70V Smart Gate Driver with Servo (FOC) Controller in HW and Buck Converter

TMC9660

General Description

The TMC9660 is a highly integrated monolithic gate driver and motor controller IC with buck converter.

It includes a smart gate driver, a high-performance motion controller with hardware-based field-oriented control (FOC) and servo controller (velocity, position, ramp generator), motor position feedback interfaces (A/B/N encoder, hall), an analog signal processing block for bottom shunt current measurement (programmable current-sense amplifiers [CSAs] and analog-to-digital converters [ADCs]). It also includes a powerful and flexible power management unit (PMU) along with a buck converter and programmable low-dropout (LDO) regulators. For overall control and communication with an external processor through SPI or UART, a preprogrammed 32-bit microcontroller is embedded. The processor system supports either low-level direct register access to all motor control peripherals or higher-level parameter mode access for extended functionality and ease of use. For initial configuration of system hardware connections and software selection, a bootloader is available also supporting permanent storage of this configuration in the one-time-programmable (OTP) memory.

Applications

- Robotics
- Power Tools
- Gardening
- Automated Guided Vehicles (AGV)/Warehouse Automation
- Pump (e.g., Peristaltic)
- Industrial 3D Printing
- Factory Automation
- Desktop Manufacturing
- E-Bike/Light Electric Vehicles or LEV

Benefits and Features

- Three-Phase Permanent Magnet Synchronous Motors (PMSM)/Brushless DC (BLDC), Two-Phase Stepper Motor, and Brushed DC Motor Support
- 7.7V to 70V Single-Supply Operating Voltage Range
- Smart Gate Driver with Adjustable Strength up to 1A/2A Source/Sink
- Field-Oriented Controller/FOC in Hardware for Wide Bandwidth Current Control Loop
- Position, Velocity, and Torque Controller in Hardware for Fast and Precise Control
- 8-Point Ramp Generator with Ramp Calculation in Real Time in Hardware
- Fast Space Vector Pulse Width Modulation (SVPWM) Engine (2kHz …100kHz) with 120MHz Clock
- Feedback Position Sensor Support (Hall, A/B/N, SPI)
- Bottom Shunt Current Measurement (Programmable CSA and ADCs)
- 5.8V/600mA Buck Converter
- 2x Configurable LDOs (2.5V, 3.3V, or 5V) for Supply of External Circuits with each having 150mA current
- Charge Pump with Voltage Doubler
- Trickle Charge Pump for 100% PWM Duty Cycle
- Integrated Preprogrammed 32-Bit/40MHz Microcontroller Supporting Initial Configuration (OTP) of the Device and Direct Hardware/Register or Parameter Mode Access
- SPI, UART Interfaces for Communication with Main/Application Controller
- Parameter Storage (Optional) in SPI flash memory or I2C EEPROM
- Internal Oscillator with Phase Locked Loop (PLL) and Optional External Crystal or Clock Support
- Watchdog with Separate Internal Oscillator
- Low-Power Hibernation Mode with Wake-Up Button and Timer Support
- Compact Monolithic Solution, 64-Pin, 9mm x 9mm TQFN Package

Ordering Information appears at end of data sheet.

19-101926; Rev 1; 11/24

Simplified Block Diagram

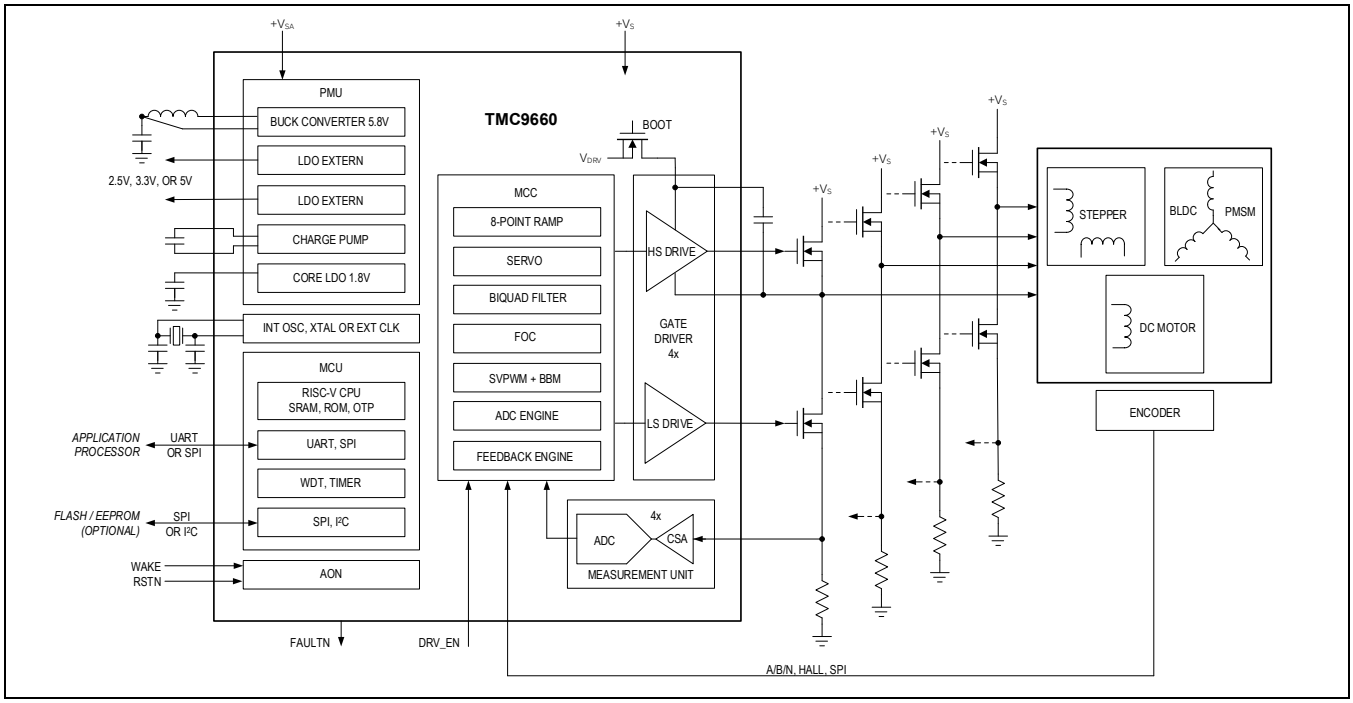

Figure 1. TMC9660 Simplified Block Diagram

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Absolute Maximum Ratings

V_S, V_{SA} to GND .....-0.3V to +87.0V

SW to GND..... -0.3V to V_{VSA} + 0.3V

V_{BUCK} to GND .....-0.3V to +6.5V

CPI to GND .....-0.3 V to V_{VBUCK} + 0.3V

CPO to GNDV_{VBUCK} - 0.3V to min(V_{VDRV} + 0.3, V_{VBUCK} + 6.5)V

V_{DRV} to GND .....V_{CPO} - 0.3V to min(+13.0, V_{CPO} + 6.5)V

V_{EXT1}, V_{EXT2} to GND.....-0.3V to min(+6.0, V_{VBUCK} + 0.3)V

1V8, A1V8 to GND .....-0.3V to min(+2.2, V_{VBUCK} + 0.3)V

V_{CC_IO} to GND .....-0.3V to +6.0V

GPIO_ to GND .....-0.3V to V_{CC_IO} + 0.3V

AIN_ to GND .....-0.3V to V_{CC_IO} + 0.3V

DRV_EN to GND.....-0.3V to V_{CC_IO} + 0.3V

FAULTN, RSTN to GND.....-0.3V to 6V

WAKE to GND .....-0.3V to V_{SA} + 0.3V

GNDA, GNDP, EP to GND..... -0.3V to 0.3V

BM_ to GND max((BS_ -16) , (V_S - 87)) V to min(V_S + 6, 87)V

BS_ to GND .....-0.3V to min(BM_ + 16, 87)V

HS_ to GND..... BM - 0.3V to BS_ + 0.3V

HS_ to BM_ .....-0.3V to BS_ + 0.3V

LS_ to GND ..... -2.7V to V_{DRV} + 0.3V

CSP_ to GND ..... -2.7V to 2.7V

CSN_ to GND ..... -2.7V to 2.7V

Gate Driver Source Current (HS_/LS_)..... Internally limited

Gate Driver Sink Current (HS_/LS_) ..... Internally limited

Continuous Power Dissipation (Multilayer Board) (T_A = +70°C, derate 43.9 mW/°C above +70°C.).....2412mW

Operating Temperature Range (*Note 1*).....-40°C to 125°C

Junction Temperature ..... +150°C

Note 1: Junction temperature greater than +125°C degrades operating lifetimes.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Table 1. Recommended Operating Conditions

PARAMETER	SYMBOL	CONDITION		TYPICAL RANGE	UNIT
Motor Supply Voltage (V_S)	V_S			7.7 to 70	V
Core Supply Voltage (V_{SA})	V_{VSA}	DCR_{LBUCK}<300mΩ		7.7V to 80.0	V
Logic Input Supply Voltage	V_{CC_IO}			2.5 to 5	V
High-Side Gate Drive Current	IGATE_HS			0 to 25	mA
Low-Side Gate Drive Current	IGATE_LS			0 to 25	mA
External Load Current on V_{EXT1}	IEXT1			0 to 150	mA
External Load Current on V_{EXT2}	IEXT2			0 to 150	mA
PWM Drive Frequency	FPWM			1.83 to 100	kHz
Trickle Charge Pump Maximum External Load Current per Pin	ITKCHP			2	µA

Ambient Temperature Range	TA			-40 to 125	°C

Package Information

TQFN 64 – 9mm x 9mm	
Package Code	T6499+2C
Outline Number	*21-100060*
Land Pattern Number	*90-100053*
THERMAL RESISTANCE, FOUR-LAYER BOARD	
Junction-to-Ambient (θ_{JA})	22.8°C/W
Junction-to-Case Thermal Resistance (θ_{JC})	0.69°C/W

For the latest package outline information and land patterns (footprints), go to *https://www.analog.com/en/resources/packaging-quality-symbols-footprints/package-index.html*. Note that a "+", "#", or "-" in the package code indicates RoHS status only. Package drawings may show a different suffix character, but the drawing pertains to the package regardless of RoHS status.

Package thermal resistances were obtained using the method described in JEDEC specification JESD51-7, using a four-layer board. For detailed information on package thermal considerations, refer to *https://www.analog.com/en/resources/technical-articles/thermal-characterization-of-ic-packages.html*.

Pin Configurations

The package is a TQFN 64-pin, 0.5mm pitch 9mm x 9mm, and the pins are positioned as follows:

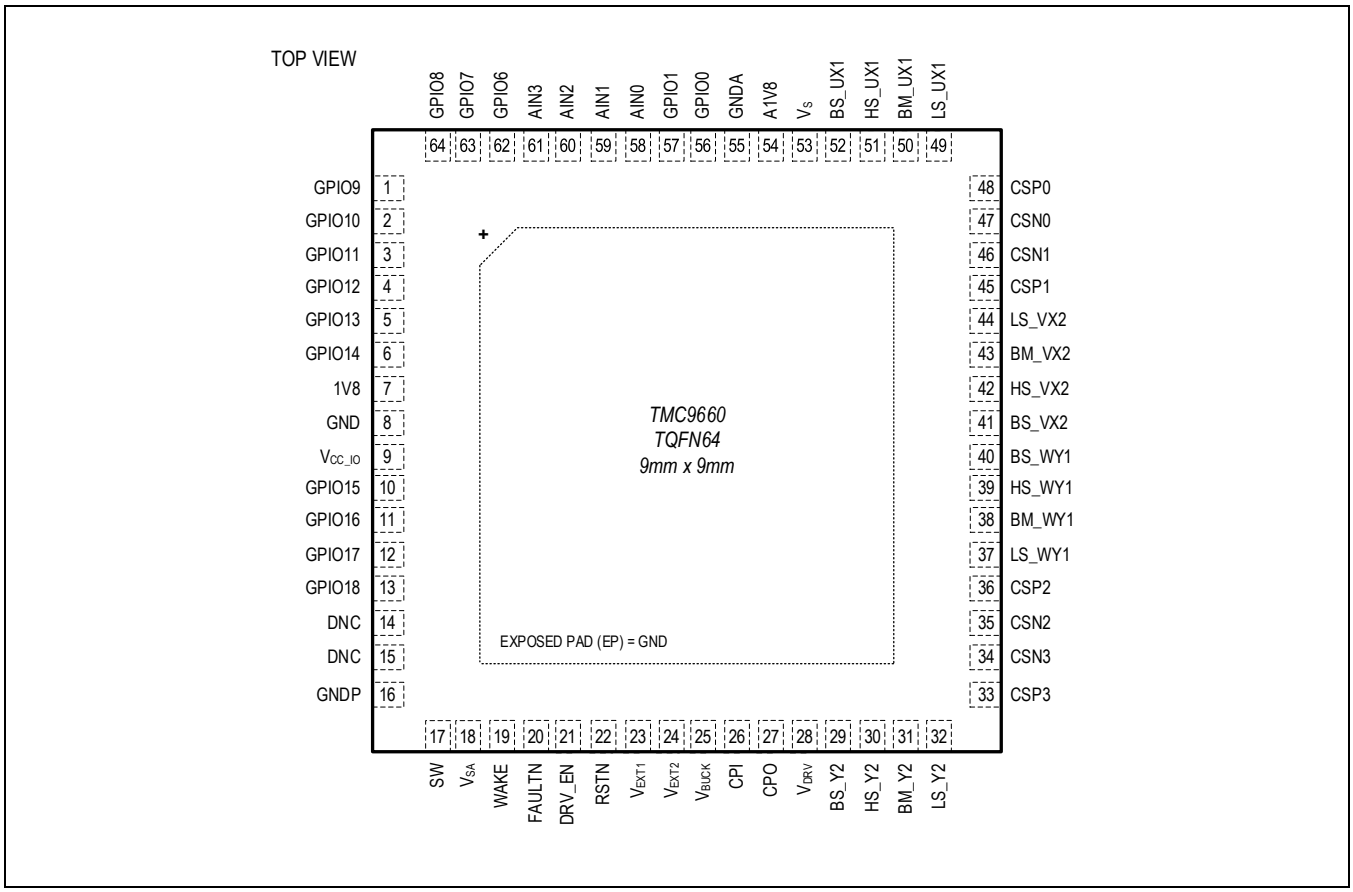

Figure 2. TMC9660 Pinout

Pin Descriptions

PIN	NAME	FUNCTION	REF SUPPLY	Type
PMU				
53	V_S	Gate Driver Supply Voltage Input. Connect at least a 1µF V_{VS_RNG} rated ceramic capacitor from V_S to GND as close to the IC as possible.	VS	POWER
18	V_{SA}	Core Circuit and Buck Supply Voltage Input. Connect at least a 0.5µF V_{VSA_RNG} rated low ESL ceramic capacitor from V_{SA} to GNDP as close to the IC as possible. Also, place a large 10µF V_{VSA_RNG} rated bypass capacitor close to the IC to limit V_{SA} ripple.	VSA	POWER
17	SW	Buck Switching Output. Connect an inductor L_{BUCK} from SW to V_{BUCK} according to what is recommended in the *Applications Information* section.		
25	V_{BUCK}	Buck Output Voltage, Charge-Pump and LV LDOs Supply Voltage Input. Connect a ceramic capacitor C_{VBUCK} from V_{BUCK} to GND as close to the IC as possible and according to what is recommended in the *Applications Information* section.	VBUCK	POWER
28	V_{DRV}	Charge-Pump Output Voltage, Gate Driver Supply Voltage Input. OTP programming voltage input. Connect a ceramic capacitor C_{VDRV} from V_{DRV} to GND as close to the IC as possible and according to what is recommended in the *Applications Information* section.	VDRV	POWER
26	CPI	Charge-Pump Fly Capacitor Low-Voltage Side. Connect a ceramic capacitor C_{FLY} from CPI to CPO as close to the IC as possible and according to what is recommended in the *Applications Information* section.		
27	CPO	Charge-Pump Fly Capacitor High-Voltage Side. Connect a ceramic capacitor C_{FLY} from CPI to CPO as close to the IC as possible and according to what is recommended in the *Applications Information* section.		
23	V_{EXT1}	5.0V/3.3V/2.5V output for supply of external circuits including encoder, sensor, switches, microcontroller, etc. Connect at least 1.6µF 6V rated ceramic capacitor from V_{EXT1} to GND as close to the IC as possible.		
24	V_{EXT2}	5.0V/3.3V/2.5V output for supply of external circuits including encoder, sensor, switches, microcontroller, etc. Connect at least 1.6µF 6V rated ceramic capacitor from V_{EXT2} to GND as close to the IC as possible.		
54	A1V8	Analog Core Supply Voltage Input. Connect a 10µF 2V rated ceramic capacitor from A1V8 to GND as close to the IC as possible.	A1V8	POWER
7	1V8	Digital Core Supply Voltage Input. Connect a 10µF 2V rated ceramic capacitor from 1V8 to GND as close to the IC as possible.	1V8	POWER
55	GNDA	Analog Ground		GND
16	GNDP	Buck Power Ground		GND
8	GND	Ground		GND
9	V_{CC_IO}	Logic Supply Voltage Input. Connect a 1µF, at least V_{VCCIO_RNG} rated, ceramic capacitor from V_{CC_IO} to GND as close to the IC as possible.	VCCIO	POWER
GATE DRIVER				
52	BS_UX1	Boost Flying Capacitor Connection. Connect a V_{DRV} rated ceramic capacitor from BS_UX1 to BM_UX1 as close to the IC as possible for the high-side MOSFET driver supply.		
51	HS_UX1	High-Side Gate Driver Output. Driver output for the high-side MOSFET gate.		
50	BM_UX1	Source Connection for High-Side MOSFET. Also serves as the return for the high-side driver.		
49	LS_UX1	Low-Side Gate Driver Output. Driver output for the low-side MOSFET gate.		

41	BS_VX2	Boost Flying Capacitor Connection. Connect a V_{DRV} rated ceramic capacitor from BS_VX2 to BM_VX2 as close to the IC as possible for the high-side MOSFET driver supply.		
42	HS_VX2	High-Side Gate Driver Output. Driver output for the high-side MOSFET gate.		
43	BM_VX2	Source Connection for High-Side MOSFET. Also serves as the return for the high-side driver.		
44	LS_VX2	Low-Side Gate Driver Output. Driver output for the low-side MOSFET gate.		
40	BS_WY1	Boost Flying Capacitor Connection. Connect a V_{DRV} rated ceramic capacitor from BS_WY1 to BM_WY1 as close to the IC as possible for the high-side MOSFET driver supply.		
39	HS_WY1	High-Side Gate Driver Output. Driver output for the high-side MOSFET gate.		
38	BM_WY1	Source Connection for High-Side MOSFET. Also serves as the return for the high-side driver.		
37	LS_WY1	Low-Side Gate Driver Output. Driver output for the low-side MOSFET gate.		
29	BS_Y2	Boost Flying Capacitor Connection. Connect a V_{DRV} rated ceramic capacitor from BS_Y2 to BM_Y2 as close to the IC as possible for the high-side MOSFET driver supply. Note: Tie this pin to V_{DRV} in case HS_Y2 is used to drive a low-side MOSFET gate.		
30	HS_Y2	High-Side Gate Driver Output. Driver output for the high-side MOSFET gate.		
31	BM_Y2	Source Connection for High-Side MOSFET. Also serves as the return for the high-side driver. Note: Tie this pin to GND in case HS_Y2 is used to drive a low-side MOSFET gate.		
32	LS_Y2	Low-Side Gate Driver Output. Driver output for the low-side MOSFET gate.		
GATE DRIVER				
CSA				
48	CSP0	Channel 0 current-sense amplifier positive analog input. Also serves as the return for the low-side driver.		
47	CSN0	Channel 0 current-sense amplifier negative analog input.		
45	CSP1	Channel 1 current-sense amplifier positive analog input. Also serves as the return for the low-side driver.		
46	CSN1	Channel 1 current-sense amplifier negative analog input.		
36	CSP2	Channel 2 current-sense amplifier positive analog input. Also serves as the return for the low-side driver.		
35	CSN2	Channel 2 current-sense amplifier negative analog input.		
33	CSP3	Channel 3 current-sense amplifier positive analog input. Also serves as the return for the low-side driver.		
34	CSN3	Channel 3 current-sense amplifier negative analog input.		
Other Analog				
56	GPIO0	GPIO0 (input with pull-down at power-up)	VCCIO	AIDIOpud
57	GPIO1	GPIO1 (input with pull-down at power-up)	VCCIO	AODIOpud
58	AIN0	Analog input 0 (high-Z at power-up)	VCCIO	AIDIOpud
59	AIN1	Analog input 1 (high-Z at power-up)	VCCIO	AIDIOpud
60	AIN2	Analog input 2 (high-Z at power-up)	VCCIO	AIDIOpud
61	AIN3	Analog input 3 (high-Z at power-up)	VCCIO	AIDIOpud
GPIO and CONTROL SIGNALS				

22	RSTN	External System Reset Input (active low). The device remains in reset while this pin is in its active state. This pin has an internal pull-down resistor.		DIpd
14	DNC	Do not connect – internally connected	VCCIO	
15		Do not connect—internally connected—or tie to GND		
21	DRV_EN	Driver enable input (active high). This pin has an internal pull-down resistor.	VCCIO	DIpd
62	GPIO6	GPIO6 (input with pull-up at power-up)	VCCIO	DIOpud
63	GPIO7	GPIO7 (input with pull-up at power-up)	VCCIO	DIOpud
64	GPIO8	GPIO8 (input with pull-up at power-up)	VCCIO	DIOpud
1	GPIO9	GPIO9 (input with pull-up at power-up)	VCCIO	DIOpud
2	GPIO10	GPIO10 (input with pull-up at power-up)	VCCIO	DIOpud
3	GPIO11	GPIO11 (input with pull-up at power-up)	VCCIO	DIOpud
4	GPIO12	GPIO12 (input with pull-up at power-up)	VCCIO	DIOpud
5	GPIO13	GPIO13 (input with pull-up at power-up)	VCCIO	DIOpud
6	GPIO14	GPIO14 (input with pull-up at power-up)	VCCIO	DIOpud
10	GPIO15	GPIO15 (input with pull-up at power-up)	VCCIO	DIOpud
11	GPIO16	GPIO16 (input with pull-up at power-up)	VCCIO	DIOpud
12	GPIO17	GPIO17 (input with pull-up at power-up)	VCCIO	DIOpud
13	GPIO18	GPIO18 (input with pull-up at power-up)	VCCIO	DIOpu
20	FAULTN	FAULT output signal (open drain). Indicates busy state during bootstrapping or severe error (e.g., 1V8 UVLO) when the integrated processor is not able to communicate error/error details anymore.		DOod
19	WAKE	Drive this pin high in order to enable power-up and exit from hibernation mode. When WAKE pin is not shorted to V_{SA}, an external pull-down resistor is recommended.	VSA	DI

Functional Diagrams

TMC9660

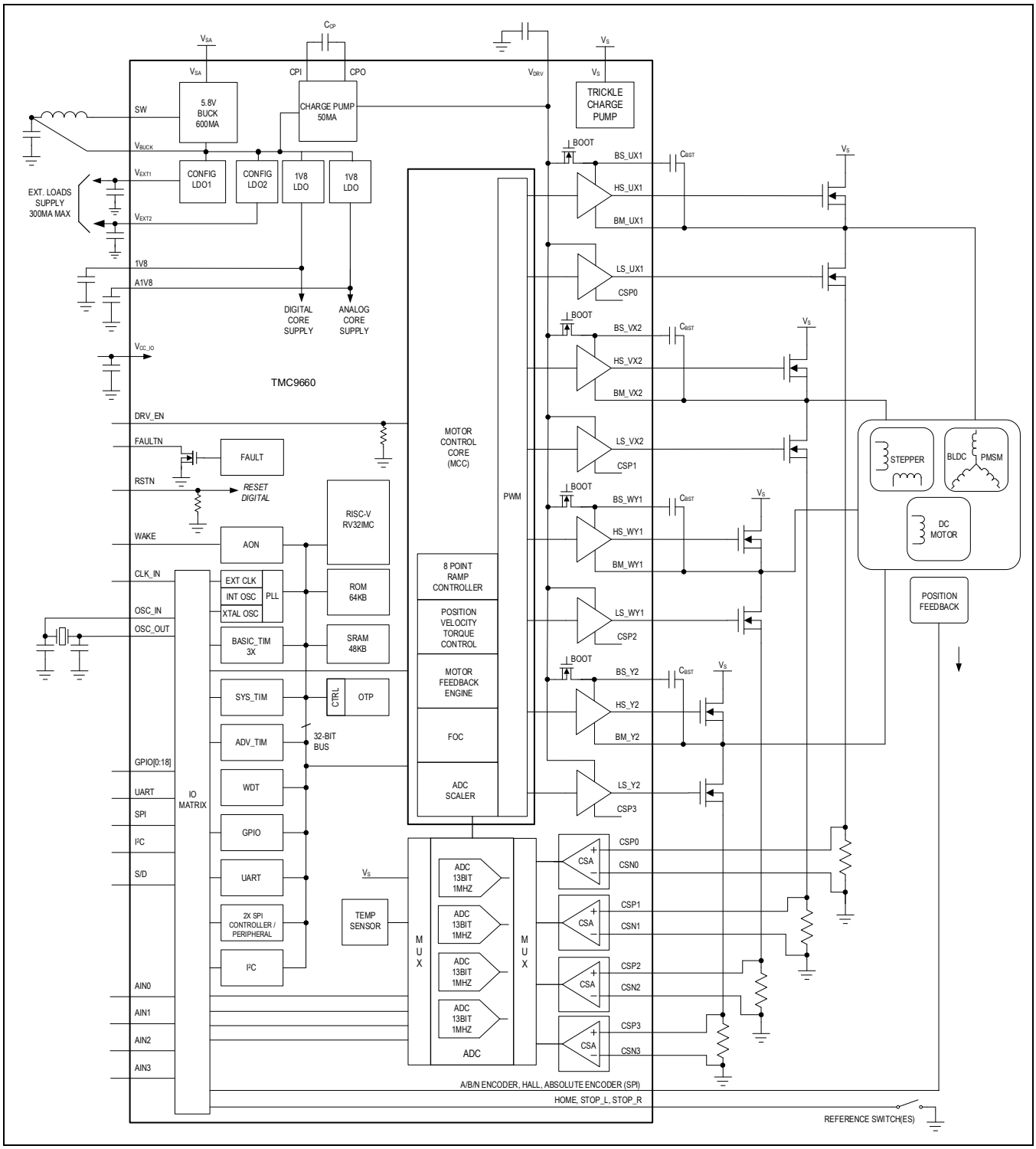

Figure 3. Detailed Block Diagram

Application Modes

The TMC9660 offers two different application modes through its integrated microcontroller – a higher-level parameter mode and a lower-level register mode. Selection of the application mode is part of the chip *Chip Bootstrapping* sequence.

The register mode offers direct access to the hardware registers without any extra features. The parameter mode offers managed access to the hardware with additional software-backed features extending the system capabilities.

Table 2. TMC9660 Supported Features per Application Mode

FEATURE	PARAMETER MODE	REGISTER MODE
FOC motor control in hardware	Y	Y
Hall and ABN feedback	Y	Y
Secondary AB feedback	Y	N
SPI encoder feedback	Y	N
BrakeChopper support	Y	N
Mechanical brake support	Y	N
StepDir	Y	N
Extended gate driver fault handling	Y	N
Communication heartbeat monitoring	Y	N
Motor parameter storage	Y	N
Script execution from external memory	Y	N
Automated GDRV bootstrapping	Y	N
Automated ADC calibration	Y	N
Configurable supply voltage monitoring	Y	N
ABN encoder initialization	Y	N
Basic reference switch support	Y	Y
Automated reference switch homing	Y	N

Power-Saving Modes

The TMC9660 offers several options for reducing power consumption by switching off unused blocks. In addition, a hibernation mode is available. Upon entering this mode everything inside the TMC9660 is switched off and put into reset, including the driver stage, the buck regulator, all voltages available externally, the microcontroller part with memory and peripherals, and the analog blocks apart from the always-on (AON) block. The AON block contains its own regulator and internal 32.768kHz oscillator in addition to 16 registers for preserving status information. AON registers content and settings are reset when V_{SA} is removed. The AON block supports "push button" and timer-based power on. The dedicated WAKE input pin connected to the AON block has to be pulled high in order to enable start up of the device or exit hibernation mode. This pin can withstand voltages up to main supply voltage (V_{SA}). In case timer wake-up has been configured, the device automatically powers up again after the predefined amount of time.

The typical power-down/wake-up procedure is as follows:

- Processor configures wake-up method—through external WAKE pin or wake-up timer.
- Processor powers down all components including itself, and the digital part is put into reset/hibernation mode in order to reduce power consumption to minimum.
- Depending on the selection, either a rising edge on the WAKE pin or an overflow of the wake-up timer puts the device out of reset and hibernation mode and powers it up again.

Power Supply—Electrical Characteristics

(V_{SA} = from 7.7V to 80V, V_S = from 7.7V to 70V, V_{CC_IO} = from 2.2V to 5.5V, L_{BUCK} = 27µH DCR<300mΩ, C_{BUCK} = 8µF, CPP1 = 220nF, CVDRVE = 4.7µF, Junction Temperature = from -40ºC to 125ºC, Unless otherwise noted, Limits over the operating temperature range and relevant supply voltage range are guaranteed by design and characterization. Specifications marked "GBD" are guaranteed by design and not production tested.)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS	GL
OPERATING VOLTAGE RANGE								
Core Supply Voltage Range (V_{SA})	V_{VSA_RNG}			7.7		80	V	II
Driver Operating Voltage Range (V_S)	V_{VS_RNG}	12V gate drive – Buck = 5.8V		7.7		70	V	II
GPIO Supply Voltage Range (V_{CC_IO})	$V_{VCC\_IO\_RNG}$			2.2		5.5	V	II
CURRENT CONSUMPTION								
V_{SA} Quiescent Current Consumption	I_{VSA_Q}	PMU, GDU, MU and digital core enabled. Quiescent operation (not switching).	V_{VSA} = 24V		13		mA	VI
			V_{VSA} = 48V		10			VI
V_S Quiescent Current Consumption	I_{VS_Q}	GDU enabled. Quiescent operation (not switching).	V_{VS} = 24V Entire Chip Enabled		2.1	3	mA	II
V_{BUCK} Quiescent Current Consumption	I_{VBUCK_Q}	PMU, GDU, MU and digital core enabled. Quiescent operation (not switching).	Entire Chip Enabled		36	47	mA	II
V_{DRV} Quiescent Current Consumption	I_{VDRV_Q}	GDU enabled. Quiescent operation (not switching).	V_{DRV} = 11.6V		1.4	2	mA	II
Quiescent Current Consumption V_{CC_IO}	I_{VCC_IO}	GPIOx stable PU disabled			3.8	15	µA	II
V_{SA} Current Consumption - Hibernate Mode	I_{VSA_HIB}	Hard hibernate mode	Buck and Driver and Logic shut down.		12.7	40	µA	II
V_S Supply Leakage Current - Hibernate Mode	I_{VS_HIB}	Hard hibernate mode	Buck and Driver and Logic shut down.			6	µA	II

GUARANTEE LEVEL	DESCRIPTION
I	Production Tested @ Multiple Temps
II	Production Tested @ Room Temp, Characterized @ Multiple Temps
IIsc	Production Tested via Scan @ Room Temp, Characterized via Scan @ Multiple Temps
III	Sample Tested

IV	Not Production Tested, Characterized by ATE
V	Not Production Tested, Characterized by Bench (GBDC)
VI	Internal Design Target
VII	Production Tested, Internal Only
VIII	Production Tested @ Hot, Characterized @ Multiple Temps

General-Purpose Input/Output Description

The TMC9660 offers 19 general-purpose digital inputs and outputs (GPIO). Each GPIO pin can be configured individually as digital input or as output (push-pull). Inputs can be configured individually with internal pull-up resistors to V_{CC_IO} or pull-down resistors to GND. GPIO pins are shared with alternate digital inputs or outputs from peripheral blocks and analog inputs which can be selected per pin.

At power-up, all four analog inputs AIN0-3 are selected as default for pin 58-61 and GPIO configured as inputs for the others. For GPIO0 and GPIO1, internal pull-down resistors are enabled, while for all other GPIO6-18 pull-ups are enabled. Note that the analog inputs do not have pull-up or pull-down enabled. In case these pins are not used, they should be connected to defined voltage levels externally, e.g., GND.

At first-time power-up, the integrated bootloader configures the UART and the SPI peripheral interfaces and accepts commands on either interface. For UART communication, serial data in signal UART_RX is selected instead of GPIO7 for Pin 63 and serial data out signal UART_TX instead of GPIO6 for Pin 62. The bootloader listens for any incoming command through UART_RX. In case a valid command is received, a reply message is sent out through UART_TX.

For SPI communication, the SPI peripheral block is initialized and listens on SPI0_MOSI, SPI0_SCK, and SPI0_CS0 for incoming data, clock, and chip select signals and switches SPI0_MISO to output and sends out data as soon as SPI0_CS0 is pulled down externally. The SPI0_MISO is selected instead of GPIO9 for Pin 1, SPI0_MOSI instead of GPIO10 for Pin 2, SPI0_SCK instead of GPIO11 for Pin 3, and SPI0_CS0 instead of GPIO12 for Pin 4.

The bootloader supports selection of alternate pin functionality according to application circuit requirements. The configuration may be written to internal OTP memory for permanent storage. This configuration is then used as default for all future power-ups.

Figure 4 and *Figure 5* show the equivalent input diagram for a GPIO and an analog input GPIO. When the pin is configured as analog input, its range is limited to 0V to 1.2V and if more than 1.8V is applied to the analog pin, the analog switch is open to avoid damages at the ADC and a fault is triggered.

Table 3. Alternate Function Mapping

PIN	AF0 (DEFAULT*)	AF1	AF2	AF3	AF4
56	GPIO0 (input with pull-down)	OSC_IN	CLK_IN	UART_TX	
57	GPIO1 (input with pull-down)	OSC_OUT	UART_RX		ENC_B
58	AIN0	GPIO2	UART_TXEN	STOP_L	HALL_U
59	AIN1	GPIO3	TIM_ADV_OUT1	STOP_R	HALL_V
60	AIN2	GPIO4	I2C_SCL	HOME	HALL_W
61	AIN3	GPIO5	I2C_SDA	ENC_A	
62	GPIO6 (input with pull-up)	UART_TX*	TIM_ADV_IN0	SPI0_SCK	
63	GPIO7 (input with pull-up)	UART_RX*	TIM_ADV_IN1	HOME	HALL_U
64	GPIO8 (input with pull-up)	UART_TXEN	SPI0_CS1	ENC_A	HALL_W
1	GPIO9 (input with pull-up)	SPI0_MISO*	TIM_ADV_IN2		HALL_U
2	GPIO10 (input with pull-up)	SPI0_MOSI*	TIM_ADV_OUT0		HALL_W
3	GPIO11 (input with pull-up)	SPI0_SCK*	I2C_SDA	TIM_ADV_IN1	
4	GPIO12 (input with pull-up)	SPI0_CS0*	I2C_SCL	STOP_L	
5	GPIO13 (input with pull-up)	SPI0_CS1	I2C_SCL	ENC_B	
6	GPIO14 (input with pull-up)	I2C_SDA	SPI1_SCK	ENC_N	
10	GPIO15 (input with pull-up)	TIM_ADV_IN0	SPI1_CS0	HOME	HALL_V
11	GPIO16 (input with pull-up)	TIM_ADV_IN1	SPI0_CS2	STOP_L	ENC_N

12	GPIO17 (input with pull-up)	TIM_ADV_IN2	SPI1_MISO	ENC_A	HOME
13	GPIO18 (input with pull-up)	TIM0_BASIC_PWM	SPI1_MOSI	STOP_R	ENC_B

*Defaults in hardware are overridden by the bootloader on power-up.

Table 4. Peripheral Pin Description

PIN GROUP	DESCRIPTION
GPIO0..18	General Purpose Digital Inputs and Outputs. These signals can be set to input with optional pull-up or pull-down resistor or digital output (push-pull). In parameter mode, the two-point hysteresis-based control signal for an external brake chopper circuit for limiting the supply voltage can be configured for any available GPIO output pin.
AIN0..3	Analog inputs connected to the ADC inputs of the measurement unit (MU). It is sampled with current motor pulse-width modulation (PWM) frequency. Values are available through the motor control core (MCC). The parameter mode supports external temperature sensor needs to be connected to AIN3.
OSC_IN, OSC_OUT	Connect a crystal between these two pins when using an external crystal oscillator instead of the internal oscillator.
CLK_IN	External clock input when using an external clock instead of the internal oscillator
UART_TX, UART_RX, UART_TXEN	UART serial transmit data out (Tx) and receive data in (Rx). For half-duplex operation, the transmitter enable signal (TxEN) may be connected to an external transceiver (e.g., RS485). The UART interface may be used for communication with the application processor (bootloader configuration).
I2C_SCL, I2C_SDA	I^2C interface, controller mode with clock output only. An I^2C EEPROM may be connected externally for program and parameter storage (bootloader configuration).
STOP_L, STOP_R, HOME	Stop left/right and Home switch inputs connected to the internal 8-point ramp generator of the MCC.
ENC_A, ENC_B, ENC_N	Incremental encoder A/B/N input channels connected directly to the feedback engine of the internal MCC.
HALL_U, HALL_V, HALL_W	Hall sensor U/V/W input channels connected directly to the feedback engine of the internal MCC.
TIM_ADV_IN0, TIM_ADV_IN1, TIM_ADV_IN2, TIM_ADV_OUT0, TIM_ADV_OUT1	Inputs and Outputs of the advanced timer peripheral block. In parameter mode, TIM_ADV_IN0/TIM_ADV_IN1 can be configured as Step/Direction inputs or TIM_ADV_IN0/TIM_ADV_IN1 as second incremental encoder A/B channel inputs with processor/software support (parameter mode).
TIM0_BASIC_PWM, TIM_ADV_OUT0, TIM_ADV_OUT1	In parameter mode, the PWM control signal for an external electromagnetic brake driver can be configured for one of the timer outputs.
SPI0_MISO, SPI0_MOSI, SPI0_SCK, SPI0_CS0, SPI0_CS1, SPI0_CS2	SPI block 0 supporting controller mode (SPI0_MISO: serial data in, SPI0_MOSI: serial data out, SPI0_SCK: clock output, SPI0_CSx: chip select outputs/low active). An SPI flash memory may be connected externally for program and parameter storage (bootloader configuration). In parameter mode, this interface may be used for connecting an external absolute encoder with SPI with processor/software support (parameter mode). Note: SPI block 0 may be replaced with SPI peripheral block. In this case, signals change direction (SPI0_MISO: serial data output, SPI0_MOSI: serial data input, SPI0_SCK: clock input and SPI0_CS0: chip select input). This interface may be used for communication with the application processor as fast alternative to UART communication (bootloader configuration). For an SPI-Flash connection, chip select signal generation is done in software for more flexibility. Any available GPIO pin may be configured as chip select signal and not just the dedicated signals supported by the SPI peripheral block 0 in hardware (SPI0_CSx).
SPI1_MISO, SPI1_MOSI, SPI1_SCK, SPI1_CS0	SPI block 1 supporting controller mode (SPI1_MISO: serial data in, SPI1_MOSI: serial data out, SPI1_SCK: clock output, SPI1_CS0: chip select output/low active). An SPI flash memory may be connected externally for program and parameter storage (bootloader configuration). In parameter mode, this interface may be used for connecting an external absolute encoder with SPI with processor/software

	support (parameter mode). Note: SPI block 1 may be replaced with SPI peripheral block. In this case, signals change direction (SPI1_MISO: serial data output, SPI1_MOSI: serial data input, SPI1_SCK: clock input and SPI1_CS0: chip select input). This interface may be used for communication with the application processor as fast alternative to UART communication (bootloader configuration). For an SPI-Flash connection, chip select signal generation is done in software for more flexibility. Any available GPIO pin may be configured as chip select signal and not just the dedicated signal supported by the SPI peripheral block 1 in hardware (SPI1_CS0).

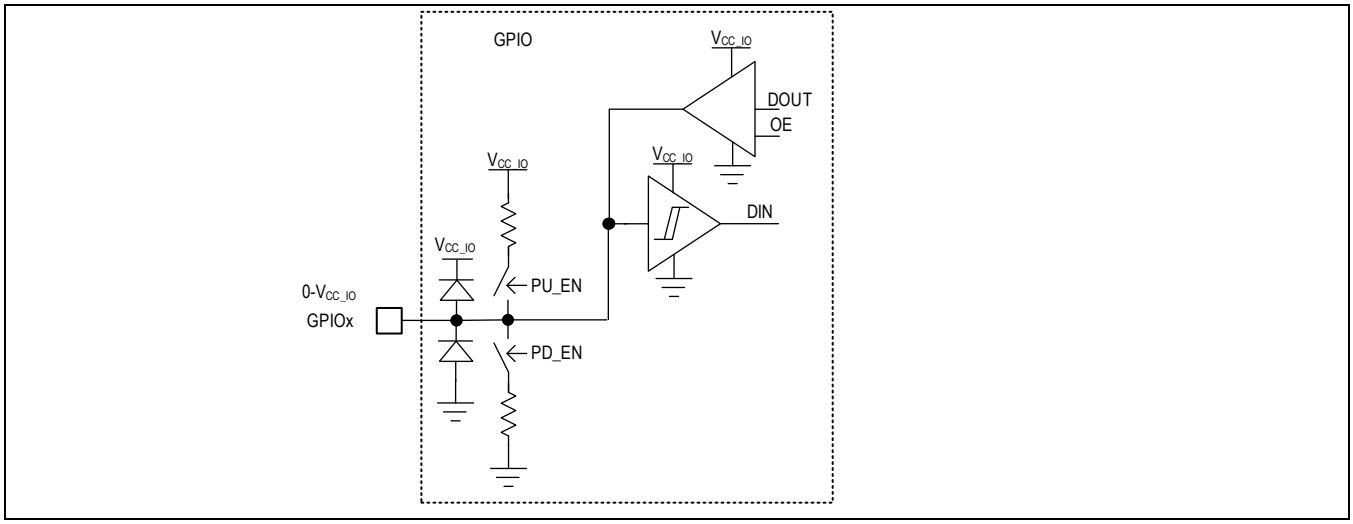

Figure 4. GPIOx—General-Purpose Digital Input/Output Block Diagram

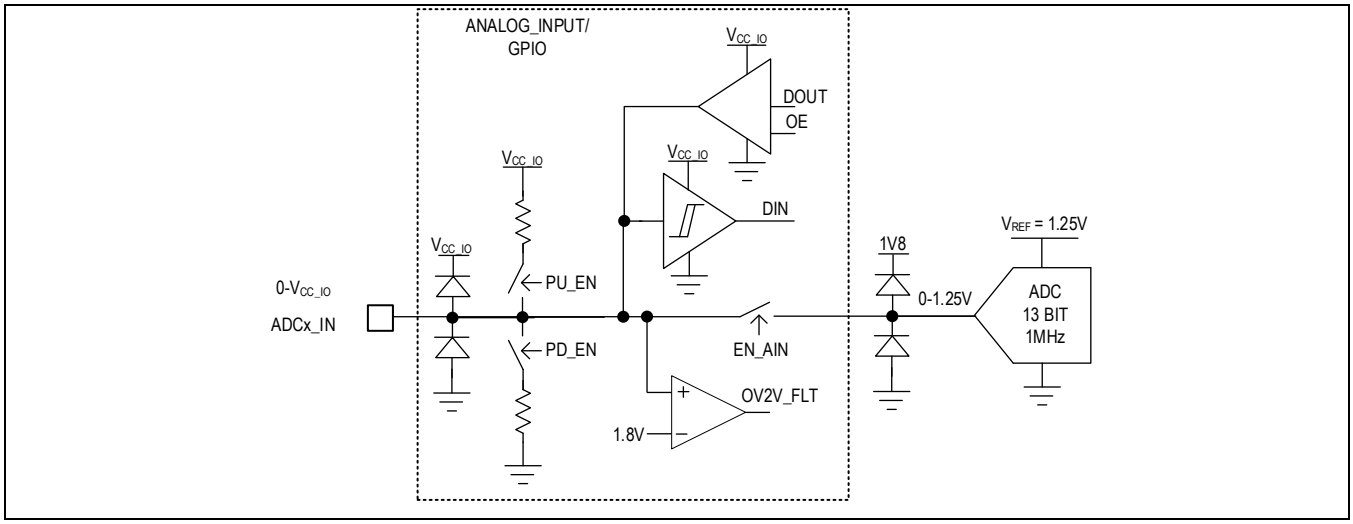

Figure 5. AINx—Analog Input and General-Purpose Digital Input/Output Block Diagram

GPIO Electrical Characteristics

(V_{SA} = From 7.7V to 80V, V_{CC_IO} = 2.2V to 5.5V, Junction Temperature = -40ºC to 125ºC, Limits over the operating temperature range and relevant supply voltage range are guaranteed by design and characterization. Specifications marked "GBD" are guaranteed by design and not production tested.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS	GL
GPIO/Input Configuration							
Resistive Pull-Up	R_{PU}		60	100	140	kΩ	II
Resistive Pull-Down	R_{PD}		60	100	140	kΩ	II
Logic Input Leakage Current	I_{LEAK}		-1		1	µA	II
Input Voltage Level High	V_{IH}		2			V	II
Input Voltage Level Low	V_{IL}				0.8	V	II
Hysteresis	HYST			0.27		V	II
Logic Input Leakage Current	I_{LEAK}	PU/PD disable	-1		1	µA	II
GPIO/Output Configuration							
Logic High Output Voltage	V_{OH}	IPD = 5mA	V_{CC_IO} - 0.4			V	II
Logic Low Output Voltage	V_{OL}	IPU = 5mA			0.4	V	II

GUARANTEE LEVEL	DESCRIPTION
I	Production Tested @ Multiple Temps
II	Production Tested @ Room Temp, Characterized @ Multiple Temps
IIsc	Production Tested via Scan @ Room Temp, Characterized via Scan @ Multiple Temps
III	Sample Tested
IV	Not Production Tested, Characterized by ATE
V	Not Production Tested, Characterized by Bench (GBDC)
VI	Internal Design Target
VII	Production Tested, Internal Only
VIII	Production Tested @ Hot, Characterized @ Multiple Temps

Microcontroller Unit Description

The TMC9660 includes a preprogrammed embedded microcontroller system (microcontroller unit/MCU) with a 32-bit RISC-V microprocessor running at 40MHz. This system also includes memory with SRAM (48KB), OTP, and read-only memory (ROM) with the bootloader, direct register access and parameter firmware, and a set of peripherals for communication (SPI, UART, and I^2C), motor control (including MCC, smart gate driver, and measurement unit/MU), and several timer units. After power-on, the bootloader is started automatically from ROM.

Timer

All timer units for motor control related PWM generation, position counting with feedback system decoding, and sample time generation for the MUs with ADCs and CSAs are integrated into the MCC. In addition, there are separate timer peripherals supporting program execution in general or used for additional functionality.

System Timer

The system timer unit comprises a 64-bit counter. As soon as the processor starts executing instructions after power-up, the timer starts counting upwards using the system clock. This timer is used by bootloader, register, and parameter mode firmware and not available for other purposes.

Basic Timer

The TMC9660 contains three basic timer units (TIM0_BASIC, TIM1_BASIC, and TIM2_BASIC). The basic timer unit comprises one 32-bit auto-reload counter with 32-bit counter limit register and a 32-bit compare register. For the first unit, limit and compare register may be used to generate a PWM signal available on TIM0_BASIC_PWM.

Advanced Timer

The advanced timer unit (TIM_ADV) comprises one 32-bit auto-reload up-/down-counter with 32-bit compare register and two auxiliary up-counters similar to the basic counter. In addition, there are three countervalue capture and two countervalue compare registers. Up to three digital inputs (TIM_ADV_IN0, TIM_ADV_IN1, and TIM_ADV_IN2) are available to control counting and trigger capture events. The results from the two compare registers are available as output signals TIM_ADV_OUT0 and TIM_ADV_OUT1.

In parameter mode, the advanced timer block is used for supporting Step/Direction input signals or a second A/B incremental encoder on TIM_ADV_IN0 and TIM_ADV_IN1, and not available for other purposes.

Watchdog Timer

The TMC9660 includes a 32-bit watchdog timer that uses an internal 32kHz oscillator shared with the AON block and separated from the internal 15MHz oscillator used for the system clock. This watchdog timer is configured and enabled at start-up in register and parameter mode firmware and cannot be disabled/modified until next power cycle. In case, the watchdog is not reset on a regular base as expected by the ROM firmware and the counter exceeds the preset limits a watchdog event is generated. This resets the processor and all peripherals, including the MCC and PWM unit, and disables the driver stage and, therefore, turn off any power supply to the motor.

UART

The UART supports full-duplex data exchange with external devices using industry standard NRZ asynchronous serial data format. The UART offers a wide range of baud rates up to 5Mbit/s (with x8 oversampling) using a fractional baud rate generator and supports automatic baud (autobaud) rate detection as an option.

Table 5. UART Baud Rates

BAUD RATE	MANTISSA (x16)	FRACTION (x16)	MANTISSA (x8)	FRACTION (x8)
300	8333	5	16666	5
600	4166	10	8333	2
1200	2083	5	4166	5
2400	1041	10	2083	2
4800	520	13	1041	5
9600*	260	6	520	6

19200*	130	3	260	3
38400*	65	1	130	1
57600*	43	6	86	6
115200*	21	11	43	3
230400*	10	13	21	5
250000*	10	0	20	0
460800	5	6	10	6
500000	5	0	10	0
921600	2	11	5	3
1000000	2	8	5	0
2000000	1	4	2	4
2500000	1	0	2	0
4000000	-	-	1	2
5000000	-	-	1	0

*Automatic baud rate detection supported with 40MHz system clock

UART bidirectional communication requires a minimum of two pins: receive data in (UART_RX) and transmit data out (UART_TX). Serial data is transmitted and received through these pins during communication. After initial power-on, the bootloader configures Pin 63 as UART_RX instead of GPIO7 and the UART listens for incoming messages. Idle state for UART communication is expected to be “1.” One communication frame always comprises a start bit (“0”), 8 bits of data (LSB first), and one stop bit (“1”). For communication with the bootloader command and reply, messages always comprise 8 communication frames/bytes.

The bootloader configures the UART block with autobaud switched on. As first character of a command message, 0x55 is expected. This pattern is used for baud rate detection. Next characters are the device address (configurable), one command byte, four data bytes (32-bit value) with the most significant byte transmitted first, and finally a cyclic redundancy check (CRC) checksum byte. In case the distance between two bytes of a command message is longer than approximately 100ms, the message is discarded, and all bytes already received are removed from the input queue in expectation of a new command.

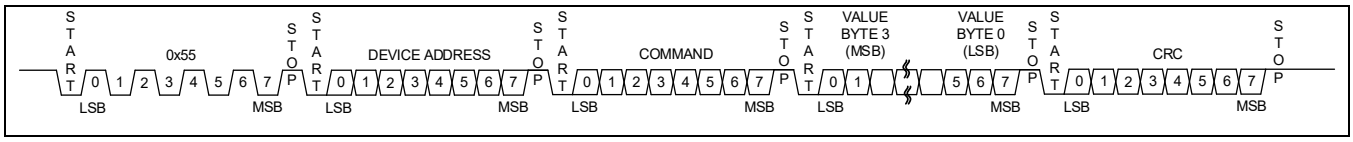

Figure 6. Bootloader UART Command Message (Received)

Bootloader communication relies on a command and reply handshake mechanism for flow control. Next command should not be sent before the reply datagram has been received.

The reply message starts with the controller address (default 0xff), the device address, one status byte, four data bytes (32-bit value) with the MSB first and finally a CRC checksum byte.

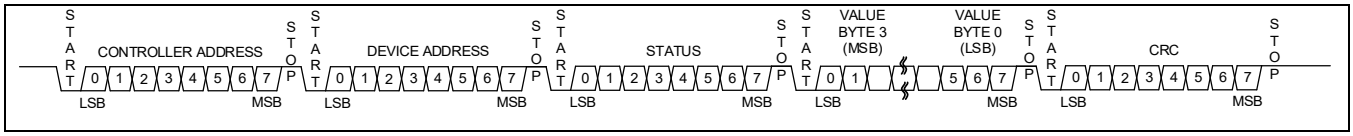

Figure 7. Bootloader UART Reply Message (Transmitted)

For more details on the content of these bytes, see the *Bootloader* section.

At 40MHz system clock, standard baud rates between 9600 bits and 250000 bits can be expected to work with the autobaud detection mechanism. Higher baud rates usually require fixed baud rate settings and a more precise external clock or crystal clock instead of the internal clock generator.

To support multi-axis solutions with a minimum of additional components and wires, the UART and bootloader also support unidirectional communication for connecting the TMC9660 to a serial bus (e.g., RS485). The UART offers the additional signal UART_TXEN to control an external transceiver directly in hardware.

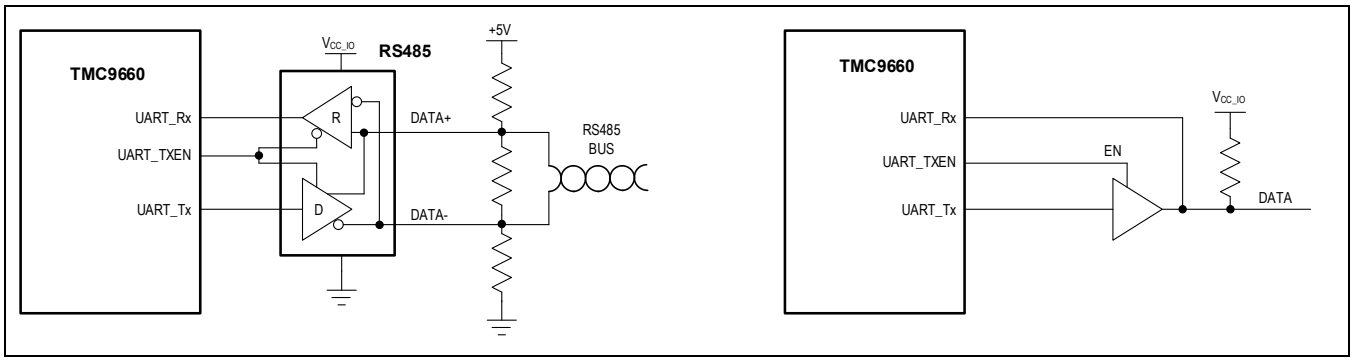

Figure 8. UART with External Transceiver

The active high transceiver enable signal UART_TXEN can be activated before the start bit is sent out through UART_TXEN (t_{PRE}) and kept active for some time after the stop bit (t_{POST}) to support glitch-free transition between transmit and idle/receive mode. The pre-delay (t_{PRE}) and the post-delay (t_{POST}) can be configured individually in the bootloader. The UART_TXEN signal can be mapped to either Pin 64 (GPIO8) or Pin 58 (AIN0).

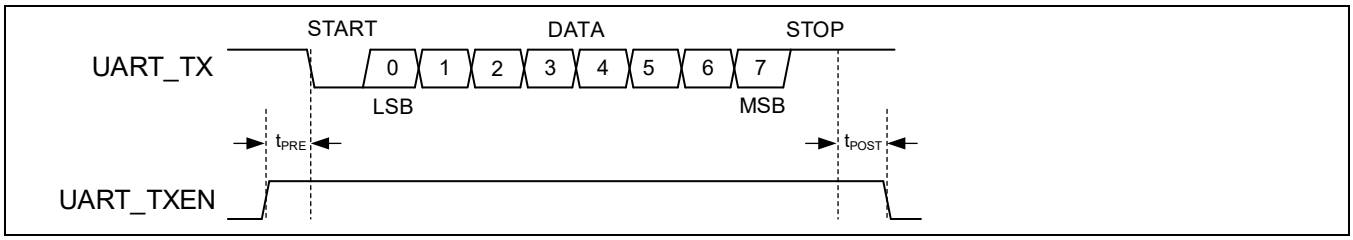

Figure 9. UART Transceiver Enable Signal UART_TXEN with Pre- and Post-Delay Configuration Options

Controller and device address are part of the bootloader protocol. This way, a system with one controller and multiple motor axis/devices (TMC9660) can be set up with a minimum of additional components and connections.

SPI

The TMC9660 includes two SPI block (SPI0/SPI1) for controlling external SPI peripherals (generating the necessary chip select and clock signals) and one SPI block for connecting the TMC9660 as peripheral to an external application processor. In this case, the application processor operates as controller generating the necessary chip select and clock signals for SPI communication. This peripheral SPI block can be mapped to either SPI0 or SPI1 alternate function signals in the GPIO matrix with only CS0 supported in both cases. The bootloader supports a fixed datagram length of 40 bits with the MSB transmitted first and uses SPI mode 3 for communication—sampling incoming data (MOSI) on the rising edge and shifting out data (MISO) on the falling edge of the clock signal (SCK). SPI clock frequencies up to 10MHz are supported. At initial power-on, SPI peripheral block is mapped to SPI0 with SPI0_MISO connected to Pin 1 instead of GPIO9, SPI0_MOSI connected to Pin 2 instead of GPIO10, SPI0_SCK connected to Pin 3 instead of GPIO11, and SPI0_CS0 connected to Pin 4 instead of GPIO12. The bootloader listens on signals SPI0_MOSI, SPI0_SCK, SPI0_CS0 and activates SPI0_MISO output and sends out data as soon as SPI0_CS0 is pulled low.

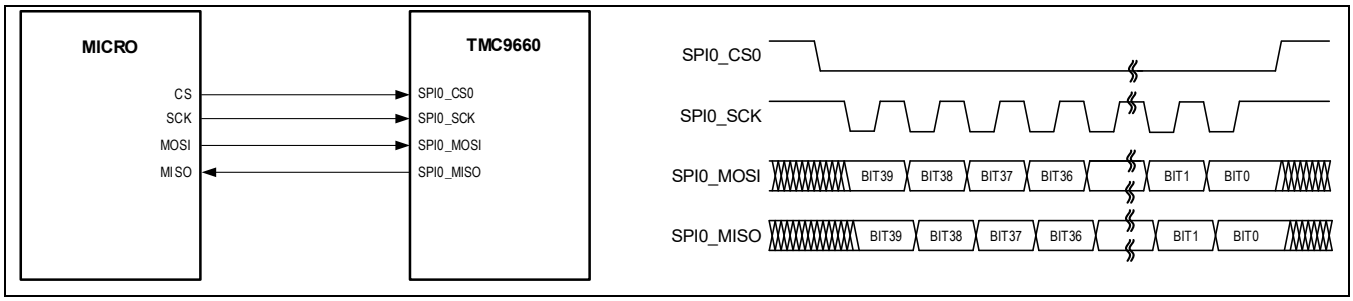

Figure 10. TMC9660 Connected as SPI Peripheral to Application Processor and SPI Communication in Bootloader Mode

A 40-bit datagram from the external micro to TMC9660 comprises an 8-bit command byte followed by 32-bit or 4 bytes of data with the most significant bit/byte transferred first. The 40-bit reply for this command from TMC9660 toward the external micro is always transferred with the next following SPI datagram. The reply datagram starts with 8-bit status information followed by 32-bit or 4 bytes of data with the most significant bit/byte transferred first.

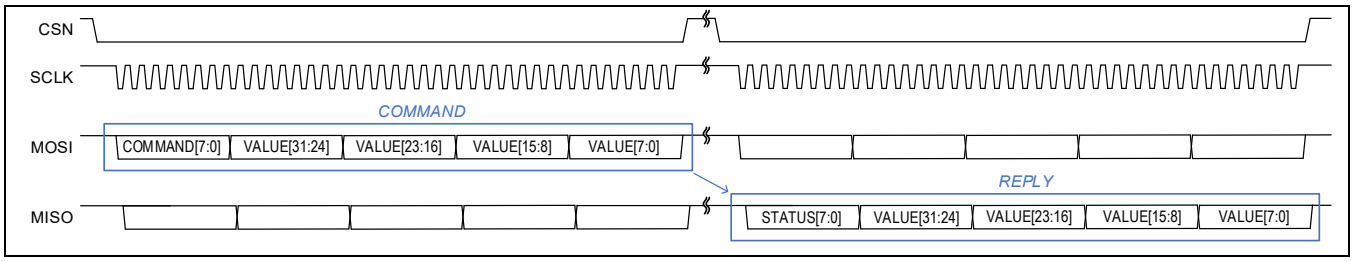

Figure 11. SPI Command and Reply Datagram in Bootloader Mode

The first reply datagram after power-up contains the SESSION_START status code in the status[7:0], the bootloader version information with the major number in value[31:24], and the minor number in value[23:16].

For more details on the content of these datagrams, see the *Bootloader* section.

As peripheral, an external flash memory may be connected to the SPI0 or SPI1 controller blocks for storing parameters or programs with the parameter mode firmware. The bootloader already supports configuration of the flash memory, including pin assignment and SPI mode selection. Optionally, the hardware CS_ can be replaced with any other GPIO.

As external Flash Windbond W25X40CL or similar devices with a compatible instruction set are supported.

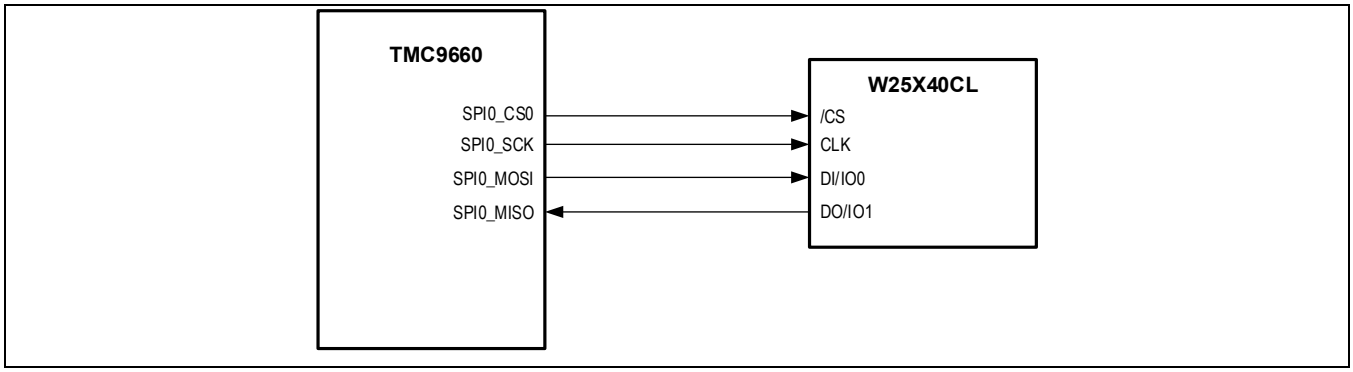

Figure 12. External Flash Memory Connected through SPI

Both SPI controller blocks support all four different SPI modes. They include 16-byte/128-bit write buffer and read buffer each—supporting transfer of up to 128 bits within one datagram. There are different chip select signals available in the GPIO Matrix with programmable settings and supported in hardware. The SPI1 offers a reduced set of functionalities compared to SPI0. SPI0 offers, in addition, automatic SPI transactions of up to two datagrams on a regular base e.g., triggered by the motor PWM in hardware. This functionality may be used to read out absolute encoder information from an external SPI encoder.

I²C

The TMC9660 includes one I²C controller for connection of external I²C peripherals e.g., I²C EEPROM. When selected, both I2C_SCL and I2C_SCL are open-drain signals and require external pull-up resistors to V_{CC_IO} for proper operation. The I²C interface supports data rates with 100kbit/s, 400kbit/s, and up to 1 Mbit/s. An external I²C EEPROM may be used for parameter and program storage with the parameter mode firmware. The bootloader already supports configuration of the EEPROM, including pin reservation/configuration. The 24LC256 or similar EEPROM devices with I²C control code 0xa, chip select bits set to zero, and separate address—low and high byte—are supported.

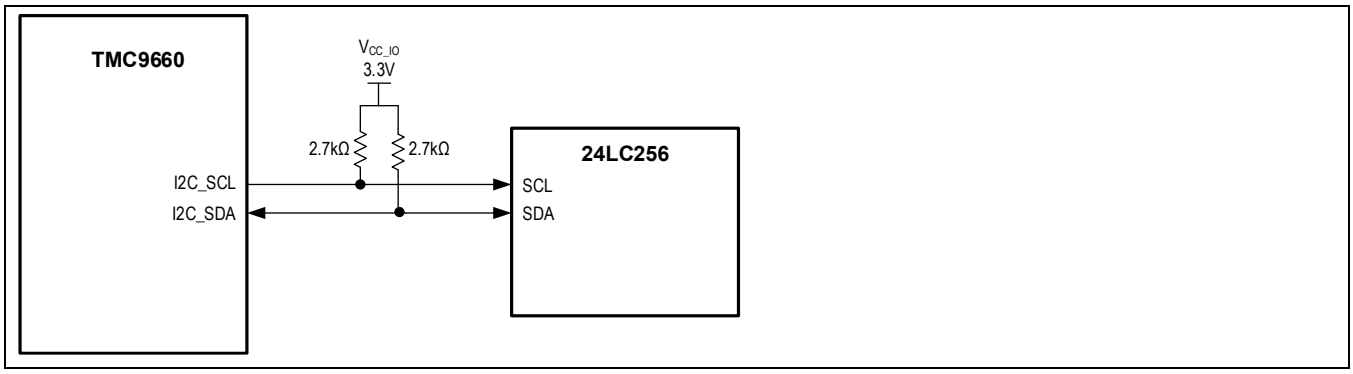

Figure 13. External EEPROM Connected through I²C

SPI Peripheral—Timing Diagram

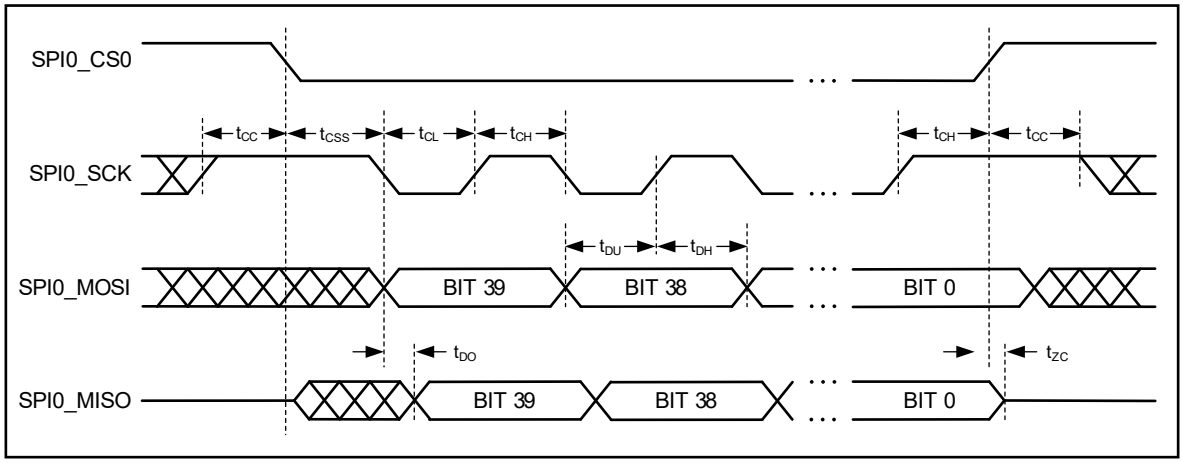

Figure 14. SPI Peripheral Timing Diagram in Bootloader Mode

SPI Peripheral—Electrical Characteristics

(V_{CC_IO} = from 3V to 5.5V, V_{SA} = from 7.7V to 80V, Junction Temperature = from -40ºC to 125ºC, SPI timings are guaranteed by design)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS	GL
SCK Valid Before or After Change of CS_	t_{CC}		T_{SCK}			ns	VI
CS_ High Time	t_{CSH}		4x T_{CLK}			ns	VI
CS_ Setup Time	t_{CSS}		100			ns	VI
SCK Low Time	t_{CL}		20			ns	VI
SCK High Time	t_{CH}		20			ns	VI

(V_{CC_IO} = from 3V to 5.5V, V_{SA} = from 7.7V to 80V, Junction Temperature = from -40ºC to 125ºC, SPI timings are guaranteed by design)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS	GL
SCK Frequency	f_{SCK}				10	MHz	VI
MOSI_ Setup Time Before SCK Rising Edge	t_{DU}		20			ns	VI
MOSI_ Hold Time After SCK Rising Edge	t_{DH}		20			ns	VI
MISO Valid Time After SCK Falling Edge	t_{DO}			21	40	ns	VI

GUARANTEE LEVEL	DESCRIPTION
I	Production Tested @ Multiple Temps
II	Production Tested @ Room Temp, Characterized @ Multiple Temps
IIsc	Production Tested via Scan @ Room Temp, Characterized via Scan @ Multiple Temps
III	Sample Tested
IV	Not Production Tested, Characterized by ATE
V	Not Production Tested, Characterized by Bench (GBDC)
VI	Internal Design Target
VII	Production Tested, Internal Only
VIII	Production Tested @ Hot, Characterized @ Multiple Temps

Motor Control Core Description

The MCC inside the TMC9660 is implemented in hardware with a high degree of software configurability. It is based on a field-oriented current controller (FOC). It supports either a three-phase brushless motor (FOC3), a two-phase stepper motor (FOC2), or a DC motor. On top of the FOC controller, it includes a complete cascaded servo control loop architecture with velocity and position proportional and integral (PI) controller and 8-point position ramp generator. For motor feedback, it includes an ADC engine interfacing to the on-chip measurement block for motor current measurement, and an encoder and hall sensor feedback engine for rotor position feedback. The MCC supports control loop speed and PWM frequencies from about 2kHz up to 100kHz.

Features Overview

- Field-oriented controller/FOC for torque and flux (PI) in hardware
  - for three-phase brushless motor (FOC3)
  - for two-phase stepper motor (FOC2)
- Velocity PI controller in hardware
- Position PI controller in hardware
- Biquad target torque filter
- Biquad velocity filter
- 8-point ramp profile generator
- ADC engine for current measurement
- Encoder feedback engine supporting ABN encoder signals and digital hall signals (including scaling and extrapolation)

Gate driver and ADCs are internally connected to the MCC. Incremental encoder signals ENC_A/ENC_B/ENC_N and digital hall sensor signals HALL_U/HALL_V/HALL_W are directly connected to the encoder feedback engine and can be configured for different package pins using the GPIO matrix. The integrated ramp profile generator directly evaluates stop and reference switch input signals STOP_L, STOP_R and HOME. These signals can also be configured for different package pins.

Field-Oriented Control

Field-oriented control/FOC—also referred to as vector control—belongs to the class of variable frequency drive (VFD) control method for brushless motors with coil windings in the stator. With the help of the rotor position angle, the motor currents through the stator coil windings are transformed into two linear time-invariant orthogonal components (vectors) for torque (I_Q) and magnetic flux (I_D) (*Figure 15*). This transformation allows using standard proportional-integral (PI) controllers and a control architecture already well-known from DC motors. The outputs of the controllers are transformed back to voltage levels and converted to PWM signals for driving the half bridges—one for each motor coil connection.

FOC is regarded as superior in terms of motor performance and energy efficiency. It supports smooth operation over the entire speed range, full torque at zero speed, and high dynamics with fast acceleration and deceleration.

While the necessary transformations required a separate digital controller in the past, the TMC9660 makes high-performance FOC-based motor control available together with the high-voltage gate driver, motor current measurement, and motor position feedback processing on a monolithic device.

For best control performance, the rotor position must be measured simultaneously with the motor current. While the internal processing delay is compensated inside the TMC9660, it is important to make sure the encoder feedback system can measure the rotor position precisely. This usually requires that the encoder is mounted directly to the motor axis and not after a gearbox or belt system which might introduce hysteresis during start/stop or when changing directions.

For three-phase BLDC/PMSM motors, the Clarke transformation converts between the three-phase signals and the two-phase signals required for the Park transformation. For two-phase stepper motors, this transformation is not necessary and bypassed when selecting the motor type. For brushed DC motors, both transformations are bypassed.

For torque control of the motor, the torque reference value (I_{QREF}) may be set directly from external bypassing the rest of the cascaded control loop architecture inside the MCC. For highest efficiency in motor control, the flux reference value is usually set to zero. In case the focus is on high-speed motor control, the additional field weakening controller may be used which then replaces the fixed reference value with the controller output.

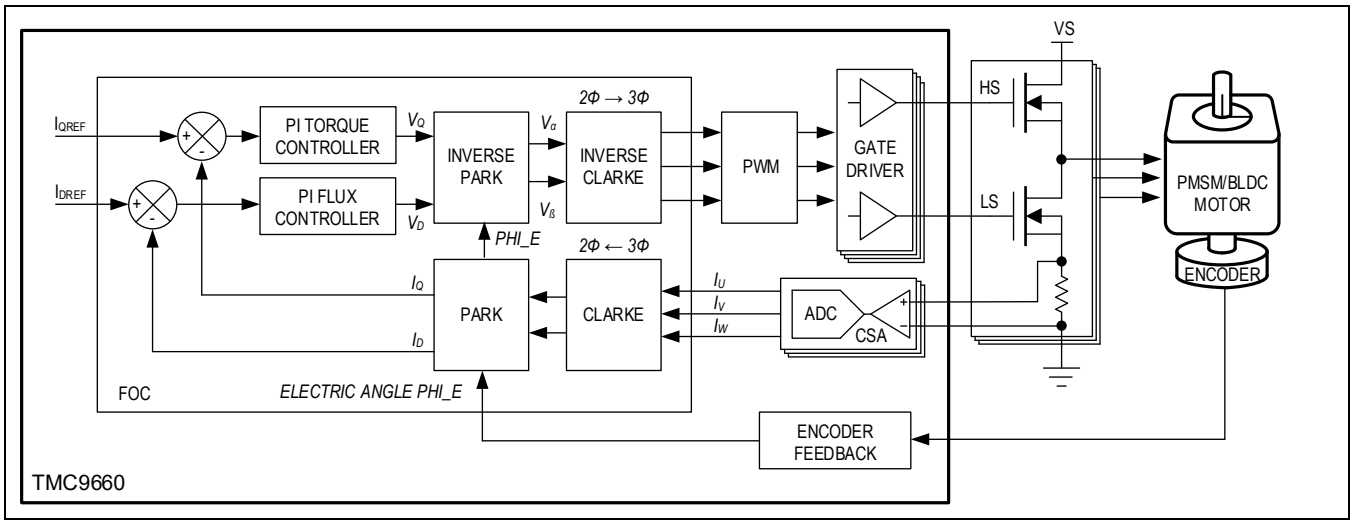

Figure 15. Field-Oriented Current Controller (FOC)

Servo Controller

The MCC inside the TMC9660 includes a servo controller with cascaded chain of position and integral (PI) controllers for torque, velocity, and position (*Figure 16*) for simplified setup and tuning. In addition, a ramp generator is available to enable smooth and fast transitions when target positions or velocities are changed.

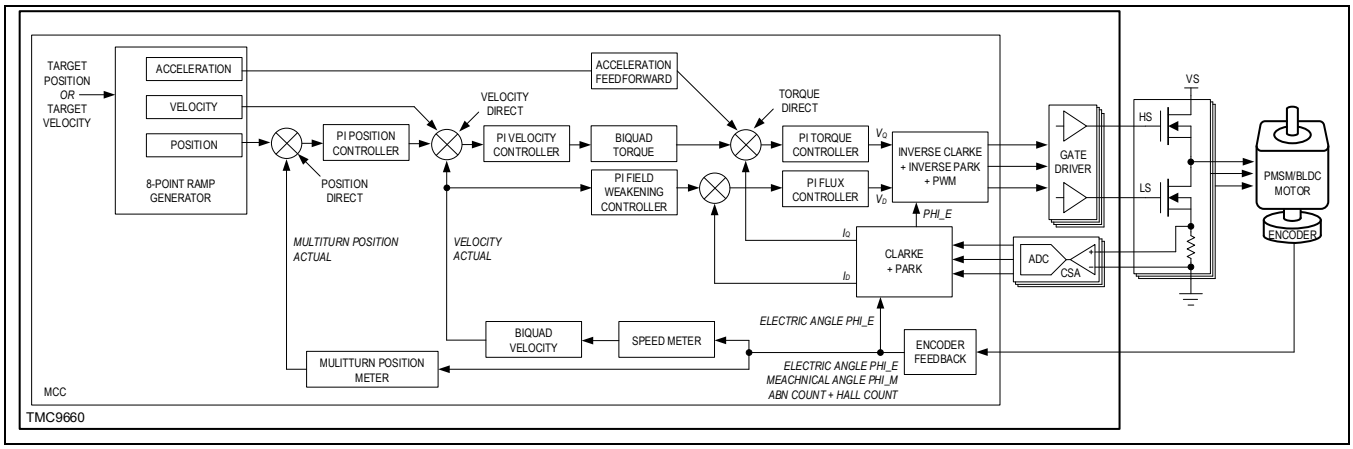

Figure 16. MCC Servo Controller

When using the complete chain after initialization, a single command from the application processor is sufficient to initiate a movement to the new target position using closed loop control and following the calculated ramp. As soon as a new target position is provided, a new ramp is calculated on-the-fly. The ramp may also be used to support the transition between different target velocities in velocity mode to avoid overshoots/oscillation of the PI controllers. However, it is not mandatory to use all servo controller parts—torque, velocity, and position target values can be set directly bypassing the other controller stages.

With the implementation in hardware, the control loop frequency of all three PI controllers can be set to the same value as the PWM frequency—up to 100kHz control loop frequency. This enables fast reaction times, high bandwidth, and superior dynamics. It is also possible to just set the torque/current controller to the same frequency as the PWM generator and the update rate of the velocity and position controllers to lower values—a fraction of the PWM frequency.

While the FOC controller requires the electric angle of the rotor for the PARK transformation, the velocity PI controller relies on the measurement of the current motor velocity (speed meter). The current motor velocity is derived from the

encoder position counter—either measuring the counting frequency (usually preferred at higher velocities) or the time interval between position updates (useful for lower velocities)—with a moving average filter for noise reduction. An additional programmable biquad filter (2nd order IIR filter) in the velocity measurement loop may be used for further reducing measurement noise.

Another programmable biquad filter (biquad target torque) is available between the velocity controller and the torque controller to help dampening resonance frequencies.

The position controller requires the multiturn absolute encoder position for position movements. The encoder used for the position controller may be the same already used for torque and velocity—or a second, separate one mounted at a different position e.g., after a gearbox (second encoder supported in parameter mode).

Ramp Generator

For position movements or motor operated in velocity mode, a ramp generator is available (*Figure 17*). This ramp generator can be configured with a start velocity (VSTART), a maximum of three different segments for the acceleration phase (A1, A2, AMAX), and three different segments for the deceleration phase (D1, D2, DMAX), and finally a stop velocity (VSTOP). The segments may have different accelerations and programmable transition velocities (V1, V2) for maximum flexibility. For high-velocity applications, the acceleration settings at high speed may be set to a lower value as motor torque declines at higher velocity. At start-up to overcome friction and inertia of the system, acceleration may be also set to a lower level while in the middle section acceleration typically can be set to a higher value for time critical movements. Also, taking friction into account, deceleration may be shorter in time than the acceleration phase. This may result in asymmetric pseudo S-shaped ramp as shown in the example diagram.

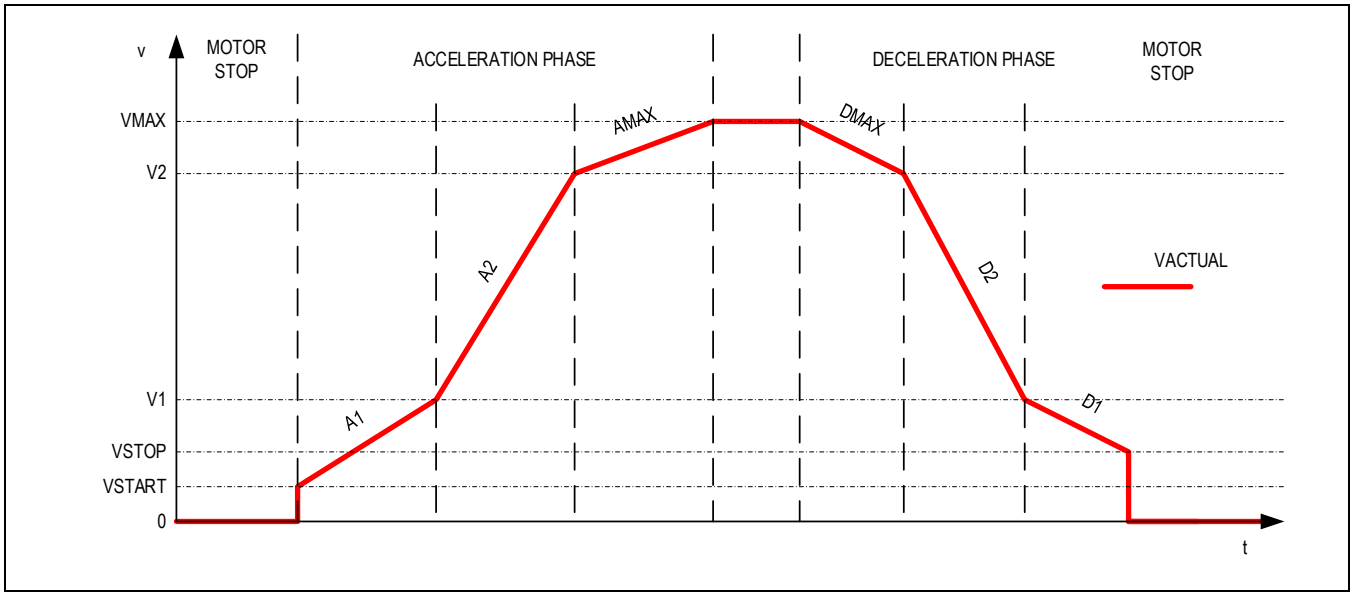

Figure 17. 8-Point Ramp Generator Configured with Pseudo S-Shaped Velocity Profile

The actual ramp is calculated on-the-fly as soon as a new target position or velocity is provided. Depending on the distance/velocity change, not all different acceleration phases may be used or the maximum velocity during movement is not reached. In case the motor is moving when a new target value is set, the ramp calculation takes the current velocity into account and calculate from thereon in real time.

The ramp generator supports up to two stop switch inputs—one for each direction (STOP_L, STOP_R). In case a stop switch is enabled and activated, the motor either decelerates with current ramp parameters to stop or stops immediately (hard stop) and any further movement in this direction is prohibited. There is a third switch input for reference search to initialize the multiturn encoder counter. The switch evaluation and ramp calculation are implemented in hardware.

Encoder Feedback

The Encoder Engine inside the MCC directly processes incremental encoder A/B/N signals (ENC_A, ENC_B, ENC_N) and digital hall sensor signals (HALL_U, HALL_V, HALL_W) (*Figure 18*).

The A/B/N decoder has a small input filter that can be enabled to ignore input pulses shorter than three system clock cycles (40MHz). Otherwise, input signals up to 10MHz are processed. The decoder counts edges on the incoming A and B channels and increments or decrements the internal encoder counter accordingly. The count direction may be inverted to compensate for different encoder-mounting options. The optional encoder N channel is expected to generate one pulse per mechanical revolution. The signal is necessary for measuring the absolute rotor angle of the motor axis for repeatability during initialization. As an alternative, hall sensor signals may be used as they also provide an absolute position information of the rotor. The mechanical angle calculation includes 24-bit counter supporting encoder resolutions up to 16,777,216 (2^24) encoder ticks or 4,194,304 (encoder ticks/4) lines per revolution. From the mechanical angle, the electrical angle is derived by dividing it with the number of motor pole pairs. A maximum of 255 pole pairs are supported. Both mechanical and electrical angles are scaled to 16-bit integer values regardless of the encoder resolution. The electrical angle is required by the PARK vector transformation as part for the FOC controller, and the mechanical angle may be used for motor velocity control.

The Encoder Feedback engine also supports digital hall sensors (HALL_U, HALL_V, HALL_W). For the hall sensor signals, a programmable digital input filter is available. Order and polarity may be changed in software. While the hall sensors provide a rather coarse granularity of the rotor position and have therefore limited use for applications requiring positioning, they are often readily available as part of BLDC motors. To allow using FOC and sinusoidal commutation also with hall sensors, the encoder engine offers an integrated hall angle extrapolator to generate higher resolution position signal. This extrapolator is suitable for higher velocities and is automatically de-activated below 60 electrical rpm.

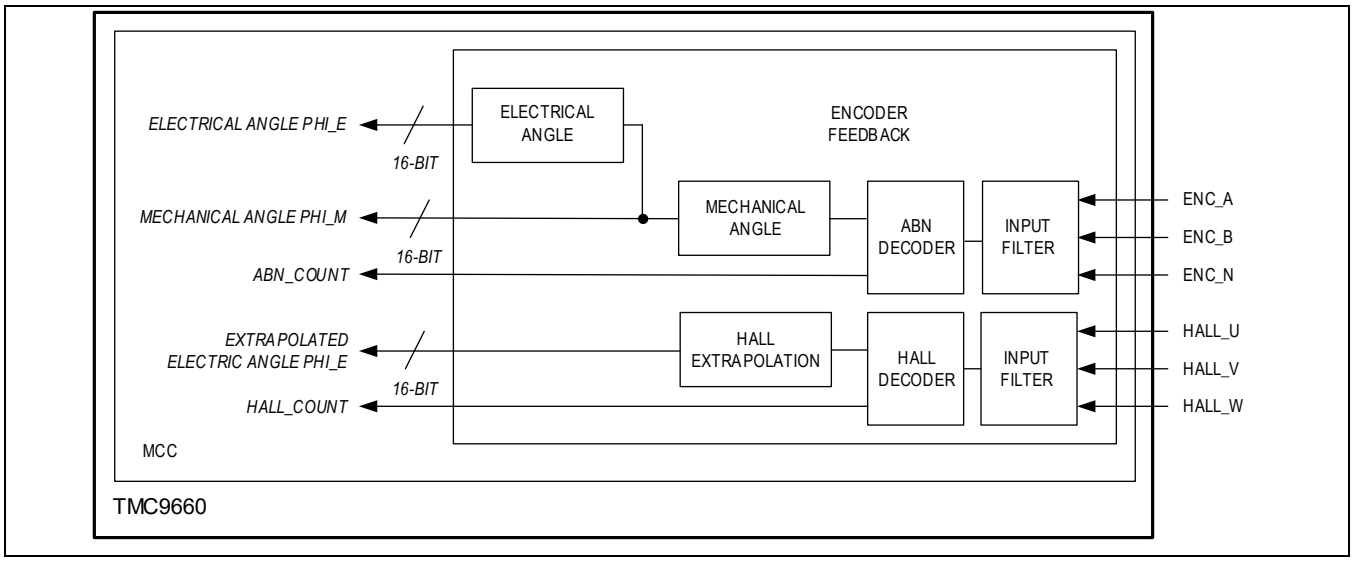

Figure 18. Encoder Feedback Engine

The encoder feedback engine focus on the encoder mounted to the motor shaft for the FOC controller. The TMC9660 also supports a second incremental A/B encoder as an option for the position controller. This must be connected to alternate function pins TIM_ADV_IN0 (encoder Channel A) and TIM_ADV_IN1 (encoder Channel B). As an alternative to the incremental A/B/N encoder, an absolute position encoder with SPI may be used. This can be connected to SPI peripheral block 0 or 1 with signals SPIx_MOSI, SPIx_MISO, SPIx_SCK, and one of the available chip-select signals SPIx_CS. For three-wire SPI, SPIx_MOSI may be omitted. These encoder options are all available in parameter mode.

Motor Types

The TMC9660 supports DC-, BLDC/PMSM-, and stepper-motors. *Table 6* shows their corresponding connection to the mid-point of the half-bridges and their needed shunt connection to the CSA. Simultaneously, only one motor is supported. If a motor type is used, which does not use all four half-bridges, the remaining half-bridge can be utilized as a connection

for electromagnetic brakes and/or a brake chopper resistor to dissipate energy while ramping down a connected motor fast.

Table 6. Motor Connection Mapping

HALF-BRIDGE	CSA CONNECTION**	DC-MOTOR	BLDC/PMSM-MOTOR	STEPPER-MOTOR
UX1	CSN0/CSP0	DC1	U	X1
VX2	CSN1/CSP1	DC2	V	X2
WY1	CSN2/CSP2		W	Y1
Y2	CSN3/CSP3	Brake/EM-Brake*	Brake/EM-Brake*	Y2

*Brake/EM-Brake are directly possible through the separated low-/high-sides. The section on the *Gate Driver Unit Description* includes an example circuit (*Figure 23*).

**This is the default assignment. The ADC assignment is changeable through the MCC.

Power Management Unit Description

Introduction

In its main configuration, the TMC9660 requires the motor supply voltage only to operate. All the voltages required by the device are efficiently generated internally by the power management unit (PMU).

Two separate supply voltage input pins, V_S and V_{SA}, are available, respectively to supply the gate driver outputs stage and the analog and digital core circuitry. This makes it possible to cut off the motor supply (V_S) for safe torque off (STO) purposes while the rest of the circuit keeps operating.

The PMU integrates a DC/DC converter (buck), a charge pump, two configurable LDOs to supply external loads, and two 1.8V LDOs to supply internal circuitry.

PMU List of Features

- Synchronous buck regulator. Up to 80V input, fixed 5.8V output, and 600mA current capability
- Charge pump (doubler). 50mA-rated load aimed to supply the GDU
- 2x configurable LDOs. 150mA-rated load each, for external loads supply
- 2x Fixed 1.8V. Regulators for internal use only

PMU Diagram TMC9660

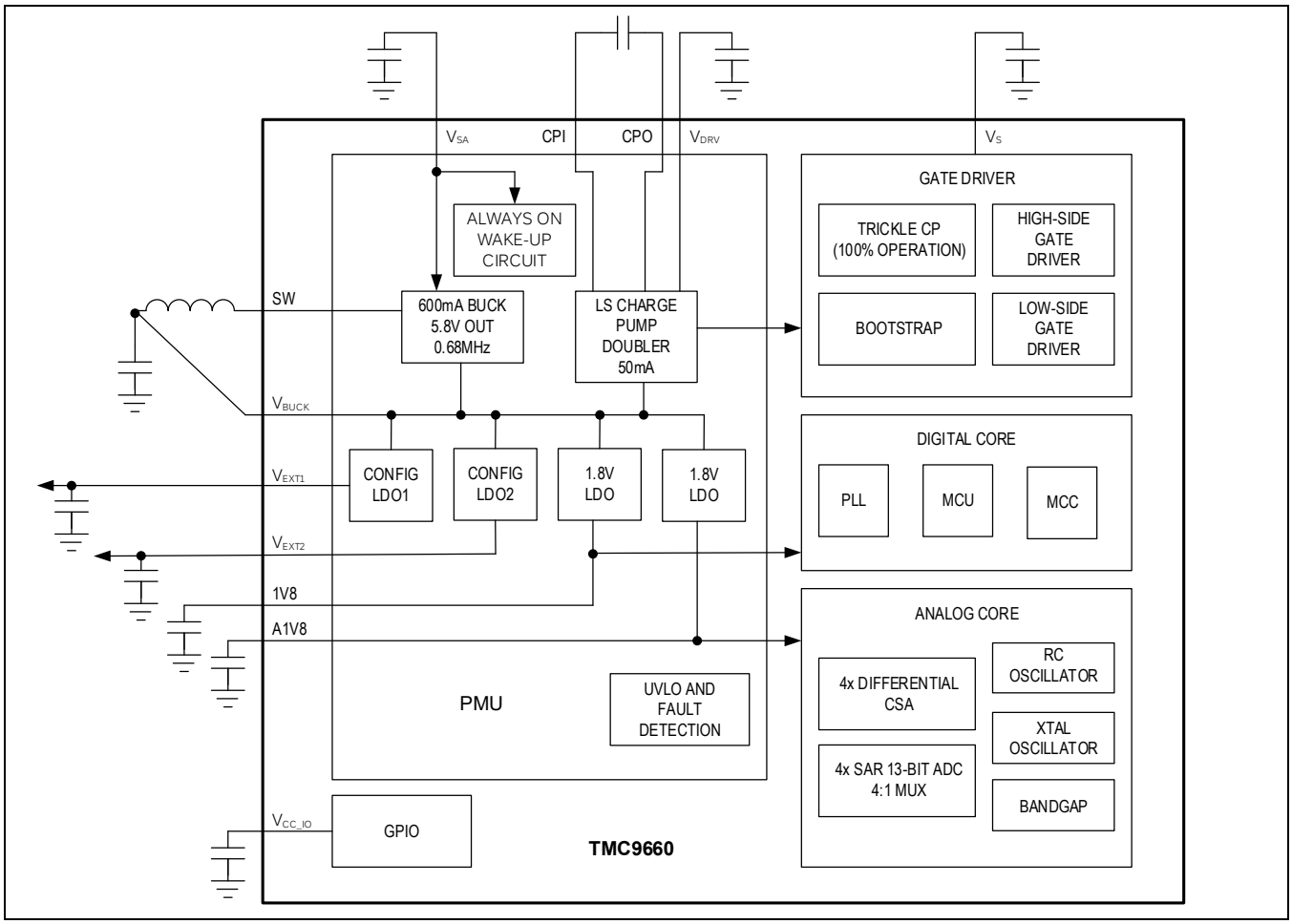

Figure 19. Power Management Unit

Power Management Unit

Table 7. PMU Regulators

V_{BUCK}	V_{EXT1}	V_{EXT2}	V_{DRV}	1V8/A1V8 INTERNAL LOAD
600mA	150mA	150mA	50mA	-
5.8V	2.5V/3.3V/5.0V	2.5V/3.3V/5.0V	11.6V-drop	1.8V

Input Supply Voltages (V_S, V_{SA}, V_{CC_IO})

The TMC9660 requires typically only two external supply voltages to operate: VS_ and V_{CC_IO}.

V_{SA} is the input supply voltage pin of the DC/DC converter and AON block. All the internal supplies and references are internally generated from buck output voltage (V_{BUCK}).

V_S is the motor voltage. V_S and V_{SA} can together be externally connected and provided from the same supply. Alternatively, a safety switch can be connected to cut V_S while keeping V_{SA} operating.

Undervoltage lockout (UVLO) comparators are integrated to put the driver in safe mode if either V_S or V_{SA} drop below the UVLO thresholds. When V_{SA} drops below its UVLO voltage threshold, the DC/DC converter is disabled, MCU is reset, all blocks (PMU, GDRV, MU) are disabled, and FAULTN pin is driven low. If V_{SA} is restored above its UVLO threshold, the system restarts enabling the DC/DC converter and then removing reset to MCU once all internal supplies are settled.

V_S UVLO disables only the gate driver, outputs are three-stated and a signal is generated for the MCU.

V_S UVLO threshold must be configured by VS_UVLO_LVL bits from 4V to 8V before enabling the gate driver. For achieving full functionality of trickle charge pump, it is recommended to set V_S UVLO rising voltage above typical 7.7V.

V_{CC_IO} is the supply voltage input for the logic input circuitry and can be supplied by the internal LDOs.

If V_{CC_IO} drops below the UVLO threshold, the FAULTN pin is driven low.

DC/DC Buck Converter (V_{BUCK}): The TMC9660 PMU integrates a DC/DC step-down (buck) regulator which efficiently converts the HV supply down to low voltages.

The buck converts the V_{SA} input voltage down to a fixed 5.8V output and can deliver up to 600mA.

It operates as a synchronous switching regulator at about 682kHz (forced PWM operation and DCM operation just in soft start).

The overall buck current load can be calculated as follows:

$$I_{LOAD\_BCK} = I_{VBUCK\_Q} + I_{LOAD\_LDO1} + I_{LOAD\_LDO2} + I_{CRGPMP}$$

where

I_{VBUCK_Q} is the total quiescent consumption due to digital and analog core LV LDOs (typ 36mA).

$I_{LOAD\_LDO\_}$ is the output current load of the VEXT_ LDO when enabled.

I_{CRGPMP} is the current due to V_{DRV} charge pump operation, and it can be calculated as follows:

$$I_{CRGPMP} = 2 \times \frac{I_{VDRV\_Q} + I_{GDRV\_SW}}{\eta_{CRGPMP}}$$

where

I_{VDRV_Q} is the quiescent consumption of gate driver unit (typ 2mA).

I_{GDRV_SW} is the switching current consumption due to external power FETs commutation (see *Gate Driver Unit Description* section for its calculation).

η_{CRGPMP} is the V_{DRV} charge pump efficiency.

The DC/DC provides by itself a robust overcurrent protection (Buck OCP) scheme that protects the device under overload and output short-circuit conditions. The Buck OCP scheme protects the device by using a hysteretic control of the inductor current that avoids the inductor current run-away condition. In hysteretic control, whenever the inductor peak current exceeds an internal peak current limit of typical 1A ($I_{BCKPKILIM\_PK}$), the high-side MOSFET is turned off and the low-side MOSFET is turned on. When the inductor current reduces by 0.4A (typ), the low-side MOSFET is turned off and the high-side MOSFET is turned on. In addition, if the V_{BUCK} node voltage drops below 3.48V (V_{BUCK_FBUV}) due to a fault condition any time after soft-start is completed, hiccup mode is activated. In hiccup mode, the converter is protected by suspending switching for a hiccup timeout period of 96ms. Once the hiccup timeout period expires, a soft-start is attempted again. The hiccup mode of operation ensures low-power dissipation under output short-circuit conditions.

If V_{BUCK} drops below the V_{BUCK_UVLO} threshold, then the MCU is reset and all blocks (external LDO, charge pump, gate driver, MU) are disabled. When V_{BUCK} voltage is above UVLO threshold and DC/DC converter is functional, the MCU reset is removed, FAULTN open-drain pin is released, and bootloader operation occurs.

LV Regulators: Two LV linear regulators are integrated to supply the digital core and analog core, respectively, of the TMC9660.

Bypass capacitors must be connected to the corresponding pins 1V8 and A1V8.

These regulators are intended to supply internal blocks only and should not be used to supply external loads.

If A1V8 drops below UVLO threshold (VA1V8_UVLO), the MCU is reset and all other MCC core blocks (gate driver, charge pump, external LDOs) are disabled.

While if 1V8 also drops below its UVLO threshold (V1V8_UVLO), then the DC/DC converter is disabled.

Since 1V8 supplies the digital core of the TMC9660, if 1V8 drops below the power-on reset threshold (V1V8_POR) then all logic is reset.

Charge Pump and Bootstrap (V_{DRV}, BS): An integrated 50mA capable charge pump doubles the DC/DC converter output to provide 11.6V gate drive voltage to the gate drive unit.

V_{DRV} supplies the low-side gate driver and recharges the bootstrap capacitors of the high-side gate driver (see *Gate Driver Unit Description* section).

The integrated charge pump allows full drive capability of the external FETs in an extended input voltage range (from 7.7V to 80V).

It is possible to disable the charge pump and supply the GDU from an external independent supply voltage. In this use case, a voltage supply equal or greater than V_{BUCK} must be connected to pin V_{DRV} to not exceed Absolute Maximum rating of V_{DRV} pin. The flying capacitor between pins CPI and CPO must be removed.

The V_{DRV} voltage is internally monitored. If V_{DRV} drops below the V_{DRV} UVLO (UVLO_VDRV), the GDU is disabled, and the output FETs are kept passively off.

Similarly, if the bootstrap floating voltage drops below the BST UVLO (BST_VDRV), the GDU is disabled, and the output FETs are kept off with a 12kΩ active pull-down resistor until BST voltage is above 1.2V. To restart the gate driver after a BST UVLO event or at the initial startup after power-up or hibernate mode, the bootstrap capacitor C_{BST} must be fully charged. To charge the bootstrap capacitors, the BST UVLO protection needs to be disabled and LS FETs must be kept on until their charge through the internal bootstrap switch is completed. In parameter mode, this sequence is handled automatically.

To avoid overloading V_{DRV} charge and consequently the buck converter, the internal bootstrap switch has a current limitation I_{LIMBS} that can be programmed from a minimum of 11mA to a maximum of 0.38A (typical values). For the initial charge of $C_{BST,}$ it is recommended to use the minimum current setting and select the maximum current limitation before starting gate driver operation. Initial bootstrap recharge time can be calculated as follows:

$$T_{BSCRG} = C_{BST} \frac{VDRV}{I_{LIMBS}}$$

For 1µF C_{BST} and minimum bootstrap current limitation, recharge time is about 1ms, but consider using twice this value 2ms to include components, parameters, and temperature variation.

All UVLO faults are signaled to internal MCU for fault handling.

Configurable LDOs (VEXT1, VEXT2): The PMU includes two configurable 150mA-capable LDOs which operate under the buck regulator. They are aimed to supply external loads connected to the Output pins VEXT1 and VEXT2.

These regulators can be independently enabled and disabled in case the external loads must be shut down to save power. The LDO outputs are user configurable and can be set independently at 2.5V, 3.3V, and 5V.

To limit the in-rush current on VBUCK due to the charge of LDOs output capacitance, the initial start-up of the two LDOs are digitally controlled to ramp up one at the time with a configurable slew rate. The ramp-up time (t_{RAMP}) of the LDOs can be programmed from 0.37ms to 3ms to not exceed DC/DC converter's maximum current limitation.

$$I_{in-rush} = C_{LDO} \frac{V_{LDO}}{t_{RAMP}}$$

LDOs are short circuit protected with autoretry circuitry. Moreover, thermal shutdown protections are integrated. The temperature is sensed close to the LDO output stages by local thermal sensors.

The Logic Supply Input pin (VCC_IO) which sets the reference voltage for logic I/O can be provided by an external voltage supply or it can be connected to either V_{EXT1} or V_{EXT2} in standalone single-supply application. In the second case, bootloader OTP must be previously programmed to enable the LDOs and the RSTN pin must be pull-up from VBUCK or by a small external POR circuit. The *Reset (RSTN)* section gives useful information about its supply.

Power Management Unit Electrical Characteristics

(V_{SA} = from 7.7V to 80V, V_S = from 7.7V to 70V, Junction Temperature = from -40ºC to 125ºC, Unless otherwise noted, Limits over the operating temperature range and relevant supply voltage range are guaranteed by design and characterization. Specifications marked "GBD" are guaranteed by design and not production tested)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS	GL
BUCK REGULATOR (V_{BUCK})							
Output Voltage Regulation	V_{BUCK_REG}		5.71	5.80	5.89	V	II
Output Current Load	I_{BUCK_LOAD}		0.0		0.6	A	II
Buck PWM Switching Frequency	$f_{BUCK\_SW\_PWM}$	External CLK/XTAL selected	675.0	681.8	688.6	kHz	II
		Internal Clock selected	650	681.8	725		II
Power Efficiency	P_{BCK_EFF}	I_{VBUCK} = 600mA VSA = 24V		91		%	VI
Low-Side FET On Resistance	$R_{BUCK\_ON\_LS}$	I_{SW} = -0.3A		240	435	mΩ	II
High-Side FET On Resistance	$R_{BUCK\_ON\_HS}$	I_{SW} = +0.3A		465	870	mΩ	II
SW Leakage Current	$I_{BUCK\_SW\_LKG}$	Buck disabled, V_{VSA} = 80V, V_{SW} = [0, 80]V	-50		+50	µA	II
Soft-Start Time	t_{BUCK_SS}			1.0		ms	II
Peak Limit Peak Current Threshold	$I_{BUCK\_PKILIM\_PK}$	I_{SW} rising	0.85	1.00	1.15	A	II
Peak Limit Valley Current Threshold	$I_{BUCK\_PKILIM\_VLY}$	I_{SW} falling	530	600	670	mA	II
Undervoltage Threshold Causing HICCUP	V_{BUCK_FBUV}	V_{VBUCK} falling	0.575* V_{BUCK_REG}	0.600* V_{BUCK_REG}	0.625* V_{BUCK_REG}	V	II
	$V_{BUCK\_FBUV\_HYS}$	V_{VBUCK} rising	0.6*V_{BUCK_REG}	0.66*V_{BUCK_REG}	0.733* V_{BUCK_REG}		II
HICCUP Retry Timeout	$t_{BUCK\_HCUP\_RETRY}$			96		ms	II
V_{BUCK} UVLO Threshold	V_{BUCK_UV}	V_{VBUCK} rising	2.65	2.70	2.75	V	II
			2.60	2.65	2.70		II
CHARGE PUMP (V_{DRV})							
Output Voltage	V_{CP_VDRV}	C_{FLY} = 220nF effective value; I_{VDRV} = 0mA		2*V_{BUCK}		V	II
Output Current Load	I_{CP_LOAD}		0		50	mA	II

(V_{SA} = from 7.7V to 80V, V_S = from 7.7V to 70V, Junction Temperature = from -40ºC to 125ºC, Unless otherwise noted, Limits over the operating temperature range and relevant supply voltage range are guaranteed by design and characterization. Specifications marked "GBD" are guaranteed by design and not production tested)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS	GL
Output Impedance	R_{CP_DRV}	C_{FLY} = 220nF effective value			7.0		Ω	II
Switching Frequency	f_{CP_SW}				$f_{BUCK\_SW\_PWM}$		kHz	II
Soft-Start Time	t_{CP_SS}	C_{TANK} = 4.7µF I_{VDRV} = 0mA	Charge pump settling time-up to 2x V_{BUCK}		360		µs	II
Power Efficiency	P_{CP_EFF}	I_{VDRV} = 50mA	C_{FLY} = 220nF effective value		95.9		%	VI
V_{DRV} UVLO Threshold	V_{VDRV_UV}	Rising		6.3	6.45	6.6	V	II
		Falling		6.1	6.3	6.4		II
BST UVLO Threshold	V_{BST_UV}	Rising		5.2	6.0	6.52	V	II
		Falling		4.9	5.8	6.24		II
LINEAR REGULATORS FOR EXTERNAL LOADS (1V8, AV18)								
1V8 Voltage Regulation	V_{1V8}			1.8	1.9	2	V	II
A1V8 Voltage Regulation	V_{A1V8}			1.72	1.82	1.92	V	II
1V8, A1V8 UVLO Threshold	$V_{A/1V8\_UV}$	Rising		1.6	1.64	1.7	V	II
		Falling		1.53	1.58	1.63		II
1V8 POR Threshold	V_{1V8_POR}	Falling		1.08	1.21	1.35	V	II
LINEAR REGULATORS FOR EXTERNAL LOADS (VEXT1, VEXT2)								
Output Voltage	V_{OUT}	2.5V setting		2.4375	2.5	2.5625	V	II
		3.3V setting		3.2175	3.3	3.3825		II
		5V setting		4.75	5	5.125		II
Output Voltage Accuracy	V_{OUT_ACC2}	V_{BUCK} = 5.8V, I_{OUT} = 0.1mA to I_{OUT}(MAX)		-2.5		2.5	%	II
Maximum Output Current	I_{OUT}(MAX)			150			mA	II
Load Regulation		V_{BUCK} = 5.8V, I_{OUT} = 0.1mA to I_{OUT}(MAX)			0.1		%	II
Output Current Limit	I_{LIM}	V_{OUT} = 90% of V_{OUT}(TARGET)		150	175	225	mA	II
Overcurrent Protection Shutdown Time	T_{OVC}	From overcurrent detection to LDO shutdown			200		µs	II
Overcurrent Protection Autoretry Time	T_{RETRY}	Rising Threshold			2		ms	II
EXTERNAL SUPPLY UNDERVOLTAGE LOCKOUT (UVLO)								
V_{SA} UVLO Threshold	V_{VSA_UV}	Rising		7.35	7.50	7.65	V	II
		Falling		7.15	7.30	7.45		II
V_S UVLO Threshold Configurable Range	V_{VS_UV}	Rising		4.0		8.0	V	II
	V_{VCCIO_UV}	Rising		1.2	1.6	2.0	V	II

(V_{SA} = from 7.7V to 80V, V_S = from 7.7V to 70V, Junction Temperature = from -40ºC to 125ºC, Unless otherwise noted, Limits over the operating temperature range and relevant supply voltage range are guaranteed by design and characterization. Specifications marked "GBD" are guaranteed by design and not production tested)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS	GL
V_{CC_IO} UVLO Threshold		Falling	1.1	1.5	1.8		II

GUARANTEE LEVEL	DESCRIPTION
I	Production Tested @ Multiple Temps
II	Production Tested @ Room Temp, Characterized @ Multiple Temps
IIsc	Production Tested via Scan @ Room Temp, Characterized via Scan @ Multiple Temps
III	Sample Tested
IV	Not Production Tested, Characterized by ATE
V	Not Production Tested, Characterized by Bench (GBDC)
VI	Internal Design Target
VII	Production Tested, Internal Only
VIII	Production Tested @ Hot, Characterized @ Multiple Temps

Gate Driver Unit Description

Introduction

The TMC9660 features a 70V application specific smart gate drive circuit tailored for motor control applications. It can drive up to four external NMOS+NMOS half-bridges supporting BLDC/PMSM, Stepper, or DC-Motors Drive applications.

It features user configurable slope control for EMI mitigation, a smart sequencer for anti-dV/dt induced turn on, overcurrent and gate short protections.

The gate driver topology results in high efficiency and performances.

GDU List of Features

- 70V gate driver for up to four external half-bridges with NMOS+NMOS topology
- 11.6V gate drive FETs
- Up to 1A (source)/2A (sink) current
- Programmable current drive for slew rate control and EMC reduction
- Smart gate drive sequencing with anti-dV/dt induced turn ON
- Efficient bootstrap-based topology with integrated bootstrap switch
- Trickle charge pump to prevent bootstrap voltage drop in static conditions
- Adaptive break before make circuitry minimizes dead zone
- Matched high-side and low-side propagation delay
- Overcurrent and short protections
- Overcurrent protections based on shunt resistor sensing
- Gate short detection

Detailed Description

Gate Driver Topology

Figure 20 shows a simplified diagram of one single gate driver channel integrated into the TMC9660.

The gate driver is intended to drive a pair of external power NMOS connected in half-bridge configuration. The controller outputs the logic signals to dynamically change the strength of the gate driver voltage and control the slope of the motor phase voltage as described in the following sections.

A common voltage supply node (V_{DRV}) supplies both the low-side gate driver and the high-side gate driver ensuring the same voltage drive for both the external FETs. V_{DRV} is generated by an internal charge pump which doubles the buck output voltage. Thanks to the doubler circuit, the TMC9660 gate driver unit can operate down to very low V_S input supply voltages.

The high-side gate drive topology is based on bootstrap. The bootstrap capacitor is recharged from V_{DRV} regulator every time the BM_ node (half-bridge output) is commanded low. The bootstrap capacitor supplies the high-side gate driver circuitry. An internal bootstrap switch automatically disconnects the bootstrap capacitor from the V_{DRV} regulator when the output is commanded high.

This topology ensures optimal performances and low PWM distortion as it results in excellent matching between high-side and low-side drive circuitry. It also results in higher efficiency compared with topologies based on high-voltage charge pumps.

Since the bootstrap capacitor is recharged only when the output is driven low, the TMC9660 MCC controller can be configured to limit the PWM duty cycle when the motor spins to ensure full recharge and correct current sensing measurements.

The 100% duty cycle operation is still supported thanks to the fully integrated trickle charge pump. The trickle charge pump delivers a small current which is not sufficient to supply the high-side gate driver during normal operation but, instead, is aimed to prevent bootstrap voltage drop due to the leakage currents of the driver and external FETs under static conditions. The maximum tolerable leakage is 2µA per phase.

The required gate current for each FET can be approximatively calculated by the following formula:

$$I_{GATE} = f_{PWM} \times Q_G$$

where

Q_G is the total gate charge and f_{PWM} is the chopping frequency.

Since the doubler, which generates the drive voltage supply (V_{DRV}), is capable of up to 50mA, the following relation must be satisfied:

$$N \times f_{PWM} \times Q_G \leq 50mA$$

where

N is the total number of external FETs (in particular N = 6 for three-phase BLDC/PMSM motors and N = 8 for two-phase steppers).

For example, when driving a BLDC motor (N = 6) at f_{PWM} = 25kHz, the Equivalent Total Gate Charge must be less than 300nC.

Integrated active and passive pull-down resistors avoid crossbar conduction of the external half-bridge when the gate driver outputs are three-stated or when the part is in low-power mode. No external pull-down resistors are needed.

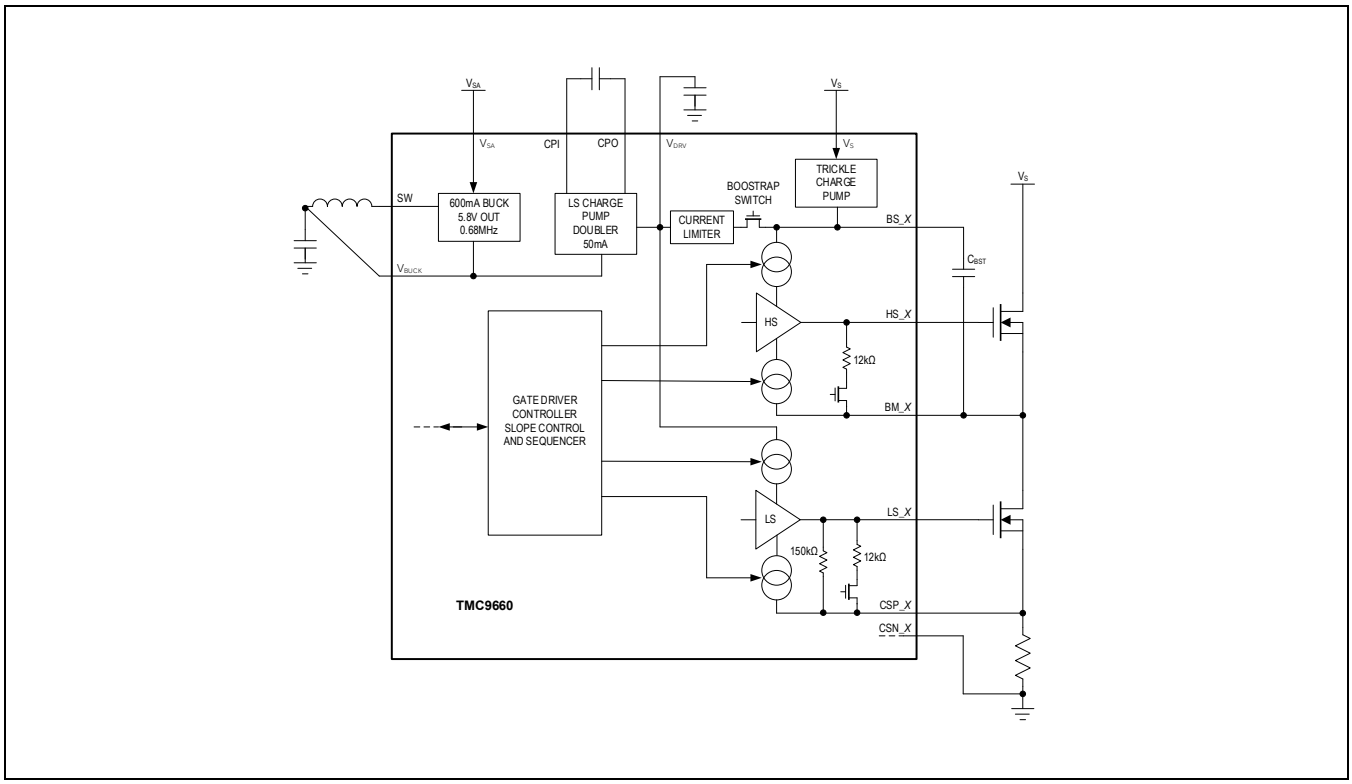

Figure 20. Gate Driver Topology Showing One Half-Bridge

Slew Rate Control

The gate driver output currents are user programmable to enable slope control of the motor phase voltages. This eliminates the external gate resistors which are commonly used for this purpose. Moreover, source and sink currents can be independently configured for maximum flexibility.

In general, slower edges result in less radiated emission (less harmonic content of the PWM signal) but worse efficiency (greater switching losses). The TMC9660 integrated programmable slope control permits design optimization in software without PCB modifications.

The TMC9660 gate drive unit operates approximately as a programmable current source. Driver currents can be chosen among 16 levels ranging from 50mA to 2A and 25mA to 1A for the Sink current and Source current, respectively. See *Table 8* and *Table 9*.

Output UX1, VX2, and WY1 share the same drive current setting whereas output Y2 can be programmed independently from the others.

Table 8. I_{SOURCE} Table

SETTING	I_{SOURCE} [mA]	SETTING	I_{SOURCE} [mA]
0	25	8	290
1	50	9	360
2	80	10	430
3	105	11	500
4	135	12	625
5	160	13	755
6	190	14	885
7	215	15	1000

Table 9. I_{SINK} Table

SETTING	I_{SINK} [mA]	SETTING	I_{SINK} [mA]
0	50	8	580
1	100	9	720
2	160	10	860
3	210	11	1000
4	270	12	1250
5	320	13	1510
6	380	14	1770
7	430	15	2000

Gate Driver Sequencer

A dedicated state machine controls the gate driver operation resulting in optimal performances and great robustness. This is obtained by dynamically changing the gate driver strength during the commutation sequence. The preprogrammed Source and Sink current (I_{SOURCE}, I_{SINK}) are output only when necessary to determine the output slope. Otherwise, weak and strong settings are used to improve performances and robustness of the solution.

Figure 21 and *Figure 22* show the TMC9660 gate driver operation when the adaptive dead time/DT is activated and not activated, respectively. With reference to the figures, see the following:

I_{SOURCE} = preprogrammed Source current (ranging from 25mA to 1A)

I_{SINK} = preprogrammed Sink current (ranging from 50mA to 2A)

WEAK PU = 10mA

WEAK PD = 35mA

STRONG PD = 2A

For the sake of simplicity, the figures assume that the external power stage is driving a purely resistive load. The actual behavior with inductive loads, such as DC motors, is different because the reactive load and the fly-back currents would influence the rising and/or the falling transitions.

Under steady state, the gate drive strength is set to a minimum. Weak pull-up/down currents ensure the correct output status. This condition protects the part in case of external short circuits on the gate output pins and reduces internal power consumption.

If adaptive mode is enabled, when a commutation signal is received, the gate of the formerly-ON external FET is first discharged with the preset I_{SINK} current. An ultra-fast comparator monitors the gate voltage and outputs a logic signal when the gate voltage gets close to zero or the programmed TIMEOUT_SNK time expires.

This signal enables the next phase where an additional programmable dead time can be added to increase the safety margin before starting the gate charge of the opposite FET. The programmable DT can be programmed from 8ns to 2.1µs with 256 steps. This adaptive dead-zone method ensures safe operation and avoid crossbar conduction while minimizing the deadzone compared with fixed time, dead-time approaches. During the dead time, weak pull downs are enforced on both the high-side and the low-side FETs.

The gate driver can also be configured in nonadaptive mode. In this case, the gate driver controller does not monitor the gate voltage and the sequencer enforces dead time after the programmed TIMEOUT_SNK (see *Figure 22*).

TIMEOUT_SNK can be programmed from 42ns to 4.3µs.

High-side and low-side gate drive propagation delays are matched which also helps minimizing dead-zone requirements. The minimum fixed dead time is 65ns.

Subsequently, the gate of the formerly OFF external FET is charged with the preset I_{SOURCE} current. During this phase, a strong pull down is applied to the opposite FET to prevent dV/dt induced turn ON.

This phase ends when a programmable TIMEOUT_SRC has elapsed.

It must be observed that TIMEOUT_ does not limit the minimum PWM duty cycle. A change in the control signal coming prior to the end of the TIMEOUT_ causes the sequencer to move to the next state.

Timeout shall be set long enough to permit fully charge/discharge of the gate. TIMEOUT is also used as blanking time to detect gate short conditions as discussed in detail in the *Gate Protection—VGS Protection* section.

When the charge phase is completed, the steady state status is resumed and with weak pull-up/down currents are applied.

Output UX1, VX2, and WY1 share the same drive current setting whereas output Y2 can be programmed independently from the others.

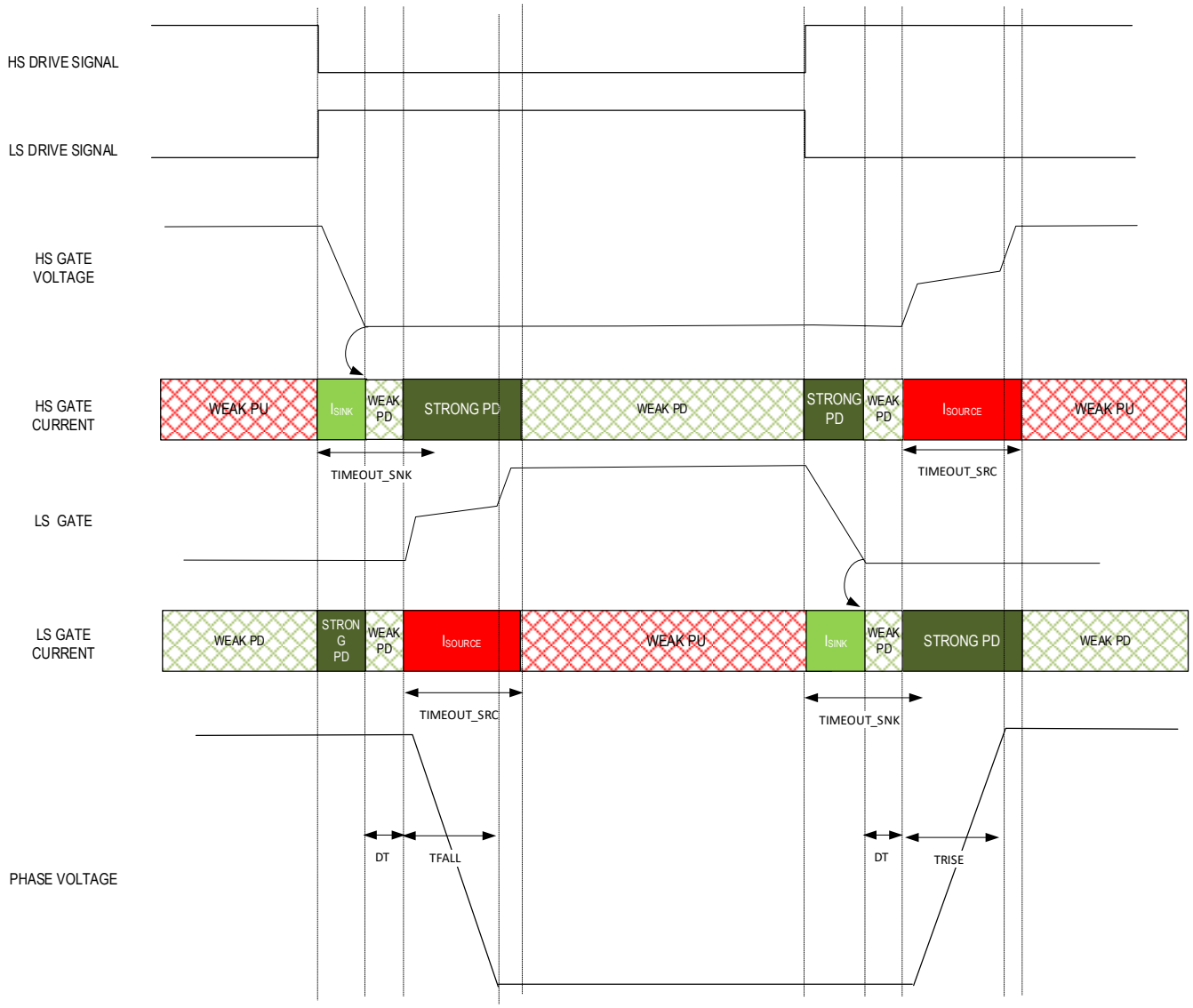

Figure 21. Gate Driver Sequencer Diagram—Adaptive

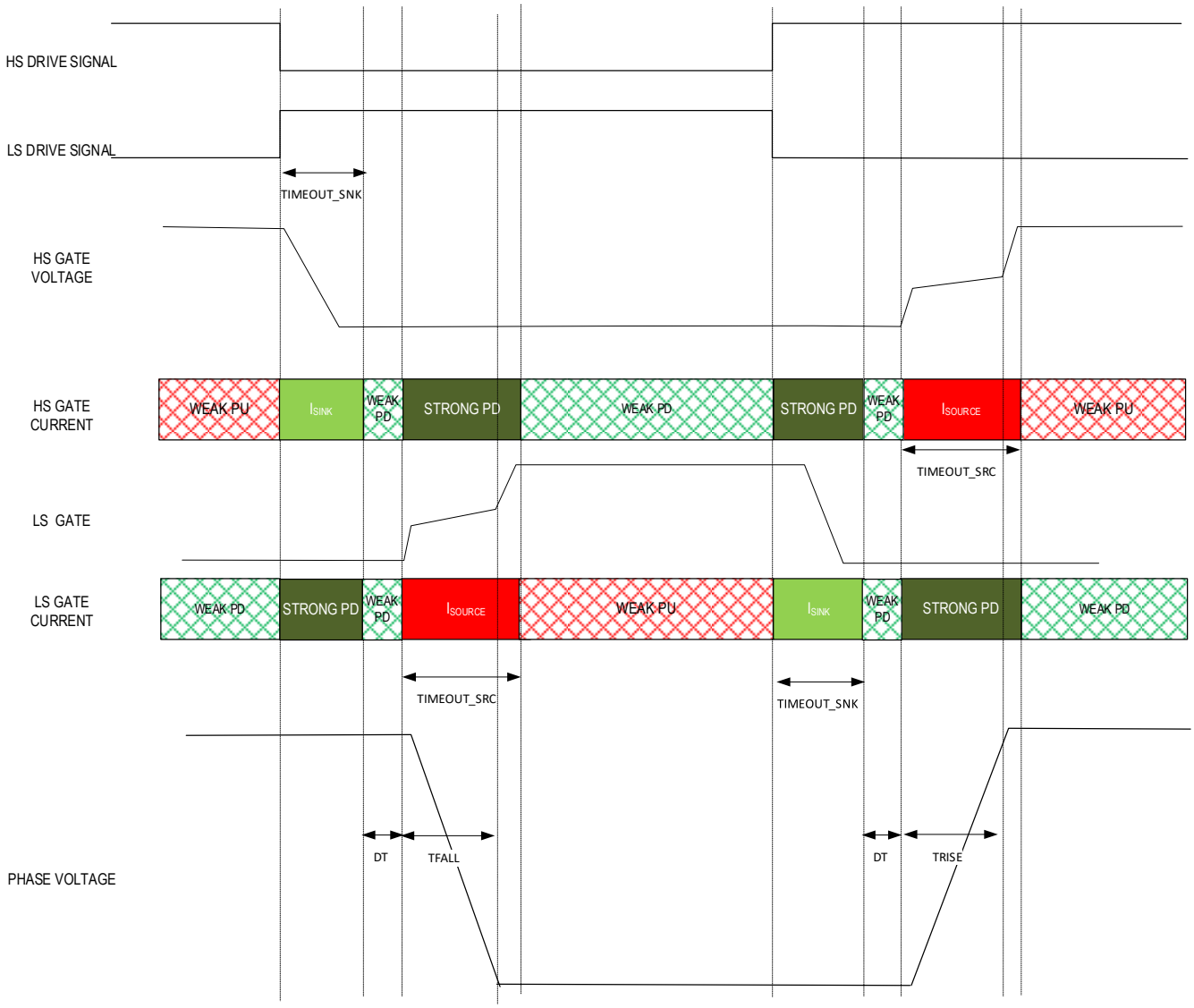

Figure 22. Gate Driver Sequencer Nonadaptive

Implementing Brake Chopper and Electromechanical Brake/Release Drive with TMC9660

In three-phase motor applications (BLDC/PMSM) or DC motor applications, the fourth half-bridge gate drive channel (Y2) can be conveniently used to drive a motor brake/release and/or a brake chopper. A typical application diagram is shown in *Figure 23*.

The brake chopper is intended to dump the motor supply in case it gets pumped up above a critical voltage threshold because of fast decelerations or regenerative braking.

The motor electromechanical brake/release is used for instance in Robotics to block the actuator in a position which is safe for the operator.

To address these use cases, the Smart Gate State machine (see *Gate Driver Sequencer*) can be disabled and the channel Y2 can be used to drive two independent LS NFET as shown in *Figure 23*.

Brake Chopper

With reference to the figure, HS_Y2 is used to drive an external NFET of the chopper circuit. The resistor is aimed to dissipate the motor regenerating power. Note that in this use case, BS_Y2 is directly connected to V_{DRV} and BM_Y2 is connected to GND.

In its simpler implementation, the brake chopper is available in parameter mode as follows:

- The supply voltage (V_S) is continuously monitored and measured.
- As soon as the supply is greater than a programmable threshold (VS_max), the external FET is turned ON and the energy gets dissipated by the external resistor.
- When the supply voltage drops below the negative threshold (VS_max - VS_Hysteresis), the FET is turned off and normal operations are restored.

Electromechanical Brake Drive

Figure 23 also shows how LS_Y2 can be used to drive an electromechanical brake/release circuitry. The coil current can be monitored by sensing the drop on an external sense resistor. A two-level current control scheme (excitation and hold current) can be used to optimize the efficiency. This feature is natively supported in parameter mode.

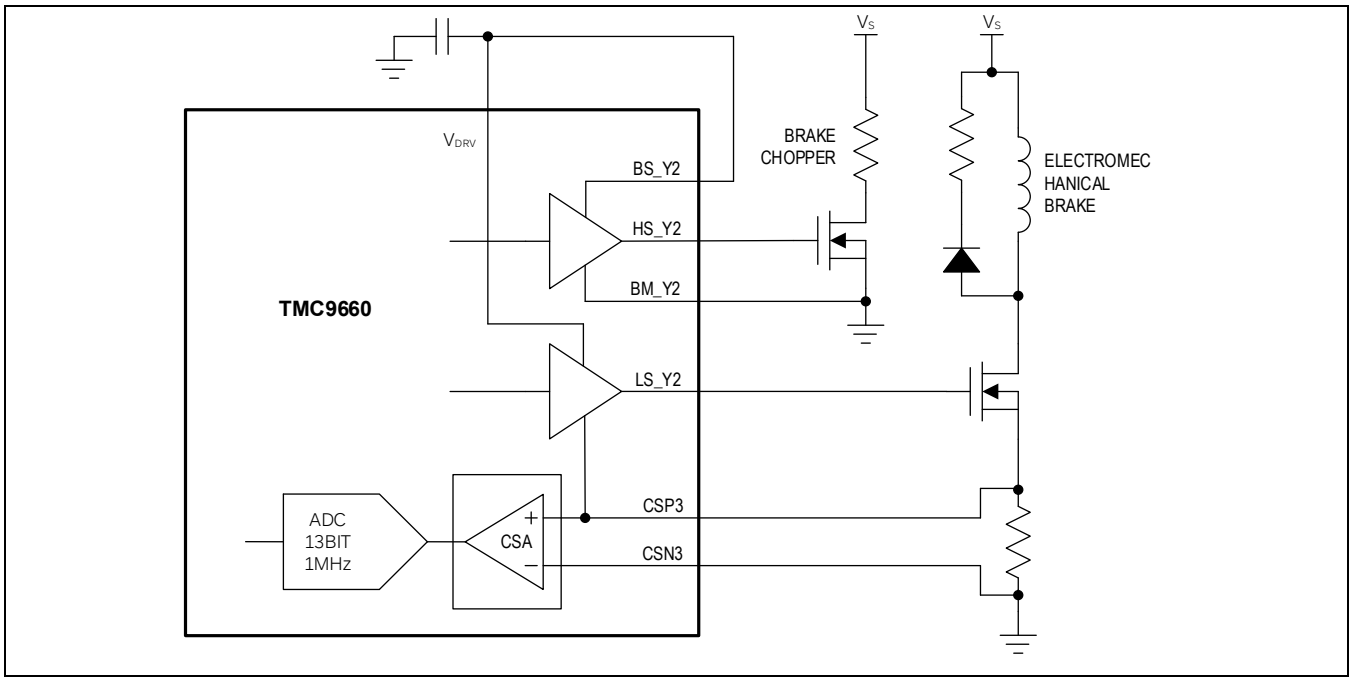

Figure 23. Brake Chopper and Electromechanical Brake

Driver Protections

The gate driver unit integrates hardware protection circuitries for safe operations in a harsh environment. These include highly configurable overcurrent protections of the external FETs as well as short protections of the gate driver outputs. See *Protections Description* section for a detailed description.

Gate Driver Unit Electrical Characteristics

(V_{SA} = from 7.7V to 80V, V_S = from 7.7V to 70V, Junction Temperature = from -40ºC to 125ºC, Limits over the operating temperature range and relevant supply voltage range are guaranteed by design and characterization. Specifications marked "GBD" are guaranteed by design and not production tested)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS	GL
GATE DRIVER UNIT (GDU)								
Drive Output Voltage	V_{DRV}	IDRV = 0 to 50mA - Total charge pump current	V_{BUCK} = 5.8V	11	11.3	12	V	II
HS Gate Output Voltage	VGD_HS	Gate Current 1mA			BS_ - 0.2		V	II
Bootstrap Switch Resistance	$R_{ONBS\_}$				15	30	Ω	II
Bootstrap Switch Maximum Current Limit	$I_{LIMBS\_}$			0.2	0.38		A	II
HS Gate Output Voltage	VGD_HS	Trickle Charge pump enable All HS ON 100%		7.5		11.5	V	II
LS Gate Output Voltage	VGD_LS	Gate Current 1mA			V_{DRV} - 0.2		V	II
Gate Output Sink Current	I_{SINK}	ISINK MIN			50		mA	II
		ISINK MAX			2		A	II
Gate Output Source Current	I_{SRC}	ISRC_MIN			25		mA	II
		ISRC MAX			1		A	II
Weak Pull-Down Current	I_{WEAKPD}				35		mA	II
Weak Pull-Up Current	I_{WEAKPU}				10		mA	II
LS Passive Pull-Down Resistor	R_{PDLS}			90	145	200	kΩ	II
LS Three-State Pull-Down Active Resistor	R_{PDALS}	All drivers disabled, 1V8 > POR1V8, GDRV_PD_DIS = 0		7	11	15.5	kΩ	II
HS Three-State Pull-Down Active Resistor	R_{PDAHS}	BS_- BM_ > 2V		8	12	16	kΩ	II
Maximum Configurable Dead Time	DT_MAX	DEAD_TIME[7:0]=0xFF			2.1		µs	IIsc
Maximum Configurable TIMEOUT	TIMEOUT_MAX	TIMEOUT[7:0]=0xFF			4.3		µs	IIsc

GUARANTEE LEVEL	DESCRIPTION
I	Production Tested @ Multiple Temps
II	Production Tested @ Room Temp, Characterized @ Multiple Temps
IIsc	Production Tested via Scan @ Room Temp, Characterized via Scan @ Multiple Temps
III	Sample Tested
IV	Not Production Tested, Characterized by ATE
V	Not Production Tested, Characterized by Bench (GBDC)
VI	Internal Design Target

VII	Production Tested, Internal Only
VIII	Production Tested @ Hot, Characterized @ Multiple Temps

Measurement Unit Description

The measurement unit (MU) includes sensing, signal conditioning, and analog to digital circuitry to measure and convert analog signals.

Fast and accurate phase current sense measurements are necessary for high performance FOC control of the motor. For this purpose, the MU includes low offset and high bandwidth Current Sense amplifiers as well as 13 bits ADCs to measure the phase current on each of the motor phases.

In addition, IC temperature, Voltage supply and up to four external low bandwidth analog signals can be measured.

MU List of Features

- 4x high-bandwidth, differential bidirectional current sense amplifiers (CSA) for bottom shunt current measurement
  - Less than 250µV equivalent input offset
  - Programmable gain (x5, x10, x20, x40)
  - Bypass mode (Gain = -1)
  - Gain-bandwidth product up to 15MHz
  - Configurable analog filters
- 4x 1MS/s 13-bit ADCs
  - Current measurements A/D conversion from integrated CSA
  - V_S supply measurement
  - Chip temperature measurement
  - A/D conversion of up to four analog signals from the outside

Programmable Current-Sense Amplifiers

The TMC9660 integrates low-offset, high-bandwidth, gain-programmable differential CSAs tailored for motion control applications.

The extremely low equivalent input offset and the Gain selection flexibility allow the user to adopt low value external sense resistances which result in cost and power saving.

The high bandwidth and fast settling time enables reliable current sampling even at extreme duty cycle and high PWM frequency resulting in optimal performances of the FOC controller.

CSA Programmable Filter

For each Gain setting, the user can configure an internal filter which reduces the CSA-equivalent bandwidth/BW. Four different filter settings are possible. This function allows the user to find the optimal compromise between settling time and RMS_Noise. For PWM frequencies higher than 50kHz, a fast settling time is desirable and hence the filter must be set to minimum levels. Vice versa, for PWM frequencies less than 25kHz, a more robust filtering is recommended to reduce RMS noise and improve overall FOC performances. As a matter of example, *Table 10* shows equivalent RMS input noise and settling time for different settings of the CSA filter with Gain = 20. The RMS noise is integrated in the entire bandwidth.

The positive input of the CSA (pins CSP_) is also used as power connection of the low-side gate driver. For this reason, external R-C filters in which resistors are connected between the terminals of the shunt resistor and the CSA input pins are not recommended. However, the integrated filter can also be used to filter out electric noise from the outside.

Table 10. RMS Noise vs. Settling Time

PARAMETER	CONDITIONS	TYP	UNIT
Settling Time	No Filter	0,715	µs
	Filter 1	0,81	µs
	Filter 2	1,31	µs
	Filter 3	1,88	µs

Bandwidth (-3dB)	No Filter	9.5	MHz
	Filter 1	7.8	MHz
	Filter 2	6.7	MHz
	Filter 3	6	MHz
Equivalent Input RMS Noise (integrated in the entire BW)	No Filter	0.18	mV
	Filter 1	0.1	mV
	Filter 2	0.08	mV
	Filter 3	0.065	mV

Gain Settings and Input Voltage Range

Four gain settings for the CSA are supported: 5X, 10X, 20X, and 40X.

The CSA-recommended differential maximum input voltage is calculated as follows:

$$V_{diff\_max} = {}^{1.0V}/_{GAIN}$$

The external shunt resistor must be chosen so that its value satisfy the following relation:

$$R_{SENSE} < \frac{V_{diff\_max}}{I_{MAX}}$$

where

I_{MAX} is the maximum (peak) phase current.

High values of R_{SENSE} result in better sensitivity and performances but higher power dissipation.

A direct mode is also supported for using external CSAs. In this direct mode, the internal CSAs are bypassed, and the CSA input pins are connected directly to ADC multiplexer (mux) inputs. The maximum differential input voltage is equal to ADC full scale 1.2V. The gain is -1 in this case.

Measurement Unit Block Diagram

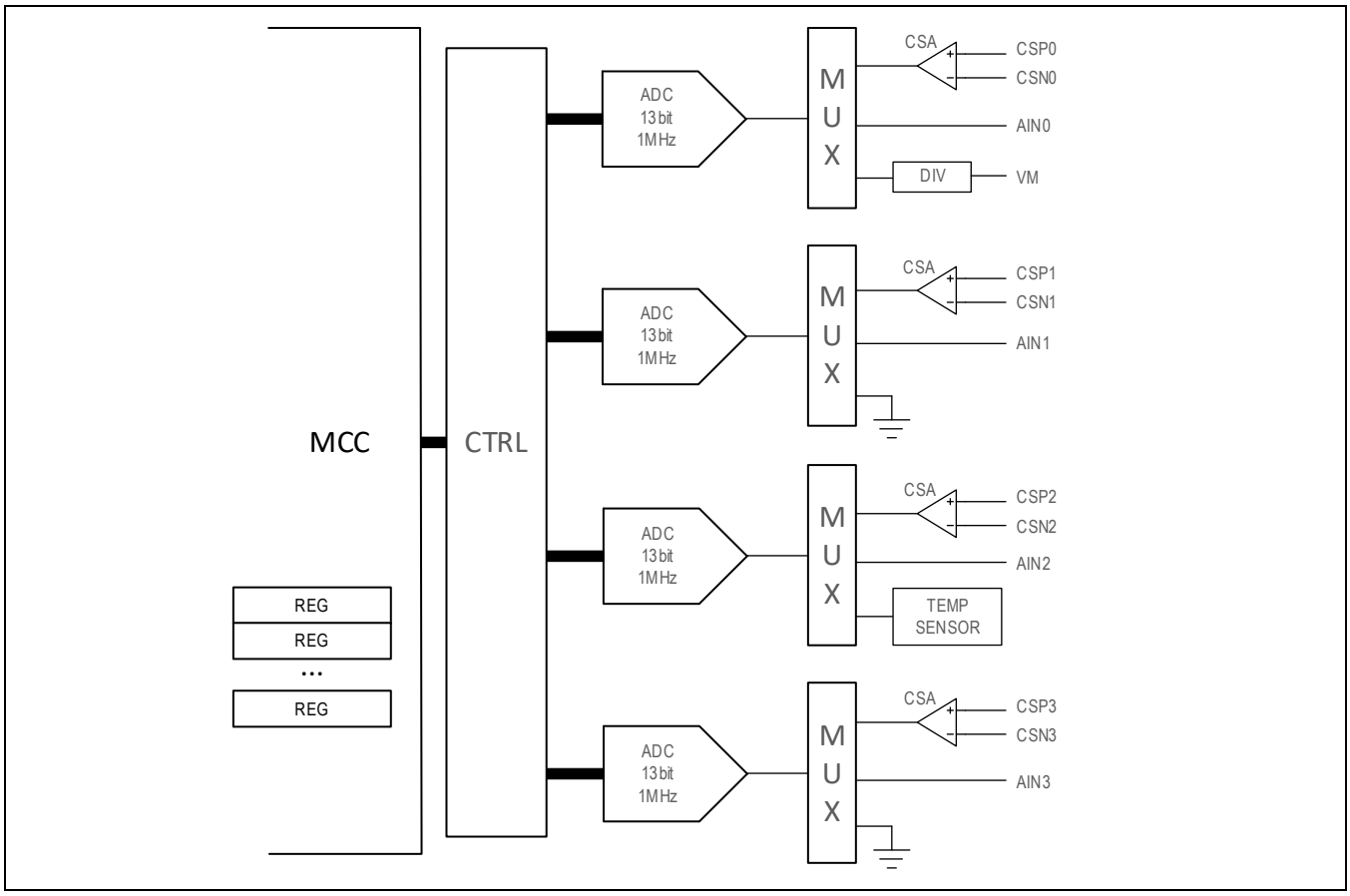

Figure 24. Measurement Unit Block Diagram

Analog Input Measurement

Pins AINx can be configured as analog input to measure and digitize external slow analog voltage signals.

Figure 25 shows a typical application diagram.

Although analog input pins are tolerant up to V_{CC_IO}, the analog signal must be limited from 0V to 1.2V to fulfil the ADC analog input range requirement. An integrated comparator protects the internal circuitry in case the input signal exceeds 1.8V disabling the analog input internal switch and signaling the fault to MCU.

The equivalent input impedance is a few pF.

In general, it is recommended having an external buffer followed by an anti-aliasing filter.

The analog inputs are sampled with a sampling rate which is equal to the programmed PWM chopping frequency.

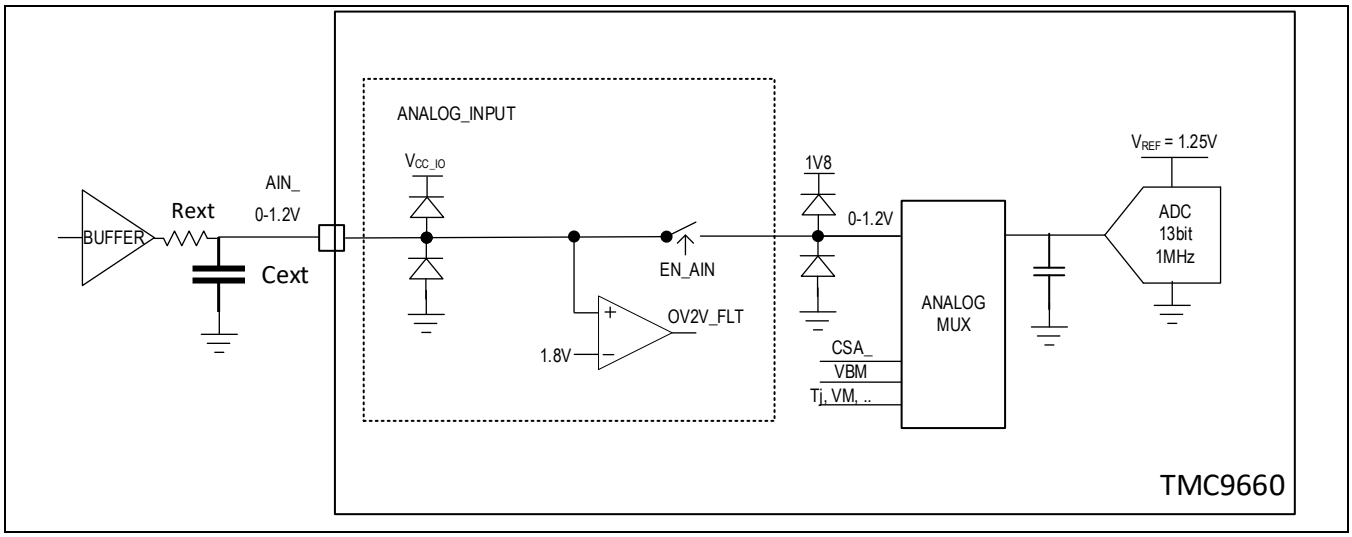

Figure 25. Analog Input Measurement Diagram

Analog-to-Digital Converter

The TMC9660 integrates 4x 13bit 1Msps analog-to-digital converters/ADCs. They are primarily intended for accurate and fast A/D conversion of CSA output signals and motor phase voltages.

Additionally, the ADCs are time-multiplexed and used to convert other low-speed analog signals. The analog multiplexer inputs are connected to four analog pins and several internal analog signals, such as power supply measurement, on-chip temperature measurement, etc.

IC Temperature Measurement

The TMC9660 IC temperature is internally measured and converted to digital by the integrated 13-bit ADCs. The temperature value can be calculated from the following formula:

$$T_J(\text{°C}) = K_{TJ} \times ADC_{TEMP} - 268.15$$

where

ADC_{TEMP} is the register value in decimal.

K_{TJ} is a proportional constant (see the *Measurement Unit Electrical Characteristics* table).

Supply Voltage Measurement

The motor supply voltage (V_S) is internally measured. The supply voltage can be calculated with the following formula from the register value:

$$VS = K_{VS} \times VS_{REG}$$

where

K_{VS} is a constant (see the *Measurement Unit Electrical Characteristics* table).

VS_{REG} is the register value in decimal.

Measurement Unit Electrical Characteristics

(V_{SA} = from 7.7V to 80V, V_S = from 7.7V to 70V, V_{CC_IO} = from 2.2V to 5.5V, Junction Temperature = from -40ºC to 125ºC, Limits over the operating temperature range and relevant supply voltage range are guaranteed by design and characterization. Specifications marked "GBD" are guaranteed by design and not production tested)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS	GL
CURRENT-SENSE AMPLIFIER								
Equivalent Input Offset Voltage	VOS					250	µV	V
Programmable Gain	GAIN	GAIN[1:0]=00			5		V/V	II
		GAIN[1:0]=01			10			II
		GAIN[1:0]=10			20			II
		GAIN[1:0]=11			40			II
Gain Bandwidth Product	GBW	CSA_FILT[1:0]=00			9.5		MHz	VI
		CSA_FILT[1:0]=01			7.8			VI
		CSA_FILT[1:0]=10			6.7			VI
		CSA_FILT[1:0]=11			6			VI
Input RMS Noise	N_{rms}	CSA_FILT[1:0]=00	GAIN[1:0]=01		180		µVrms	VI
		CSA_FILT[1:0]=01	GAIN[1:0]=01		105			VI
		CSA_FILT[1:0]=10	GAIN[1:0]=01		81			VI
		CSA_FILT[1:0]=11	GAIN[1:0]=01		68			VI
Common Mode Rejection Ratio	CMRR	GAIN[1:0]=01			70		dB	VI
Common Mode Operating Input Range	V_{COM}			-0.5		0.5	V	II
Common Mode Tolerant Input Range	VCOM_tolerant				+/-2		V	VI
Differential Input Voltage Range	V_{DIFF}			-1/GAIN		1/GAIN	V	II
Settling Time	TSET	Step response within 0.1% GBW = 9.5MHz GAIN = 20V/V			0.7		µs	VI
ANALOG-TO-DIGITAL CONVERTER/ADC								
ADC Resolution	ADC_RES				13		Bits	II
ADC LSB	LSB				0.3049		mV	II
Internal Reference Voltage	V_{REF}			1.2375	1.249	1.2625	V	II
Full Scale Input, Single Ended	VFS_SE					1.25	V	II
Differential Nonlinearity[2]	DNL					+/-1	LSB	VI
Integral Nonlinearity[2]	INL					+/-3	LSB	VI
SAR Clock Frequency	ADC_F				60		MHz	II

(V_{SA} = from 7.7V to 80V, V_S = from 7.7V to 70V, V_{CC_IO} = from 2.2V to 5.5V, Junction Temperature = from -40ºC to 125ºC, Limits over the operating temperature range and relevant supply voltage range are guaranteed by design and characterization. Specifications marked "GBD" are guaranteed by design and not production tested)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS	GL
Gain Error[2]	ADC_GAIN		-0.85		+0.85	%	V
Offset Error[2]	VOFF_ADC	Analog input.	-1.5		0.4	mV	VI
CURRENT MEASUREMENT (CSA+ADC)							
Total Gain Error[2]	GAINERR	Ta = 25°C	-1.1		+1.1	%	V
		Ta = -40°C, 125°C	-1.2		+1.8		V
Total Gain Temperature Drift[2]	ΔGAIN	Ta = -40°C (Max/min is average drift +/- 6 sigma)	-0.26	0.34	0.95	%	V
		Ta = 125°C (Max/min is average drift +/- 6 sigma)	-0.66	-0.06	0.55		V
SUPPLY VOLTAGE MEASUREMENT							
V_S Voltage Measurement Coefficient	KVS	VS = KVS × VSREG	2.41	2.44	2.47	mV/LSB	II
V_S Voltage Measurement Offset Error	OSVS		-0.4		0.4	V	II
TEMPERATURE MEASUREMENT							
Junction Temperature Measurement Coefficient	KTJ	Tj = KTJ × ADCTEMP -268.15		16.15		mºC/LSB	II

GUARANTEE LEVEL	DESCRIPTION
I	Production Tested @ Multiple Temps
II	Production Tested @ Room Temp, Characterized @ Multiple Temps
IIsc	Production Tested via Scan @ Room Temp, Characterized via Scan @ Multiple Temps
III	Sample Tested
IV	Not Production Tested, Characterized by ATE
V	Not Production Tested, Characterized by Bench (GBDC)
VI	Internal Design Target
VII	Production Tested, Internal Only
VIII	Production Tested @ Hot, Characterized @ Multiple Temps

Protections Description

Introduction

The TMC9660 is designed for robustness and safe operation in harsh environments. It offers a full set of hardware protections and diagnostic. These include protection against driver faults (motor overcurrent, external FET gate short), thermal fault and warning, internal and external power supply faults (short circuits, overcurrent, undervoltages).

Protections are highly configurable. Depending on application requirements, the user can program intervention thresholds, take different actions when fault occurs and adopt several strategies to recovery from fault, configure different blanking and deglitch timings to ensure faults are not triggered by environmental electrical noise, etc.

The following sections describe the hardware protection blocks integrated into the device.

List of Features

Output Stage Overcurrent and Short-Circuit Protection

- Configurable high-/low-side NFET overcurrent protection based on VDS measurement
- Configurable low-side NFET overcurrent protection based on RSHUNT

Gate Driver Output Short-Circuit Protections

- High-/low-side gate shorted high
- High-/low-side gate shorted low

Thermal Shutdown and Warning Protections

- Global thermal shutdown and warning
- Linear regulators (V_{EXT1}, V_{EXT2}, 1V8, A1V8) local thermal protection
- Buck converter (V_{BUCK}) local thermal protections

PMU Overcurrent and Short-Circuit Protections

- Linear regulators current limit (V_{EXT1}, V_{EXT2}, 1V8, A1V8)
- V_{DRV} charge pump short protection
- Buck converter overload and shorts protections

Undervoltage Lockout/UVLO

- V_S, V_{SA} UVLO, and warning
- V_{CC_IO} UVLO
- V_{BUCK}, V_{DRV}, and bootstrap (BST_) UVLO

Driver Protection Diagram

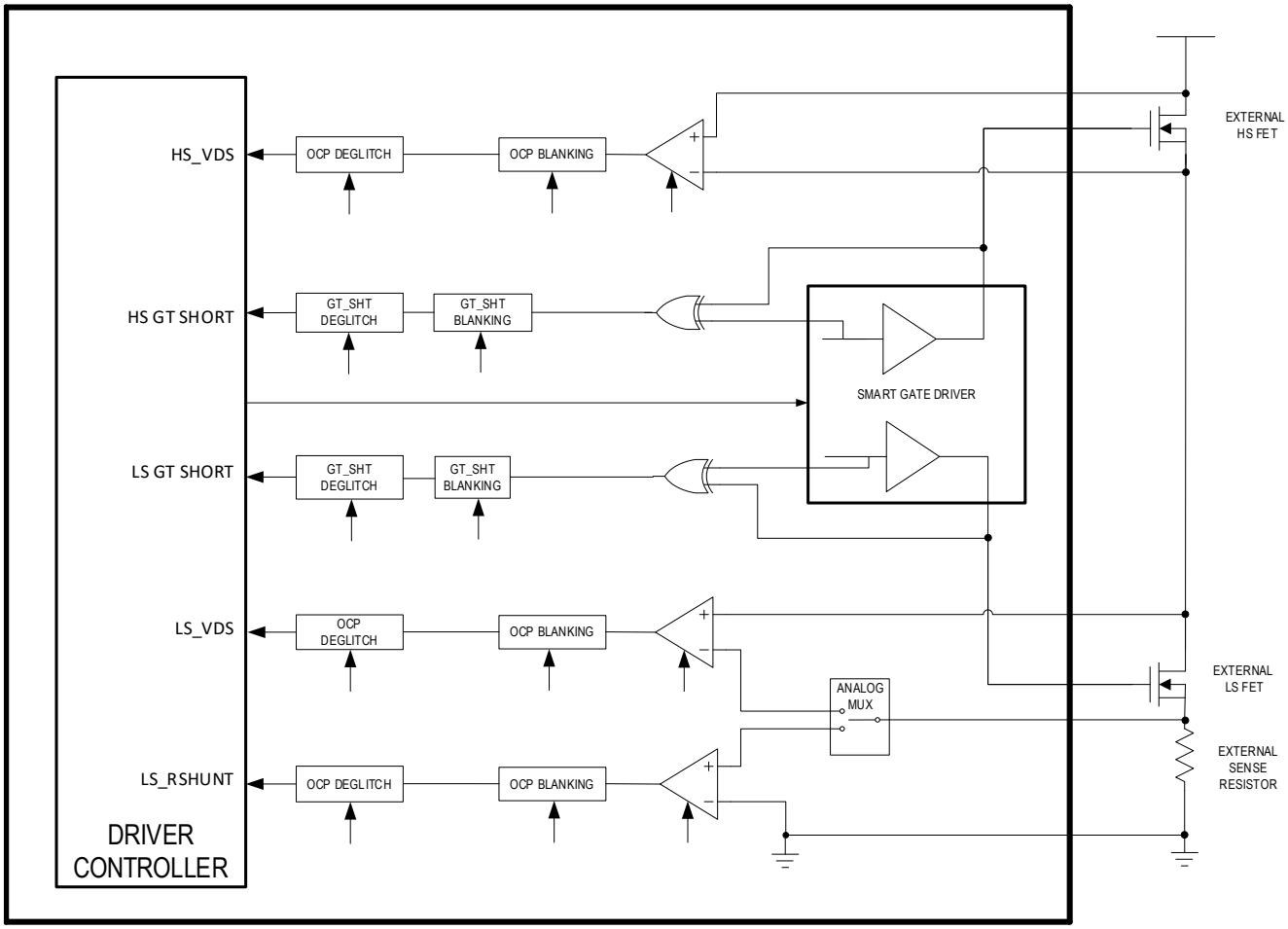

Figure 26. Overcurrent and Gate Short Protection Diagram

Output Stage Overcurrent and Short-Circuit Protection

The gate driver integrates circuitry to detect and manage overcurrent failure events.

The HS protection circuitry senses the drain source voltage (VDS) of the external HS FET and compare it with a programmable threshold.

The LS protection circuitry senses either the drain source voltage (VDS) of the external LS FET or the voltage across the bottom shunt resistor (RSHUNT) and compare it with a programmable threshold.

The VDS-based current measurements strongly depend on temperature and production stray of external MOSFET and is typically not accurate and reproducible. RSHUNT-based current measurement is more accurate and reproducible and allows a finer tune of current protection for the motor and power stage.

The VDS-based or RSHUNT-based protection thresholds are programmable over a wide range. Possible settings are shown in *Table 11*.

To avoid spurious intervention of the overcurrent and short-circuit protections due to electrical noise in harsh environments, the TMC9660 features configurable blanking and deglitch timings.

Blanking timing starts from the logic PWM command which initiates the output transition whereas deglitch timing starts from the commutation of the overcurrent protection comparator. The sum of blanking and deglitch time must be set longer than the phase voltage transition time (rise and fall edges) to avoid false fault indication during rise/fall edges.

Programmable blanking times and deglitch filter values are shown in *Table 14*.

Thresholds, blanking, and deglitch settings are common for UX1, VX2, WY1 channels whereas these can be different for the Y2 channel.

When an overcurrent fault event occurs, the TMC9660 controller can three-state the individual half-bridge or all the four half-bridges. Different recovery schemes (autoretry or latched) are supported as summarized in *Table 13*.

The parameter mode supports more advanced handling options.

Table 11. Overcurrent Protection Thresholds

SETTING	HIGH SIDE	LOW SIDE	
	HS_VDS	LS_RSHUNT	LS_VDS
0	63mV	80mV	63mV
1	125mV	165mV	125mV
2	187mV	250mV	187mV
3	248mV	330mV	248mV
4	312mV	415mV	312mV
5	374mV	500mV	374mV
6	434mV	582mV	434mV
7	504mV	660mV	504mV
8	705mV	125mV	705mV
9	0.94V	250mV	0.94V
10	1.18V	375mV	1.18V
11	1.41V	500mV	1.41V
12	1.65V	625mV	1.65V
13	1.88V	750mV	1.88V
14	2.11V	875mV	2.11V
15	2.35V	1V	2.35V

Table 12. Overcurrent Protection

MNEMONIC	PROTECTION	DESCRIPTION	CONFIGURABILITY	SUPPORTED ACTION	FAULT TYPE AND RECOVERY
HS_VDS_S2G	High Side ext FET Short-Circuit Protection	Short is detected by sensing the voltage drop on the HS external power FET	• 16 levels - Programmable Thresholds • 8 value - Programmable Blanking Time • 8 value - Programmable Deglitch Time • Function Enable/Disable	• Individual Half-Bridge is Threestated • MCU signaling is generated (interrupt) • All Half-Bridges are Threestated	• Autoretry with Tretry • 3 times Autoretry and Latch • Latched Fault

LS_VDS_S2G	Low Side ext FET Short-Circuit Protection	Short is detected by sensing the voltage drop on the LS external power FET	• 16 levels - Programmable Thresholds • 8 value - Programmable Blanking Time • 8 value - Programmable Deglitch Time • Select between VDS-based or RSHUNT-based • Function Enable/Disable	• Individual Half-Bridge is Tristated • MCU signaling is generated (interrupt) • All Half-Bridges are Threestated	• Autoretry with Tretry • 3 times Autoretry and Latch • Latched Fault

Gate Protection—VGS Protection

The LS_GT_SHT and HS_GT_SHT circuitry detect malfunctioning of the gate driver (low side and high side, respectively). Fault is detected and action taken if, after a programmable TIMEOUT, the gate output voltage is found not consistent with the commanded signal (see *Table 13*). This may occur because of short-circuit or overload conditions at the gate output voltage.

The TIMEOUT parameter is used by the gate driver sequencer as discussed in the *Gate Driver Sequencer* section. It follows that TIMEOUT must be set long enough to allow the gate voltage to settle at the final level.

Configurable deglitch filter and blanking time prevent from false triggering caused by short glitches on the measured gate voltage (see *Blanking and Deglitch* section).

When a gate short fault is detected, the corresponding half-bridge is three-stated to protect the gate driver. It is also possible to three-state all the half-bridges.

Table 13. Gate Driver Output Protection

MNEMONIC	PROTECTION	DESCRIPTION	CONFIGURABILITY	SUPPORTED ACTION	FAULT TYPE AND RECOVERY
HS_GT_SHT_OFF/ON	High-Side Gate Short	Fault is detected when HS Gate voltage is not consistent with Gate command	• 8 value - Programmable Timeout • 4 value - Programmable Blanking Time • 8 value - Programmable Deglitch Time • Function Enable/Disable	• Individual Half-Bridge is Threestated • MCU signaling is generated (interrupt) • All Half-Bridges are Threestated	• Latched Fault (requires reset) • No Latched Fault
LS_GT_SHT_OFF/ON	Low-Side Gate Short	Fault is detected when LS Gate voltage is not consistent with Gate command	• 8 value - Programmable Timeout • 4 value - Programmable Blanking Time • 8 value - Programmable Deglitch Time • Function Enable/Disable	• Individual Half-Bridge is Threestated • MCU signaling is generated (interrupt) • All Half-Bridges are Threestated	• Latched Fault (requires reset) • No Latched Fault

Blanking and Deglitch

As shown in *Table 14*, blanking and deglitch timings can be configured to avoid the protection to be triggered by spurious or transitory signals. This applies to both the Overcurrent Protections and Gate Output Protections.

Figure 27 shows how the blanking and deglitch timer operates. This diagram refers to the overcurrent protection function but can be easily extended to the gate output protection function.

In this example, the controller outputs a PWM signal commanding the external FET to be turned ON. This causes the blanking time counter to start.

In the example, the overcurrent analog comparator outputs a spurious fault signal (event “1”). This could be due to the motor phase output still transitioning or due to switching noise which is erroneously detected as an OCP event. Being the event “1” shorter than the blanking time (TSHT_BLK), this spurious fault signal is ignored.

Other spurious OCP events occur (namely number “2” and “3”) and are shown in the example as well. They occur when the driver output is supposed to be in steady state. Every time the OCP comparator detects an OCP event, the Deglitch counter starts counting. Both the events “2” and “3” are ignored as they are shorter than the deglitch time (TSHT_DEG). Notice that the deglitch time is reset every time the comparator output is zeroed (namely no fault event detected).

Finally, a fourth OCP event is detected (“4”). Since it lasts longer than the deglitch time (TSHT_DEG), this is recognized as a valid fault condition and cause the driver to enter in protection mode and switch off the output transistor.

Possible values for Blanking and Deglitch timings are shown in *Table 14*.

Table 14. Blanking and Deglitch Possible Values

	POSSIBLE VALUES
OCP Blanking Time	0, 0.25µs, 0.5µs, 1µs, 2µs, 4µs, 6µs, 8µs
OCP Deglitch Time	0, 0.25µs, 0.5µs, 1µs, 2µs, 4µs, 6µs, 8µs
Gate Short Blanking Time	0, 0.25µs, 0.5µs, 1µs
Gate Short Deglitch Time	0, 0.25µs, 0.5µs, 1µs, 2µs, 4µs, 6µs, 8µs

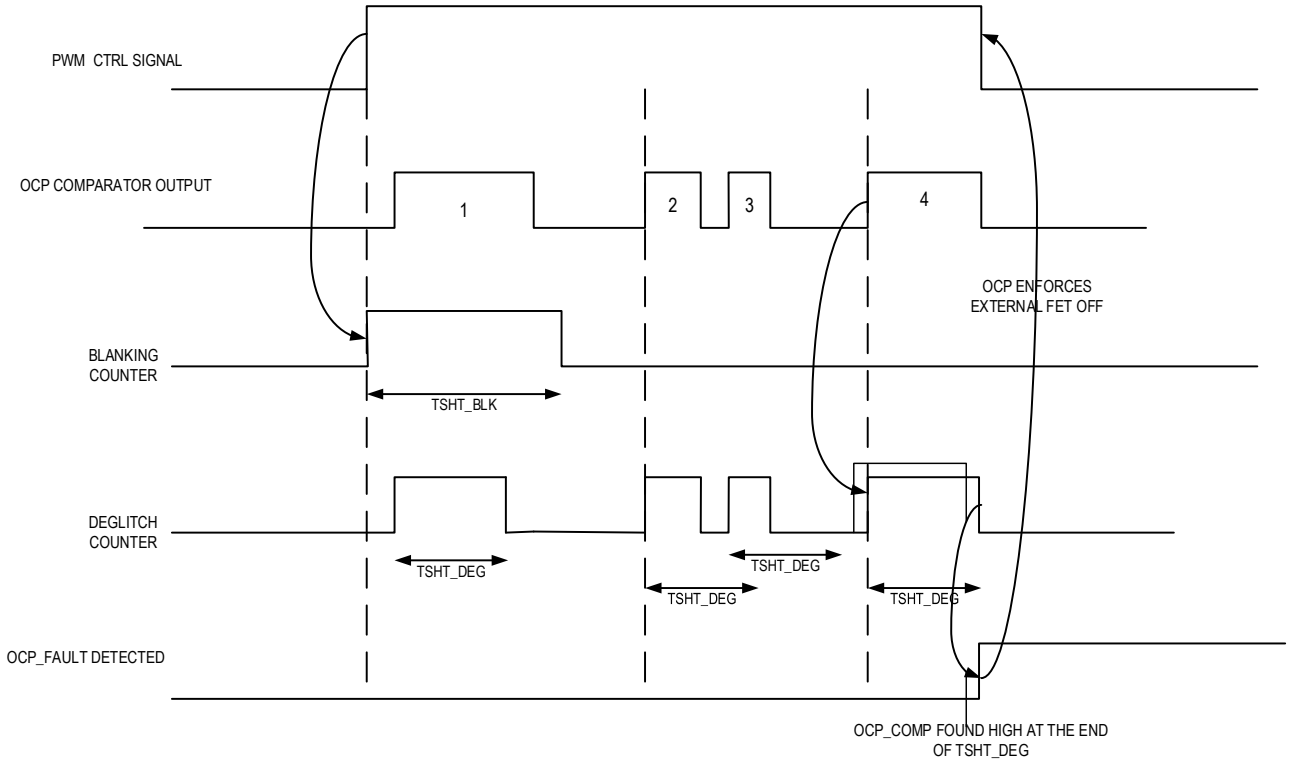

Figure 27. Blanking and Deglitch

Thermal Protections

The TMC9660 features integrated thermal warning and protection circuits.

Thermal warning is triggered when the die temperature exceeds the TWARN threshold. When this happens, the Thermal Warning bit is set. No additional actions are taken, and the device continues to function. If the die temperature falls below the hysteresis point, the Thermal Warning bit clears automatically.

Thermal shutdown is triggered when the die temperature exceeds the TSDN threshold. When this happens, the gate driver, V_{DRV} charge pump, and external LDO are shut down and no operation is possible. If the die temperature falls below the hysteresis point, normal operations are restored.

External LDOs (V_{EXT1}, V_{EXT2}) are locally thermal protected. They are individually shut off when junction temperature exceeds 165°C.

The buck converter as well as the 1V8 and the A1V8 linear regulators are also locally thermal protected with thermal sensor near the LDO output stages. When the junction temperature exceeds 165°C, the corresponding regulator is shut off and the logic is reset.

Undervoltage Lockout

Both the external supply voltages (V_S, V_{SA}, V_{CC_IO}) and the internally generated voltages (1V8, A1V8, VDRV,BS_) are monitored and UVLO comparators ensure safe operation if their values drop below the minimum operating threshold.

See the *Power Management Unit* section for a detailed description of the UVLO functionality for each supply.

Fault Pin (FAULTN)

The TMC9660 offers a dedicated pin to signal faults as well as completion of bootstrapping steps. The pin is an open-drain output which pulls the pin low when asserting. When the TMC9660 powers on, the FAULTN pin starts asserted. Once the chip has successfully completed its power-on sequence, the pin deasserts. Some fault events always assert the FAULTN when detected. Some other fault events can be user-configured whether to assert the FAULTN pin upon detection. The bootstrapping sequence can also be configured to temporarily assert the FAULTN pin during processing of various operations.

Protections—Electrical Characteristics

(V_{SA} = from 7.7V to 80V, V_S = from 7.7V to 80V, Junction Temperature = from -40ºC to125ºC, Limits over the operating temperature range and relevant supply voltage range are guaranteed by design and characterization. Specifications marked "GBD" are guaranteed by design and not production tested)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS	GL
OVERCURRENT PROTECTIONS							
R_{sense}-Based—Overcurrent Protection Voltage Configurable Range	LS_RSNS_OCP	OCP_THRES_ = 0	125		1000	mV	II
HS/LS FET-Based—Overcurrent Protection Voltage Configurable Range	VDS_OCP		63		2350	mV	II
OCP Blanking Time Configurable Range	TBLANK_R	Steps: 0, 0.25µs, 0.5µs, 1µs, 2µs, 4µs, 6µs, 8µs	0		8	µs	VI
OCP Deglitch Time Configurable Range	TDEG_R	Steps: 0, 0.25µs, 0.5µs, 1µs, 2µs, 4µs, 6µs, 8µs	0		8	µs	VI
OCP_Autoretry	TRETRY	OCP Autoretry Function enabled. Default setting		4		ms	II
GATE SHORT PROTECTIONS							
HS/LS FET VGS ON Threshold Voltage	VGS_ON	VGS = rising	BS-1.2	BS-0.9		V	II

(V_{SA} = from 7.7V to 80V, V_S = from 7.7V to 80V, Junction Temperature = from -40ºC to125ºC, Limits over the operating temperature range and relevant supply voltage range are guaranteed by design and characterization. Specifications marked "GBD" are guaranteed by design and not production tested)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS	GL
HS/LS FET VGS OFF Threshold Voltage	VGS_OFF	VGS = falling		1.2	1.6	V	II
VGS Protection Blanking Time Range	TBLANK_VGS_R	Programming steps: 0µs, 0.25µs, 0.5µs, 1µs (Note: Tdrive acts as additional blanking)	0		1	µs	VI
VGS Protection Deglitch Time Range	TDEG_VGS_R	Programming Steps: 0, 0.25µs, 0.5µs, 1µs, 2µs, 4µs, 6µs, 8µs	0		8	µs	VI
THERMAL PROTECTIONS							
Thermal Shutdown Threshold	TSDN			165		°C	II
Thermal Hysteresis	THYST			30		°C	II
Thermal Warning Threshold	TWARN			135		°C	II

GUARANTEE LEVEL	DESCRIPTION
I	Production Tested @ Multiple Temps
II	Production Tested @ Room Temp, Characterized @ Multiple Temps
IIsc	Production Tested via Scan @ Room Temp, Characterized via Scan @ Multiple Temps
III	Sample Tested
IV	Not Production Tested, Characterized by ATE
V	Not Production Tested, Characterized by Bench (GBDC)
VI	Internal Design Target
VII	Production Tested, Internal Only
VIII	Production Tested @ Hot, Characterized @ Multiple Temps

Clock and Control Description

This section describes the timing references (clock lines) and how they are generated. These references are used as time base for the control blocks and analog-to-digital conversion inside the TMC9660.

It also describes some global control functions such as Global RESET, Driver enable function, and Wake-up function.

Clock

The TMC9660 supports three different clock sources one at a time for the main/system clock which are as follows:

- Internal oscillator (INT_OSC 15MHz)
- External clock input (CLK_IN)
- On-chip crystal oscillator with external crystal (connected to OSC_IN, OSC_OUT)

In addition, there is an independent secondary clock source:

- Internal 32kHz low-power oscillator, which drives the timer inside the AON block and the watchdog.

After a system reset, the MCU always starts on the internal clock executing the bootloader program from ROM. The bootloader configures and enables the internal PLL to generate the 120MHz for the PWM block, 60MHz for the ADCs, and 40MHz for the MCU and MCC (among others). After initialization, the clock source may be changed to external clock input or to on-chip crystal oscillator based on bootloader OTP configuration. The crystal oscillator requires an external crystal for operation.

For the different clock options, frequencies between 1MHz and 32MHz are supported (whole numbers). As a valid clock signal is essential for the sanity of the overall system, an external clock or the clock from the crystal oscillator is permanently monitored with the help of the internal oscillator, and in case of a clock deviation error, all digital blocks are reset.

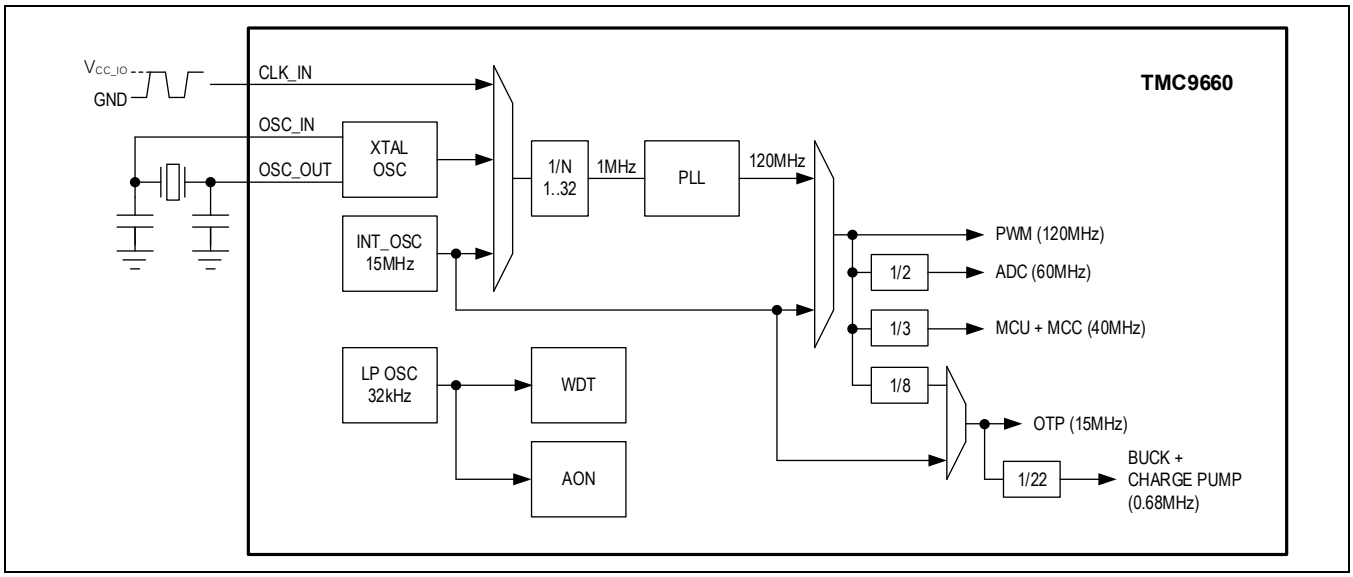

Figure 28. TMC9660 Clock Options and Clock Tree

Crystal Oscillator

The on-chip crystal oscillator offers a programmable output current for different gain settings optimized for five different crystal frequencies 8MHz, 16MHz, 24MHz, 25MHz, and 32MHz.

Table 15. XTAL Output Current vs. Crystal Frequency

XTAL_CFG	CONDITIONS	XTAL_OUT (µA)	f_{XTAL} (MHz)
1	ESR <250R and CL = 9pF	75µA	8MHz
3	ESR < 70R and CL = 9pF	225µA	16MHz
5	ESR < 70R and CL = 9pF	375µA	24MHz or 25MHz
6	ESR < 60R and CL = 9pF	450µA	32MHz

ESR is the equivalent series resistance given by the crystal manufacturer. As load capacitance, CL = 9pF is recommended. The XTAL oscillator also offers a boost feature to support tiny crystals with a larger ESR than listed in the table. This increases the output current of the oscillator to 900µA for a limited amount of time (65k clock cycles) to increase gain during startup.

Reset (RSTN)

The TMC9660 offers an integrated power-on reset (POR) on 1V8 digital core supply. In addition, a dedicated external RSTN pin is available that can be used to reset the microcontroller of TMC9660 manually. This input is active low and must be pulled high to release the internal logic from reset condition. In case of standalone supplied operation where the supply outputs V_{EXT1} or V_{EXT2} of the integrated regulators are connected to V_{CC_IO}, these supplies and, hence, V_{CC_IO} are not available until configured by the bootloader inside the TMC9660. Consequently, a pull-up resistor between the RSTN input to VCCIO is not sufficient. Instead, the RSTN input can be pulled up to the buck regulator output. Typical configuration:

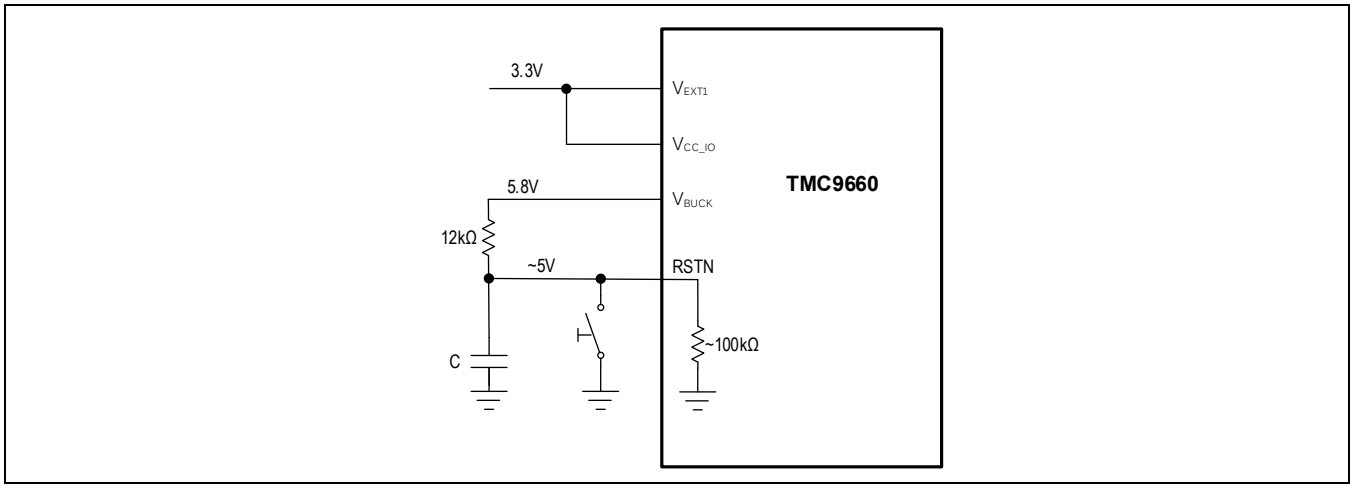

Figure 29. TMC9660 Reset Pin (RSTN) External Circuit

There are additional events that generate a reset:

- watchdog reset
- entering hibernation mode
- software reset
- clock loss detection
- critical hardware fault

Driver Enable (DRV_EN)

The TMC9660 offers a dedicated driver enable input pin DRV_EN. This input offers an internal pull-down resistor. When left unconnected or actively pulled “low” from external resistor, the driver stage is switched off regardless of software settings. The power stage gets three-stated and meanwhile, the BST caps might be discharged.

When this input is pulled “high,” it is possible to activate the driver stage in software.

In case this function is not required in the application, this pin should be connected to V_{CC_IO}. Otherwise, it is not possible to enable the driver stage in software.

Note: Pulling this pin low and, therefore, turning off the driver stage while the motor is still rotating might lead to overvoltage and damaging of the driver stage due to the back-EMF from the motor, especially in field-weakening operation mode.

Wake-Up (WAKE)

The TMC9660 offers a dedicated WAKE pin. If this pin is driven high, the part exits hibernate mode.

Note: Like the Driver Enable, the driver stage gets deactivated as well, while the part enters hibernation mode. Turning off the driver stage while the motor is still rotating might lead to overvoltage and damaging of the driver stage due to the back-EMF from the motor, especially in field-weakening operation mode.

FAULT Output (FAULTN)

The TMC9660 offers a dedicated FAULT output pin. This pin is an active-low, open-drain (limited to V_{CC_IO}) output, which is activated in case of critical errors that might also prevent communication with the TMC9660 through standard SPI or UART interfaces to read out internal status information and error flags. *Figure 30* shows a subset of the available internal signals for an overview.

In addition to indicating fault conditions, this output is pulled low after power-up until the TMC9660 is ready for communication and accepting commands.

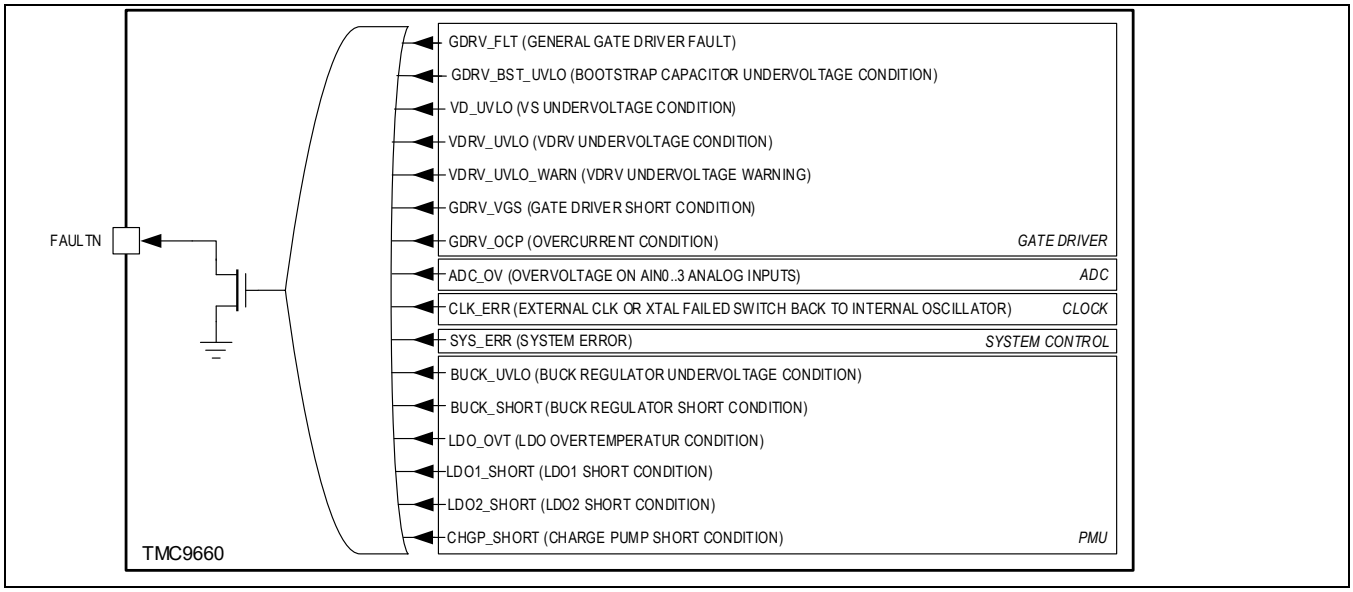

Figure 30. FAULTN Output Signal Options

Clock and Control—Electrical Characteristics

(V_{SA} = from 7.7V to 80V, V_S = from 7.7V to 70V, Junction Temperature = from -40ºC to 125ºC, Limits over the operating temperature range and relevant supply voltage range are guaranteed by design and characterization. Specifications marked "GBD" are guaranteed by design and not production tested)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS	GL
Power-Up/Wake-Up Time to Application	TWAKE	From power off or hibernation until application is ready for communication	Self-test enabled		44		ms	VI
	TWAKE_FAST	From power off or hibernation until application is	Self-test disabled		10.5			VI

(V_{SA} = from 7.7V to 80V, V_S = from 7.7V to 70V, Junction Temperature = from -40ºC to 125ºC, Limits over the operating temperature range and relevant supply voltage range are guaranteed by design and characterization. Specifications marked "GBD" are guaranteed by design and not production tested)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS	GL
		ready for communication						
CLOCK								
Internal Oscillator Frequency	I_{CLK}			14.2	15	15.8	MHz	II
External Oscillator Frequency Range	ECLK			1		32	MHz	II
32K Oscillator Frequency	CK32			30.4	32	33.8	kHz	II
Internal PLL Output Frequency	PLL_CLK				120		MHz	II
MCU Clock Frequency	MCU_CLK				40		MHz	II
PLL Startup Time	tPLLSTRT	Enable PLL to commit goes low			200		µs	II
CONTROL PINS								
Resistive Pull-Down	RPD	RSTN, DRV_EN		60	100	140	kΩ	II
Logic Input Leakage Current	I_{LEAK}			-1		1	µA	II
Input Voltage Level High	V_{IH}	RSTN,DRV_EN		2			V	II
Input Voltage Level Low	V_{IL}	RSTN,DRV_EN				0.8	V	II
Hysteresis		RSTN,DRV_EN			0.2		V	II
Input Voltage Level High	V_{IH}	WAKE		890			mV	II
Input Voltage Level Low	V_{IL}	WAKE				650	mV	II
Hysteresis		WAKE			80		mV	II
DIN Propagation Delay		RSTN,DRV_EN			0.5	1	µs	II
Logic Low Output Voltage	Vol	Iol = 5mA	FAULTN			0.4	V	II
Time from Hibernation to gate driver output		From boot until open loop commutation / gate driver output			49.4		ms	VI
Time from Hibernation to gate driver output without self-test		From boot until open loop commutation / gate driver output			15.9		ms	VI

GUARANTEE LEVEL	DESCRIPTION
I	Production Tested @ Multiple Temps
II	Production Tested @ Room Temp, Characterized @ Multiple Temps
IIsc	Production Tested via Scan @ Room Temp, Characterized via Scan @ Multiple Temps
III	Sample Tested
IV	Not Production Tested, Characterized by ATE

V	Not Production Tested, Characterized by Bench (GBDC)
VI	Internal Design Target
VII	Production Tested, Internal Only
VIII	Production Tested @ Hot, Characterized @ Multiple Temps

Bootloader

The TMC9660 offers a bootloader to bootstrap the system and configure low-level settings.

The following features are available.

- Internal memory self-test on startup
- System configuration through UART, RS485, or SPI
  - Clock source configuration
  - GPIO function selection
  - Motor control mode selection
- System configuration storage in OTP memory
- External memory access to connected SPI Flash or I2C EEPROM
- Launching the motor control system

All bootloader settings can be configured at runtime or burnt into OTP to occur on power-on.

Note: For the entire TMC9660 bootloader as described in this section, the UblTools software is offered, which allows communication with the TMC9660 from a PC. For using the TMC9660 bootloader from a microcontroller, the UblTools C library is offered.

Chip Bootstrapping Overview

The TMC9660 bootstrapping is performed after entering the bootloader. For a not-yet-configured chip, this only requires the chip to be fully powered on (V_{SA} and V_{CC_IO} supplies externally provided) and pulled out of reset. For a chip configured to start the motor control system after power-on, the return-to-bootloader sequence must be completed first (see the *Reentering Bootloader* section in the parameter data sheet).

Once the bootloader is reachable, the bootstrapping can be continued. It is advised to first evaluate all settings using the *Runtime Reconfiguration* before committing the configuration to OTP memory. See the *Configuration List* for a full list of bootloader settings.

Bootloader FAULTN Signaling

The TMC9660 bootloader uses the FAULTN pin to signal its status.

There are three different configuration bits in the *Bootstrap Configuration* controlling how the bootloader uses the pin—BL_READY_FAULT, BL_EXIT_FAULT, and BL_CONFIG_FAULT. By default, only the BL_EXIT_FAULT option is enabled.

BL_READY_FAULT controls whether the bootloader shall assert or deassert the FAULTN pin once it is ready to receive commands. This allows to detect once the power-up sequence is completed, when a configuration task is completed, or when the motor control system has finished exiting back to the bootloader.

On power-on, the FAULTN pin gets asserted. Once the bootloader is ready, it uses the BL_READY_FAULT configuration to assert/deassert the pin. By default, on an unprogrammed chip, this causes the pin to deassert to signal readiness.

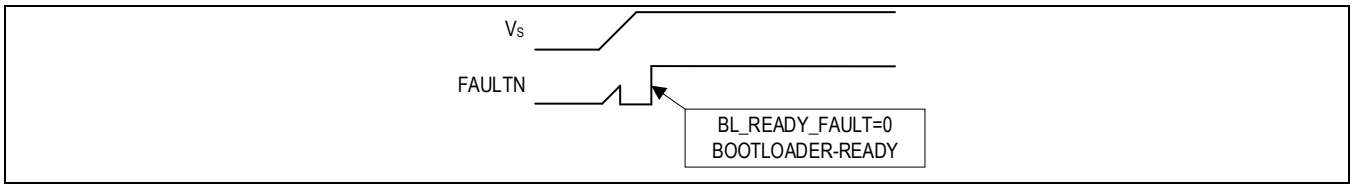

Figure 31. Bootloader FAULTN Pin: Power-On to Bootloader-Ready

When the motor control system starts, it deasserts the FAULTN pin to signal readiness. When exiting back into the bootloader, the bootloader applies BL_READY_FAULT. When configured to assert the pin, this causes the pin to signal the completion of the return to bootloader.

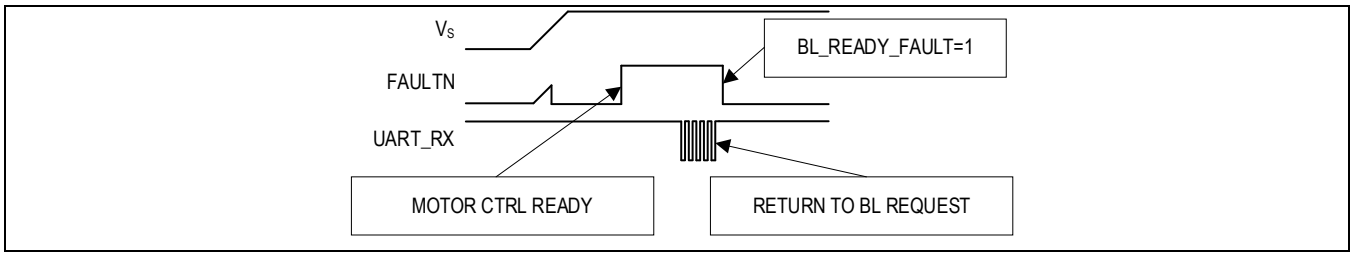

Figure 32. Bootloader FAULTN Pin: Return to Bootloader from Motor Control System

Note: In *Figure 32*, the motor control system is configured to start on power-on (START_MOTOR_CONTROL=1 in the stored configuration).

When updating the *Configuration*, the BL_CONFIG_FAULT option can be used in combination with the BL_READY_FAULT option to allow signaling the completion of a configuration update.

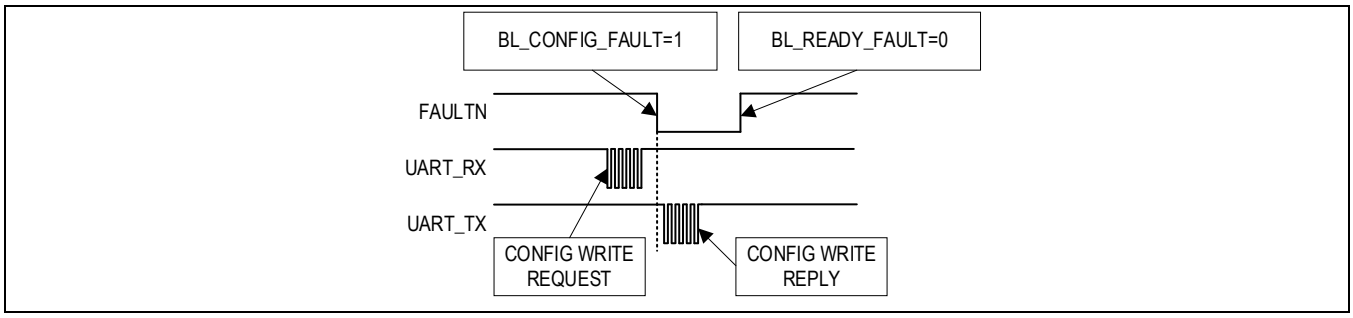

Figure 33. Bootloader FAULTN Pin: Configuration Updates

The BL_EXIT_FAULT option can be used to signal the completion of starting the motor control system. However, note that using BL_CONFIG_FAULT instead is a more reliable option to do so.

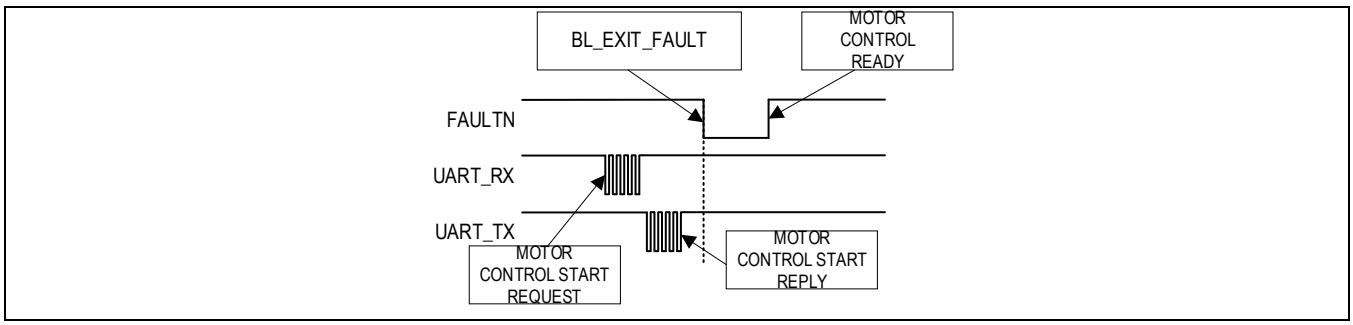

Figure 34. Bootloader FAULTN Pin: Motor Control System Starts Using BL_EXIT_FAULT

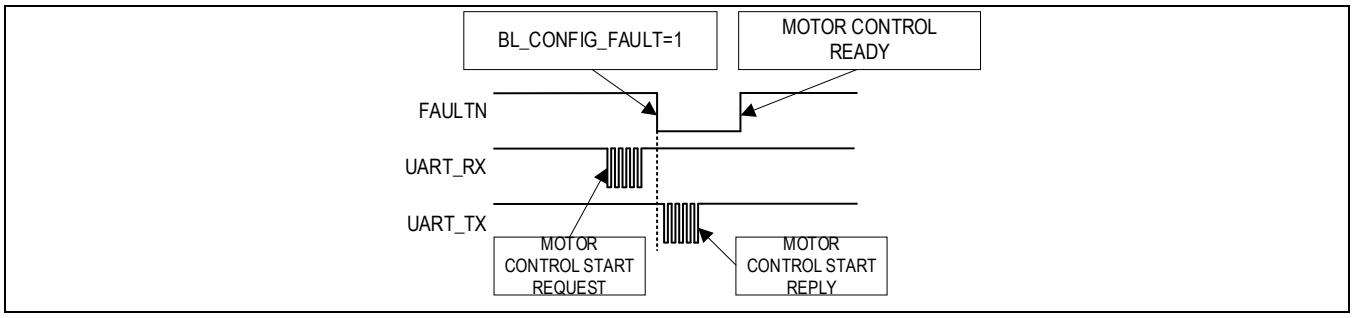

Figure 35. Bootloader FAULTN Pin: Motor Control System Starts Using BL_CONFIG_FAULT

When using BL_EXIT_FAULT, a delay between receiving the response and observing the FAULTN pin assertion is required. When using BL_CONFIG_FAULT instead, such a delay is not needed, as the pin asserts before the reply is sent.
Note: The “Motor control start request” refers to setting the START_MOTOR_CONTROL bit in the *Bootstrap Configuration*.

Communication Protocol

The TMC9660 bootloader communication comprises request/response datagrams sent through UART or SPI. Common between both busses is each request sending a command byte and four data bytes, and each reply returning a status byte and four data bytes.

UART Communication

Communication through UART comprises 8 bytes sent for each request and 8 bytes received subsequently for each reply.

Table 16. Bootloader UART Request Communication Byte Format

REQUEST BYTE	NAME	DESCRIPTION
1	Sync byte	This byte is always 0x55.
2	Device address	The device address configured for the chip. Default is 1.
3	Command byte	The command byte. See the *Commands* section.
4-7	Data bytes	The data bytes. The highest byte is sent first.
8	Checksum byte	The CRC8 checksum byte. See *CRC8 Calculation* section for calculating the CRC8 checksum byte.

Table 17. Bootloader UART Reply Communication Byte Format

REPLY BYTE	NAME	DESCRIPTION
1	Host address	The host address configured for the chip. Default is 255.
2	Device address	The device address configured for the chip. Default is 1.
3	Status byte	The Status byte. See the *Commands* section.
4-7	Data bytes	The data bytes. The highest byte is sent first.
8	Checksum byte	The CRC8 checksum byte. See *CRC8 Calculation* section for calculating the CRC8 checksum byte.

For the device and host address, see the *UART Configuration* section for details.

Any datagram with an invalid sync byte gets ignored.

Any datagram with a different device address gets ignored.

Any datagram with an invalid CRC checksum gets ignored.

When using autobaud, the baud rate detection gets restarted after each reply sent and after any datagram gets ignored.

If less than 8 bytes of data are sent, after 10ms the bootloader drops all bytes received until then.

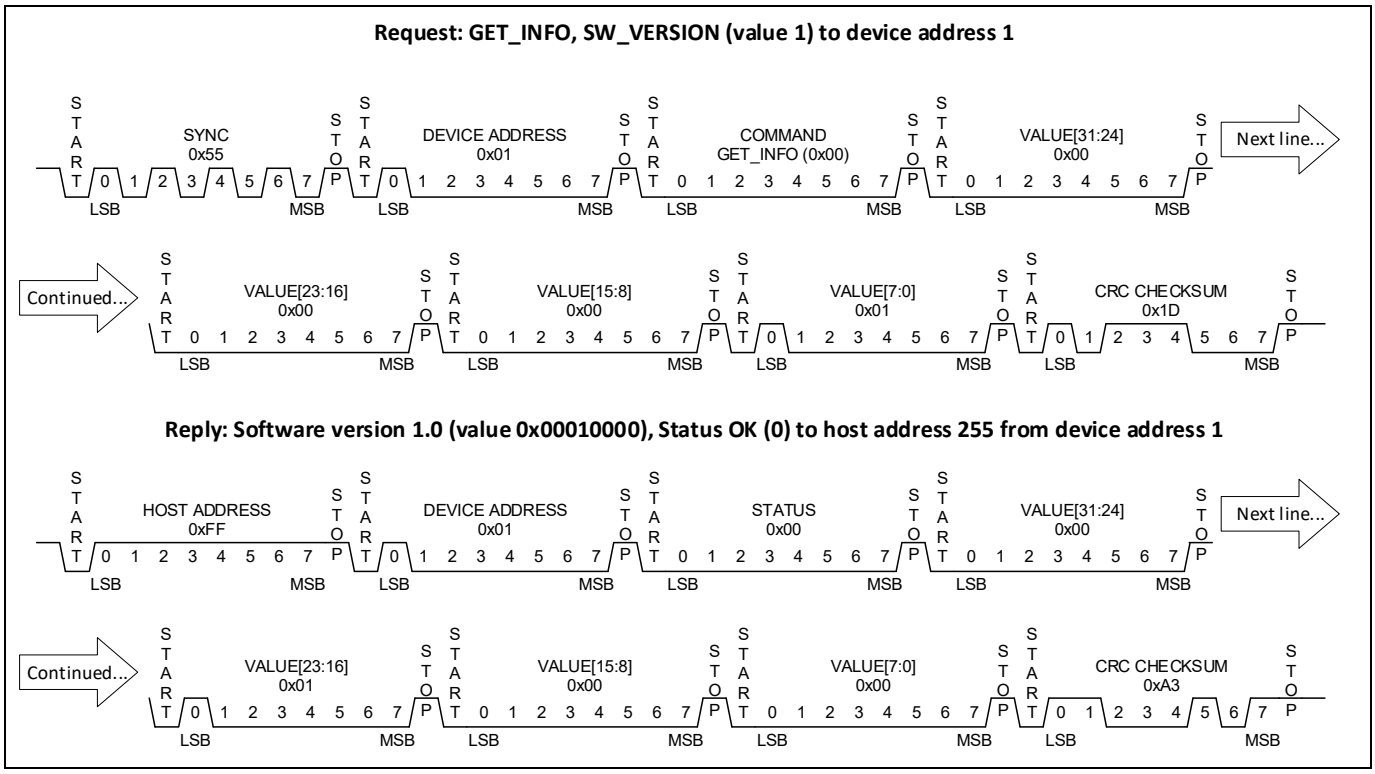

Figure 36. UART Example Request and Reply Pair

RS485 Communication

Communication using RS485 is based on the UART communication with the addition of a TX_EN signal. For every response sent by the TMC9660, the configured TX_EN pin asserts to a high logic level before the reply is sent out through the configured UART_TX pin and deasserts after the reply is finished.

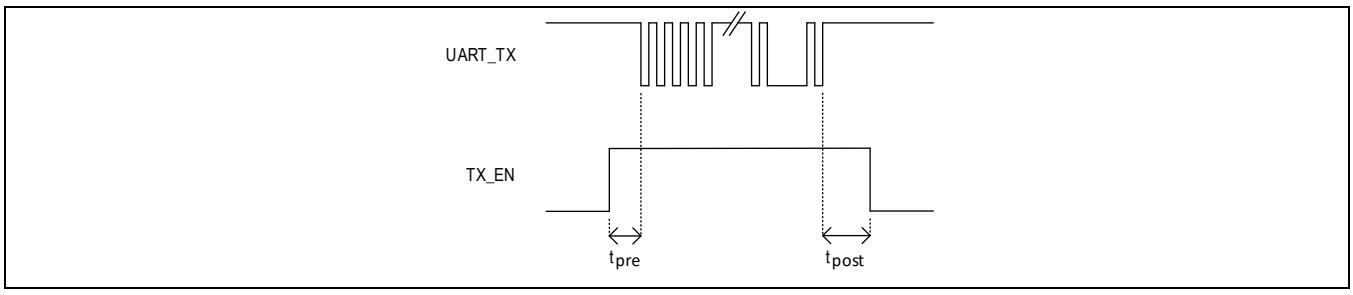

Figure 37. UART Timing Diagram

The pre-delay (t_{Pre}) and the post-delay (t_{Post}) can be individually configured. The TX_EN pin can be configured to GPIO2 or GPIO8. See *Configuration* section for more details.

Bootstrapping RS485

The TMC9660 requires configuration of some settings before being able to correctly communicate over RS485. To allow easy bootstrapping of RS485 connections for a not-yet bootstrapped TMC9660, a special command is available that sets all settings needed. For RS485 connections, this must be the very first command sent.

The BOOTSTRAP_RS485 command configures the following UART and RS485 settings at once:

The TX_EN pin to use (BL_UART_TXEN).

The pre- and post-delay (UART_TXEN_PRE_DELAY, UART_TXEN_POST_DELAY).

The device address (DEVICE_ADDRESS).

The host address (HOST_ADDRESS).

These settings get applied as soon as the BOOTSTRAP_RS485 command gets received before the reply gets sent back out.

Note that this command sets the same value for the pre- and post-delay. Set them to the higher of the two needed values to ensure both timings are valid, then update the faster delay after the RS485 bootstrap step is completed.

After this bootstrapping, the communication works over RS485 and the TMC9660 can be fully configured as usual.

SPI Communication

Communication through SPI comprises 40-bit datagrams. Each datagram, a request gets sent to the TMC9660 and the reply of the previous request gets sent back. When more than 40 bits are sent, the first bits received are shifted out again to allow daisy-chaining of multiple SPI devices.

Table 18. Bootloader SPI Request Communication Byte Format

REQUEST BYTE	NAME	DESCRIPTION
1	Command byte	The command byte. See the *Commands* section.
2-5	Data bytes	The data bytes. The highest byte is sent first.

Table 19. Bootloader SPI Reply Communication Byte Format

REPLY BYTE	NAME	DESCRIPTION
1	Status byte	The status byte. See the *Commands* section.
2-5	Data bytes	The data bytes. The highest byte is sent first.

To receive the reply of a command, send another command. A special NO_OP command is available that does not perform any operation—this allows receiving a status without sending another command with side effects.

The very first command sent receives the special SESSION_START status with the value containing the software version—the upper 16 bits holding the major and the lower 16 bits holding the minor version.

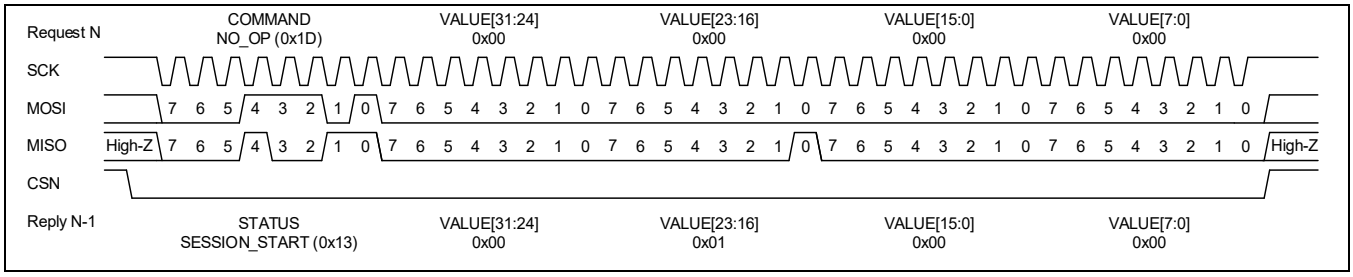

Figure 38. SPI Datagram Example of the Very First Datagram

Memory Access

The TMC9660 bootloader allows access to memory to bootstrap the system. The different types of memory available, referred to as memory banks, are selected using the SET_BANK/GET_BANK commands.

Table 20. Bootloader Memory Bank Decode

NAME	NUMBER	DESCRIPTION

SPI	1	An external SPI Flash
I2C	2	An external I²C Flash
OTP	3	A page of internal OTP memory. Requires extra commands to load and store pages.
CONFIG	5	The configuration memory. Writing to this reconfigures the TMC9660. See the *Configuration* section for more details.

To read or write a memory bank, the TMC9660 bootloader holds a memory address, which can be controlled using the SET_ADDRESS/GET_ADDRESS commands.

Reading and writing is done using a selection of READ_* and WRITE_* commands. They support 32-, 16-, or 8-bit accesses. Additionally, the commands with the _INC suffix increment the memory address by the number of bytes accessed to allow quicker-streamed reading or writing.

For example, the READ_16_INC command reads 16 bits of data and then increases the memory address by 2.

External Memory

The TMC9660 can be used with external SPI Flash and external I2C EEPROM. For either of these external memories, the TMC9660 bootloader requires the memory to be configured, connected, and partitioned.

An external memory is considered as follows:

Configured if the corresponding configuration settings have been made. See to the SPI and I2C configuration in the *Configuration* section for more details.

Connected if the external memory responds to the TMC9660.

Partitioned if a valid partition header is located at the start of the external memory.

The configured and connected external memory states can be queried using the MEM_IS_CONFIGURED, MEM_IS_CONNECTED commands. The partitioned state can be determined using the GET_INFO command with the SPI_MEM_PARTITIONS and I2C_MEM_PARTITIONS values.

The bootloader reloads the partition table information after any external memory configuration change or when requesting the partition count using GET_INFO.

Additionally, an external memory can be busy. This occurs when the external memory is occupied performing a write operation. The busy state can be queried using the MEM_IS_BUSY command.

External Memory Partitions

The TMC9660 requires a partition descriptor at the start of the external memory. The partition descriptor comprises a partition header followed by one or more partition entries.

Table 21. External Memory Partition Header Addresses

OFFSET	SIZE	NAME	DESCRIPTION
0	4	Size	Size of the Partition descriptor in bytes. The partition header is 16 bytes, each partition entry is 24 bytes. The partition description size is therefore $size = 16 + 24 * N_{partitions}$
4	4	Checksum	The CRC32 checksum of the partition description starting at the magic number.
8	2	Magic number	The value 0xCAFE.
10	1	Major version number	The major version of the partition format: 1.
11	1	Minor version number	The minor version of the partition format: 1.
12	1	Size index	The size of the external memory stored in the format: $memory\ size = 2^{size\ index}$ If the size is unknown, the size index is 0.

13	1	Sector index	The size of the external memory erase sectors, stored in the following format: $sector\ size = 2^{sector\ index}$ If the sector size is unknown, the sector index is 0. See *I2C* and *SPI* memory sections for details.
14	2	RESERVED	Reserved. Set to 0.

Table 22. External Memory Partition Entry Addresses

OFFSET	SIZE	NAME	DESCRIPTION
0	12	Partition name	The name of the partition encoded in UTF-8. Unused bytes are zero.
12	1	Partition type	The lower 7 bits contain the partition type: 2: Parameter data to be applied on startup in parameter mode 4: Script data to be run by the motor control application The MSB contains the writable bit, indicating whether the partition is writable at runtime. Note that this bit only provides a hint—the bootloader can always bypass this bit with its direct memory access.
13	3	RESERVED	Reserved. Set to 0.
16	4	Offset	Start of the partition data in the external memory. Must be a multiple of the sector size.
20	4	Size	Size of the partition data in the external memory. Must always be a multiple of the sector size. For Parameter data, it must be at least 0x400 bytes.

SPI Flash

The bootloader accesses SPI Flash using SPI mode 0 and an active-low chip select. All bytes are sent and received most significant bit first.

The bootloader considers an SPI Flash configured if the config bit SPI_FLASH_EN is set. Note that for a successful connection, the other SPI Flash configuration parameters must be set correctly as well:

SPI Flash access frequency

SPI pins (SCK, CSN, MOSI, MISO)

See the *SPI Flash Configuration* for more details.

The bootloader determines whether an SPI Flash chip is connected using the following sequence of datagrams:

Send a chip enable command byte (0xAB).

Send a chip ID read command byte (0x90) followed by five zero bytes. If the fourth byte received is neither 0x00 nor 0xFF, the chip is considered connected, and the sequence stops.

Send a JEDEC ID command byte (0x9F) followed by three bytes. If the second byte received is neither 0x00 nor 0xFF, the chip is considered connected, and the sequence stops.

Send a chip enable command byte (0xAB) followed by four bytes. If the fifth byte received is neither 0x00 nor 0xFF, the chip is considered connected.

Otherwise, the chip is considered not connected.

The bootloader determines whether an SPI Flash is busy by sending a STATUS_READ command byte (0x05) followed by a zero byte. If the LSB of the second response byte is one, the external memory is considered busy.

To read from an SPI Flash, the bootloader uses the read command byte 0x03, followed by three big-endian address bytes, followed by as many bytes as it tries to read.

To write to an SPI Flash, the bootloader first sends datagram with the WRITE_ENABLE byte (0x06), then sends a second datagram with the PAGE_WRITE byte (0x02), followed by three big-endian address bytes, followed by up to four bytes of data to be written.

To erase an SPI Flash sector, the bootloader first sends datagram with the WRITE_ENABLE byte (0x06), then sends a second datagram with the SECTOR_ERASE byte (0x20), followed by three big-endian address bytes.

The bootloader only accesses memory within the bounds defined by the partition's chip size index value. If not partitioned, or if the chip size index is 0, the maximum size of 224 bytes is assumed.

I²C EEPROM

The bootloader considers an I2C EEPROM configured if the config bit I2C_EEPROM_EN is set. Note that for a successful connection, the other I2C EEPROM configuration parameters must be set correctly as well:

I2C EEPROM access frequency

I2C pins (SCL, SDA)

I2C EEPROM chip address

See the *I2C EEPROM Configuration* for more details.

All bytes sent and received are most significant bit first.

The bootloader determines whether an I2C EEPROM chip is connected using the following sequence of datagrams:

Send a 1 byte read request at address 0, then 1, then 2, then 3. If any request succeeds, the chip is considered connected, and the sequence stops.

If all four accesses fail, the chip is considered not connected.

The bootloader determines whether an I2C EEPROM is busy by sending a write command and checking whether an ACK signal is sent back. No write data is sent after the write command.

The I2C EEPROM accesses utilize a control byte comprising seven address bits and one read/write bit. The highest four address bits are fixed to 0b1010, the lower three bits are configurable. The read/write bit is the LSB of the control byte. It is 1 for reads, 0 for writes. For example, when configuring all configurable address bits to 0 and performing a read, the control byte is 0b10100001.

To write to an I2C EEPROM, the bootloader sends the following sequence:

Start condition

Control byte (RW bit is 0)

Address high byte

Address low byte

Write bytes

Stop condition

To read from an I2C EEPROM, the bootloader sends the following sequence:

Start condition

Control byte (RW bit is 0)

Address high byte

Address low byte

Start condition (RW bit is 1)

Read bytes

Stop condition

For the read and write sequences, if any byte receives a NACK response, the bootloader sends a stop condition and abort the sequence.

To check if an I2C EEPROM is busy, the bootloader sends the following sequence and checks whether it receives an ACK or NACK, representing not busy and busy, respectively:

Start condition

Control byte (RW bit is 0)

Stop condition

The bootloader only accesses memory within the bounds defined by the partition's chip size index value. If not partitioned, or if the chip size index is 0, the maximum size of 216 bytes is assumed.

OTP Memory

The OTP memory is organized into 61 pages of 64 bytes plus a three-bit page tag. The OTP offers ECC for each page to correct single-bit errors and detect two-bit errors.

To access these pages, the bootloader offers a 64-byte memory region accessible through the OTP memory bank as well as the OTP_LOAD and OTP_BURN commands to load a page into the region and burn the region into a page, respectively.

The first four pages are reserved for storing the bootloader configuration, see the *Configuration Storage* for more details.

The pages can be burnt in any order except the last page. Once the last OTP page is burnt, no other burn operations are possible.

Applications

The TMC9660 bootloader can start the motor control application located in ROM.

Launching Applications

The {{TM01_A} bootloader supports launching applications in multiple ways—the power-on boot sequence and the manually triggered boot sequence steps.

Both the power-on sequence and the manually triggered boot sequence follow the same steps based on the boot configuration register. The power-on sequence runs if a configuration is burnt into OTP. The manually triggered sequence is run when the boot configuration register is written to. See the *Configuration* section for more details.

Commands

This section details information about all available bootloader commands.

Unless otherwise noted, all commands return an OK status.

Unless otherwise noted, all multiple-byte values are treated as little-endian.

Command Overview

Table 23. Bootloader Command Overview and Decode

NAME	NUMBER	REQUEST VALUE	REPLY		SHORT DESCRIPTION
			POSSIBLE ERRORS	VALUE	
GET_INFO	0	INFO selection	INVALID_VALUE, CMD_NOT_AVAILABLE	INFO value	Get various basic information about the connected TMC9660
GET_BANK	8	d/c	n/a	Bank number	Get the currently selected memory bank
SET_BANK	9	Bank number	INVALID_VALUE	Request value	Set the memory bank
GET_ADDRESS	10	d/c		Memory address	Get the current memory address
SET_ADDRESS	11	Memory address	INVALID_VALUE	Request value	Set the memory address
READ_32	12	d/c	INVALID_ADDR	Read data	

READ_32_INC	13				Reads data from the selected memory bank at the selected memory address.
READ_16	14				
READ_16_INC	15				
READ_8	16				
READ_8_INC	17				
WRITE_32	18	Write data	INVALID_ADDR	Request value	Writes data to the selected memory bank at the selected memory address.
WRITE_32_INC	19				
WRITE_16	20				
WRITE_16_INC	21				
WRITE_8	22				
WRITE_8_INC	23				
NO_OP	29	d/c	n/a	Request value	Do nothing. Useful for SPI communication to retrieve responses for the last command.
OTP_LOAD	30	OTP page	INVALID_VALUE, OTP_ERROR	OTP page error count and address	Read a programmed OTP page
OTP_BURN	31	OTP page, address	INVALID_VALUE, OTP_ERROR	OTP status code	Burn an OTP page. Note: This command has an Erratum – use it only with the workaround described in *Erratum 1: Bootloader OTP_BURN Command*.
MEM_IS_CONFIGURED	32	Memory bank	n/a	Configured status	Check whether an external memory bank is configured
MEM_IS_CONNECTED	33	Memory bank	MEM_UNCONFIGURED	Connected status	Check whether an external memory is connected
FLASH_SEND_CMD	36	Command data	MEM_UNCONFIGURED	Command data	Send arbitrary commands to an external flash
FLASH_ERASE_SECTOR	37	Sector address	MEM_UNCONFIGURED	Request value	Send a sector erase command to an external flash
MEM_IS_BUSY	40	Memory bank	MEM_UNCONFIGURED	Busy status	Check whether an external memory is busy
BOOTSTRAP_RS485	255	RS485 settings	INVALID_VALUE	Request value	Set up RS485 settings. This is used to bootstrap communication to use the TX_EN pin.

Reply Status Codes

NAME	NUMBER	DESCRIPTION
OK	0	Command executed successfully
CMD_NOT_FOUND	1	The request has an invalid command number.
INVALID_ADDR	3	The memory address is not valid for the requested command.
INVALID_VALUE	4	The request has an invalid value.
INVALID_BANK	14	The memory bank is not valid for the requested command.

BUSY	15	This status code is only for the SPI communication. Indicates that the bootloader has not yet finished processing the last command.
MEM_UNCONFIGURED	17	The external memory is not configured.
OTP_ERROR	18	The OTP command has failed. Refer to the OTP command for more details.
SESSION_START	19	This status code is only for the SPI communication. This status code is sent during the first SPI datagram after power-on, when no prior command has yet been processed and therefore no prior status exists.
CMD_NOT_AVAILABLE	20	The command is currently not available.
BOOTLOADER_RESUMED	21	This status code is only for the SPI communication. This status code is sent during the first SPI datagram after returning to the bootloader from the motor control system.

GET_INFO

The GET_INFO command allows readout of various basic information about the connected TMC9660.

Table 24. Bootloader Command GET_INFO Decode

NUMBER	NAME	DESCRIPTION
0	CHIP_TYPE	Get the Chip type. Returns 0x544D0001.
1	BL_VERSION	Returns the version of the software bootloader. The upper 16 bits hold the major version, the lower 16 bits the minor version.
2	FEATURES	Returns what feature groups are available. Each bit corresponds to one feature: Bit 0: SRAM support Bit 1: ROM Bit 2: OTP Bit 3: SPI flash external memory Bit 4: I^2C EEPROM external memory
12	GIT_INFO	Returns the Git version control information: Bit 28: Dirty bit - Firmware build has uncommitted changes Bits 27-0: 7-digit hex commit hash. For example: A reply of 0x0D00BA81 refers to commit d00ba81 with no local changes (no dirty flag).
13	CHIP_VERSION	Silicon revision TMC9660 reports revision 1.
14	CHIP_FREQUENCY	System frequency in MHz. See the Clock Configuration section for details.
17	CONFIG_MEM_START	Returns the starting address of the CONFIG memory
18	CONFIG_MEM_SIZE	Returns the size of the CONFIG memory
19	OTP_MEM_SIZE	Returns the size of one OTP memory page
20	I2C_MEM_SIZE	Returns the memory size of the connected, partitioned I^2C memory, or the maximum possible size (2^{16} bytes) if unknown. Note: The memory size is stored inside the external memory partition header. See *External Memory Partitions* section for details.
21	SPI_MEM_SIZE	Returns the memory size of the connected, partitioned SPI memory, or the maximum possible size (2^{24} bytes) if unknown. Note: The memory size is stored inside the external memory partition header. See *External Memory* section for details.
22	PARTITION_VERSION	Returns the version of the external memory partition format. Bits 15-8: Major version

		Bits 7-0: Minor version The TMC9660 reports a version of 1.1 (value = 0x00000101).
25	SPI_MEM_PARTITIONS	Returns the number of partitions available of the connected, partitioned SPI memory. If the memory is not configured, connected, or partitioned, returns CMD_NOT_AVAILABLE error. This command requires that the currently selected *memory bank* is *SPI*.
26	I2C_MEM_PARTITIONS	Returns the number of partitions available of the connected, partitioned I^2C memory. If the memory is not configured, connected, or partitioned, returns CMD_NOT_AVAILABLE error. This command requires that the currently selected *memory bank* is *I2C*.
28	CHIP_VARIANT	Returns the chip variant. The TMC9660 reports a value of 2.

GET_BANK

Returns the current memory bank.

See *Memory Access* section for more details.

SET_BANK

Sets the memory bank.

When the previously selected memory bank set is already active, nothing happens.

When the previously selected memory bank is different, the memory bank gets updated, and the memory address gets set to the start of the newly selected memory.

When an invalid memory bank number is requested, the command returns INVALID_VALUE.

See *Memory Access* section for more details.

GET_ADDRESS

Returns the current memory address.

See *Memory Access* section for more details.

SET_ADDRESS

Sets the memory address.

When the requested memory address is out of bounds of the selected memory bank, the command returns INVALID_VALUE.

See *Memory Access* section for more details.

READ_32

Reads 32 bits of data from the selected memory bank at the selected memory address.

If the memory address is not aligned to 4 bytes, the command returns INVALID_ADDR.

If the memory address is not within the bounds of the selected memory, the command returns INVALID_ADDR.

READ_32_INC

Reads 32 bits of data from the selected memory bank at the selected memory address and increments the memory address by 4.

If the memory address is not aligned to 4 bytes, the command returns INVALID_ADDR and doesn't increment the memory address.

If the memory address is not within the bounds of the selected memory, the command returns INVALID_ADDR and doesn't increment the memory address.

READ_16

Reads 16 bits of data from the selected memory bank at the selected memory address.

The upper 16 bits of the response value are 0.

If the memory address is not aligned to 2 bytes, the command returns INVALID_ADDR.

If the memory address is not within the bounds of the selected memory, the command returns INVALID_ADDR.

READ_16_INC

Reads 16 bits of data from the selected memory bank at the selected memory address and increments the memory address by 2.

The upper 16 bits of the response value are 0.

If the memory address is not aligned to 2 bytes, the command returns INVALID_ADDR and doesn't increment the memory address.

If the memory address is not within the bounds of the selected memory, the command returns INVALID_ADDR and doesn't increment the memory address.

READ_8

Reads 8 bits of data from the selected memory bank at the selected memory address and increments the memory address by 1.

The upper 24 bits of the response value are 0.

If the memory address is not within the bounds of the selected memory, the command returns INVALID_ADDR.

READ_8_INC

Reads 8 bits of data from the selected memory bank at the selected memory address. The upper 24 bits of the response value are 0.

If the memory address is not within the bounds of the selected memory, the command returns INVALID_ADDR ADDR and doesn't increment the memory address.

WRITE_32

Writes 32 bits of data to the selected memory bank at the selected memory address.

If the memory address is not aligned to 4 bytes, the command returns INVALID_ADDR.

If the memory address is not within the bounds of the selected memory, the command returns INVALID_ADDR.

WRITE_32_INC

Writes 32 bits of data to the selected memory bank at the selected memory address and increments the memory address by 4.

If the memory address is not aligned to 4 bytes, the command returns INVALID_ADDR and doesn't increment the memory address.

If the memory address is not within the bounds of the selected memory, the command returns INVALID_ADDR and doesn't increment the memory address.

WRITE_16

Writes 16 bits of data to the selected memory bank at the selected memory address.

The upper 16 bits of the request value are ignored.

If the memory address is not aligned to 2 bytes, the command returns INVALID_ADDR.

If the memory address is not within the bounds of the selected memory, the command returns INVALID_ADDR.

WRITE_16_INC

Writes 16 bits of data to the selected memory bank at the selected memory address and increments the memory address by 2.

The upper 16 bits of the request value are ignored.

If the memory address is not aligned to 2 bytes, the command returns INVALID_ADDR and doesn't increment the memory address.

If the memory address is not within the bounds of the selected memory, the command returns INVALID_ADDR and doesn't increment the memory address.

WRITE_8

Writes 8 bits of data to the selected memory bank at the selected memory address.

The upper 24 bits of the request value are ignored.

If the memory address is not within the bounds of the selected memory, the command returns INVALID_ADDR.

WRITE_8_INC

Writes 8 bits of data to the selected memory bank at the selected memory address and increments the memory address by 1.

The upper 24 bits of the request value are ignored.

If the memory address is not within the bounds of the selected memory, the command returns INVALID_ADDR and doesn't increment the memory address.

NO_OP

This command does nothing.

It is intended to be used with SPI communication to retrieve the reply of the previous command. See the *SPI Communication* section for more details.

OTP_LOAD

Load an OTP page into the OTP memory bank.

The request value selects which OTP page to load. The return value consists of the OTP bit error count in bits 15-8 and the OTP page tag in bits 7-0.

If the selected OTP page exceeds the amount of available OTP pages, the command returns INVALID_VALUE.

If the loading of the OTP page failed, the command returns OTP_ERROR.

See the *OTP Memory* section for more details.

OTP_BURN

Note: This command has an Erratum – use it only with the workaround described in *Erratum 1: Bootloader OTP_BURN Command*.

Burn the contents of the OTP memory bank into an OTP page.

The request value selects which OTP page to burn with bits 7-0 and what OTP page address to write into the OTP page with bits 15-8.

If the selected OTP page exceeds the amount of available OTP pages, the command returns INVALID_VALUE.

If the burning of the OTP page failed, the command returns OTP_ERROR and the reply value contains additional error information:

ERROR CODE	DESCRIPTION
-1	The OTP page number is invalid.
-2	The last OTP page has been burnt and no more burn operations are possible.
-3	Setting up the internal OTP charge pump failed.
-4	The burn procedure failed. Note: Whether the OTP page has erroneous data burnt in must be manually checked using OTP_LOAD.
-5	Internal clock setup for OTP operation failed
-6	Restoring original clock setup after OTP operation failed.

See the *OTP Memory* section for more details.

MEM_IS_CONFIGURED

Returns whether an external memory is configured. See the *External Memory* section for more details.

The request value selects the memory bank to check.

The reply value is 1 if the external memory is configured or 0 otherwise. All other memory types always return a reply value of 0.

MEM_IS_CONNECTED

Returns whether an external memory is connected. See the *External Memory* section for more details.

The request value selects the memory bank to check.

The reply value is 1 if the external memory is connected or 0 otherwise. All other memory types always return a reply value of 0.

If the requested external memory is not configured, the command returns MEM_UNCONFIGURED.

FLASH_SEND_CMD

Send arbitrary commands through to an external SPI flash. This command uses an internal buffer of 6 bytes to construct and send datagrams and to receive and store the replies for reading.

Bits 31-28 control what action this command takes:

0: Load bits 23-0 into the buffer. Bits 27-24 control at what byte offset the buffer shall be filled.
1: Read out 3 bytes from the buffer. Bits 27-24 control at what byte offset the buffer shall be read. The response value consists of:

- Bits 31-24: The request bits 31-24.
- Bits 23-16: The first byte read out from the buffer at byte offset + 0.
- Bits 15-8: The second byte read out from the buffer at byte offset +1.
- Bits 7-0: The third byte read out from the buffer at byte offset + 2.

2: Send a datagram with the current buffer contents and override the buffer with the reply. Bits 27-24 control how many bytes to transmit.

For example, to read out the JEDEC manufacturer ID of a flash, the command byte 0x9F followed by one more byte must be sent. The response will contain the manufacturer ID in the second byte sent back. To do this, the following bootloader commands must be sent:

1. FLASH_SEND_CMD, value 0x009F0000. This fills the internal buffer with the bytes 0x9F, 0x00.
2. FLASH_SEND_CMD, value 0x22000000. This causes two bytes to be sent and received.
1. 3. FLASH_SEND_CMD, value 0x10000000. This reads out the first two bytes of the internal buffer. The response will hold the manufacturer ID in the response bits 15-8.

FLASH_ERASE_SECTOR

Erase an external SPI Flash sector.

This command sends the SPI Flash the command byte 20h followed by the 24-bit address in big-endian format. For example, a request to delete the sector at address 0x010200 would send the bytes 0x20 0x01 0x02 0x00 to the SPI flash. See the *SPI Flash* section for more details.

If the requested external memory is not configured, the command returns MEM_UNCONFIGURED.

MEM_IS_BUSY

Returns whether an external memory is busy. See the *External Memory* section for more details.

The request value selects the memory bank to check.

The reply value is 1 if the external memory is busy or 0 otherwise. All other memory types always return a reply value of 0.

If the requested external memory is not configured, the command returns MEM_UNCONFIGURED.

BOOTSTRAP_RS485

Configures the TMC9660 to respond through RS485. See the *RS485 Communication* section for more details.

The request contains all the configuration needed for using the TX_EN pin for sending back replies. This command must be sent as the very first command for RS485 connected ***{Variable not found in the variables list}*** chips so that replies can be received.

The request value contains the configuration for the TX_EN usage:

- Byte 0 contains what pin to use for TX_EN: 1 for GPIO8, 2 for GPIO2.
- Byte 1 contains how long to wait between controlling the TX_EN pin and sending data.
- Byte 2 contains the host address to use.
- Byte 3 contains the device address to use.

If the TX_EN pin selection is invalid, the command returns INVALID_VALUE. Note that due to the nature of RS485, this means the TX_EN pin is not yet configured, and the reply will not be able to travel over RS485.

Configuration

The TMC9660 bootloader allows flexible configuration of various settings, which are as follows:

- LDO outputs (VEXT1, VEXT2)
- Clock settings
- Communication interfaces
- GPIO pin usage
- External memory (SPI Flash, I^2C EEPROM)
- Motor systems*:
  - Feedback systems
  - StepDir input
  - Brakechopper
  - Mechanical brake
  - Motor parameter storage
  - Script control

*The motor systems related configurations are detailed in the TMC9660 parameter mode data sheet.

The configuration can be written to the TMC9660 OTP storage to automatically be applied at power-on, or it can be written at runtime to evaluate each setting individually without using up the limited OTP burn cycles. It is suggested to first evaluate the configuration options using the runtime mechanism before committing the configuration to OTP memory.

Runtime Reconfiguration

To reconfigure the TMC9660 while it is active, the bootloader offers the CONFIG memory bank. It exposes a 64-byte memory region located at address 0x00020000. Writing to this memory region using WRITE_* commands cause the bootloader to update the TMC9660 configuration.

The location and size of the CONFIG memory bank can also be queried using the GET_INFO command with the CONFIG_MEM_START and CONFIG_MEM_SIZE values.

For example, to change the DEVICE_ADDRESS to 3 and the HOST_ADDRESS to 4, run the following commands:

- SET_BANK: 5 (CONFIG)
- SET_ADDRESS: 0x00020002 (Base address 0x00020000 + Offset 2: Device and host address config)
- WRITE_16: 0x0403*

*When communicating over UART, the reply gets sent before applying the configuration change. This means the reply to this example WRITE_16 still replies with the prior device and host addresses.

Note that different sized WRITE_* commands are permitted—setting just the HOST_ADDRESS may also be done by setting the memory address to 0x00020003 and sending a WRITE_8 command. Each WRITE_* command triggers a reconfiguration, do not split up bigger writes into smaller ones if it would create an invalid intermediate configuration. The alignment limitations specified in the WRITE_* command descriptions still apply.

Configuration Storage

A given configuration can be stored in OTP memory to be applied automatically on bootup. This enables automatically starting the TMC9660 in the desired configuration.

The configuration can be stored in OTP memory to automatically apply it on power-on. To store a configuration, they can be burnt into one of the first four pages of the OTP with a page tag value of 4.

The bootloader then checks the first four pages for such a configuration, going backwards from page 3 to page 0. The first valid configuration found is then applied. This allows burning the pages from 0 to 3 for a total of four times, allowing the updating of the burnt configuration.

If the bootloader finds a 2-bit ECC error before finding a valid configuration, it will assert the FAULTN pin and enter bootloader mode with the default configuration. The following table shows examples of how the bootloader determines its initial configuration:

PAGE 0	PAGE 1	PAGE 2	PAGE 3	BOOTLOADER BEHAVIOR
Empty	Empty	Empty	Empty	Load default configuration values
Configured	Empty	Empty	Empty	Load configuration from page 0
Configured	Configured	Empty	Empty	Load configuration from page 1
2-Bit Error	2-Bit Error	2-Bit Error	Configured	Load configuration from page 3
Empty	Empty	2-Bit Error	Empty	Fault and load default configuration
Configured	2-Bit Error	Empty	Empty	Fault and load default configuration
2-Bit Error	Configured	Configured	2-Bit Error	Fault and load default configuration

Configuration List

LDO Configuration

NAME	OFFSET	BITS	DESCRIPTION
VEXT1	0	0-1	Sets the voltage the LDO shall output on the VEXT1 pin: 0: LDO disabled 1: 2.5V 2: 3.3V 3: 5.0V
VEXT2	0	2-3	Sets the voltage the LDO shall output on the VEXT2 pin: 0: LDO disabled 1: 2.5V 2: 3.3V 3: 5.0V
SS_VEXT1	0	4-5	Sets the VEXT1 voltage slope speed, controlling the startup time: 0: 3ms 1: 1.5ms 2: 0.75ms 3: 0.37ms
SS_VEXT2	0	6-7	Sets the VEXT2 voltage slope speed, controlling the startup time: 0: 3ms 1: 1.5ms 2: 0.75ms 3: 0.37ms
LDO_SHORT_FAULT	0	8	Controls whether a detected LDO short asserts the FAULTN pin. Default: 0

Bootstrap Configuration

NAME	OFFSET	BITS	DESCRIPTION
BOOT_MODE	8	0-1	Selects the motor control mode: 0: RESERVED 1: Register mode 2: Parameter mode 3: RESERVED
BL_READY_FAULT	8	2	Configures whether the FAULTN pin asserts when the bootloader is ready to communicate. Default: 0
BL_EXIT_FAULT	8	3	Configures whether the FAULTN pin asserts when the bootloader launches the motor application. Default: 1
DISABLE_SELFTEST	8	8	If set, disables the self-test of internal ROM and SRAM memory. This saves 34ms (typical) of boot time when launching the motor control system directly from power-on (START_MOTOR_CONTROL=1). Note: This bit only affects the power-on self-test, and therefore is only relevant when permanently burning a config into the OTP. Reconfiguring this bit at runtime has no effect. Default: 0
BL_CONFIG_FAULT	8	9	Configures whether the FAULTN pin asserts during the application for a configuration option. Default: 0
START_MOTOR_CTRL	8	12	Start the motor control, based on the BOOT_MODE selection. Default: 0

UART Configuration

NAME	OFFSET	BITS	DESCRIPTION
DEVICE_ADDRESS	2	0-7	Device address Default: 1
HOST_ADDRESS	2	8-15	Host address Default: 255
BL_DISABLE_UART	6	0	Disable the bootloader UART connection Default: 0
BL_UART_RX	6	3	Select the UART TX pin: 0: GPIO7 1: GPIO1
BL_UART_TX	6	4	Select the UART TX pin: 0: GPIO6 1: GPIO0
BL_UART_BAUDRATE	6	7-9	Select the UART baudrate: 0: 9600 1: 19200 2: 38400 3: 57600 4: 115200 5: 1000000 6: Autobaud detection at 8x sampling 7: Autobaud detection at 16x sampling

These settings get applied when they get changed, when the system frequency (f_{System}) changes, or when the motor control is started and exited.

RS485 Configuration

The RS485 configuration is based on the UART configuration, with the following extra options:

NAME	OFFSET	BITS	DESCRIPTION
UART_TXEN_POST_DELAY	4	0-7	Delay between end of TX transmission and TX_EN deassertion. (t_{Post}) Default: 0
UART_TXEN_PRE_DELAY	4	8-15	Delay between TX_EN assertion and beginning of TX transmission. (t_{Pre}) Default: 0
BL_UART_TXEN	6	5-6	Select the UART_TXEN pin: 0: UART_TXEN not used 1: UART_TXEN on GPIO8 2: UART_TXEN on GPIO2 3: RESERVED

These settings get applied when they get changed, when the system frequency (f_{System}) changes, or when the motor control is started and exited.

SPI Communication Configuration

NAME	OFFSET	BITS	DESCRIPTION
BL_DISABLE_SPI	6	1	Disable the bootloader SPI connection Default: 0
BL_SPI_SELECT	6	2	Select which SPI interface to use for the bootloader connections: 0: SPI0 for bootloader connection 1: SPI1 for bootloader connection Note: This bit is shared with the SPI flash configuration. If both flash and bootloader SPI communication are used, they must always be on separate SPI interfaces for bootloader usage.
BL_SPI0_SCK	6	10	Select which SCK pin to use for SPI0: 0: GPIO6 1: GPIO11 Note: This is bit is only required for the SPI bootloader communication if it uses SPI0 (BL_SPI_SELECT=0)

These settings get applied when they get changed, when the system frequency (f_{System}) changes, or when the motor control is started and exited.

SPI Flash Configuration

NAME	OFFSET	BITS	DESCRIPTION
SPI_FLASH_EN	10	0	Enable the usage of SPI Flash Default: 0
BL_SPI_SELECT	6	2	Select which SPI interface to use for the SPI Flash: 0: SPI1 for SPI Flash 1: SPI0 for SPI Flash Note: This bit is shared with the SPI bootloader communication. If both flash and bootloader SPI communication are used, they must always be on separate SPIs.
BL_SPI0_SCK	6	10	Select which SCK pin to use for SPI0: 0: GPIO6 1: GPIO11 Note: This is bit is only required for the SPI flash usage if it uses SPI0 (BL_SPI_SELECT=1)
SPI_FLASH_CS	10	3-7	GPIO pin number of the CS pin to use Default: 0
SPI_FLASH_FREQ	10	8-11	Select the SPI flash frequency: $f_{SPIFlash} = \frac{f_{System}}{SPI\_FLASH\_FREQ + 1}$

			The system frequency (f_{System}) is 40MHz. The maximum frequency is 10MHz, requiring a minimum value for SPI_FLASH_FREQ of 3 when using SPI flash. Default: 0

These settings get applied when they get changed, when the system frequency (f_{System}) changes, or when the motor control is started and exited.

I²C EEPROM Configuration

NAME	OFFSET	BITS	DESCRIPTION
I2C_EEPROM_EN	12	0	Enable the usage of I²C EEPROM Default: 0
I2C_EEPROM_SDA	12	1-2	Select which SDA pin to use: 0: GPIO5 1: GPIO11 2: GPIO14 3: RESERVED
I2C_EEPROM_SCL	12	3-4	Select which SCL pin to use: 0: GPIO4 1: GPIO12 2: GPIO13 3: RESERVED
I2C_EEPROM_ADDR	12	5-7	The configurable bits of the I2C device address. See the *I2C EEPROM* memory section for details. Default: 0
I2C_EEPROM_FREQ	12	8-10	Select what I²C frequency to use: 0: 100kHz 1: 200kHz 2: 400kHz 3: 800kHz 4: RESERVED 5: RESERVED 6: RESERVED 7: RESERVED

These settings get applied when they get changed, when the system frequency (f_{System}) changes, or when the motor control is started and exited.

Clock Configuration

NAME	OFFSET	BITS	DESCRIPTION
RESERVED_1	24	0-6	Reserved. Must always stay set to 99. Default: 99
EXT_NOT_INT	24	8	Select whether to use internal oscillator or external clock source for the PLL input. 0: Internal 15MHz oscillator 1: External clock source selected by EXT_NOT_XTAL.
XTAL_CFG	24	9-11	Drive current selection for external oscillators based on their frequency: 0: RESERVED 1: 8MHz 2: RESERVED 3: 16MHz 4: RESERVED 5: 24MHz-25MHz 6: 32MHz 7: RESERVED
XTAL_BOOST	24	12	If enabled, the TMC9660 drives the maximum external oscillator current while the oscillator is starting to speed up oscillator start-up.

			Default: 0
EXT_NOT_XTAL	24	13	Select whether to use an external clock or an external oscillator: 0: External oscillator 1: External clock
PLL_OUT_SEL	24	16-17	Select whether to use the internal oscillator or the PLL as the system clock: 0. Use the internal oscillator 1: Use the PLL 2: RESERVED 3: RESERVED
RDIV	24	18-22	Divider of the PLL input frequency. Must be set to the input frequency in MHz minus one. The internal oscillator has a frequency of 15 MHz. For Internal oscillator input (EXT_NOT_INT=0), RDIV must be set to 14. Default: 14
SYS_CLK_DIV	24	23-24	Select the system clock frequency: 0: 40MHz 1: RESERVED 2: RESERVED 3: 15MHz Note: Only change this setting during the workaround for *Erratum 1: Bootloader OTP_BURN Command*.
PLL_STATUS	24	30	This bit is read-only. When clock configuration is started, it is cleared. When it completes successfully, it is set.

Any change to the clock configuration takes multiple milliseconds to apply. Wait for it to complete before sending any further datagrams. See the *Bootloader FAULTN Signaling* section on how to configure the FAULTN pin to signal completion of this update.

It is advised to read back the PLL_STATUS bit after updating the clock configuration to verify correct reconfiguration. If the clock update failed, verify your circuit and that your settings match that circuit.

Note: Running the motor control system is only supported with system clock configured to 40 MHz (PLL_OUT_SEL=1 with a valid PLL configuration and SYS_CLK_DIV=0).

Note: The clock configuration should be written with a single WRITE_32 or WRITE_32_INC command.

GPIO Configuration

NAME	OFFSET	BITS	DESCRIPTION
GPIOx_OUT	14	0-15	Output level of GPIO outputs (GPIOx_OUT_EN=1). GPIOs 0-15 are set using bits 0-15 respectively at offset 14. GPIOs 16-18 are set using bits 0-2 respectively at offset 22. Default: 0 for all GPIOs
	22	0-2	
GPIOx_OUT_EN	16	0-15	Direction of GPIOs: 0: Input 1: Output GPIOs 0-15 are set using bits 0-15 respectively at offset 16. GPIOs 16-18 are set using bits 3-5 respectively at offset 22.
	22	3-5	
GPIOx_PU	18	0-15	Enable the GPIO pull-up resistor. GPIOs 0-15 are set using bits 0-15 respectively at offset 18. GPIOs 16-18 are set using bits 9-11 respectively at offset 22. Default: 0 for GPIOs 0-5, 1 for GPIOs 6-18
	22	9-11	
GPIOx_PD	20	0-15	Enable the GPIO pull-up resistor. GPIOs 0-15 are set using bits 0-15 respectively at offset 20. GPIOs 16-18 are set using bits 6-8 respectively at offset 22. Default: 1 for GPIOs 0-1, 0 for GPIOs 2-18
	22	6-8	

GPIOx_ANALOG_EN	22	12-15	Configures the GPIO to an analog input. GPIOs 2-5 are set using bits 12-15 respectively. Default: 0 for GPIOs 0-1, 0 for GPIOs 2-18

To use a GPIO as input/output, no other function must be configured on that pin.

When both Pull-up and Pull-down are configured, the pull-down has priority.

When a GPIO is configured as analog input, the pull-up/pull-down settings are ignored—no pull resistors are active.

Checksum Calculations

The TMC9660 bootloader utilizes two different cyclic redundancy checks (CRC) to verify data integrity—a CRC8 for the UART/RS485 communication and a CRC32 for data stored in memory.

CRC8 Calculation

The CRC8 calculation is used to secure the UART datagram requests and replies.

The polynomial for the checksum is $x^8+x^2+x^1+x^0$. The 8th byte of the UART datagram holds the CRC value calculated over the prior 7 bytes. There are no bit inversions (binary exclusive-or with 1) before or after the calculation. Each input byte is bit-flipped, and the result is also bit-flipped:

Example datagram: GET_INFO (command 0), SW_VERSION (value 1) request to device address 1

The final datagram looks like this:
Sync byte: 0x55
Address byte: 1
Command byte: 0
Data bytes: 0, 0, 0, 1
CRC checksum: 0x1D (See calculation below)

To calculate this CRC value, first concatenate all 7 input bytes, each byte having its least significant bit first (e.g. the sync byte with a binary value of 01010101 must be written as 10101010):

Sync Address Command Data
10101010 10000000 00000000 00000000 00000000 00000000 10000000
CRC Calculation step 1 - Append eight 0 bits:
10101010 10000000 00000000 00000000 00000000 00000000 10000000 00000000

CRC calculation step 2 – Create the binary polynomial representation:
For each exponent in the polynomial, set the corresponding bit to 1.
For the polynomial of x^8 + x^2 + x^1 + x^0 this means bits 8, 2, 1 and 0 are 1:
Bit index: 876543210
Polynomial: 100000111
Note: The polynomial for an 8 bit CRC is 9 bits long!

CRC Calculation step 3 – XOR the polynomial with the input data, with the leftmost 1 bit of the input aligned with the leftmost 1 bit of the polynomial. Repeat this until all input bits before the 8 appended bits are zero:

10101010100000000000000000000000000000000000000010000000000000000
^ 100000111
00101001000000000000000000000000000000000000000010000000000000000
100000111
00001001111000000000000000000000000000000000000010000000000000000
100000111
00000001110110000000000000000000000000000000000010000000000000000
100000111
00000000110111110000000000000000000000000000000010000000000000000
100000111
00000000010111001000000000000000000000000000000010000000000000000
100000111

00000000000011101010000000000000000000000000000010000000000000000
100000111
00000000000011010011000000000000000000000000000010000000000000000
100000111
00000000000001010000100000000000000000000000000010000000000000000
100000111
00000000000000010001010000000000000000000000000010000000000000000
100000111
00000000000000000010011000000000000000000000000010000000000000000
100000111
00000000000000000000110111000000000000000000000010000000000000000
100000111
00000000000000000000010111111000000000000000000010000000000000000
100000111
00000000000000000000000111100100000000000000000010000000000000000
100000111
00000000000000000000000011100011000000000000000010000000000000000
100000111
00000000000000000000000011000010100000000000000010000000000000000
100000111
00000000000000000000000000100001010000000000000010000000000000000
100000111
00000000000000000000000000000011010000000000000010000000000000000
100000111
00000000000000000000000000000010100111000000100000000000000000000
100000111
00000000000000000000000000000000100100100000100000000000000000000
100000111
00000000000000000000000000000000000100011000100000000000000000000
100000111
00000000000000000000000000000000000000011111100000000000000000000
100000111
00000000000000000000000000000000000000011111111000000000000000000
100000111
00000000000000000000000000000000000000011111001000000000000000000
100000111
00000000000000000000000000000000000000011110101000000000000000000
100000111
00000000000000000000000000000000000000011101101000000000000000000
100000111
00000000000000000000000000000000000000011011101000000000000000000
100000111
00000000000000000000000000000000000000010111101000000000000000000
100000111
00000000000000000000000000000000000000001111101000000000000000000
100000111
00000000000000000000000000000000000000000111100110000000000000000
100000111
00000000000000000000000000000000000000000011100001000000000000000
100000111
00000000000000000000000000000000000000000001100010100000000000000
100000111
00000000000000000000000000000000000000000000100011010000000000000
100000111

00000000000000000000000000000000000000000000000000000000011101

CRC Calculation step 4: Take the remaining 8 bits, with the most significant bit first. This is the CRC checksum. For this example datagram, the CRC checksum is 00011101 (0x1D)

Figure 39. Checksum Calculation for Commands

CRC32 Calculation Example

The CRC32 calculation is used to secure memory contents. It is used for the external memory partition table description and the internal ROM self-test.

This CRC32 calculation is equivalent to the one used in the Ethernet protocol. The polynomial used for it is $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x^1+x^0$. The first four input bytes are inverted (exclusive-or with 1), all input bytes are bit-flipped (LSB-first), and the result of the polynomial division is both inverted and bit-flipped:

Example calculation: Checksum of a memory partition with the following values:
Partition length: 40 (1 partition)
Checksum: 0x6CA0265D (see calculation below)
Chip size index: 19 (512 KiB)
Sector index: 12 (4 KiB)
Partition name: “motor_cfg”
Partition type: 2 (parameter data)
Partition offset: 0x00001000
Partition size: 0x00001000

The final partition data structure looks like this:
00: 28 00 00 00 5D 26 A0 6C
08: FE CA 01 01 13 0C 00 00
10: 6D 6F 74 6F 72 5F 63 66
18: 67 00 00 00 82 00 00 00
20: 00 10 00 00 00 10 00 00

Note: The calculation example has some of the bits in the middle omitted due to the number of bits present in this calculation. Additionally, not the full length of step 5 is shown for the same reason. For a full calculation example without omitted bits, refer to the CRC8 example calculation above.

For the partition data structure, the checksum is calculated starting at byte offset 8 and ending at the end of the partition structure:

Field:	Magic number	Version number	size index	Last partition byte
Offset:	0x08	0x09	0x0A	0x0B	0x0C ... 0x27
Data (hex):	0xFE	0xCA	0x01	0x01	0x13 ... 0x00

Data (binary): 11111110 11001010 00000001 00000001 00010011 00000000

CRC calculation step 1 – Concatenate all the input bytes, each byte having its least significant bit first (e.g. the first magic number byte with a binary value of 11111110 must be written as 01111111):
01111111 01010011 10000000 10000000 11001000 ... 00000000

CRC calculation step 2 - Append 32 zero bits:
01111111 01010011 10000000 10000000 11001000 ... 00000000000000000000000000000000000000000000

CRC calculation step 3 – Invert the first 32 bits:
10000000 10101100 01111111 01111111 11001000 ... 00000000000000000000000000000000000000000000

CRC calculation step 4 – Create the binary representation of the polynomial:
For each exponent in the polynomial, set the corresponding bit to 1.
Polynomial: x^32 + x^26 + x^23 + x^22 + x^16 + x^12 + x^11 + x^10

+ x^8 + x^7 + x^5 + x^4 + x^2 + x^1 + x^0

Bit index: 33222222222221111111110000000000
21098765432109876543210987654321 0
Binary polynomial: 100000100110000010001110110110111
Note: The polynomial for a 32-bit CRC is 33 bits long!

CRC calculation step 5 – XOR the polynomial with the input data, with the leftmost 1 bit of the input aligned with the leftmost 1 bit of the polynomial. Repeat this until all input bits before the 32 appended bits are zero:

01111111010100111000000010000000011001000 ... 00000000
100000001010110001111111011111111111001000 ... 000000000000000000000000000000000000000000
100000100110000010001110110110111
00000010110011001111000110100100010010000 ... 000000000000000000000000000000000000000000
100000100110000010001110110110111
00000000110001010111001110011111100100110 ... 000000000000000000000000000000000000000000

...

00000000000000000000000000000000000000000000000010011000001001100100100000011101100000
100000100110000010001110110110111
00000000000000000000000000000000000000000000000011010010001101100011011100000100000
100000100110000010001110110110111
00000000000000000000000000000000000000000000000010100000101011010111001110111110
100000100110000010001110110110111
00000000000000000000000000000000000000000000000010001011001101111111010110011001

CRC calculation step 6 – Take the remaining 32 bits from the previous step and flip the bit direction – the least significant bit (LSB) first:
32 result bits (MSB first): 01000101100110111111101011001001
32 result bits (LSB first): 10010011010111111101100110100010

CRC calculation step 7 – Invert the 32 bits to get the CRC checksum. For this example calculation, the CRC checksum is 0x6AA0265D:
32 result bits (LSB first): 10010011010111111101100110100010
32 result bits (LSB first, inverted): 01101100101000000010011001011101 = 0x6CA0265D

Figure 40. Checksum Calculation for Commands

Applications Information

Bootstrap Capacitor Selection

Neglecting internal losses, the Voltage Ripple on the Bootstrap Line is given by the following formula:

$$\Delta V = \frac{Q_{GATE}}{C_{CBST}}$$

where

Q_{GATE} is the Total Gate capacitance of the external High-Side NFET.

Assuming a maximum tolerable ripple of ~0.5V, CBST can be sized based on the following formula:

$$C_{BST\_MIN} > \frac{Q_{GATE}}{\Delta V} = \frac{Q_{GATE}}{0.5V} = 2 \times Q_{GATE}$$

For example for Q_{GATE}=50nC the formula provides C_{BST_MIN}=100nF capacitor. This value has to be considered as the minimum tolerated value. Considering process spread, voltage derating and ER and ESL drops, larger capacitors are normally used. As a rule of thumb, a factor 10 can be used.

$$C_{BST} \sim 10 \times Q_{GATE}$$

CBST must be placed as close to the device's BS_ and BM_ pins. Choose low ESR ceramic capacitors with voltage rating >16V or 25V.

BUCK Recommended Inductor and Capacitor

RECOMMENDED INDUCTOR	RECOMMENDED CAPACITOR
L_{BUCK} = 27µH	12V rated
DCRLBUCK< 300mΩ	C_{VBUCK} = 8µF (effective value after derating at 5.8V)
I_{SAT}>1.5A	low ESR ceramic capacitor

Charge Pump Recommended External Capacitors

CFLY: (Connected between CPI and CPO)

- 0603 recommended to minimize the charge pump output impedance
- 12V rated low-voltage coefficient
- 220nF effective value after derating at 5.8V
- Smaller caps can possibly be considered depending on the gate driver current requirements.

CDRIVE: (Connected between V_{DRV} and GND)

- At least 16V rated
- 4.7µF recommended (effective value after derating at 12V)

Shunt Resistor Signal Routing

The TMC9660 requires external low-side shunt resistors for motor current feedback. CSP_ is also the return path for the gate drive current. Therefore, the impedance of CSP must be minimized. In particular, external RC filters in which resistors are placed bewteen the source of the external LS FET and the CSP pin must be avoided. Keep the traces short and low impedance.

Connect CSN_ to RSHUNT ground pin to allow accurate current sensing.

Gate Driver Unit Gate-Signal Routing

CSP_ is the return current path for LS_ as BM_ is for HS_. The loop area for those signals should be minimized in a parallel and compact layout. Keep the gate-signal traces short and low impedance.

In case of switching off the MOSFET, the sink current will take this route and the voltage drop must be low. This is especially important for MOSFETs with high gate charge (max. driver strength) and low Vth for example in power tools applications.

Gate Driver Unit Protection Signal Routing

The Gate Driver Unit provides different protection features.

MOSFET overcurrent protections are based on voltage measurements across the MOSFETs. On the high-side MOSFET the voltage VS to BM_ is measured, and on the low-side BM_ to CSP_. Keep the traces short and low impedance for a low voltage drop.

For gate-short and overcurrent protection the connection between CSP_ and the low-side MOSFET Source pin must be low-impedance.

Typical Application Circuits

The TMC9660 supports a variety of applications due to its flexible IOs and support for multiple motor types. This section provides an overview about the basic applications based on the motor type and feedback.

Depending on the used IOs and the Application mode, multiple features can be combined, and the examples extended or even the motor types switched.

BLDC/PMSM with Hall Sensor

A minimal BLDC/PMSM Motor application with a hall-sensor is shown in *Figure 41*.

The motor phases UVW are connected to the corresponding mid-points of the half-bridges UX1, VX2, and WY1 with their shunt resistor. A hall-sensor—required for the closed-loop FOC regulation—is powered by an internal LDO and connected via the alternate functions of the GPIOs directly to the Motion Control Core. Both LDO-voltage and alternate function need initial configuration through the bootloader.

If V_{CC_IO} matches the supply-voltage of the hall-sensor a direct connection is possible. Use a level shifter otherwise. It is recommended to add a RC-Lowpass if long sensor wires are used to reduce induced noise.

A hall-sensor is not recommended for positioning tasks due to its limited resolution.

Used features:

- Integrated DC/DC buck and LDO regulator delivers supply voltage for the hall-sensor (e.g., +3.3V or +5V)
- Digital hall (e.g., +3.3V or +5V) sensor supported directly (with optional integrated extrapolation)
- Configurable voltage for two LDOs
- Configurable alternate functions for GPIOs
- 3x half-bridge gate driver (up to 1A source/2A sink, adjustable slew-rate control)
- 3x bottom shunt current measurement (programmable CSA and ADCs)

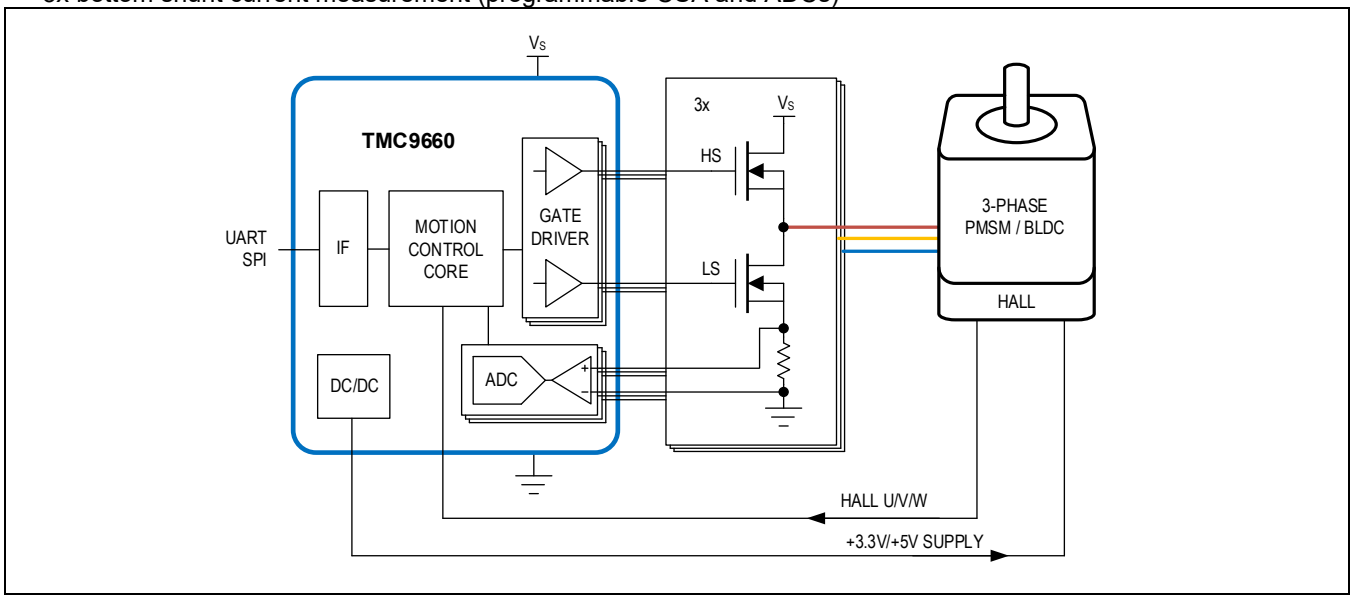

Figure 41. BLDC/PMSM with Hall Sensors Application Diagram

BLDC/PMSM with Encoder

A minimal BLDC/PMSM Motor application is shown in *Figure 42*.

The motor phases UVW are connected to the corresponding mid-points of the half-bridges UX1, VX2, and WY1 with their shunt resistor. A Hall- and ABN-Encoder—required for the closed-loop FOC regulation—are powered by an internal LDO and connected through the alternate functions of the GPIOs directly to the MCC. The Hall-Sensor is optional but can

improve the offset calibration of the ABN-Encoder. Both LDO-voltage and alternate function need initial configuration through the bootloader.

If V_{CC_IO} matches the supply-voltage of the Hall/ABN Sensor a direct connection is possible. Use a level shifter otherwise. It is recommended to add a RC-Lowpass if long sensor wires are used to reduce induced noise.

An ABN Sensor is recommended for positioning tasks due to its increased resolution.

Used features:

- Integrated DC/DC buck and LDO regulator delivers supply voltage for the hall-/ABN-sensor (e.g., +3.3V or +5V)
- Digital ABN (e.g., +3.3V or +5V) sensor supported directly
- Optional additional Digital Hall (e.g., +3.3V or +5V) sensor supported directly (with optional integrated extrapolation)
- Configurable voltage for two LDOs
- Configurable alternate functions for GPIOs
- 3x half-bridge gate driver (up-to 1A source/2A sink, adjustable slew-rate control)
- 3x bottom shunt current measurement (programmable CSA and ADCs)

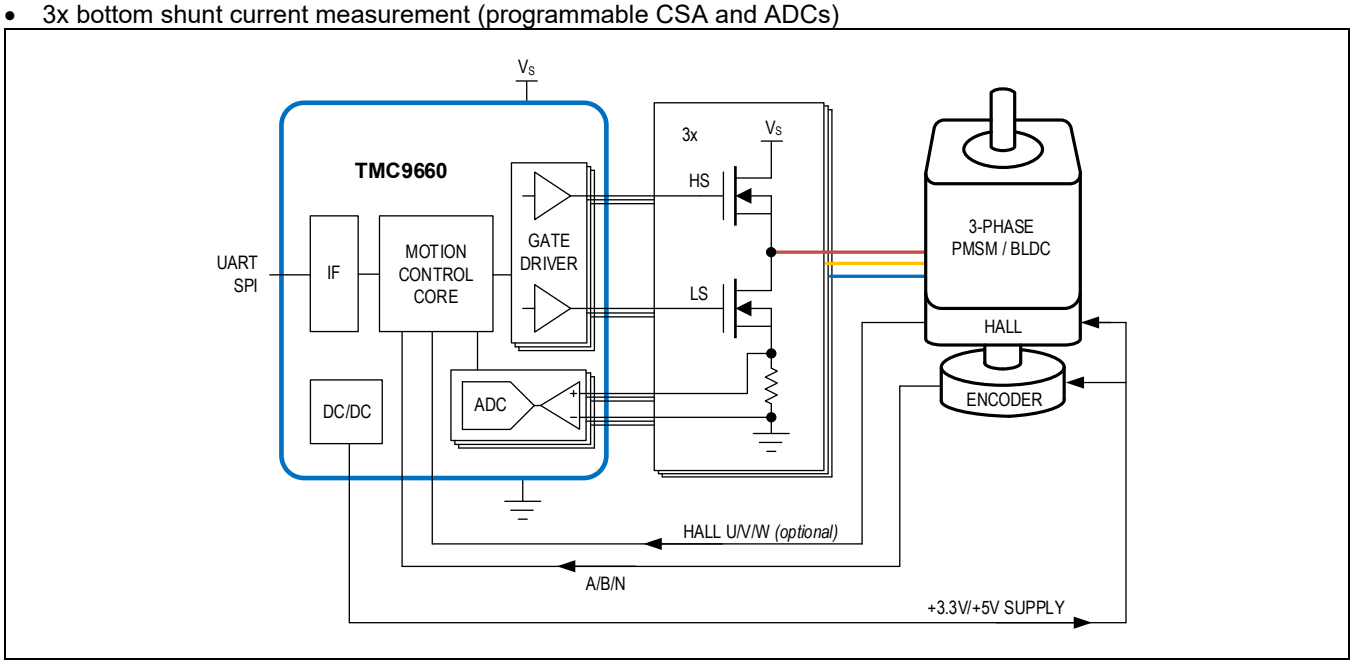

Figure 42. BLDC/PMSM with Encoder Application Diagram

In BLDC/PMSM motor operation, only three of the four half-bridges are used (*Figure 43*), The integrated additional fourth half-bridge gate driver can be used for the following:

- Brake chopper with integrated supply voltage measurement
  - High-side used as low-side
  - Only natively supported in parameter mode after bootloader configuration
- Electromechanical brake with optional current measurement
  - Only natively supported in parameter mode after bootloader configuration

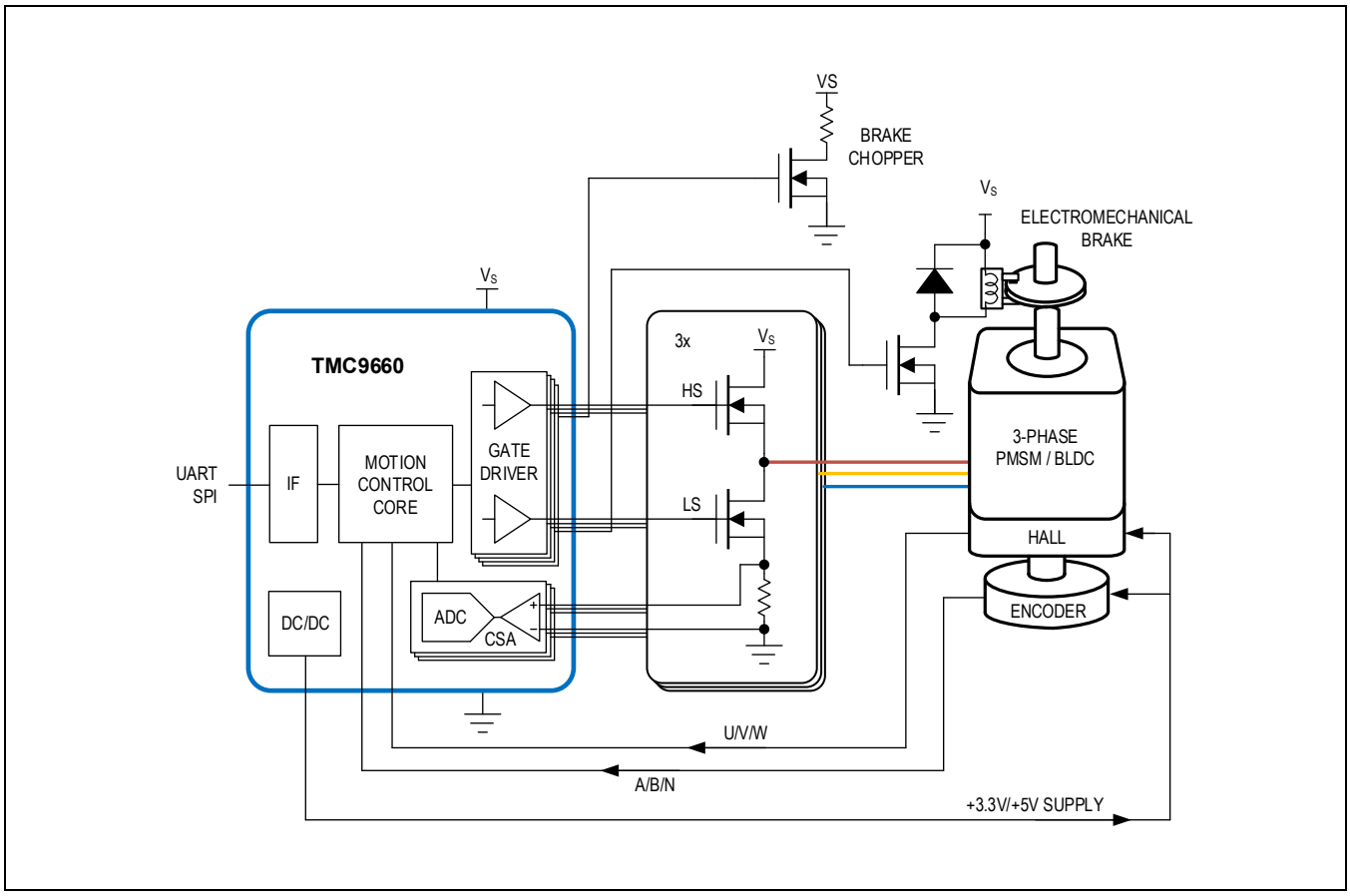

Figure 43. Encoder, Incremental A/B/N, Brake Chopper, and Electromechanical Brake

BLDC/PMSM with Encoder + 2nd Absolute Encoder

A minimal BLDC/PMSM motor application for positioning tasks based on an absolute encoder and limit switches is shown in *Figure 44*.

The motor phases UVW are connected to the corresponding mid-points of the half-bridges UX1, VX2, and WY1 with their shunt resistor. An ABN-Encoder—required for the closed-loop FOC regulation—is powered by an internal LDO and connected through the alternate functions of the GPIOs directly to the MCC. A second SPI absolute encoder is mounted after the gearbox to allow precise feedback of the position within the application. Both LDO-voltage and alternate function need initial configuration through the bootloader.

If V_{CC_IO} matches the supply-voltage of the ABN sensor a direct connection is possible. Use a level shifter otherwise. It is recommended to add a RC-lowpass if long sensor wires are used to reduce induced noise.

Used features:

- Integrated DC/DC buck and LDO regulator delivers supply voltage for the ABN sensor (e.g., +3.3V or +5V)
- Digital SPI absolute encoder (e.g., +3.3V or +5V) supported directly in parameter mode
- Digital ABN (e.g., +3.3V or +5V) sensor supported directly
- Stop/limit switches in hardware
- FOC position control in hardware based on either of the two sensors
- Configurable voltage for two LDOs
- Configurable alternate functions for GPIOs
- 3x half-bridge gate driver (up to 1A source/2A sink, adjustable slew-rate control)
- 3x bottom shunt current measurement (programmable CSA and ADCs)

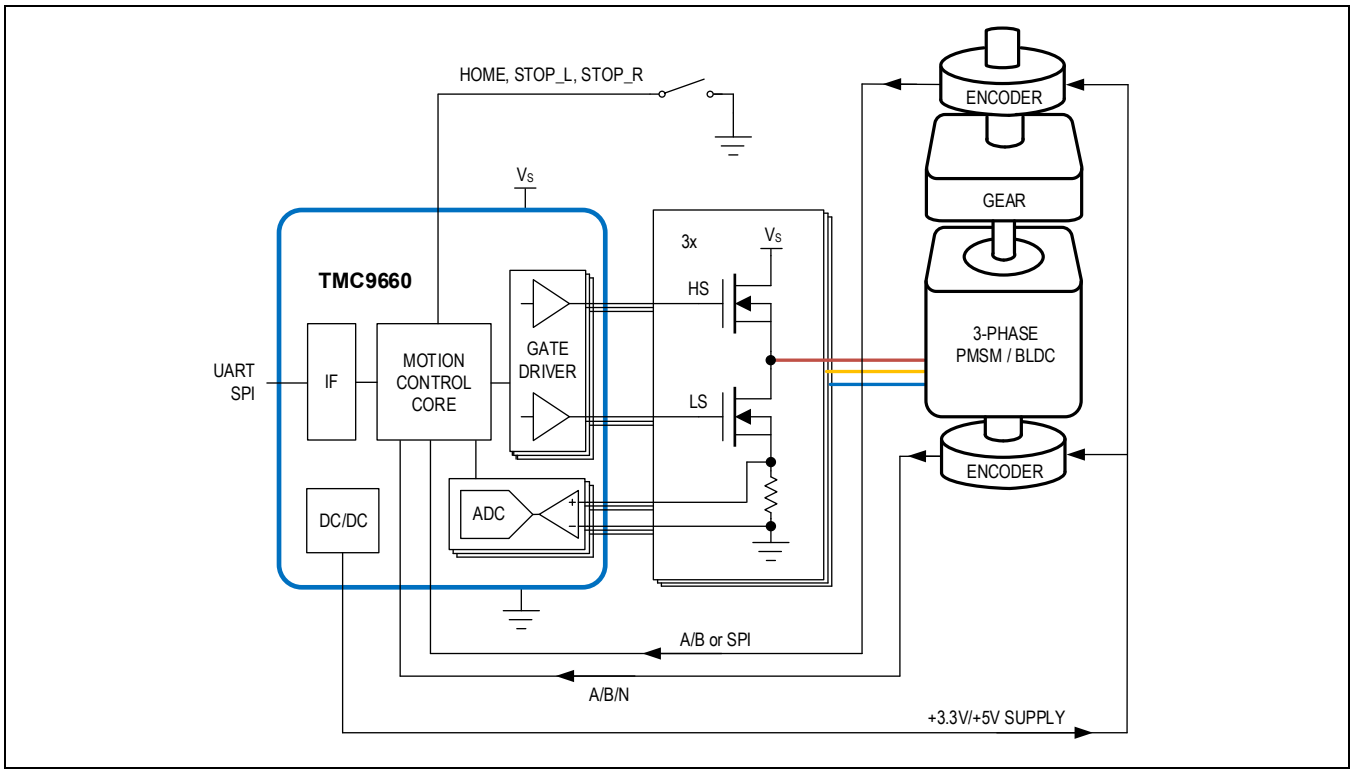

Figure 44. BLDC/PMSM with Encoder + Second Absolute Encoder on the Gear Application Diagram

STEPPER with Encoder

A minimal Stepper Motor application is shown in *Figure 45*.

The Motor coils X1, X2, Y1, and Y2 are connected to the corresponding mid-points of the half-bridges UX1, VX2, WY1, and Y2 with their shunt resistor. An ABN-Encoder—required for the closed-loop FOC regulation—is powered by an internal LDO and connected through the alternate functions of the GPIOs directly to the MCC. Both LDO-voltage and alternate function need initial configuration through the bootloader.

If V_{CC_IO} matches the supply-voltage of the ABN sensor a direct connection is possible. Use a level shifter otherwise. It is recommended to add a RC-Lowpass if long sensor wires are used to reduce induced noise.

An ABN sensor is recommended for positioning tasks due to its increased resolution.

Used features:

- Integrated DC/DC buck and LDO regulator delivers supply voltage for the ABN sensor (e.g., +3.3V or +5V)
- Digital ABN (e.g., +3.3V or +5V) sensor supported directly
- Configurable voltage for two LDOs
- Configurable alternate functions for GPIOs
- 4x half-bridge gate driver (up to 1A source/2A sink, adjustable slew-rate control)
- 4x bottom shunt current measurement (programmable CSA and ADCs)

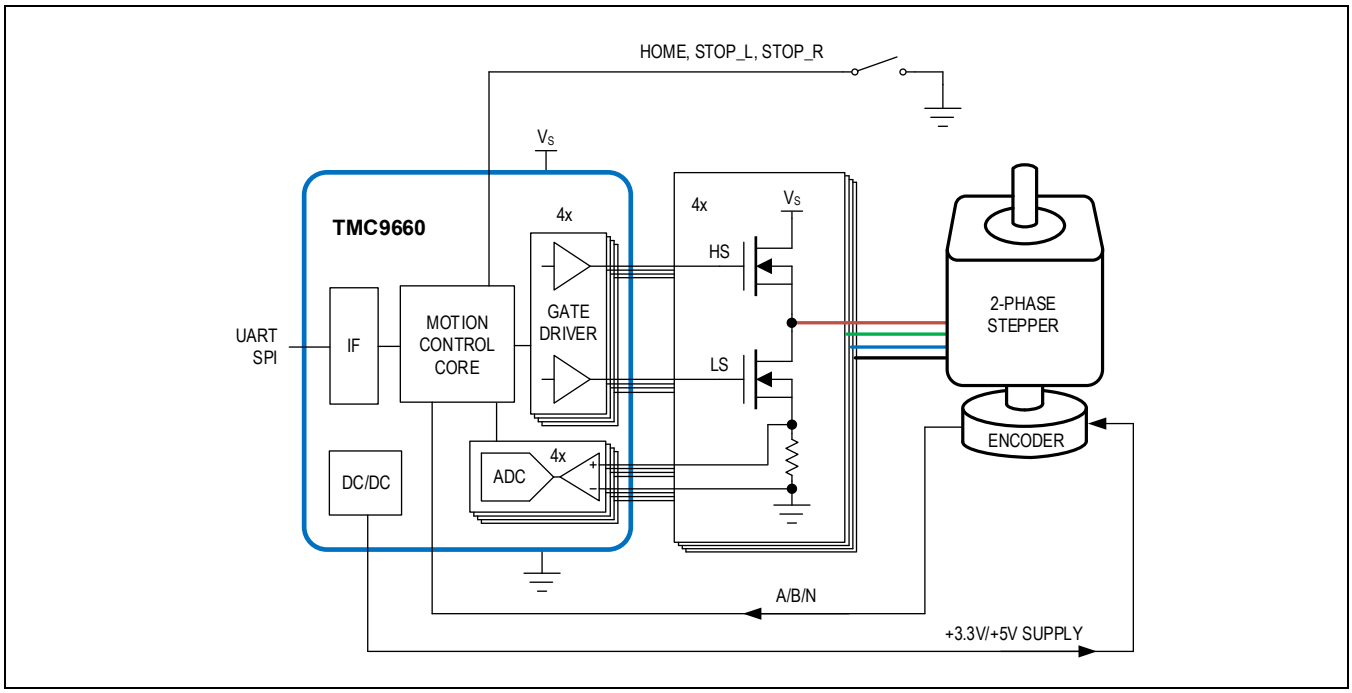

Figure 45. STEPPER with Encoder Application Diagram

Ordering Information

PART NUMBER	TEMPERATURE RANGE	PIN-PACKAGE	MOTOR TYPE	HALF-BRIDGE GATE DRIVERS	GATE DRIVE VOLTAGE (VDRV)	BUCK OUTPUT (VBUCK)	OPERATING VOLTAGE
TMC9660ATB+	-40°C to +125°C	64 TQFN 9mm x 9mm	BLDC/PMSM and STEPPERS	4	11.6	5.8	7.5V-70V
TMC9660ATB+T	-40°C to +125°C	64 TQFN 9mm x 9mm	BLDC/PMSM and STEPPERS	4	11.6	5.8	7.5V-70V

+ Denotes lead(Pb)-free/RoHS-compliance.
T Denotes tape-and-reel.

Errata

This section describes known chip issues, their restrictions, and their workarounds.

Erratum 1: Bootloader OTP_BURN Command

The OTP_BURN command has issues:

1. When the Motor system control was started, the V_{DRV} pin is charged up to 12V. Attempting to send an OTP burn request which then always fail to set up the correct voltage level on the VDRV pin quickly enough, causing the OTP burn procedure to abort with an internal 500ms timeout.

2. Any subsequent OTP_BURN commands after the first one will incorrectly always report a failure, irrespective of the actual result of the operation. It does not matter whether the first OTP_BURN command succeeded for failed. Note that a reset is not sufficient to avoid this, only a power cycle makes the first command report correctly.

To mitigate these issues, instead of just sending OTP_BURN, the following sequence of bootloader commands and extra steps must be performed when attempting an OTP burn:

- Send SET_BANK, value 0.
- Send SET_ADDRESS, value 0x4801B010.
- Send READ_32
- Clear bit 0 of the read value (0x00000001).
- Send WRITE_32, with the modified read value as write value.
- Wait for the VDRV voltage to drop below 8.4V. The duration for this depends on the attached capacitor. With a 10uF capacitor this takes 1.0s (typ).
- Send OTP_BURN.

To retrieve the status of the burn, either read back the OTP contents using OTP_LOAD or perform the following steps:

- Configure the clock settings to have the PLL active, with the SYS_CLK_DIV set to 3 (15MHz system clock).
- Send SET_BANK, value 0.
- Send SET_ADDRESS, value 0x48020014.
- Send READ_16.
- A read value of 0x80 or 0x84 indicates a successful burn operation. Any other value indicates a burn failure.
- Set SYS_CLK_DIV back to 0.

Erratum 2: SPI slave MISO operation

The TMC9660 SPI slave does not return the MISO line to high-Z after the SPI chip select signal to the TMC9660 is de-asserted. This is not an issue if the SPI connection is used exclusively for the TMC9660 with no other SPI slaves present.

Use one of the following workarounds if more SPI slaves are required:

1. Hold the TMC9660 in reset by asserting the asserting the RESETN pin during any SPI transactions to other SPI slaves.

2. Add an external component to the board design to ensure the MISO line is disconnected when the chip select signal is de-asserted. See example using an *ADG719* switch in *Figure 46*.

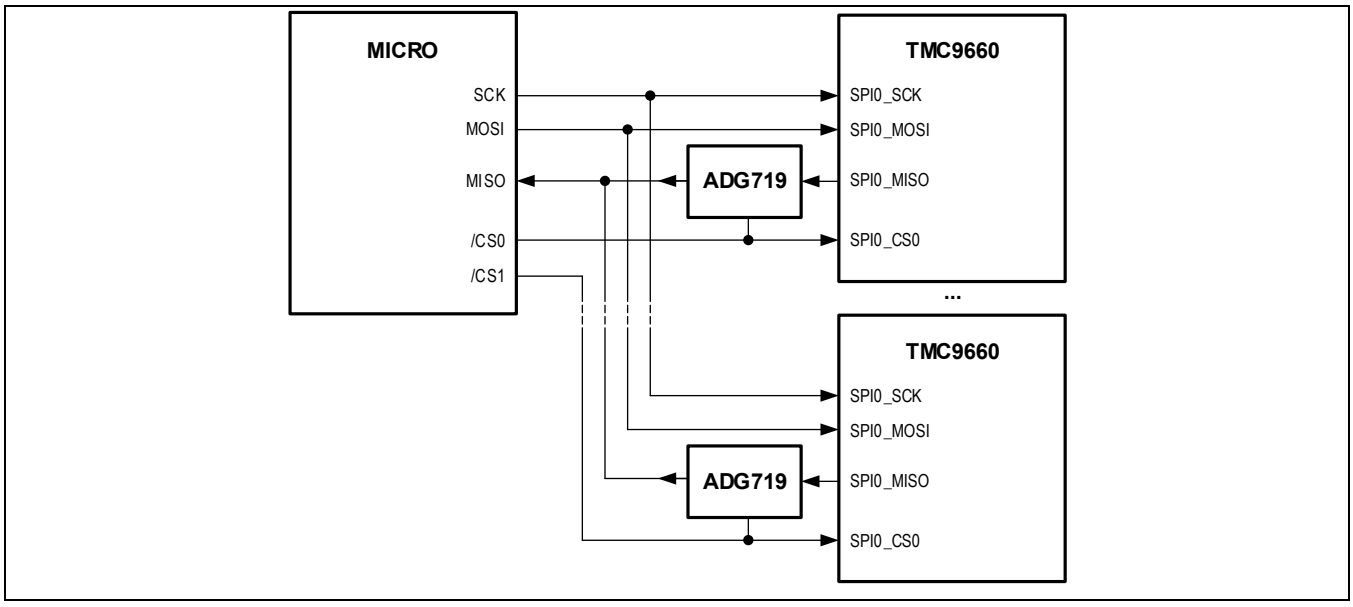

Figure 46. SPI Slave Operation, Workaround with External Component

Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	10/24	Release for Market Intro	—
1	11/24	Updated Table 24 and Ordering Information	79, 98