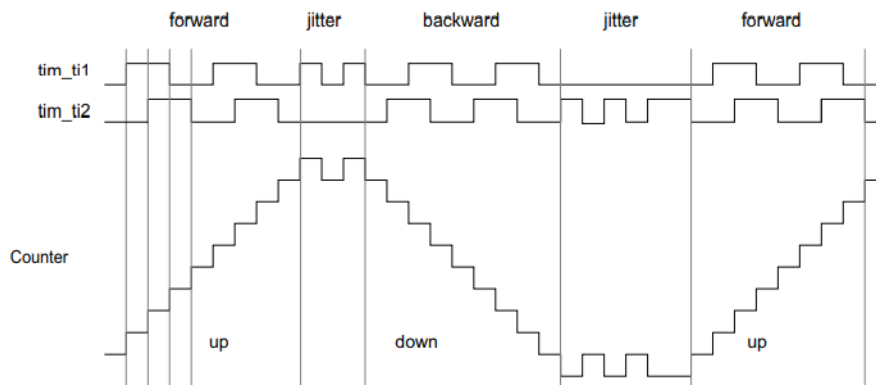
Low power/Status



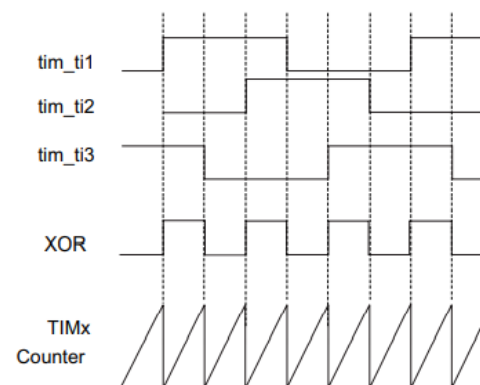
ST's Advanced Timers are designed for optimized Motor Control

- Dual motor control possible with 2 Advanced Timers
- Motor control specific timer features
 - 3x Complementary PWMs to drive high and low side of each half bridge
 - TIM XOR for hall effect speed/interrupt capture
 - Deadtime insertion, Block commutation

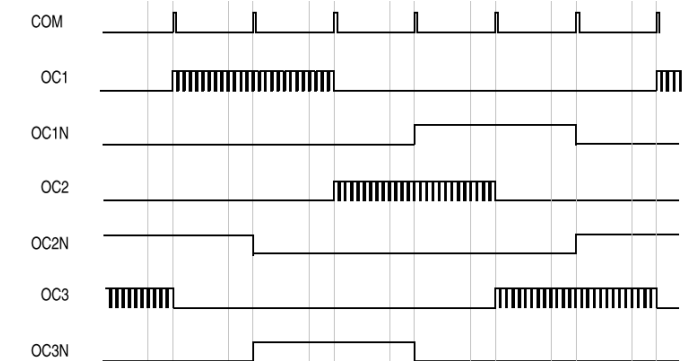
Encoder Signal Handling



Hall Signal Handling



PWM Commutation

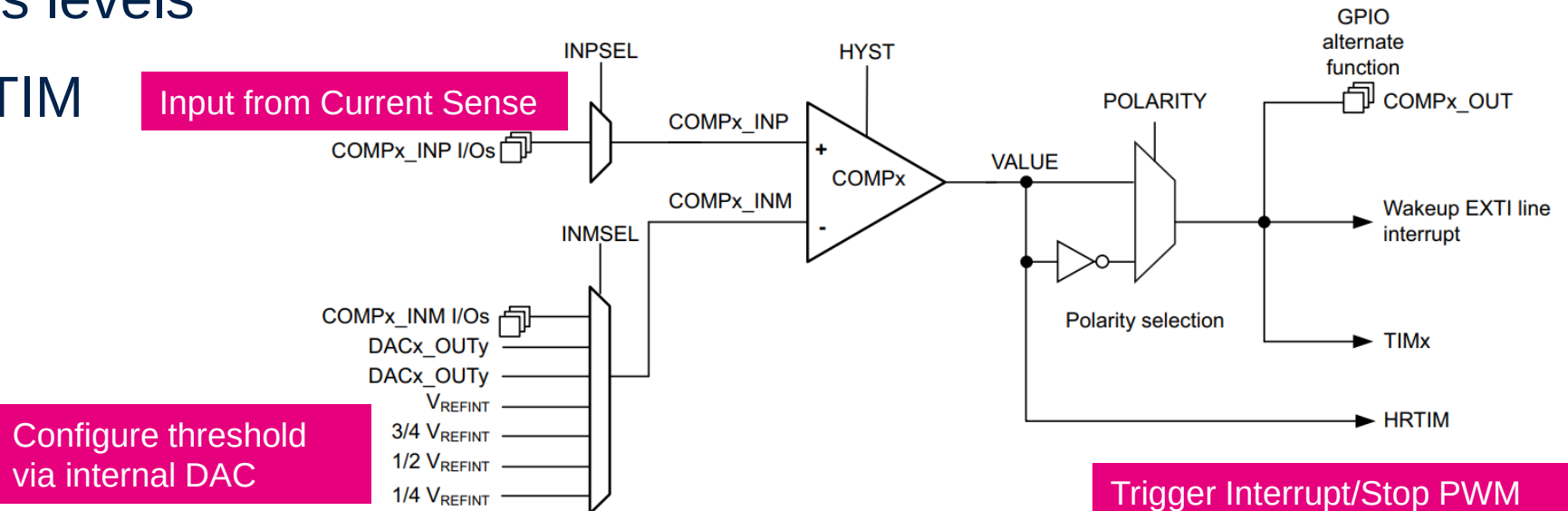
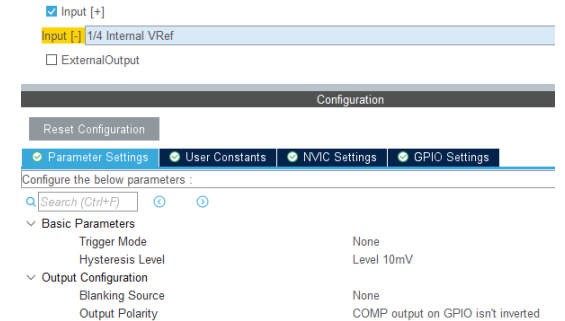




Over-Current Protection

Hardware based OCP with STM32G4 Comparator Peripheral

- Compare input vs Internal, External or DAC reference
- Output can trigger TIM Break to shutdown PWM to Motor w/o Firmware intervention
- Configurable Hysteresis levels
- Blanking via separate TIM
- Trigger configurability





Fully Integrated MCU, Power, and Gate Driver

Arm Cortex-M4

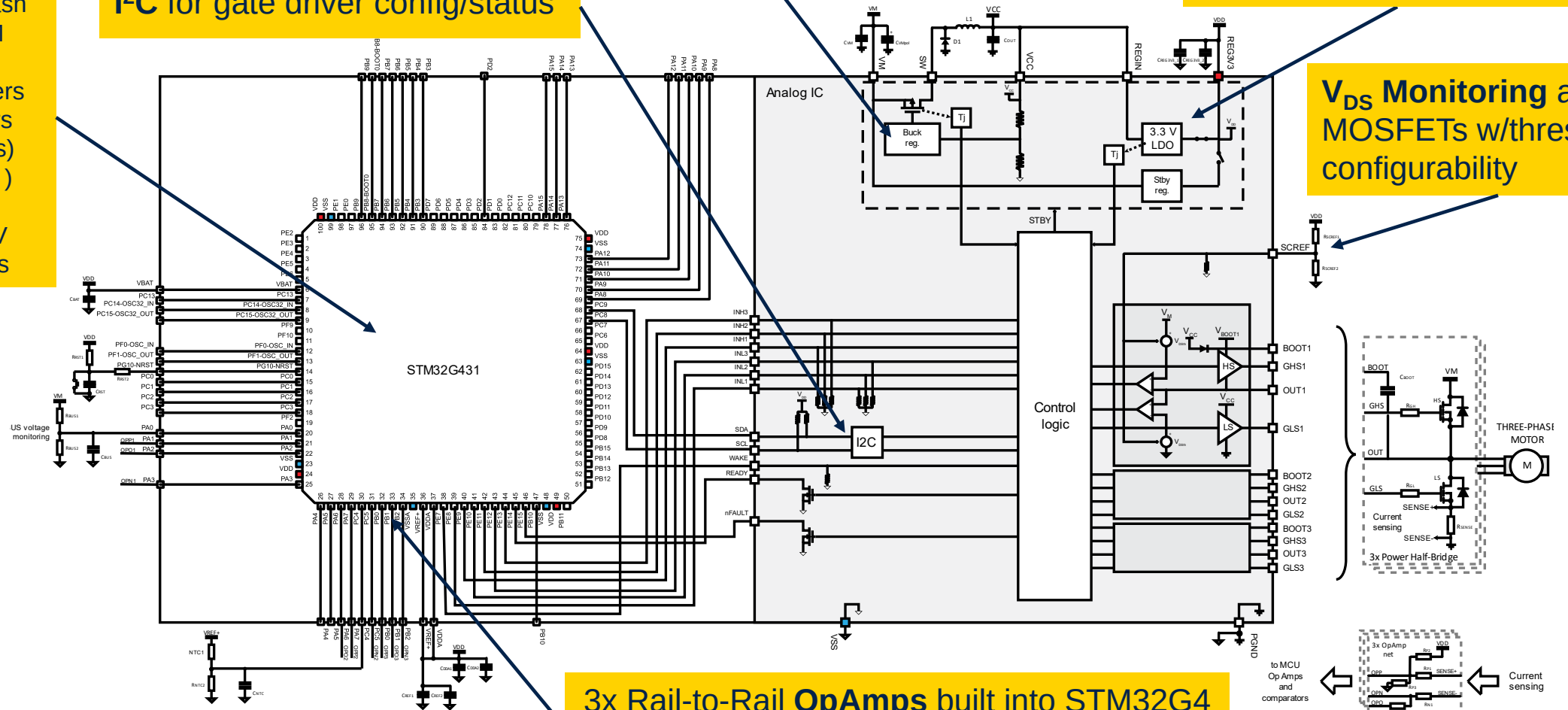
128 Kbytes of Flash
32 Kbytes RAM
170 MHz CPU
2x advanced timers
8x general timers
2x ADCs (12-bits)
4x DACs (12-bit)
Full set of I/Fs
 $V_{DD/DDA}$ 1.7-3.6V
HW accelerators

I²C for gate driver config/status

VCC Buck Converter
Up to 200 mA

3.3 V LDO Linear Regulator
Up to 150 mA

V_{DS} Monitoring across
MOSFETs w/threshold
configurability



3x Rail-to-Rail OpAmps built into STM32G4
15 MHz bandwidth, Rail-to-rail input/output, PGA



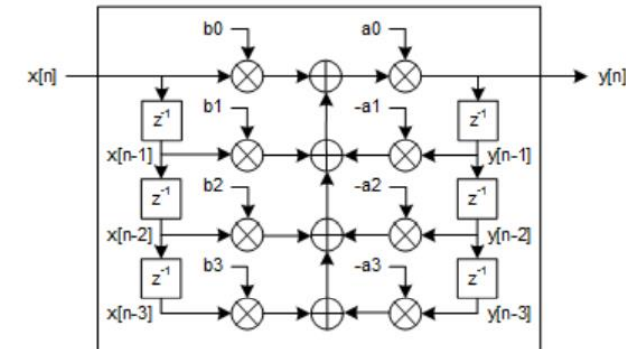
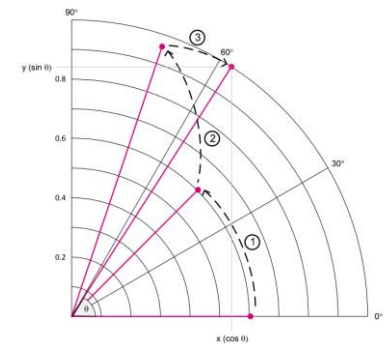
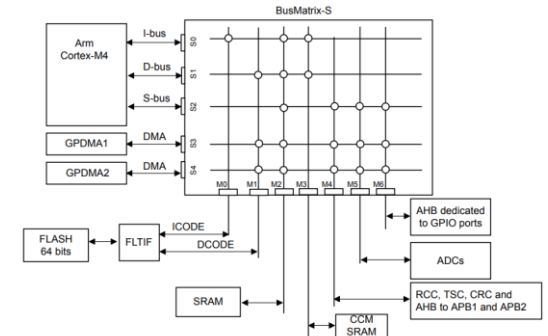
life.augmented



Performance Gains

M4 Based STM32G4 Gives Enhanced Motor Control Performance

- FOC Switching frequency with F0 (~14kHz) vs G4 (>40kHz)
- Core Coupled Memory (**CCM**) SRAM enables accelerating computationally expensive code (eg. FOC)
- Coordinate Rotational Digital Computer (**CORDIC**) Accelerator for trig functions
- Filter Math Accelerator (**FMAC**) for FIR and IIR filtering
- Dual Motor Control with 2 advance timers
- Low power and sleep modes, 15uA standby + MCU current



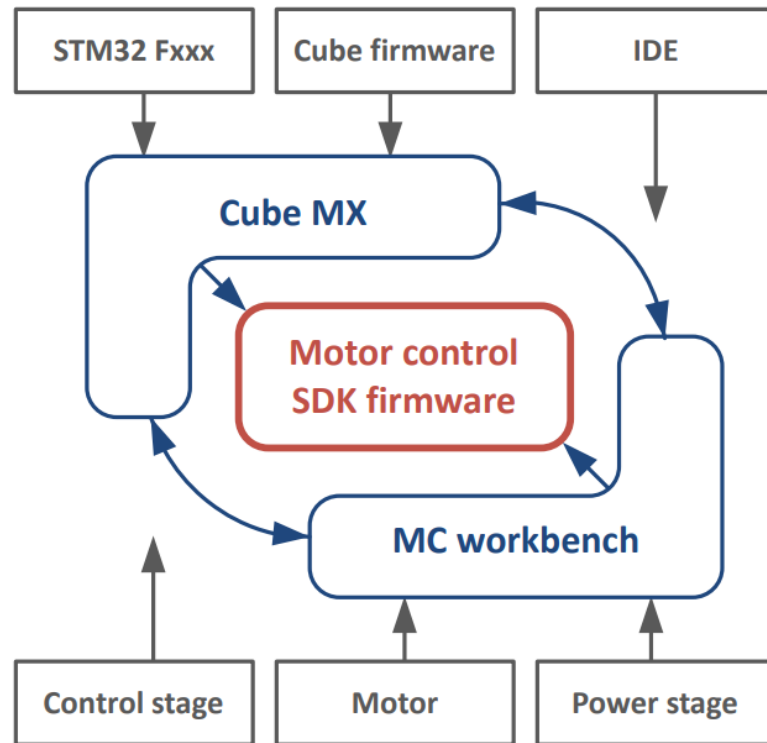


ST's Motor Control SDK



Motor Control SDK

Automated Code Generation, Highly Customizable



Field Oriented Control or 6-step

Automated Code Generation

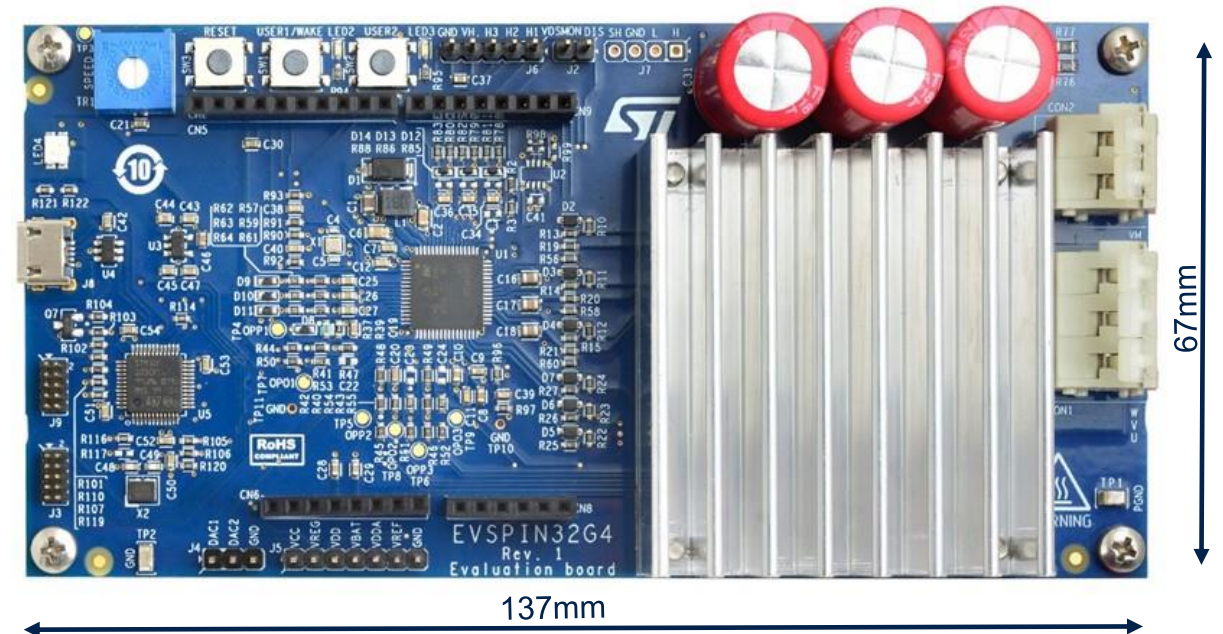
Live GUI for tuning and configuration



EVSPIN32G4 Devboard

STSPIN32G4 Evaluation Board Key Features

- Up to **75 V** and output current up to **20 A_{rms}**
 - Optional heatsink to increase power dissipation
- Three-shunt or single-shunt topology
- Embedded OPAMPs used for current sensing
- BEMF sensing circuitry for sensorless driving
- Digital Hall sensors input management
- PCB temperature monitoring (NTC)
- Bus voltage monitoring
- Arduino UNO R3 expansion connector
- CAN bus mounting option
- Embedded STLINK V2-1 programmer



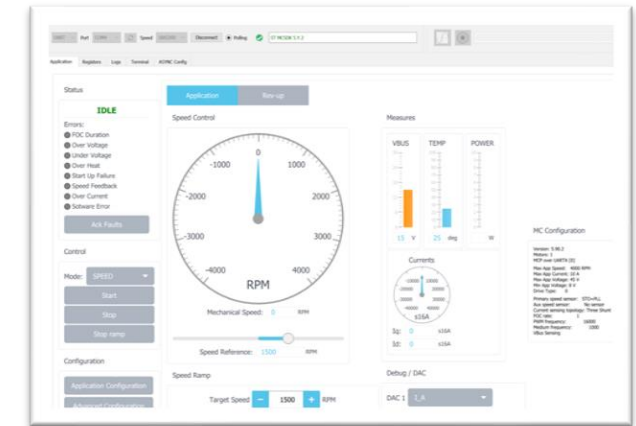
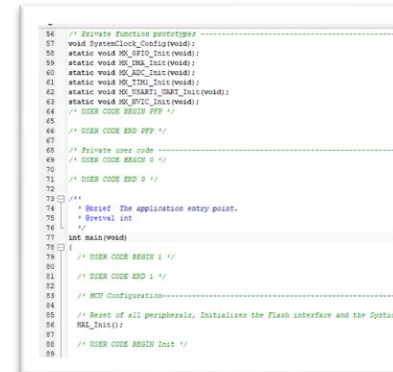
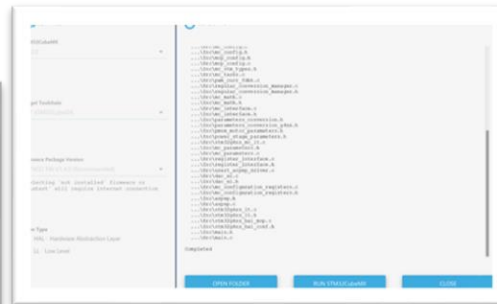
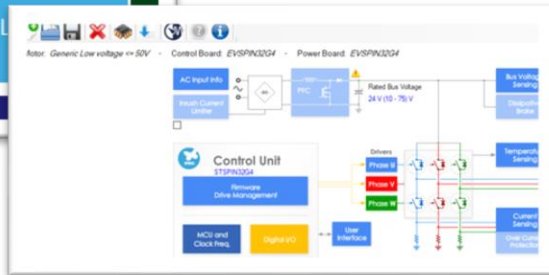
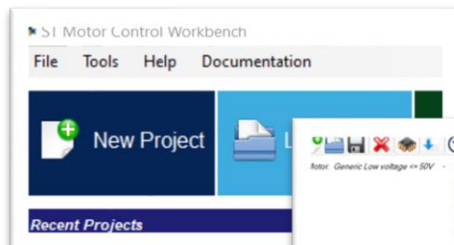
Suitable for FOC and 6-Step control



Application Design Flow

Configurability and tuning built on Cube Mx

- Supported by ST's Motor Control SDK
- Configure hardware and motor tuning via the MCWB
- Code generation and HW configuration via CubeMX
- Compile with STM32Cube/Keil/IAR/GCC
- Live tune and monitor with Motor Control Pilot



Create Project

Configure to App

Generate Code

Compile Project

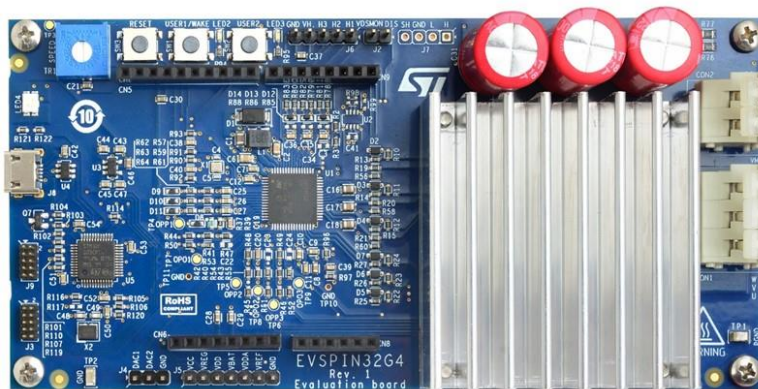
Monitor and Tune



Getting Started

Generating a new project for the STSPIN32G4 with the MC-SDK

- **STSPIN32G4 is available only in version 5.Y**
- In new project dialog window, under inverters, select EVSPIN32G4
- Support in version 6.1 coming soon!



1 Application type: Custom

2 System: Single Motor

3 Select Board: ☒ Inverter ☐ MC Kit ☐ Power & Control

Control: custom board

Power: custom board

New Project

1 Application type: Custom

2 System: Single Motor

3 Select Boards: ☒ Inverter ☐ MC Kit ☐ Power & Control

Inverter

custom board

B-G431B-ESC1

EVSPIN32F0251S1

EVSPIN32F02Q1S1

EVSPIN32F0601S1

EVSPIN32F0601S3

EVSPIN32F0602S1

EVSPIN32F06Q1S1

EVSPIN32F06Q1S3

EVSPIN32F06Q2S1

EVSPIN32G4

STEVAL-ESC001V1

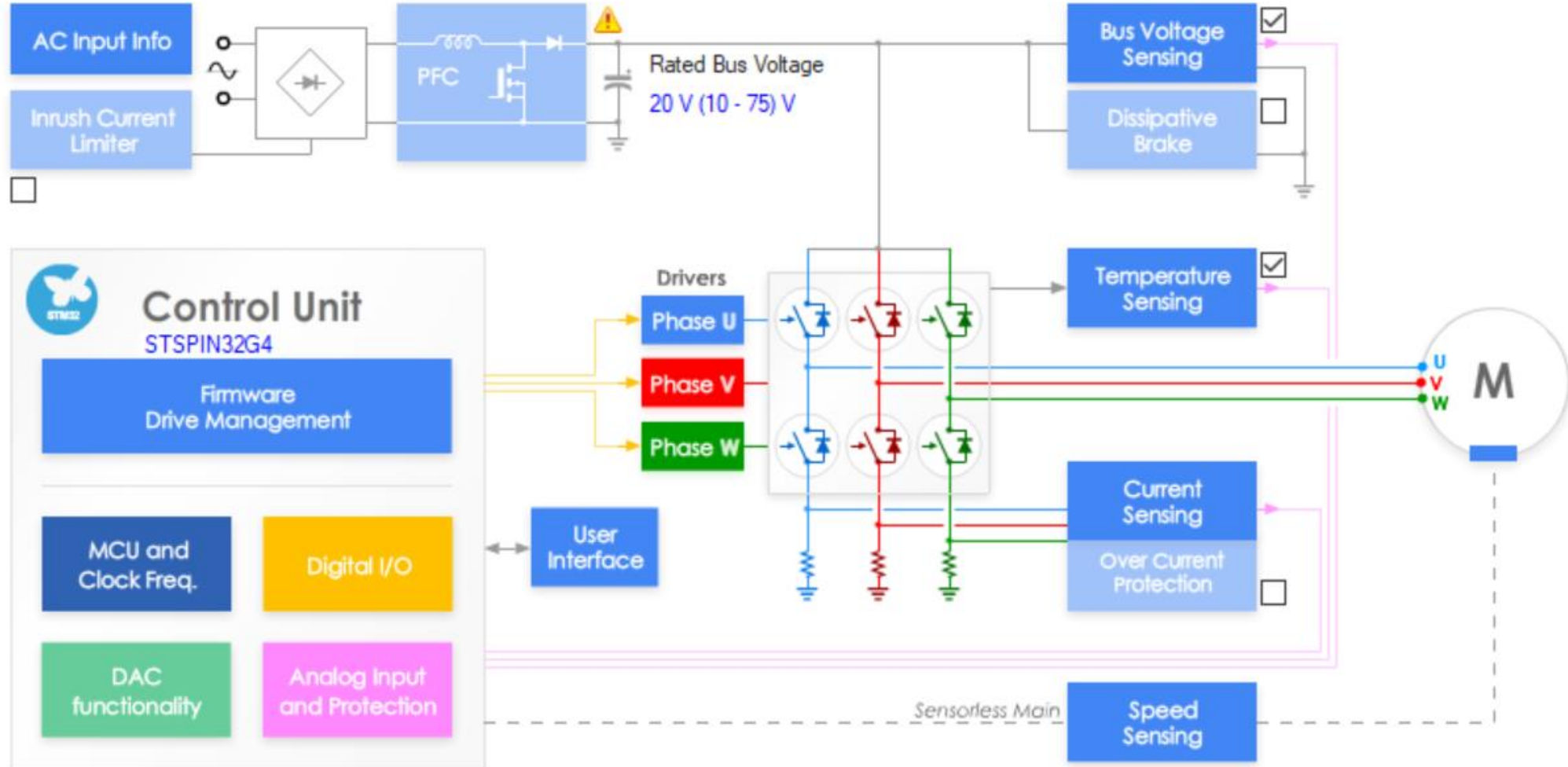
STEVAL-HK001V2

STEVAL-SPIN3201

STEVAL-SPIN3202

X-NUCLEO-IHM16+NUCLEO-F303RE

Motor Control Workbench 5.y





Setting Motor Parameters

Characterizing your Motor

- Pole Pairs
- Max Application Speed
- Nominal Current
- Nominal DC Voltage
- R_s (Stator Resistance)
- L_s (Stator Inductance)
- B-EMF constant
- Surface-Mounted or Internal Permanent Magnet Synchronous Motor

Electrical Model

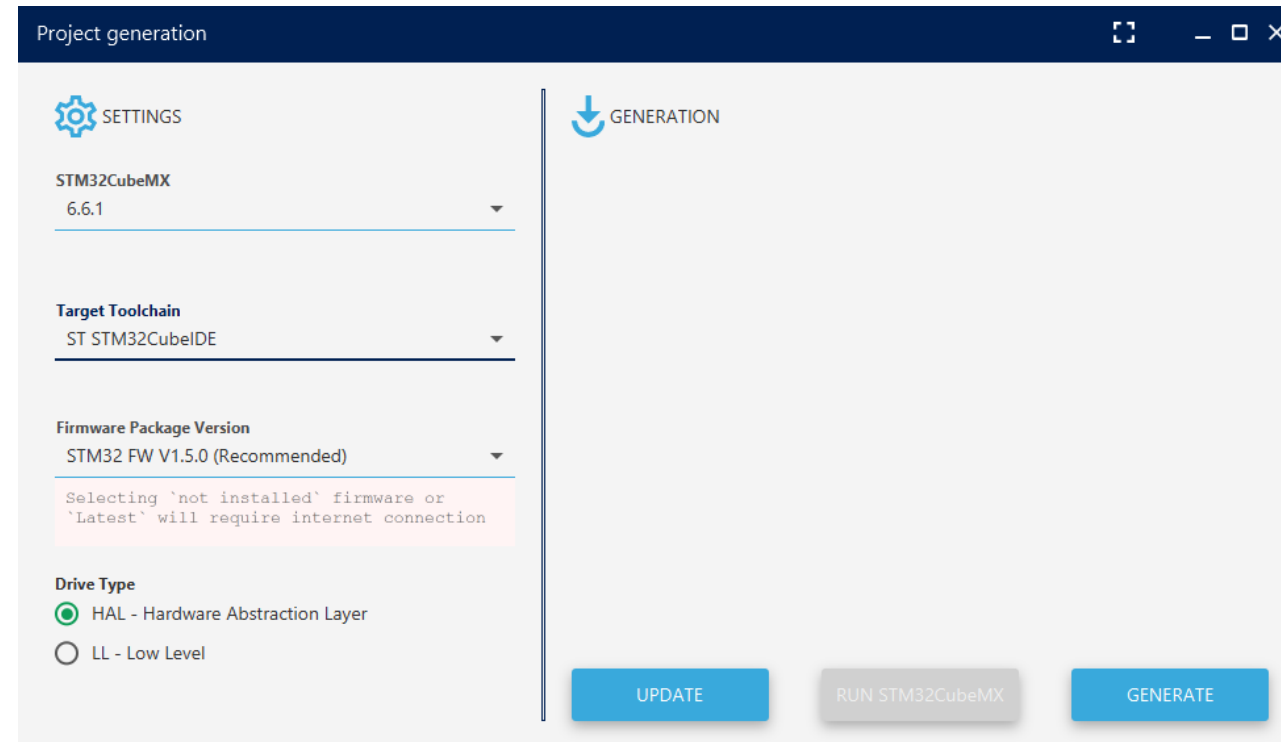




Project Generation

MC-SDK leverages CubeMx for Code Generation

- Can target many toolchains
 - IAR
 - Keil
 - STM32CubeIDE
- Select CubeMx and FW version
- Low level or HAL based code gen







MC-SDK Firmware API Overview

Quick run down of SDK firmware usage

- Core API to get started
- Control/monitor higher level parameters
- All functions defined as weak and can be overridden as needed
- Many examples included in installed directory of SDK

```
MC_API_H
MC_StartMotor1(void) declaration
MC_StopMotor1(void) declaration
MC_ProgramSpeedRampMotor1(int16_t, uint16_t) declaration
MC_ProgramSpeedRampMotor1_F(float, uint16_t) declaration
MC_ProgramTorqueRampMotor1(int16_t, uint16_t) declaration
MC_ProgramTorqueRampMotor1_F(float, uint16_t) declaration
MC_SetCurrentReferenceMotor1(qd_t) declaration
MC_SetCurrentReferenceMotor1_F(qd_f_t) declaration
MC_GetCommandStateMotor1(void) declaration
MC_StopSpeedRampMotor1(void) declaration
MC_StopRampMotor1(void) declaration
MC_HasRampCompletedMotor1(void) declaration
MC_GetMecSpeedReferenceMotor1(void) declaration
MC_GetMecSpeedReferenceMotor1_F(void) declaration
MC_GetMecSpeedAverageMotor1(void) declaration
MC_GetAverageMecSpeedMotor1_F(void) declaration
MC_GetLastRampFinalSpeedMotor1(void) declaration
MC_GetLastRampFinalSpeedMotor1_F(void) declaration
MC_GetControlModeMotor1(void) declaration
MC_GetImposedDirectionMotor1(void) declaration
MC_GetSpeedSensorReliabilityMotor1(void) declaration
```

```
MC_GetPhaseCurrentAmplitudeMotor1(void) declaration
MC_GetPhaseVoltageAmplitudeMotor1(void) declaration
MC_GetIabMotor1(void) declaration
MC_GetIabMotor1_F(void) declaration
MC_GetIalambdaMotor1(void) declaration
MC_GetIqdMotor1(void) declaration
MC_GetIqdMotor1_F(void) declaration
MC_GetIqdfMotor1(void) declaration
MC_GetIqdfMotor1_F(void) declaration
MC_GetVqdMotor1(void) declaration
MC_GetValphaMotor1(void) declaration
MC_GetElAngledppMotor1(void) declaration
MC_GetTerefMotor1(void) declaration
MC_GetTerefMotor1_F(void) declaration
MC_Clear_IqdfMotor1(void) declaration
MC_SetPolarizationOffsetsMotor1(PolarizationOffsets_t *) declaration
MC_GetPolarizationOffsetsMotor1(PolarizationOffsets_t *) declaration
MC_StartPolarizationOffsetsMeasurementMotor1(void) declaration
MC_AcknowledgeFaultMotor1(void) declaration
MC_GetOccurredFaultsMotor1(void) declaration
MC_GetCurrentFaultsMotor1(void) declaration
MC_GetSTMStateMotor1(void) declaration
MC_GetAveragePowerMotor1(void) declaration
MC_GetAveragePowerMotor1_F(void) declaration
```



Motor Control Pilot

Modern GUI for Live Application Testing and Tuning



- High resolution plotter/coms
 - Via UART interface @1.843Mbaud
 - No longer dependent on DAC/Scope for real-time monitoring
- QML based UI allows for tweaking, building, and loading custom interfaces
- Data Logging and updated Motor Control Protocol
- Separate application from workbench to allow for faster updates



STSPIN32G4 Triple Motor Drive Demo

Multi-Motor Control Drive for Vacuum Robot



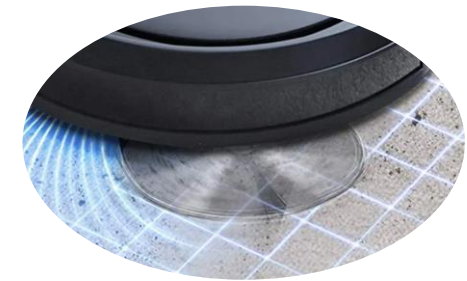
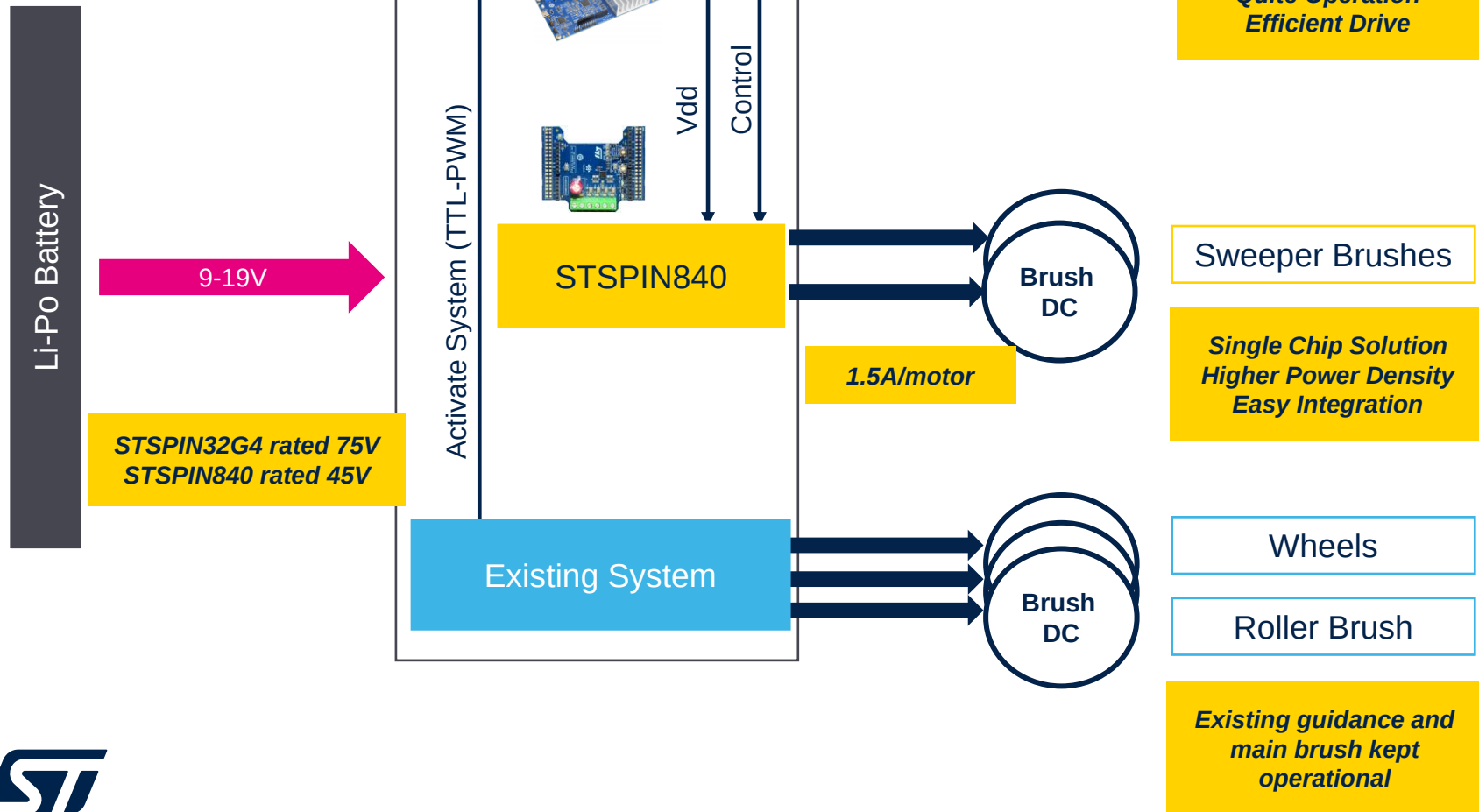
Retrofit an existing device with an STSPIN32G4

Brushed DC Motor Driver added STSPIN840

Configurable drive capabilities via MC Pilot



Updated Architecture





STSPIN32G4 Motor Controls: Key Takeaways

- **STSPIN32G4** offers a compact, integrated and economical solution
- Leverages the large existing STM32 ecosystem and toolset
- Able to take on a lead role in system design and control with more resources
- Wide operating voltage range and minimal external circuitry required with integrated buck and LDO regulator
- Higher current gate drive capability facilitating high power applications
- Automated code generation using the MC-SDK
- Extended performance potential from the STM32G4 processing accelerators
- FOC or 6-step control algorithms to handle application specific conditions

Our technology starts with You

© STMicroelectronics - All rights reserved.

ST logo is a trademark or a registered trademark of STMicroelectronics International NV or its affiliates in the EU and/or other countries.

For additional information about ST trademarks, please refer to www.st.com/trademarks.

All other product or service names are the property of their respective owners.



life.augmented