

12 V Sensorless Code-Free FOC BLDC Motor Controller

FEATURES AND BENEFITS

- Fully integrated single-shunt sensorless FOC controller (code-free)
- Power-loss brake functionality
- Universal speed curve
- Fault mode brake
- Analog/PWM/Clock mode speed control

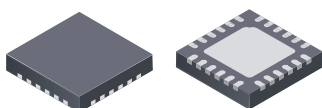
Continued on next page...

APPLICATIONS

- 12 V server fans
- Home appliance fans

PACKAGE

24-lead QFN with exposed thermal pad (suffix ES)



Not to scale

DESCRIPTION

The A89337 is a three-phase, sensorless brushless DC (BLDC) motor controller.

A field-oriented control (FOC) control algorithm is fully integrated to achieve the best efficiency and acoustic noise performance. The device also optimizes the motor startup performance in the stationary, forward, and reverse-windmill conditions.

Programmable profiles and closed-loop controls are included for constant power and constant speed operations. This eliminates the requirement for a microprocessor-based system and minimizes programming requirements. The sleep state can be entered to conserve power if enabled.

A simple I²C interface is provided for setting motor-related parameters, control-related parameters, and startup profiles, which are stored in nonvolatile memory (NVM). The I²C interface can also be used to control the motor directly and to provide speed and status feedback. Password protection is available for stored NVM data.

The A89337 is available in a 4 mm × 4 mm 24-terminal quad-flat no-lead (QFN) package with exposed thermal pad (suffix ES).

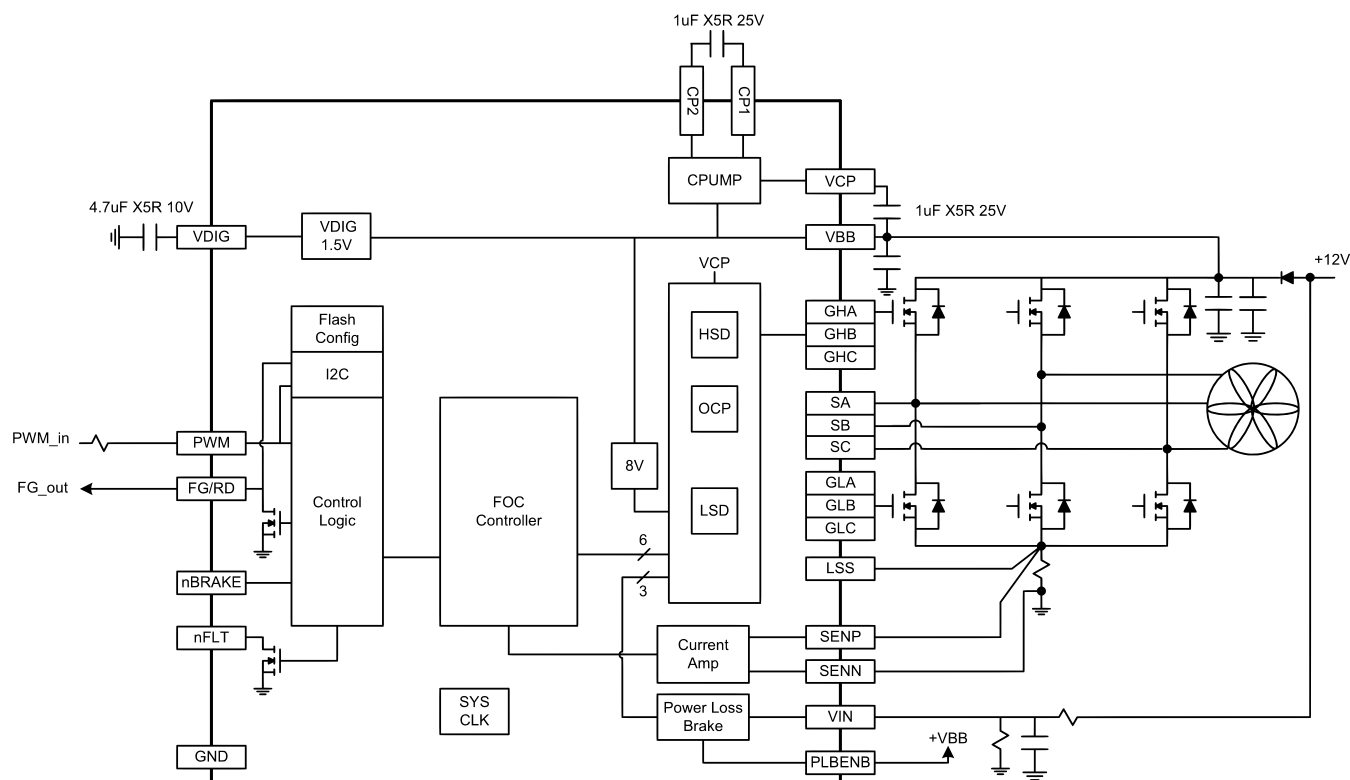


Figure 1: A89337 Block Diagram

FEATURES AND BENEFITS (cont.)

- Precision closed-loop speed control
- Configurable RD or FG output
- Slew-rate control for electromagnetic interference (EMI) mitigation
- Proprietary nonreverse fast startup
- Ultraquiet low-speed operation
- Current-control soft start
- Windmill startup operation
- I²C serial port
- Power limit protection
- Configurable current and power limit
- Lock detection
- Overcurrent limit (OCL)
- Overvoltage protection (OVP)
- Short-circuit protection (OCP)
- Logic input compatible with 3.3 V and 5 V

Table of Contents

Features and Benefits.....	1	Application Features.....	32
Description.....	1	Gate Drive.....	32
Applications.....	1	FG/RD.....	32
Package.....	1	AC Power Loss.....	40
Functional Block Diagram.....	1	Power Fan.....	40
Specifications.....	3	Forced OL.....	41
Selection Guide.....	3	Brake on Fault.....	41
Absolute Maximum Ratings.....	3	Quiet Start.....	41
Thermal Characteristics.....	3	BEMF Compensation.....	42
Pinout Diagram and Terminal List.....	4	Power-Loss Brake (PLB).....	42
Electrical Characteristics.....	5	No-Power Brake (NPB).....	43
Functional Description.....	9	Part Configuration.....	44
System Startup.....	9	Registers and Nonvolatile Memory (NVM).....	44
Numerical Representation.....	9	NVM Programming.....	46
System Limits.....	10	e-Flash Programming.....	47
Communication Interfaces.....	11	e-Flash CRC-32 Calculation.....	49
I ² C Timing Diagrams.....	13	Flash Operations.....	49
Universal Curve Controller.....	19	Flash Address Map.....	50
Control Modes.....	21	Package Outline Drawing.....	51
Motor-Control Application.....	22	Revision History.....	52
System Faults and Diagnostics.....	29		

A89337, A89337-1

12 V Sensorless Code-Free FOC BLDC Motor Controller

SELECTION GUIDE

Part Number	Description	Package	Packing
A89337GESSR	With power-fan function	24-terminal, 4 mm × 4 mm, QFN with exposed thermal pad	6,000 pieces per 13-inch reel
A89337GESSR-1	Without power-fan function	24-terminal, 4 mm × 4 mm, QFN with exposed thermal pad	6,000 pieces per 13-inch reel



ABSOLUTE MAXIMUM RATINGS [1][2]

Characteristic	Symbol	Notes	Min.	Max.	Units
Supply Voltage	V_{BB}	VBB	-0.3	30	V
Power-Loss Brake Terminals	V_I	VIN, PLBENB	-0.3	30	V
Logic Input Terminal	V_I	PWM/SPD, nBRAKE	-0.3	6	V
Logic Bidirectional Terminal	V_{IO}	FG/RD	-0.3	20	V
Logic Output Terminal	V_O	nFLT	-0.3	6	V
Output Terminal	V_O	VDIG	-0.3	2	V
Charge Pump	V_{CP}		$V_{BB} - 0.3$	$V_{BB} + 9$	V
	V_{CP1}		-0.3	$V_{BB} + 0.3$	V
	V_{CP2}		$V_{BB} - 0.3$	$V_{CP} + 0.3$	V
Motor Phase Terminals	V_{SX}	SA, SB, SC	-1	$V_{BB} + 2$	V
		$T_w < 200$ ns	-4	$V_{BB} + 4$	V
High-Side Gate Drive Output Terminals	V_{GHX}	GHA, GHB, GHC	$V_{SX} - 0.3$	$V_{CP} + 0.3$	V
Low-Side Gate Drive Output Terminals	V_{GLX}	GLA, GLB, GLC	$V_{LSS} - 0.3$	9	V
Bridge Low-Side Source Terminals	V_{LSS}	DC	-0.5	0.5	V
		$T_w < 500$ ns	-4	4	V
Current Sensing Input Terminals	V_I	SENN, SENP	-4	4	V
Maximum NVM Write Cycles	NVM_W		-	10000	cycles
Junction Temperature	T_j		-40	125	°C
Storage Temperature Range	T_s		-55	150	°C

[1] With respect to GND unless otherwise noted. Ratings apply when other circuit operating constraints are not present.

[2] Lowercase "x" in terminal names and symbols indicates a variable sequence character.

THERMAL CHARACTERISTICS: May require derating at maximum conditions

Characteristic	Symbol	Test Conditions [1]	Value	Units
Package Thermal Resistance (Junction to Ambient)	$R_{\theta JA}$	2-sided PCB, 1 in ² copper	45	°C/W
Package Thermal Resistance (Junction to Pad)	$R_{\theta JP}$		2	°C/W

[1] Additional thermal information is available on the Allegro website.

PINOUT DIAGRAM AND TERMINAL LIST

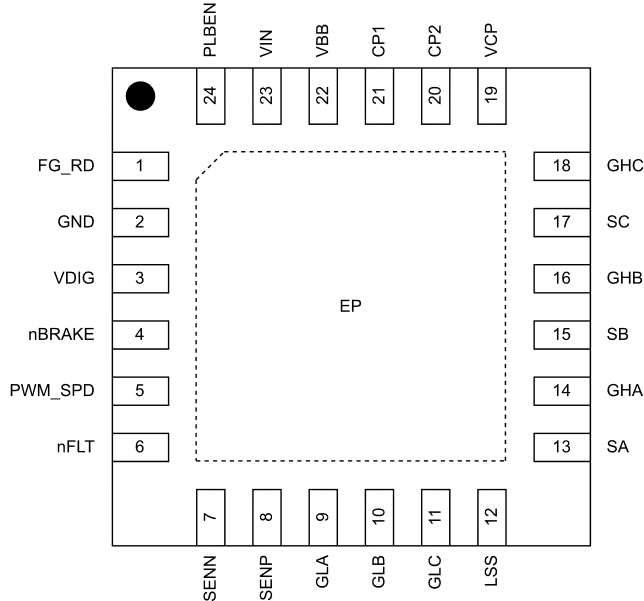


Figure 2: 24-Terminal QFN Package, Suffix ES

Table 1: Terminal List Table

Terminal		Description
Number	Name	
1	FG_RD	FG/RD is an open-drain output that can be pulled up to V_{BB} externally. It also has, by default, an internal pullup to 5 V that can be disabled via the <code>ipupFGRDdis</code> NVM parameter. Selection of the functions and the provision of either speed (FG) or rotation detection (RD) data is programmed via the NVM parameters. For more information, refer to the FG/RD section. The FG/RD terminal is also used as a serial data line (SDA) in the I ² C mode. The I ² C command can pass only when this terminal is high (open-drain).
2	GND	Device ground
3	VDIG	Voltage reference (1.5 V) to power internal digital logic. A low equivalent series resistance (ESR) ceramic capacitor with 4.7 μ F or greater is required on the terminal to stabilize the supply.
4	nBRAKE	Active low input signal, which turns on all low sides for the braking function. The brake function overrides the input control. During a thermal shutdown (TSD) event, or if ($V_{BB} > \text{ovpTh}$) or ($V_{BB} < V_{BBUVLO}$), the brake input is ignored. Care should be taken to avoid stressing the MOSFET when braking while the motor is running. When braking, the current is limited only by V_{bemf}/R_{MOTOR} . The A89337 includes an optional feature that holds off braking until the motor speed reduces to a level that is sufficiently low (configurable via the <code>brakeFreqTooHighTh</code> NVM parameter) that the braking current does not damage the MOSFET nBRAKE terminal. This optional feature also has the internal pullup to 5 V, which can be configured through <code>ipupNBRAKEdis</code> . nBRAKE can also be configured as an I ² C CLK input for use in the initial configuration of NVM parameters.

Terminal		Description
Number	Name	
5	PWM_SPD	Device control demand input. This terminal supports analog, PWM, or frequency command. It also has, by default, an internal pullup to 5 V that can be disabled via the <code>ipupPWMSPDdis</code> NVM parameter. In the I ² C mode, the PWM/SPD terminal is also used as the SCK.
6	nFLT	Open-drain output. This terminal provides the motor operation fault status. It has an internal pullup that can be configured via the NVM. When a fault is detected, the nFLT terminal transitions to low and remains active (low) until the fault is cleared according to the latched or not-latched mechanism. nFLT can be used to indicate a successful software startup after a power-on reset (POR) by remaining low for a short duration as follows: This feature can be enable/disable in config NVM (<code>nfltStartupDis</code>). nFLT can also be configured as I ² C SDA
7	SENN	Current sense negative terminal.
8	SENP	Current sense positive terminal.
9	GLA	Low-side A, gate-drive outputs for external N-channel MOSFETs.
10	GLB	Low-side B, gate-drive outputs for external N-channel MOSFETs.
11	GLC	Low-side C, gate-drive outputs for external N-channel MOSFETs.
12	LSS	Low-side return path for discharge of capacitance on the MOSFET gates, connected to the common sources of the low-side external MOSFETs through a low-impedance track.
13	SA	Motor phase A connection
14	GHA	High-side A, gate-drive outputs for external N-channel MOSFETs.
15	SB	Motor phase B connection
16	GHB	High-side B, gate-drive outputs for external N-channel MOSFETs.
17	SC	Motor phase B connection
18	GHC	High-side C, gate-drive outputs for external N-channel MOSFETs.
19	VCP	Charge pump output. A 1 μ F/25 V ceramic capacitor is required between VCP and VBB
20	CP2	Charge pump capacitor +
21	CP1	Charge pump capacitor –
22	VBB	DC power supply
23	VIN	PLB/NPB voltage input (monitoring)
24	PLBEN	PLB enable
EP		Exposed pad must be connected to ground

A89337, A89337-1

12 V Sensorless Code-Free FOC BLDC Motor Controller

ELECTRICAL CHARACTERISTICS: Valid at $T_J = 25^\circ\text{C}$, $V_{BB} = 12\text{ V}$ (unless noted otherwise) [1]

Characteristics	Symbol	Test Conditions	Min.	Typ.	Max.	Units
Power Supply Voltage Range	V_{BB}		8	12	24	V
VBB Supply Current	I_{BB1}	Standby, $V_{IN} > V_{INTH}$, $V_{BB} > V_{BBUVLO}$	–	8	13	mA
	I_{BB2}	$V_{IN} < V_{INTH}$	–	0.5	1	mA
	I_{BB3}	Sleep state	–	–	125	μA
VCP Regulator Voltage	V_{VCP}	Relative to V_{BB} , $I_{VCP} = 10\text{ mA}$, $V_{BB} = 12\text{ V}$	7	7.5	8.5	V
VDIG Regulator Voltage	V_{DIG}	$I_{EXT_LOAD} = 0\text{ mA to } 5\text{ mA}$	1.43	1.53	1.63	V
VDIG Current Limit	V_{DIGOCL}	$V_{DIG} = 0\text{ V}$, internal current limit	25	40	85	mA
GATE DRIVER						
High-Side Gate Drive Output	V_{GH}	$V_{BB} = 12\text{ V}$	7	7.5	8.5	V
Low-Side Gate Drive Output	V_{GL}	$V_{BB} = 12\text{ V}$	7	7.5	8	V
Gate Drive Source Current	I_{SRC}	pwmGDSlewRate = 0, $V_{GS} = 4\text{ V}$	–	12.5	–	mA
		pwmGDSlewRate = 1, $V_{GS} = 4\text{ V}$	–	25	–	mA
		pwmGDSlewRate = 2, $V_{GS} = 4\text{ V}$	–	50	–	mA
		pwmGDSlewRate = 3, $V_{GS} = 4\text{ V}$	–	75	–	mA
Gate Drive Sink Current	I_{SNK}	pwmGDSlewRate = 0, $V_{GS} = 4\text{ V}$	–	25	–	mA
		pwmGDSlewRate = 1, $V_{GS} = 4\text{ V}$	–	50	–	mA
		pwmGDSlewRate = 2, $V_{GS} = 4\text{ V}$	–	100	–	mA
		pwmGDSlewRate = 3, $V_{GS} = 4\text{ V}$	–	150	–	mA
GHx Passive Pull-Down	R_{GHPD}	$V_{BB} = 0$, $V_{GHx} - V_{SX} < 0.1\text{ V}$	0.5	–	1.5	$\text{M}\Omega$
GLx Passive Pull-Down	R_{GLPD}	$V_{BB} = 0$, $V_{GLx} - V_{LSS} < 0.1\text{ V}$	0.5	–	1.5	$\text{M}\Omega$

[1] The specified limits are tested at a single temperature and assured over operating temperature range by design and characterization.

A89337, A89337-1

12 V Sensorless Code-Free FOC BLDC Motor Controller

ELECTRICAL CHARACTERISTICS: Valid at $T_J = 25^\circ\text{C}$, $V_{BB} = 12\text{ V}$ (unless noted otherwise) [1]

Characteristics	Symbol	Test Conditions	Min.	Typ.	Max.	Units
PROTECTION CIRCUITS						
Lock Protection	t _{OFF}	Factory default or configured via rotStartRetryTOut	10	–	10	%
Overcurrent Limit	I _{OCL}	Factory default or configured via sfocDrvCurrLmt	10	–	10	%
Thermal Shutdown Temperature	T _{JTSD}	Relative to target value	–10	–	10	°C
VDIG Undervoltage Lockout	V _{DIGUVLO}	V _{DIG} falling	1.4	1.44	1.48	V
VDIG Undervoltage Lockout Hysteresis	V _{DIGHYS}		–	32	–	mV
VCP Undervoltage Lockout	V _{VCPUVLO}	V _{CP} falling	5	–	–	V
VCP Undervoltage Lockout Hysteresis	V _{VCPUVLOHYS}		–	0.5	–	V
VBB Overvoltage Accuracy	V _{BBOVTH}	Factory default or configured via ovpTh	–5	–	5	%
Hysteresis	V _{BBOVHYS}		–	2	–	V
VBB Undervoltage Lockout	V _{BBUVLO}	V _{BB} rising	7.25	7.6	7.95	V
VBB Undervoltage Lockout Hysteresis	V _{BBUVLOHYS}		–	1	–	V
VDS Overvoltage Accuracy	V _{DSTH}	V _{DSTH} = 0.53 V (OCPTH = 0), –40, –105°C, characterization only	–48	–	48	%
		V _{DSTH} = 0.8 V (OCPTH = 1), characterization only	–33	–	33	%
		V _{DSTH} = 1.09 V (OCPTH = 2), characterization only	–25	–	25	%
		V _{DSTH} = 1.61 V (OCPTH = 3), characterization only	–18	–	18	%
POWER-LOSS BRAKE						
VIN Logic Threshold	V _{INTH}	V _{IN} falling	–	2.5	–	V
Hysteresis	V _{INHYS}		–	300	–	mV
VIN Pull-Down Resistor	V _{INPD}		–	300	–	KΩ
VBB Regulated Boost Voltage	V _{BOOST}	V _{IN} < V _{INTH}	5.5	6.25	7	V
VBB Boost Low Threshold	V _{BBTH}		–	1	–	V
Boost Switching Frequency	f _{BOOST}		–	40	–	kHz

[1] The specified limits are tested at a single temperature and assured over operating temperature range by design and characterization.

A89337, A89337-1

12 V Sensorless Code-Free FOC BLDC Motor Controller

ELECTRICAL CHARACTERISTICS: Valid at $T_J = 25^\circ\text{C}$, $V_{BB} = 12\text{ V}$ (unless noted otherwise) [1]

Characteristics	Symbol	Test Conditions	Min.	Typ.	Max.	Units
LOGIC INPUTS AND OUTPUTS						
Logic Input Low Level	V_{IL}		–	–	0.8	V
Logic Input High Level	V_{IH}		2	–	–	V
Logic Input Hysteresis	V_{HYS}		200	300	600	mv
Logic Input Current	I_{IN}	SPD, nBRAKE	–10	<1	10	μA
Output Saturation Voltage (FG/RD, nFLT)	V_{SAT}	$I = 5\text{ mA}$	–	–	0.3	V
Output Leakage (FG/RD)	I_O	$V = 16\text{ V}$, switch OFF	–	–	5	μA
Output Leakage (nFLT)	I_O	$V = 5\text{ V}$, Switch OFF	–	–	5	μA
INTERNAL OPEN-DRAIN PULL-UP						
Internal Pull-Up Voltage			–	5	–	V
Internal Pull-Up Resistor			–	100	–	$\text{k}\Omega$
I²C TIMING						
SCL Clock Frequency	f_{CLK}		10	–	400	kHz
Bus Free Time Between Stop/Start	t_{BUF}		1.3	–	–	μs
Hold Time Start Condition	$t_{(HD,STA)}$		0.6	–	–	μs
Setup Time for Start Condition	$t_{(SU,STA)}$		0.6	–	–	μs
SCL Low Time	t_{LOW}		1.3	–	–	μs
SCL High Time	t_{HIGH}		0.6	–	–	μs
Data Setup Time	$t_{(SU,DAT)}$		100	–	–	ns
Data Hold Time	$t_{(HD,DAT)}$		0	–	900	ns
Setup Time for Stop Condition	$t_{(SU,STO)}$		0.6	–	–	ms

[1] The specified limits are tested at a single temperature and assured over operating temperature range by design and characterization.

A89337, A89337-1

12 V Sensorless Code-Free FOC BLDC Motor Controller

ELECTRICAL CHARACTERISTICS: Valid at $T_J = 25^\circ\text{C}$, $V_{BB} = 12\text{ V}$ (unless noted otherwise) [1]

Characteristics	Symbol	Test Conditions	Min.	Typ.	Max.	Units
SPEED CONTROL						
PWM Duty Input, for PWM Duty Control Mode	f_{PWM}		0.1	–	100	kHz
Duty Cycle On Threshold	DC_{ON}	Relative to programmed target	–0.5	–	0.5	%
Duty Cycle Off Threshold	DC_{OFF}	Relative to programmed target	–0.5	–	0.5	%
Analog SPD Full Range Voltage, for Analog External Command Source (Analog Control Mode)	$\text{AnaSpd}_{\text{Max}}$		–	2.4	–	V
Analog SPD Resolution			–	0.6	–	mV
Input Clock Frequency (Clock Mode)	f_{FREQIN}	Factory default or configured via clkCompRef	–	1	–	kHz
Speed Error (Not Clock Mode)	F_{SPD}	$T_J = 25^\circ\text{C}$	–1.2	–	1.2	%
		$T_J = -40^\circ\text{C}$ to 105°C (characterization only)	–4	–	4	%
Speed Setpoint (Clock Mode)	F_{SPDCLK}		–0.5	–	0.5	%
System Oscillator	f_{OSC}		–	31.25	–	MHz
SLEEP STATE						
Analog SPD Sleep Enter Threshold	$\text{AnaStb}_{\text{Ent}}$	Factory default or configured via sbyLPSanaTh	–	0	–	mV
Analog SPD Sleep Exit Threshold	$\text{AnaStb}_{\text{Ext}}$		–	730	1050	mV
ID CLOCK TIMING						
Maximum ID Clock Time	t_{idmax}	Factory default or configured via FGIdDur	–	1	–	s
ID Clock Frequency	f_{idclk}	Factory default or configured via FGIdFreq	–	300	–	Hz
VBB UVLO to ID Clock Delay	t_{dly1}		–	–	100	μs
PWM Rising Edge to End of ID Clock	t_{dly2}		–	–	100	μs
NONVOLATILE MEMORY (NVM) PROGRAMMING						
V_{BB} for Programming	$V_{(\text{NVM}, \text{Prog})}$	V_{BB}	8	–	–	V

[1] The specified limits are tested at a single temperature and assured over operating temperature range by design and characterization.

FUNCTIONAL DESCRIPTION

System Startup

System power startup after a power-on reset (POR) is illustrated in Figure 3.

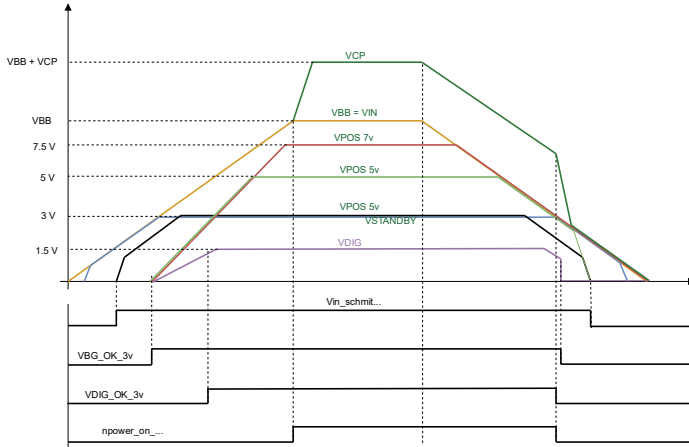


Figure 3: System Power Startup

Numerical Representation

PHASE RESISTANCE

The phase resistance is calculated according to:

Equation 1:

$$\text{sfocRs} = \text{round} \left[\frac{\text{motorRs} \times i_{\max} \times 2^{(\text{sfocRsFl}+11)}}{0.298 \times V_{bb\text{nom}}} \right]$$

where:

- sfocRs [unitless] is the stator resistance NVM setting.
- $I_{\max}$ [A] is the maximum system current.
- motorRs [Ω] is the motor phase to neutral resistance (phase to phase/2).
- sfocRsFl [unitless] is the SFOC stator inductance fractional length NVM setting (signed).
- $V_{BB\text{nom}}$ [V] = 12 V is the nominal DC link.

PHASE INDUCTANCE

For the sensorless mode of operation, the motor angle and speed data are estimated in the observer block, without need of an external sensor.

The machine parameters required in the A89337 observer are the motor phase inductance sfocLs[unitless] and the motor phase resistance sfocRs[unitless] according to:

Equation 2:

$$\text{sfocLs} = \text{round} \left[\frac{2\pi \times i_{\max} \times f_{\text{res}} \times \text{motorLs} \times 2^{(\text{sfocLsFl}+21)}}{0.298 \times V_{bb\text{nom}}} \right]$$

BEMF CONSTANT

The motor bemf constant (sfocKe) is calculated according to:

Equation 3:

$$\text{sfocKe} = \text{round} \left[\frac{2\pi \times f_{\text{res}} \times \text{motorKe} \times 2^{(14+(8+\text{sfocKeFl}))}}{0.298 \times V_{BB\text{nom}}} \right]$$

where:

- sfocKe [unitless] is the motor back-emf NVM setting.
- $I_{\max}$ [A] is the maximum system current.
- f_{res} [Hz/LSB] is the frequency resolution.
- motorKe [V/(rad/s)] is the motor back-emf constant.
- sfocKeFl [unitless] is the bemf-constant fractional-length NVM setting (signed).
- $V_{BB\text{nom}}$ [V] = 12 V is the nominal DC link voltage.

MOTOR ROTATION DIRECTION

Direction data is decoupled from $\text{mca}_{\text{command}}$. The FOC algorithm has two inputs: direction and command of the motor, which depend on:

- The direction bit ($\text{write_ref_cmd}[16]$) in the WRITE_REF_CMD register.
- The msb of $\text{mca}_{\text{command}}$, $\text{mca}_{\text{command}}[16]$: If the y values of the UCC are all positive, $\text{mca}_{\text{command}}[16] = 0$; otherwise, $\text{mca}_{\text{command}}[16] = 1$.
- The rotDirInvert configurable NVM parameter.
- The operation control mode (i.e., speed, power, torque mode, selected according to sfocVarCtrlMode).

The motor direction is defined as:

Equation 4 (torque or power control mode):

$$\text{Motor direction} = \text{mca}_{\text{command}}[16] \text{ XOR } \text{rotDirInvert} \text{ XOR } \text{ref_cmd}[16]$$

Equation 5 (speed control mode):

$$\text{Motor direction} = \text{rotDirInvert} \text{ XOR } \text{ref_cmd}[16]$$

It remains possible to configure UCC to output a negative command; however, when transferring the command to the algorithm, the MSB of $\text{mca}_{\text{command}}$ is ignored.

System Limits

MAXIMUM SYSTEM SPEED

The maximum system speed ($f_{\max}$ [Hz]) is the maximum electrical motor speed that can be commanded. It depends on frequency resolution, f_{res} , according to:

Equation 6:

$$f_{\max} = \text{round}((2^{15} - 1) \times f_{\text{res}})$$

where f_{res} [Hz/LSB] is the frequency resolution, and it is set via the `sfocFreqRes` register according to:

Equation 7:

$$f_{\text{res}} = 1/(8 \times 2^{\text{sfocFreqRes}}) \text{ ([Hz])/([LSB])}$$

NOTE: The maximum rotation speed can be limited by supply voltage (V_{BB}). Therefore, a high command speed might not result in high-speed motor rotation due to low V_{BB} voltage.

SPEED LIMIT

The NVM parameter `sfocFreqLmt` limits the maximum rotation speed. It works for all speed, torque, and power modes, and it is typically used in power and torque modes.

MAXIMUM SYSTEM CURRENT

The maximum system current is typically set to 110% of the motor rated current. A single shunt-resistor connecting `SEN_N` and `SEN_P` is used to measure the phase current for the FOC algorithm and current limit. The shunt resistor value needs to be selected based on the maximum required rated current of the system as:

Equation 8:

$$I_{\max} [A] = \text{ADC}_{\text{VREF}} / (R_{\text{shunt}} \times \text{CSA}_{\text{GAIN}})$$

where ADC_{VREF} is the reference ADC clipping voltage equal to 0.9464 V; R_{shunt} [Ω] is the board shunt resistance; and CSA_{GAIN} is the sense amplifier gain, which is equal to 8.

Use of a Kelvin sensing connection is recommended for the shunt resistor, as shown in Figure 4.

Table 2: Shunt Resistance Examples

R_{shunt} [m Ω]	$I_{\max}$ [A]
20	5.92
15	7.89
10	11.83

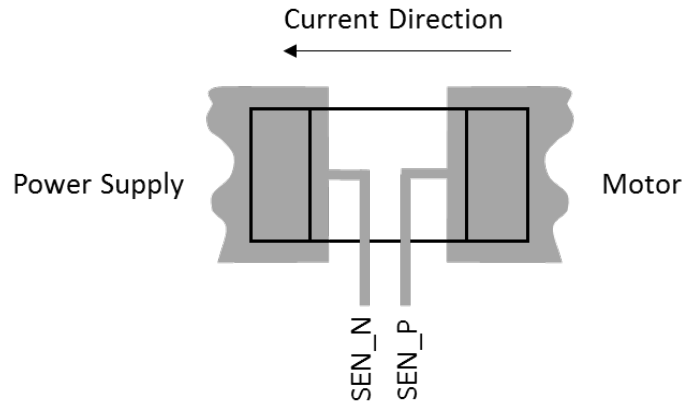


Figure 4: Shunt Resistor Layout

CURRENT LIMITS

A89337 can limit the maximum Q-axis current setting of the `sfocDrvCurrDrvLmt` NVM parameter as a percentage of the maximum current ($I_{\max}$) that the system can handle. The Q-axis current limit is also designed to help reduce the risk of demagnetization of permanent magnets. Reaching this limit is not considered a fault.

MAXIMUM ELECTRICAL POWER

The maximum electrical power value (max electrical power [W]) is calculated according to:

Equation 9:

$$P_{\max} [W] = \frac{3}{2} \times i_{\max} [A] \times \frac{\sqrt{3}}{3} \times V_{\text{BB}_{\text{nom}}} [V] = \frac{\sqrt{3}}{2} \times i_{\max} [A] \times V_{\text{BB}_{\text{nom}}} [V]$$

For example, when the maximum system current is $I_{\max} = 6.25$ A and $V_{\text{BB}_{\text{nom}}} = 12$ V, the maximum electrical power is 64.95 W.

POWER LIMIT

The A89337 monitors V_{BB} and I_{BB} for events that exceed the maximum power limit, and regulates power at approximately the NVM programmed maximum power limit (`sfocPwrLmt`). The maximum power limit is applicable to all speed, torque, and power control modes. This limit is set as a percentage of the maximum power ($P_{\max}$) that the system can manage.

A89337,
A89337-1

12 V Sensorless Code-Free FOC
BLDC Motor Controller

VBB OVERVOLTAGE VARIABLE LIMITER

If enabled (ovvlEnable), when voltage increases to greater than ovvlVoltTh, the A89337 enters the overvoltage variable limit (OVVL) condition; and, when voltage reduces to less than ovvl-VoltTh-18.75%, the A89337 exits the OVVL condition

If the voltage increases beyond ovvlVoltTh, the command is over-written by the ovvlVarLmt value. This prevents a further increase in voltage and does not trigger a fault.

The ovvlVoltTh parameter should be set to be less than the ovpTh parameter. If this is not the case, it is not possible to enter the OVVL condition (the OV fault triggers first).

Overvoltage detection is not considered to be a fault, and it is prioritized over the maximum power limit function.

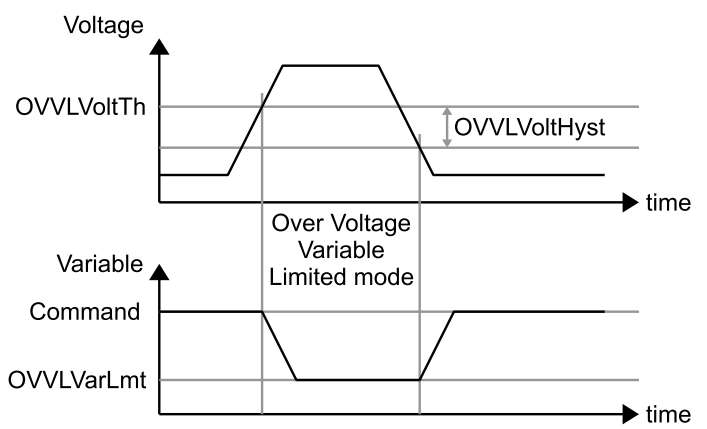


Figure 5: OVVL Timing Diagram

Communication Interfaces

The A89337 can be controlled with four different sources of command, according to the extCmdSrc NVM parameter setting.

Table 3: Communication Interfaces

extCmdSrc	Communication Interface
0x0	I2C
0x1	Analog
0x2	PWM Duty Cycle
0x3	PWM Frequency

I2C

The A89337 uses a standard fast-mode I2C serial port format to program the NVM or to control the I2C operation serially. Via the graphical user interface (GUI), the user can adjust configuration register values for a particular application before programing these parameters into e-flash. The A89337 7-bit slave address is 0x55. The I2C interface terminals are configurable by NVM parameters as described next.

Table 4: I2C SCL Source

i2cSCLsource	SCL Terminal
0x0	PWM/SPD
0x1	nBRAKE

Table 5: I2C SDA Source

i2cSDAsource	SDA Terminal
0x0	FG/RD
0x1	nFLT

If the motor is running, the FG (i2cSDAsource = 0x0) might pull the data line low while trying to initialize into serial port mode.

Because RD is typically low (NVM parameter RDActiveLevel = 0x0) when no faults are present, to establish an I²C communication when the FG/RD option is set to RD, V_{BB} must be brought to less than V_{BBUVLO} to disable the RD pull-down.

The motor control-speed reference command is defined by writing in the WRITE_REF_CMD register through I²C protocol. In this mode, the motor electrical frequency can be read through this register, as well as many other internal parameters. The value selected in the WRITE_REF_CMD register is copied into the MCA_EXTCOMMAND register. The direction, when available, is defined in the NVM and UCC.

The A89337 can load a predefined command into the MCA_EXTCOMMAND register at power-up (i.e., POR) to start the motor without need of an external command. The default command can be set in the defaultExtCmd NVM and, at power-up, its value is copied into the MCA_EXTCOMMAND register. To disable this feature, defaultExtCmd must be set to 0. The command is treated as speed, power, or torque depending on the mode selected.

When FG is in the high state (not driven), I²C communication is always available. Also, if the device has been configured for analog, PWM or frequency external command source operation,

I²C communication is always available

I²C data are mapped on the FG terminal and, because the terminal is an open-drain type, it can be controlled by the I²C when in the high state. The internal NVM register, shadow NVM, and status register can therefore be read and set during normal operation. However, for reliable communication, use of I²C is recommended when the A89337 is in the standby state (motor stopped).

NOTE:

- The internal NVM register and shadow NVM are inaccessible in application operation mode. They can be accessed only in user operation mode. For more information, refer to the Operation Modes and Customer Password Protection section.
- The I²C SCK signal on PWM or SPD is interpreted as the maximum command. This is because, during serial transactions, the terminal state is high; because the high terminal state is a valid command, it is interpreted as the maximum command. For reliable communication, it is better to use serial communication in undervoltage (UV) mode to ensure no issues with motor start and correct FG/RD terminal state.

For more information, refer to the VBB Undervoltage—VBB UVS section.

I²C Timing Diagrams

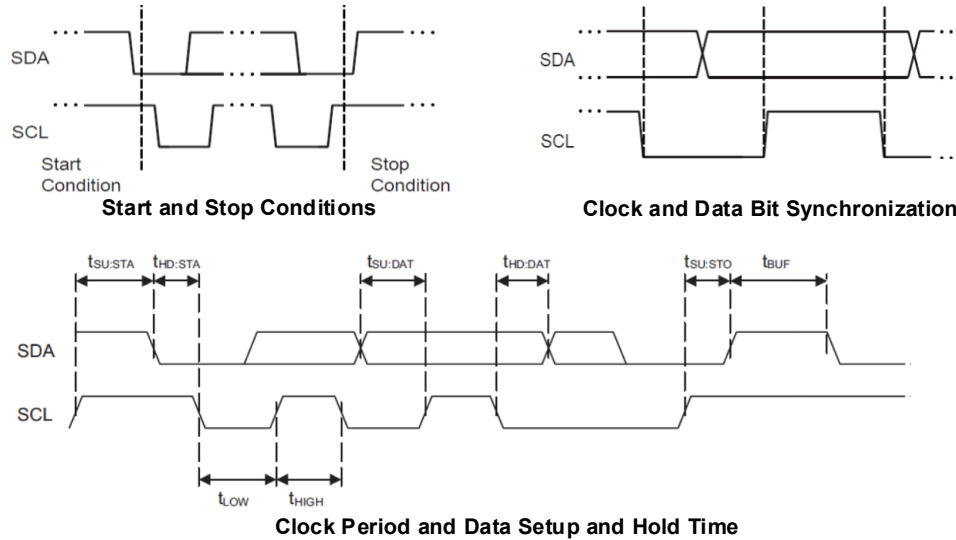


Figure 6: I²C-Compatible Timing Requirements

Write Procedure

1. Start condition
2. (1 byte) 7-bit I²C slave address (device ID) 1010101, R/W bit = 0
3. (1 byte) Fixed bit = 1, bits [6:0] of the internal register address
4. (1 byte) Fixed bit = 1, bits [13:7] of the internal register address
5. (4 bytes) Four data bytes, least significant byte first
6. Stop condition

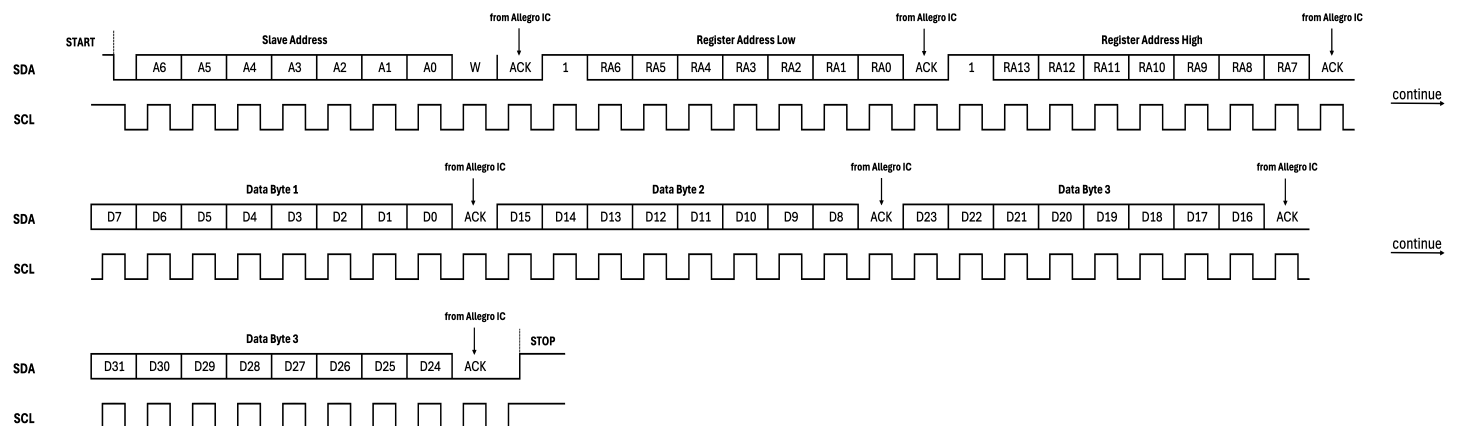


Figure 7: I²C Write Procedure

Read Procedure

The I²C read procedure can be configured in two response modes: in frame and out of frame.

Table 6: I²C Response Modes

i2cResponseMode	Response Mode
0x0	In frame
0x1	Out of frame

In-frame mode

1. Start condition
2. (1 byte) 7-bit I²C slave address (device ID) 1010101, R/W bit = 1
3. (1 byte) Fixed bit = 1, bits [6:0] of the internal register address
4. (1 byte) Fixed bit = 1, bits [13:7] of the internal register address
5. (4 bytes) read four data bytes, least significant byte first, where the last byte should be NACK by the master
6. Stop condition

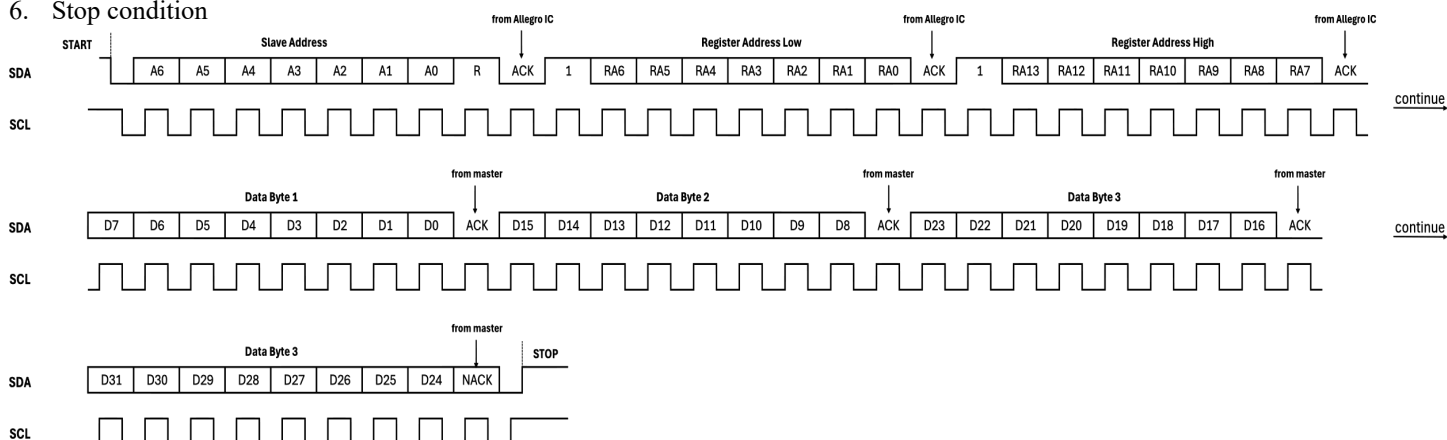


Figure 8: I²C In-Frame Read Procedure

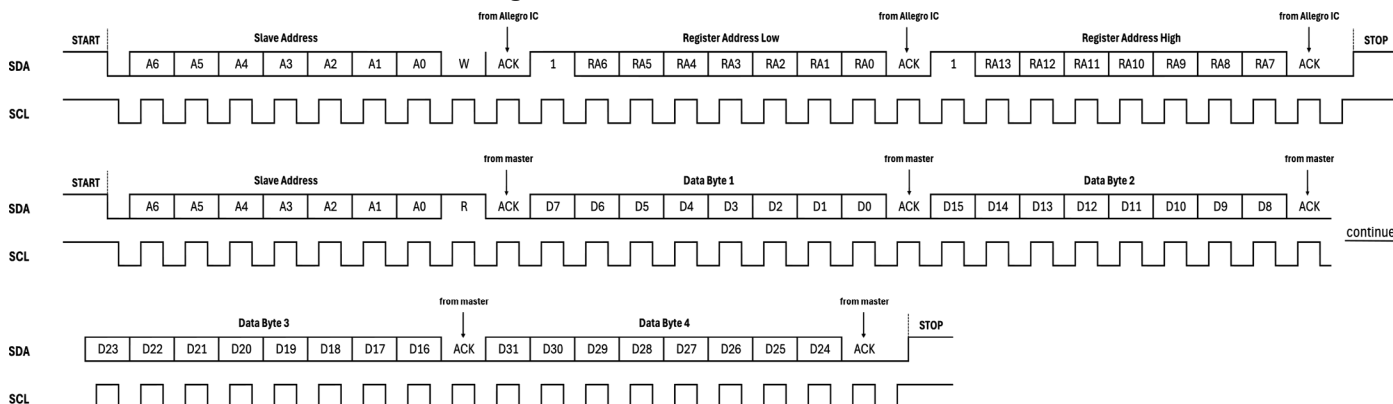


Figure 9: I²C Out-of-Frame Read Procedure

Out-of-frame mode

1. Start condition
2. (1 byte) 7-bit I²C slave address (device ID) 1010101, R/W bit = 0
3. (1 byte) Fixed bit = 1, bits [6:0] of the internal register address
4. (1 byte) Fixed bit = 1, bits [13:7] of the internal register address
5. Stop condition
6. Start condition
7. (1 byte) 7-bit I²C slave address (device ID) 1010101, R/W bit = 1
8. (4 bytes) Read four data bytes, least significant byte first, where the last byte should be ACK by the master
9. Stop condition

Table 7: Status Variables (Telemetry)

Variable Name	Address	Size	Type	Description
MPCI_VALUE_GLOBAL	0x0714	4B	uint32	Interface variable of multipurpose control interface: - Bits [15:0] are the data written into the multiprocessor communications interface (MPCI) - Bits [32:16] are the MPCI instructions (code of functionality to execute)
MCA_THETA	0x0718	2B	int16	Start rotor position, 9-bit resolution (0.3515625 degrees per bit): -2^9 (sign-extended to 16 bits) is -180 degrees 0 is 0 degrees $(2^9 - 1)$ is 179.99 degrees
MCA_FREQ	0x071a	2B	int16	Start rotor frequency, sfocFreqRes resolution, sign represents rotation direction: - Negative = clockwise (CW) - Positive = counter clockwise (CCW) - Zero = stationary
MCA_COMMAND	0x071c	2B	int16	Current value of the command after UCC; 15-bit resolution, signed: -2^{15} is the maximum negative command (-100%) 0 is the 0 command (0%) $(2^{15} - 1)$ is the maximum positive command (99.99%)
IPD_ERRSTATUS	0x071e	1B	uint8	IPD execution status: 0 = OK (initial position detection [IPD] finished successfully) 1 = STG1errorInductEqual_ipd 2 = STG2errorTimeout_ipd 3 = STG2errorEqualSaturation_ipd
WNDML_ERRSTATUS	0x071f	1B	uint8	Windmill execution status: 0 = OK (windmill finished successfully) 1 = ERBEMFsequence_wndml 2 = ERBEMFvalue_wndml
VQ_EST	0x0720	2B	int16	Q-axis voltage estimate (available in position-observer closed-loop mode of operation), signed, 14-bit resolution: $V_q = (\text{value}/2^{14}) \times (0.5 \times 1.1547) \times (\text{nominal voltage [V]})$ [V]
MCA_EXTCMD	0x0722	2B	uint16	Input of the curve transformer
V_TEMP	0x0724	2B	uint16	Code, proportional to IC temperature in °C
V_BUS_AVG	0x0728	2B	uint16	V_{BB} voltage: $V_{BB} = (\text{value}/2^{14}) \times (\text{nominal voltage [V]})$ [V]
V_CMD	0x072a	2B	uint16	Voltage command, 15-bit resolution
I_BUS	0x072c	2B	int16	I_{BB} current, signed, 11-bit resolution: $I_{BB} = 1.5 \times (\text{value}/2^{11}) \times (0.5 \times 1.1547) \times (\text{FS current [A]})$ [A]
IPD_STG2CURRDIFF	0x072e	2B	uint16	IPD stage 2 performance indication: difference in rate of rising of currents during two rounds of stage 2; value dimension depends on the configuration type of stage 2, according to: - Constant current: The difference in the times required to reach the configured amplitude of current in two rounds of stage 2, expressed as: value = TimeDifference, ns/32 ns - Constant time: The difference in the amplitudes of current spikes of stage 2, round, unsigned, 12 bit resolution, expressed as: value = $(\text{CurrDiff [A]} \times R_{\text{shunt}} (\Omega) \times \text{CSA}_{\text{gain}}) / ((V_{\text{ADCreff}} [\text{V}]) / (2^{12} - 1))$.
VBEMF	0x0730	2B	int16	Back-emf voltage (available in position-observer closed-loop mode of operation), signed, 14-bit resolution, expressed as: $V_{\text{bemf}} = (\text{value}/2^{14}) \times (0.5 \times 1.1547) \times (\text{nominal voltage [V]})$ [V]
P_MOTOR	0x0732	2B	int16	Motor power
FG_FREQ	0x0734	4B	int32	FG frequency, sfocFreqRes resolution, where the sign represents the rotation direction: - Negative = CW - Positive = CCW - Zero = stationary

Multipurpose Control Interface (MPCI)

The MPCI is a software interface that can be used for multiple predetermined purposes (e.g., to enter the customer password, to initiate NVM write-back, etc.) via simple 32-bit I²C writing into a fixed-address CPU RAM location (MPCI_VALUE_GLOBAL; for more details, refer to Table 7). The MPCI interface variable consists of two 16-bit-wide parts: data (i.e., password combination), contained in LSB bits [15:0]; and instruction, contained in MSB bits [31:16]. After execution (default value = 0), the MPCI returns the data output value in the LSBs. Functionalities currently supported by the MPCI are listed in Table 8. In any state of the motor controller, MPCI commands are checked. However, in a certain state of the motor controller, some commands might

need to be entered.

The MPCI interface variable (mpci_value_global) consists of two 16-bit wide parts:

- Instruction code is contained in the MSBs [31:16].
- Data (i.e., password combination) is contained in the LSBs [15:0]

After execution, the MPCI returns the data output value (where 0 is no error, 1 is e-flash error, and 2 is not in standby [only for NVM_WRITE_BACK]) written in the LSBs. The functionalities currently supported by the MPCI are listed Table 8.

Table 8: MPCI Functions

Functionality	Instruction Code	Example of transaction content value written into mpci_value_global variable via I2C 32-bit frame	Execution
IDLE: Idle	0x00	bits [31:16] = 0x00: instruction code bits [15:0] = any value	Idle (no action).
PASS_CHECK: Customer password check	0x01	bits [31:16] = 0x01: instruction code bits [15:0] = password to be entered	MPCI acknowledges customer password entry by rewriting instruction code and data (bits [31:0]) with zero (0x00) and performs password check The customer password entry and the check can be executed in any state of the motor controller.
NVM_WRITE_BACK: Customer NVM write back (shadow NVM to NVM copy)	0x02	bits [31:16] = 0x02: instruction code bits [15:0] = any value	Customer NVM write back: The shadow register of the CONFIG NVM is copied into the CONFIG NVM (RAM to e-flash copy) Write-back should be executed only in the standby state of the motor controller; otherwise, the command is ignored To signal completion of the write-back operation, the MPCI rewrites the instruction data (bits [31:16]) with zero (0x00) after completion of the copy procedure. The return data (bits [15:0]) indicate the following: 0 = copy successfully finished 1 = error occurred during copy 2 = controller not in the standby state (copy request ignored)

Operation Modes and Customer Password Protection

A89337 implements customer password protection access to certain regions of memory space, foremost, the copy of customer settings in the shadow registers. The A89337 operation modes are provided in Table 9.

Table 9: Operation Modes Access Rights

Name	Description
Application Mode (AM)	I ² C has access to user registers (default mode after power up)
User Mode (UM)	I ² C has access to user registers, shadow registers, and NVM configuration registers

The password length is 16 bits, and the password is determined by the customer (or it can be programmed by Allegro upon initial part configuration) by configuring the corresponding parameter (customerPassword) in the NVM map.

All combinations except 0x0000 and 0xFFFF are considered valid passwords.

If the password combination is set to 0xFFFF or 0x0000, the customer password is not configured and, after POR, the part enters user mode. In this way, when the part arrives, the customer has access to the e-flash NVM to set the user mode password (by configuring the corresponding parameter, customerPassword, in the NVM map). Once the user mode password is configured, after POR, the part enters application mode and the e-flash NVM and shadow registers are protected.

To change the operation mode, it is necessary to write into the MPC_I_VALUE_GLOBAL variable via the I²C 32-bit frame. The variable (MPC_I_VALUE_GLOBAL) consists of two 16-bit-wide parts:

1. Instruction code is contained in MSB bits [31:16]; this includes the instruction code for the customer password check is 0x01.
2. Data (i.e., password combination) is contained in the LSBs [15:0].

Table 10: Customer Password Check

Functionality	Customer Password Check
Instruction Code	0x01
Example	For transaction content value written into the MPC_I_VALUE_GLOBAL variable via the I ² C 32-bit frame: • Bits [31:16] = 0x01—instruction code • bits [15:0]—password to be entered
Execution	The device acknowledges the customer password entry by rewriting the instruction code and data (bits [31:0]) with zero (0x00), then the device performs the password check. The customer password entry and the check can be executed in any state of the motor controller.

If a valid customer password is configured, the controller remains in the application operation mode, where the user registers and the fixed-address variables are accessible until the configured password is correctly entered in MPC_I_VALUE_GLOBAL.

Once the password is entered correctly, the controller enters the user operation mode, and the user registers, fixed-address variables, FOC internal variables, shadow registers and e-flash NVM configuration registers can be accessed.

If the wrong combination is entered in MPC_I_VALUE_GLOBAL, further checks of the customer password are not performed until reboot of the controller via POR.

ANALOG

The motor command is proportional to the analog voltage applied on the PWM/SPD terminal.

Direction, when available, is defined in the NVM and UCC. For information, refer to the Motor Rotation Direction section.

The analog command is converted into the digital domain, then rescaled according to:

Equation 10:

$$mca_{extcommand} = (2^{15} - 1) \times V_{(PWM,SPD)} / AnaSpdMax$$

where AnaSpdMax is a fixed voltage equal to 2.4 V. For example:

Table 11: Analog Input Mapping

V _(PWM,SPD) [V]	mca _{extcommand} [LSB] AnaSpdMax = 2.4 V
0.5	4551
1.0	13652
2.4	32767

PWM DUTY CYCLE

Motor control is proportional to the duty cycle applied on the PWM/SPD terminal. The frequency range of the input PWM is from 100 Hz to 100 kHz.

Direction, when available, is defined in the NVM and UCC. For more information, refer to the Motor Rotation Direction section.

If the external PWM frequency is less than the minimum threshold (<100 Hz), it is observed as 0% DC.

If the external PWM frequency is greater than the maximum threshold (>100 kHz), it is observed as a terminal stuck in the high state and it is interpreted as a 100% DC command.

The duty cycle command is converted into the digital domain according to:

Equation 11:

$$mca_{extcommand} = (2^{15} - 1) \times duty$$

For example:

Table 12: PWM Duty Cycle Input Mapping

duty(PWM,SPD) [%]	mca _{extcommand} [LSB]
25	8191
50	16383
75	24575

In this mode, it is possible to enable the clock mode feature to synchronize the internal clock with the external PWM signal frequency. This can provide higher speed accuracy and can eliminate the need for a precision external resistor on the ROSC terminal. The external PWM signal is used to compensate for the inaccuracy of the integrated oscillator, allowing speed accuracy to be achieved down to $\pm 0.1\%$. When this feature is activated in NVM (via the clkCompEnable setting), the input-admissible frequencies of the external PWM signal are fixed and need to be specified via the clkCompRef setting in NVM. To detect the frequency, the input duty cycle of the PWM signal must be greater than 0% and less than 99%. If a 0% or 100% duty cycle value is applied, the clock compensation is not applied. In addition, the supplied frequency must be within $\pm 10\%$ of the frequency specified by the clkCompRef setting; otherwise, the clock compensation is not applied. Speed accuracy is directly related to the accuracy of the PWM clock frequency to the selected clkCompRef frequency.

PWM FREQUENCY

Motor control is proportional to the frequency applied on the PWM/SPD terminal. The Input frequency range needs to be between 50 Hz and 3000 Hz (duty cycle is intended to be 50%, but this mode works with any percentage except 0 or 100).

If the PWM input frequency is greater than 3000 Hz, $mca_{extcommand}$ is assumed to be the maximum value, equal to 32767 [LSB]. If the PWM input frequency is less than 50 Hz, $mca_{extcommand}$ is equal to 0.

Direction, when available, is defined in the NVM and UCC. For information, refer to the Motor Rotation Direction section.

The frequency command is converted into the digital domain according to:

Equation 12:

$$mca_{extcommand} = (pwm_freq [Hz] - 50) \times map_{coefficient}$$

Where $map_{coefficient}$ is the scaling factor equal to 11.1.

For example:

Table 13: PWM Duty Cycle Input Mapping

Frequency(PWM,SPD) [Hz]	mca _{extcommand} [LSB]
50	0
100	$(100 - 50) \times 11.1 = 555$
200	$(200 - 50) \times 11.1 = 1666$
3000	$(3000 - 50) \times 11.1 = 32745$

Universal Curve Controller (UCC)

The $mca_{extcommand}$ motor command is passed through a universal curve controller to create an arbitrary command profile, then to the FOC algorithm. The input of the UCC is unsigned ($mca_{extcommand}$ [LSB]) and the output is signed ($mca_{command}$ [LSB]), as shown in Figure 12.

The UCC is a transformer curve defined by corner points. Each point has a specific input value and its corresponding output value. The values between the points are calculated using linear interpolation. Up to 10 corner points can be defined in the NVM (ucc_x0 to ucc_y9). Not all corner points need to be defined. The number of points that must be defined is equal to the number of corner points that are needed for the desired curve. UCC points need to be configured in the standby state, written to the configuration NVM (e-flash). After configuration, a POR is required.

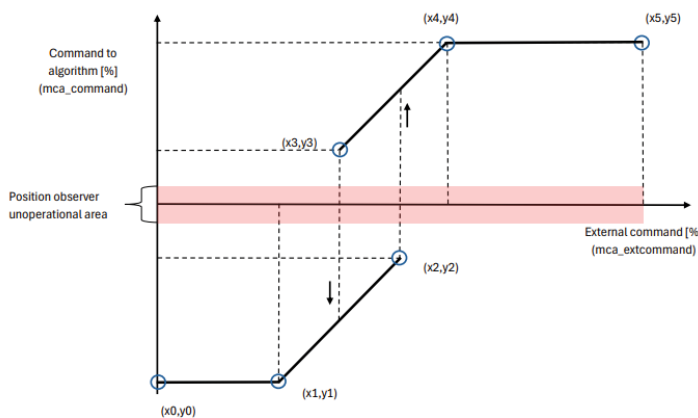


Figure 12: Universal Curve Controller (UCC)

A closed-loop nonoperational region, defined by the position observer unoperational area (poOper) NVM parameter as marked by the red hatched rectangular region in Figure 12, defines the minimum speed operation for reliable position-observer estimation and proper FOC functionality. In this region, operation is not reliable. Therefore programming of the UCC in this region is not recommended. The easiest way to avoid operation within this region it is to use hysteresis, as shown in Figure 12.

The UCC command transformer can be configured in endless ways. Several useful examples follow.

LINEAR OPERATION CONFIGURATION (DEFAULT)

For applications that do not require any transformation of an external command, it is possible to configure the UCC in such a way that commands to the control algorithm match the input. Points $(x0, y0)$ (set to 0) and $(x1, y1)$ are used for this purpose. There is not an option to disable UCC. The last-used point of UCC curve $(x1, y1)$ can be used to set the motor base speed. In that case, the $mca_{command}$ range represents the entire range achievable by the motor (100% mapping). If a direction reversal is necessary, the end point $(x1, y1)$ can be configured to be positive or negative.

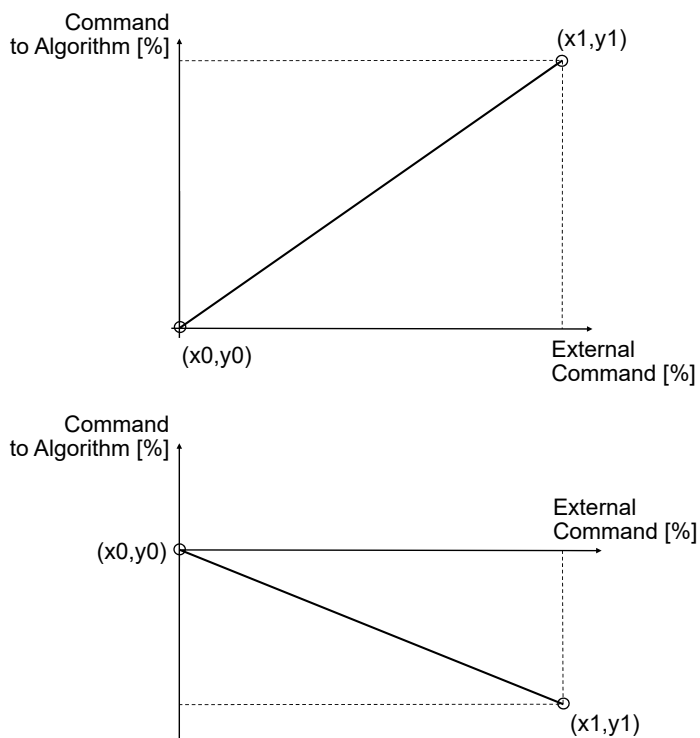


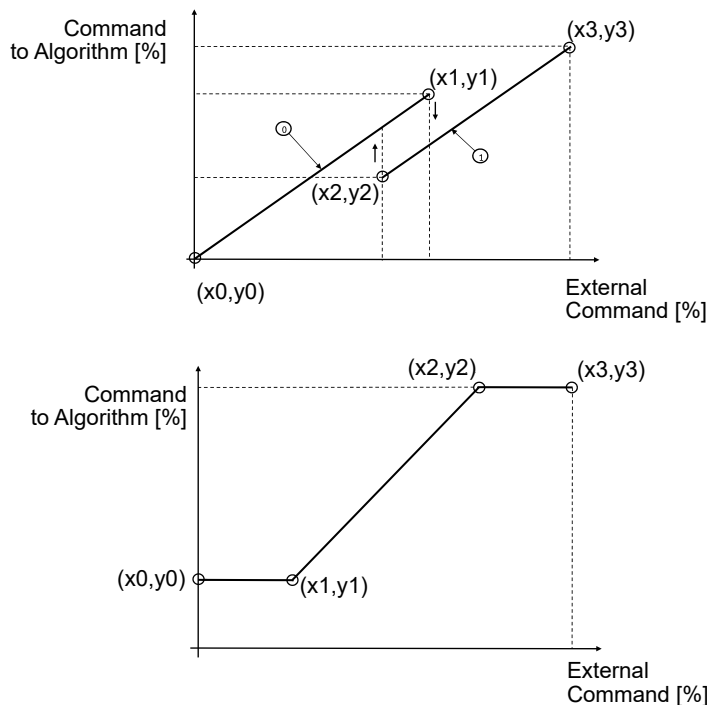
Figure 13: UCC Linear Configuration for Positive (top) and Negative (bottom) Values

HYSTERESIS CONFIGURATION

To implement the hysteresis feature, the abscissa coordinate of $(n + 1)$ -th point must be smaller than the n th point. Then, the area between those two points is considered to be the hysteresis, and the value of the command to the control algorithm is calculated based on the previous range of commands. This feature is illustrated in Figure 14. The area between $(x1, y1)$ and $(x2, y2)$ is considered to be the hysteresis because $x2 < x1$.

Principle of operation:

- If the external command is between $x0$ and $x2$ or between $x1$ and $x3$, the value of the command to control algorithm is calculated by the 0-line or 1-line coefficients, respectively
- If the external command is between $x2$ and $x1$ (the hysteresis region), the command to the control algorithm is calculated based on the range of previous values. For example, if the previous external command is greater than $x0$ and less than $x2$, and the new external command is in the hysteresis region, the command to the control algorithm is calculated by the 0-line coefficients



**Figure 14: UCC Curve Hysteresis Configuration (top),
UCC Curve Minimum and Maximum
Configuration (bottom)**

MINIMUM AND MAXIMUM COMMAND CONFIGURATION

To program the minimum/maximum external command clamp, the user must define the horizontal lines in the desired area as shown in Figure 14. Here, line $(x0, y0) - (x1, y1)$ is used to create the minimum command clamp, and line $(x2, y2) - (x3, y3)$ is used to create the maximum command clamp.

COMMAND CLIPPING CONFIGURATION

If the entire range of the command is not necessary, the x value can be set to be less than 100%. Every value of the external command greater than the last-used x value is 0.

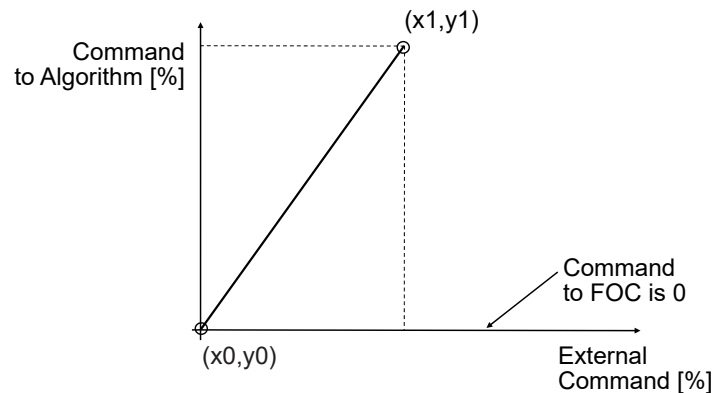


Figure 15: UCC Command Clipping

Control Modes

The A89337 includes a configurable control loop able to support three operational modes for motor control: speed, torque, and power. The control variable can be selected with the sfocVarCtrl-Mode setting in the NVM register.

Table 14: Control Mode Selection

sfocVarCtrlMode	Control Mode
0x0	Speed
0x1	Torque
0x2	Power

SPEED

In speed mode, the $mca_{command}$ variable is used as the reference for the FOC speed loop to regulate the motor speed to f_{ref} [Hz].

The $mca_{command}$ variable is scaled to the maximum frequency [Hz] according to:

Equation 13:

$$f_{ref} [Hz] = mca_{command} [LSB] \times f_{max} / (2^{15} - 1) ([Hz] / [LSB])$$

TORQUE

In torque mode, the $mca_{command}$ variable is used as the reference for the FOC current loop to regulate the motor torque proportional to I_{ref} [A].

In this mode, the FOC speed loop is bypassed, and the reference command is compared directly with the estimated q-axis current, $I_{q_{ref}}$.

The $mca_{command}$ variable is scaled to the maximum system current [A] according to:

Equation 14:

$$I_{ref} [A] = mca_{command} [LSB] \times I_{max} / (2^{15} - 1) ([A] / [LSB])$$

In torque control mode, the $mca_{command}$ variable is downscaled to the current fraction length (11 bit). For this reason, the FOC algorithm command changes only if the external command varies for more than 16 units.

POWER

In power mode the $mca_{command}$ variable is used as the reference for the FOC power loop to regulate the motor power to P_{ref} [W].

The $mca_{command}$ variable is scaled to the maximum system current [A] according to:

Equation 15:

$$P_{ref} [W] = mca_{command} [LSB] \times P_{max} / (2^{15} - 1) ([A] / [LSB])$$

In power control mode, the command is downscaled to the voltage fraction length (14 bits), so the FOC algorithm command changes only if the external command varies for more than 2 units.

A89337,
A89337-1

12 V Sensorless Code-Free FOC
BLDC Motor Controller

Motor-Control Application

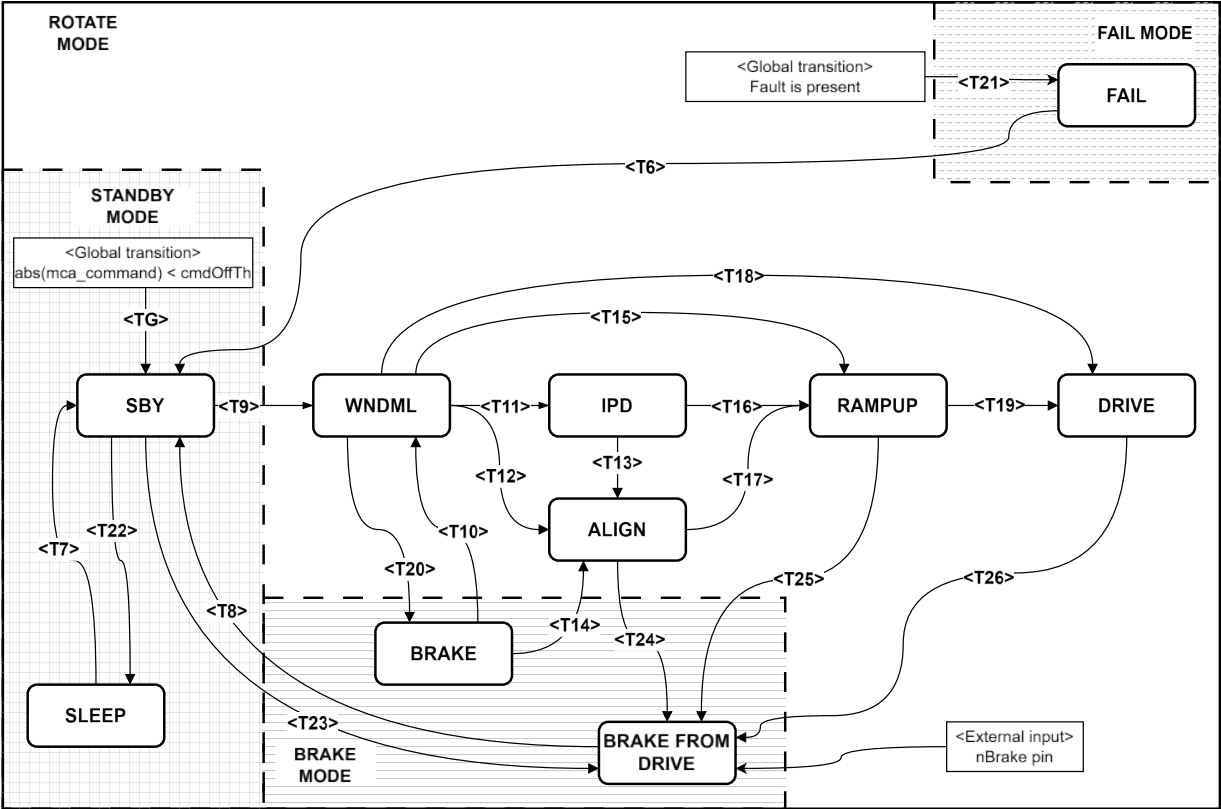


Figure 17: Motor-Control Application, Finite State Machine

Table 15: Motor-Control Application, State Table

	SBY	WNDML	IPD	ALIGN	RAMPUP	DRIVE	BRAKE	FAIL	SLEEP	BRAKE FROM DRIVE
SBY		T9						T21	T22	T23
WNDML	T1(TG)		T11	T12	T15	T18	T20	T21		
IPD	T2(TG)			T13	T16			T21		
ALIGN	T3(TG)				T17			T21		T24
RAMPUP	T4(TG)					T19		T21		T25
DRIVE	T5(TG)							T21		T26
BRAKE		T10		T14				T21		
FAIL	T6							T21		
SLEEP	T7							T21		
BRAKE FROM DRIVE	T8							T21		

Table 16: Motor-Control Application State Transitions

Transition Number	Transition Description
T1	Command mca_command is greater than the off threshold, or the rotor is already spinning (windmilling)
T2	Command mca_command is greater than the off threshold
T3	Command mca_command is greater than the off threshold and a request to brake is not present
T4	Command mca_command is greater than the off threshold and a request to break is not present
T5	Command mca_command is greater than the off threshold and a request to break is not present
T6	Faults and latched fault are cleared
T7	Analog input voltage is greater than threshold
T8	Brake procedure completed
T9	Command mca_command is greater than the on threshold
T10	Rotor is rotating in the forward direction
T11	Rotor is considered as stopped, and AlignType is configured as IPD
T12	Rotor is considered as stopped, and AlignType is configured as IPD, or windmill detection was not successful
T13	IPD was not successful
T14	Rotor is rotating in the reverse direction

Transition Number	Transition Description
T15	Rotor is rotating in the forward direction or in the reverse direction with FOC deceleration enabled; rotor speed is less than the observer threshold
T16	IPD was successful
T17	Command mca_command is greater than the off threshold
T18	Rotor is rotating in the forward direction with windmill braking disabled or in the reverse direction with FOC deceleration enabled; rotor speed is greater than the observer threshold.
T19	Command mca_command is greater than the off threshold
T20	Rotor is rotating in the forward direction with windmill braking enabled or in the reverse direction with FOC deceleration disabled
T21	Fault is present
T22	Sleep is enabled, and the sleep counter has reached the threshold
T23	Brake requested
T24	Command is less than a stop threshold, and the brake is requested
T25	Command is less than a stop threshold, and the brake is requested
T26	Command is less than a stop threshold, and the brake is requested

FOC DESCRIPTION

The full field-oriented control (FOC) algorithm is integrated in the A89337 integrated circuit (IC), resulting in highly efficient, dynamic control of a BLDC motor. The main building blocks of the Allegro FOC algorithm are shown in Figure 18. A variable-loop PI controller offers three different control modes of operation (speed, torque, and power) configurable by NVM parameter. An output of the variable-loop PI controller serves as the reference for the inner loop, namely the current PI controllers. This control loop assures stability of DQ currents. The output of the current controller is the DQ voltage. These voltages are transformed into three phases and fed through the PWM scheme. The goal is to achieve the maximum possible DC bus voltage utilization. The modulated voltage signal is converted to a PWM gate driving signal in the microcontroller PWM module, which is

Allegro-proprietary IP.

To close the feedback loop, current is measured with a single-shunt current measurement scheme. The measured current (in the three-phase ABC domain) is transformed into the DQ domain, which is used for control.

To achieve this transformation, position angle data is obtained from the position and speed estimator. This block is based on the specialized lightweight algorithm resulting in position and speed estimates without the need of an external sensor. If speed is selected, the speed estimate serves as the closed-loop feedback signal for the variable-loop PI controller. The observer is tuned via the sfocPOthetaK and sfocPOfreqK NVM settings; this function requires the phase inductance value in the parameter sfocLs.

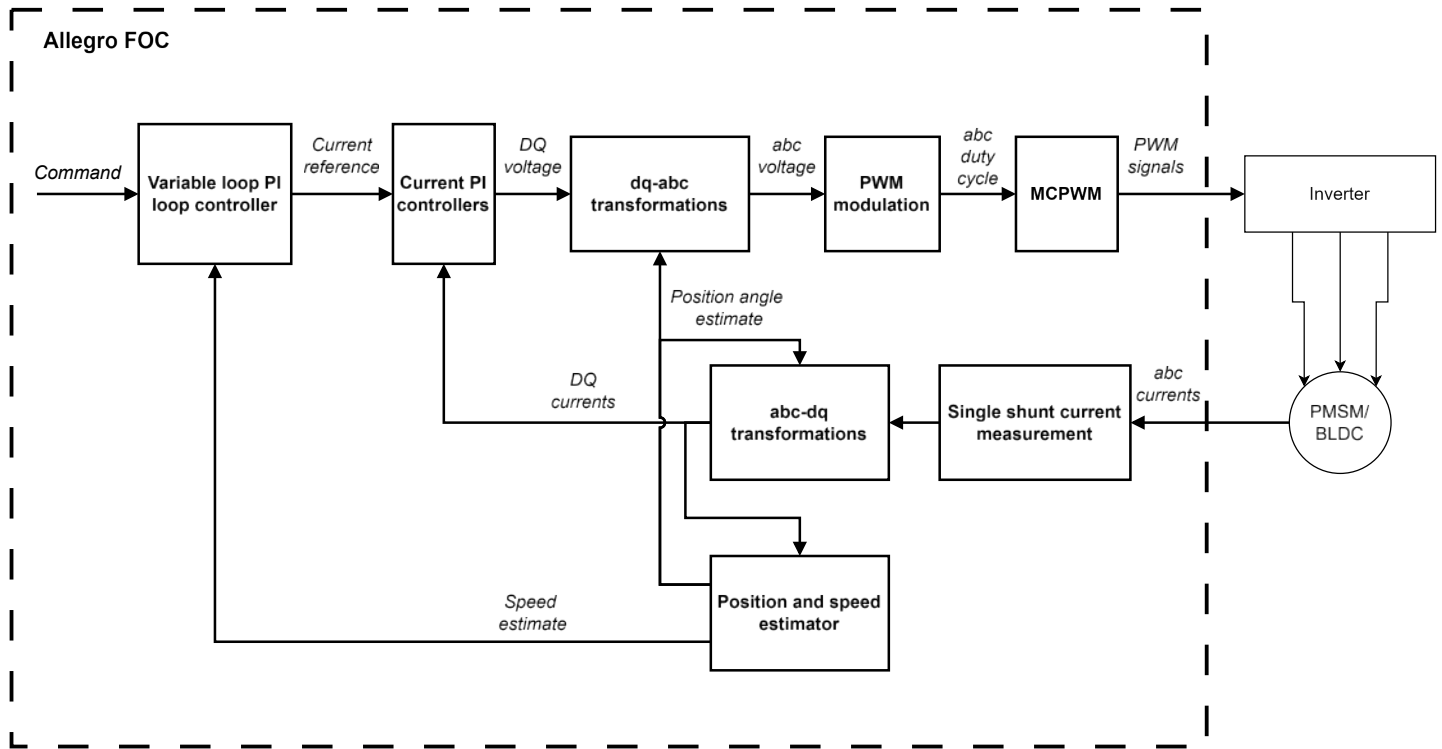


Figure 18: Allegro FOC Diagram

In speed or power mode, the $mca_{command}$ command is used as a reference set point to the speed loop, and the output of the speed loop is used as the Q-axis current reference for the current loop.

In torque mode, the $mca_{command}$ command is passed directly to the current loop as the Q-axis current reference.

The D-axis current reference can be set via the $sfocDrvDcurrRef$ parameter; however, to ensure that the stator and the rotor field are 90 degrees out of phase with each other such that the maximum torque and system efficiency are achieved, it is recommended for this setting to remain at zero.

Two current PI controllers for control of the DQ current can be tuned using the $sfocDcurrKp$, $sfocDcurrKi$, $sfocQcurrKp$, and $sfocQcurrKi$ settings in the NVM. The variable PI controller can be tuned with the $sfocVarKp$ and $sfocVarKi$ settings in the NVM.

The A89337 integrates a space-vector-modulation (SVM) technique to optimize the supply voltage and to obtain a higher base speed of the motor.

MODES

Modes serve as higher abstraction of states. They offer faster diagnostics of the current state of the device. One mode can contain several states (see Figure 17).

Table 17: MCA Modes

Mode	Description
Standby	Motor is not in rotation and a fault is not present
Rotate	Motor is being actively driven
Brake	Brake is applied
Fail	Fault is present

STATES

Standby

In the standby (SBY) state, the motor is not driven, and the power stage is in tri-state so that the motor can be externally rotated.

If the sleep state is enabled ($\text{sbyLPSEn} = 1$), the device is in low power consumption mode. As soon as a wake-up command is received, the device is ready to process an external command. Details are provided in the Sleep section.

If the sleep state is not enabled ($\text{sbyLPSEn} = 0$) the device is immediately ready to process an external command without need of a wake-up event.

Only when $\text{mca}_{\text{command}}$ exceeds the cmdOnTh threshold, the A89337 transitions to the windmill (WNDML) state. When $\text{mca}_{\text{command}}$ is less than cmdOffTh , the A89337 returns to the standby state.

Windmill

In the windmill (WNDML) state, the motor is not driven. However, bemf signals from the windings of the motor are analyzed to determine rotation speed of the motor, the position of the rotor, and the direction of the movement.

The response of the state machine varies based on the rotor speed measurement:

- If motor speed is less than the stopped threshold (stpdTh , NVM parameter), or if an illegal switching sequence is detected, the device enters the initial position detection (IPD) state or the rotor alignment (ALIGN) state to determinate the position of the rotor.
- If motor rotation exceeds the stopped threshold (stpdTh), and motor rotation is in the:
 - Intended direction, and the brake in windmill parameter in NVM is:
 - Enabled ($\text{brakeInWndmlAlwaysEn} = 1$), the device enters the brake state to reduce speed. The brake pattern is

applied until the rotor is stationary. When the speed is less than stpdTh , the device enters the standby state.

- Not enabled ($\text{brakeInWndmlAlwaysEn} = 0$), the device enters either the drive state or the ramp-up state, depending on the absolute speed value with respect to the observer operational threshold (configurable poOper parameter in NVM).
- Unintended direction, and the brake in windmill parameter in NVM is:
 - Enabled ($\text{brakeInWndmlAlwaysEn} = 1$ or $\text{brakeInReverseType} = 1$), the device enters the brake state to reduce speed. From the brake state, the device returns to the windmill state to verify the speed condition. When the speed is less than stpdTh , the device enters the standby state; otherwise, the FOC decelerates the motor. Once the motor stops, it restarts in the opposite direction.

IPD

The initial position detection parameter configurable in NVM ($\text{AlignType} = 1$) is determined by measuring the stator inductance difference caused by rotor flux. The IPD technique offers a smoother and faster startup without reverse spinning. It does this by using salient characteristics and winding saturation characteristics to determine the rotor position.

IPD is performed in two stages. In the first stage, test pulse sequences are applied to the motor. If there is enough saliency ($L_q \neq L_d$), A89337 locks up the rotor position within 30 degrees of 0° to 180° , or 180° to 360° .

When the first stage is complete, the A89337 applies a test sequence that uses the saturation property of the motor phase inductances to determine the current magnetic polarity of the rotor. Once the entire sequence is complete, A89337 can determine the rotor angle with 30-degree accuracy.

After alignment is finished, the FOC algorithm assumes that the rotor is stationary and is in the expected position, and the device enters the ramp-up state.

Align

Align mode (AlignType = 0). A89337 supports both AC and DC alignment.

- **DC alignment** is enabled by the configuration `sfocACDCAlignType = 0`. A DC current is injected for a fixed length of time configured by `sfocAlgnTime` so that the rotor aligns with the stator in a known position. The current limit during DC alignment is configured through `sfocAlgnDcurrRef`. `sfocAlgnTime` and `sfocAlgnDcurrRef` are generally related to the inertia of the system and, if the inertia is great or if a load is applied on the shaft, an increase in these parameters is advised.
- **AC alignment** is enabled by the configuration `sfocACDCAlignType = 1`. A low-current controlled driving frequency is applied to the winding to generate a known position flux in the rotor. The integrated permanent magnet rotor gravitates to the generated stator flux. The AC alignment process spins the rotor in the open-loop control mode for the configured duration (`ALGN_DUR`, `sfocAlgnTime`) with constant frequency (`AC_ALGN_FREQ`, `1/ALGN_DUR`) and current reference (`sfocAlgnDcurrRef`). The alignment period should be chosen to match the required load torque to gravitate the rotor. The chance that the rotor is caught during this process is greater for a longer alignment period than a shorter alignment period.

In the align mode, the rotor might oscillate before it settles (aligns). When IPD fails to detect the initial rotor position (IPD-FailAction), the align mode is entered as a default action.

Once IPD completes, the FOC algorithm assumes that the rotor is stationary and in expected position, and the device enters the ramp-up state.

Ramp-Up

During the ramp-up state, the driving frequency increases in open-loop mode until sufficient $bemf$ is generated to allow the observer to accurately detect the rotor speed and position to transition into the closed-loop state (drive state).

The open-loop ramp-up increases the rotor frequency up to `RAMP_UP_FREQ` based on:

- `sfocRampTime`, which defines the length of the ramp-up.
- `sfocRampStep`, which defines the acceleration (frequency per second) during the ramp-up phase only.
- `sfocRupDcurrRef`, which defines the current during ramp-up. The ramp-up speed is not affected by the value of the current. The appropriate current level depends on the motor property and the connected load.

The choice of `sfocRampStep` and `sfocRampTime` affects the frequency at which the transition between open-loop (OL) and closed-loop (CL) operation occurs.

For reliable startup, the recommended setting for the OL-CL transition frequency is faster than 35 Hz. To achieve this, an appropriate `sfocRampStep` value must be selected. This allows use of a current equal to 60% of the rated current and an `sfocRampTime` that is not too short.

A motor with larger inertia typically requires a smaller `sfocRampStep` and a longer `sfocRampTime`.

To minimize start time and achieve a fast startup, it is necessary to minimize the ramp-up time in open-loop operation. This may require an OL-CL transition frequency of less than 35 Hz at the expense of reliability. In addition, the value of `sfocRampStep` must be increased and the value of the current to start the motor must increase to 80% to 100% of the nominal current, while the `sfocRampTime` value must be reduced.

Generally, fast startup demands high current.

Drive

In the drive state, the motor spins using a closed-loop sensorless field oriented control (SFOC) technique.

This SFOC technique implements an Allegro-patented light observer to provide a sensorless estimation of the motor angle and speed.

In the drive state, the current can be further controlled and limited by the following parameters:

- `sfocDrvCurrDrvLmt` sets the maximum current driving capability in q axis.
- `sfocDrvCurrRateLmt` and `sfocDwnCurrRateLimit` limit the rate of growth and reduction of the q current.
- To prevent sudden changes in the torque applied to the motor which could result in acoustic noise and instability, the A89337 provides the option to limit the maximum rate at which the speed command can change through the `sfocRefRateLmt` parameter.
- In the case of strong deceleration, the motor can operate as a generator and transfer energy back to the power supply. If the power supply is not able to absorb the energy, V_{BB} voltage is pumped up with potential damage to the system. The A89337 can limit pump-up of the V_{BB} via the `sfocVbbLmt` and `sfocVbbLmtTh` parameters. The I_q regenerative current can be instead limited via the `sfocCurrGenLmt` and `sfocGenCurrRateLmt` parameters. For further protection, the TVS diode can be added to the application to limit the pump-up of V_{BB} that is caused by strong deceleration.

Brake

Brake mode is available to quickly slow the motor when desired. The type of brake that is applied is configurable via the NVM parameter `brakeType`:

- `brakeType = 0`: Low-side FET brake—The brake is applied by enabling all three low-side drivers at the same time, which shorts the motor `bemf`.
- `brakeType = 1`: Constant current brake: The brake is applied by FOC.

While using the low-side FET brake (`brakeType = 0`), the winding current in brake mode is limited by the motor and the resistance of the external FETs. Care must be taken to avoid excessive current in the FETs when braking. Several configuration options are available for programming in the NVM to allow for safe braking, such as pulsed braking, and braking hold-off based on motor frequency.

In the brake state, the device activates the low-side MOSFET and brakes the motor according to the parameters:

- `brakeFETonDuration`
- `brakeFEToffDuration`

If the speed of the motor exceeds `brakeFreqTooHighTh`, the motor coasts before the brake activates. This delay avoids the generation of excessive current, which can otherwise damage the MOSFET or the motor.

The device enters the brake state when one of the following conditions is verified:

- `nBRAKE` terminal is driven low
- brake bit is set in the `BRAKE_CTRL` register
- `WINDML` state is being exited, and the braking feature is required to exit the windmill state.
- Speed command is received where `mcacommand` is less than the `cmdOffTh` parameter, and `brakeIfOffEn = 1`.

NOTE: If `brakeIfOffEn = 0` the device enters the standby state and the motor coasts.

If `brakeIfOffEn = 1`, the device continues to brake the motor (i.e., low-side MOSFET in the on state) after the motor stops.

NOTE: Deceleration of the motor is possible in closed-loop operation (the drive state) if `softOffEn` is set.

Fail

When any fault is active or latched, the device enters the fail state. If a fault is latched, the A89337 handles latch reset based on the processed external command (first, less than `cmdOffTh`; then, greater than `cmdOnTh`). In the event of a stalled rotor or a motorless start condition, motor operation restarts from the standby state.

If any brake-on-fault parameter is enabled, the brake operation is performed in the fail state, where the low-side FETs brake with the same configuration as the high-side FETs.

Sleep

In the sleep state, turn-off of almost all internal circuitries occurs. This includes all internal voltage regulators. This minimizes power consumption and reduces the current absorbed by the device to the level of I_{BB3} . After power-on of the part, the device is always in the standby state before it enters the sleep state, provided that the entering condition is satisfied. The sleep state can be disabled via the `sbyLPSEn` NVM parameter.

The input condition for the sleep state varies according to the selected external command source (`extCmdSrc`).

- **PC:** The NVM parameter `cmdOnTh` defines the external command level threshold for the transition to the sleep state. If sleep is enabled (`sbyLPSEn`) and the command is less than the `cmdOnTh` parameter and the device is in the standby state, the sleep timeout (configured via the `sbyLPSTh` NVM parameter) starts; then, after the duration of the sleep time outs, the device enters sleep.
- **PWM duty or frequency:** If sleep is enabled (`sbyLPSEn`) and the PWM/SPD terminal is held at 0 and the device is in the standby state, the sleep time out (configured via the `sbyLPSTh` NVM parameter) starts and, after the duration of the sleep time out, the device enters sleep.
- **Analog:** The NVM parameter `sbyLPSanaTh` defines the external command level threshold for transition to sleep state. If sleep is enabled (`sbyLPSEn`) and the external command is less than the `sbyLPSanaTh` parameter, and the device is in the standby state, the sleep timeout (configured with `sbyLPSTh` NVM parameter) starts and, after the duration of the sleep timeout elapses, the device enters sleep. The A89337 wakes from sleep state upon a rising edge on the PWM/SPD terminal that is greater than the analog voltage threshold `AnaSpdExit`.

The sleep timeout can be set between about 0.3125 ms and 33.792 s, according to:

Upon wake-up, the A89337 reinitializes, clears the shadow NVM registers, and reloads the contents of the NVM registers according to Figure 19.



System Faults and Diagnostics

A89337 integrates advanced diagnostics for critical hardware protection and motor-control diagnostics. Hardware protections include overvoltage, undervoltage, overtemperature, and short-circuit protection. Motor-control diagnostics include lock detection (loss of synchronization) and “no-motor” (i.e., motorless) start.

In case of a fault event, the nFLT terminal is pulled low to indicate a fault condition. For most faults, braking can be masked via NVM parameters (Vcpuv, Vbbov, OCP, OVT, Lock_detect, NoMotorStart). Braking is not permitted for the V_{BBUVLO} condition.

Every fault can be enabled or disabled (masked) with the NVM parameters uvsDisable, ovtDisable, ovpEnable, ocpEnable, rotStallDetCtrl, and noMotorStartEn.

Fault data flags can be read in the software (SW) status register via I²C communication.

Table 18: Faults

Fault	Fault Action	Latched
VBB Undervoltage (VBB UVS)	Disable outputs	N
Overtemperature Protection	Disable outputs	N
VBB Overvoltage Protection (VBB OVP)	Disable outputs	N
VCP UVLO	Disable outputs	N
Overcurrent Protection (OCP)	Disable outputs	Y [1]
Loss of Sync (lock Det)	Set lock-detect timeout, disable outputs	Y [1]
No-Motor Start	Set lock-detect timeout, disable outputs	Y [1]

[1] The latch resets upon command of an off/on sequence or a power cycle.

VBB UNDERVOLTAGE—VBB UVS

The A89337 disables outputs when the V_{BB} supply voltage reduces to less than its UVLO falling threshold (typically 6.75 V). Digital logic, however, remains active until VBB reduces to less than the lower operating threshold. This UVLO features a 1 V hysteresis, with a rising threshold of 7.75 V.

Upon detection of a V_{BB} undervoltage condition, outputs are disabled, the controller enters the fail mode, and fault reporting is performed by pulling the nFLT terminal low and setting the corresponding flag in the SW status register. When V_{BB} recovers to greater than its rising threshold, the fault flag clears, and the controller transitions to the standby state. In the standby state, if the command exceeds the cmdOnTh parameter, the controller transitions to the windmill state. In the windmill state, a speed check determines if the device starts from a stationary position or with a moving rotor.

The UVLO blanking time is set by a 3-bit value, *N*, calculated as $N \times 2^{(N+7)} V_{CL}$ cycles, as provided in Table 19.

Table 19: UVLO Blanking Time

N [3-bit]	uvloBlankTime [V _{CL} cycles]
0	0
1	256
2	1024
3	3072
4	8192
5	20480
6	49152
7	114688

OVERTEMPERATURE PROTECTION—OVT/TSD

The A89337 protects itself from overheating with an internal thermal monitoring circuit. If the junction temperature exceeds the upper threshold T_{JTSD} . When the OVT is detected, the outputs are disabled, the controller enters the fail mode, and fault reporting is performed by pulling low the nFLT terminal and setting the corresponding fault flag in the SW status register. When the device temperature returns to less than the T_{JTSD} threshold minus the hysteresis level, ΔT_J , the outputs are re-enabled automatically (not latched), the fault flag is cleared, and the controller transitions to the standby state. In this state, the state controller first verifies that the command exceeds the cmdOnTh threshold, then transitions to the windmill state where a speed check is performed. The device then determines if start is to be performed from a stationary position or with a moving rotor.

VBB OVERVOLTAGE PROTECTION—VBB OVP

Overvoltage protection is intended to protect the IC and the three-phase inverter from applications in which high-velocity motor rotation is possible or rapid motor-speed decelerations are requested, where significant energy can be fed back from the motor coils to the power supply. Overvoltage protection might also be relevant when low-side FET braking is applied: since the A89337 switches the low side FETs during braking, during t_{off} , the voltage generated on the inductors of the motor force the current to flow in the body diode of the high-side FETs charging the capacitor of the DC-link and generating a pump up on VBB.

When V_{BB} voltage exceeds ovpTh, the A89337 disables the outputs. After the V_{BB} overvoltage is detected, the outputs are disabled, the controller enters the fail mode, and fault reporting is performed by pulling low the nFLT terminal and setting the corresponding fault flag in the SW status register. When the V_{BB} level returns to less than the ovpTh parameter (minus the hysteresis level), the outputs are re-enabled automatically (not latched), the fault flag is cleared, and the controller transitions to the standby state. In the standby state, the controller verifies that the command exceeds the cmdOnTh threshold, then it transitions to the windmill state. In the windmill state, a speed check is performed, and the device determines if the start is to occur from a stationary position or with a moving rotor.

The V_{BB} overvoltage protection level is programmable via the ovpTh parameter in NVM. The programmable range of this parameter is approximately 15 V to 26 V.

The V_{BB} overvoltage diagnostic is active independently by the control method and the state. If not needed, the fault can be masked via the ovpEn parameter in NVM.

VCP UNDERVOLTAGE—VCP UVLO

A V_{CP} undervoltage is intended to protect the power FETs when the charge pump does not produce enough voltage to safely drive them. If the voltage on the charge pump (VCP) terminal reduces to less than its falling UVLO threshold, the high-side and low-side drivers are disabled, and the device enters the fail state. Once the V_{CP} level increases to greater than the rising threshold, the operation transitions to the standby state.

OVERCURRENT PROTECTION—OCP (SHORT PROTECTION)

Overcurrent protection is intended to protect the power FETs from application conditions of a shorted load, a motor short to ground, and a motor short to battery that can damage the system.

The V_{DS} voltages across each power MOSFET are monitored by the A89337. If a V_{DS} exceeds a predefined threshold (ocpTh) when that MOSFET is in conduction mode (i.e., in the on state), an OCP fault is triggered, and the output is disabled.

Table 20: OCP Thresholds

ocpTh	V_{DS} Threshold (V)
0	0.53
1	0.80
2	1.09
3	1.61

If the braking is not configured, the bridge remains in the off state and, after the off command (i.e., a command less than cmdOffTh) is received, the restart procedure can be performed.

Overcurrent protection (OCP) is a latched event, and the output remains in the off state until a command is received for an off/on cycle, or until power is cycled.

LOSS OF SYNC (LOCK/STALL DETECTION)

A89337 implements three stall-detection mechanisms selected by the rotStallDetCtrl NVM parameter:

- Frequency estimation: Estimated rotation frequency by the observer f_{est} is compared to the predefined threshold configuration parameter rotStallTooLowSpeedTh. When $abs(f_{est})$ reduces to less than rotStallTooLowSpeedTh, the stall state is detected immediately.
- Vq estimation: The estimated Vq is compared to a range defined by rotStallDetHighTh and rotStallDetLowTh. A stall condition is detected if the estimated Vq remains less than rotStallDetLowTh or greater than rotStallDetHighTh for longer than the duration specified by rotStallDetTimeTh. For

this estimate, it is necessary to set the values of the motor phase resistance sfocRs and the motor bEMF constant sfocKe .

- bEMF estimation: The estimated bEMF is compared to the predefined threshold $\text{rotStallTooLowVbEMFTh}$. If abs(bEMF) is less than abs(threshold) , the stall state is detected immediately. The higher the value of $\text{rotStallTooLowVbEMFTh}$, the faster the bEMF stall detection detects the stall. However, the chances of a false-stall detection increase.

After the first rotor stall is detected, the outputs are disabled, the controller enters the fail mode, and the fault is reported, pulling low the nFLT terminal and setting the corresponding fault flag in the SW status register. The controller remains in the fail mode until the stall condition is absent; this includes during retries. Once the rotor is stalled, the controller attempts to restart the motor for $\text{rotStartRetryMaxAttempts}$. Between retry attempts, a delay equal to the duration of rotStallRetryTOut occurs.

If the rotor is not in the stall state, the outputs are re-enabled automatically, the fault flag is cleared (not latched), and the controller transitions to the standby state. In the standby state, the state controller verifies that the command exceeds the cmdOnTh threshold, then it transitions to the windmill state. In the windmill

state a speed check is performed, and the device determines if start is to occur from a stationary position or with a moving rotor.

If the motor remains stalled after completion of all retry attempts, the controller checks whether the command is less than cmdOffTh . If the command is not less than cmdOffTh , the controller remains in the fail mode and the standby state, with the flag set in the SW status register. Otherwise, the fault and the flag are cleared, and the $\text{mca}_{\text{command}}$ mode transitions to the standby state.

NOTE: During open-loop operation (i.e., during ramp-up mode), stall detection is masked. After the controller transitions to closed-loop operation (i.e., the drive state) and the rotStallBlankDur period has elapsed.

NO-MOTOR START

The “no-motor” (i.e., motorless) start diagnostic detects a fault when a command to start the motor is applied and the motor is not connected to a board. The fault-handling mechanism is the same as that of the stall detection.

APPLICATION FEATURES

Gate Drive

A89337 can control the MOSFET gate charge and discharge without external components. The output slew rate can be estimated by the Q_{GD} specification of the MOSFET and the chosen gate-drive current according to:

Equation 19:

$$dt = Q_{GD} / (I_{SRC} \text{ or } I_{SNK})$$

Also, to avoid shoot-through current in the power stage, a dead time is implemented. This dead time begins after turn-off of the low side. Turn-on of the high side is delayed until after this dead elapses.

The desired dead time is programmable via the `pwmDeadTime` parameter according to:

Equation 20:

$$DeadTime [s] = pwmDeadTime \times t_{clk} [s]$$

where `pwmDeadTime` is configurable in NVM with values between 0 and 255.

Table 21: Gate Driver Configurations

<code>pwmGDSlewRate</code>	$I_{SRC} \text{ (mA)}$ (I_{HS})	$I_{SNK} \text{ (mA)}$ (I_{LS})
0	12.5	25
1	25	50
2	50	100
3	75	150

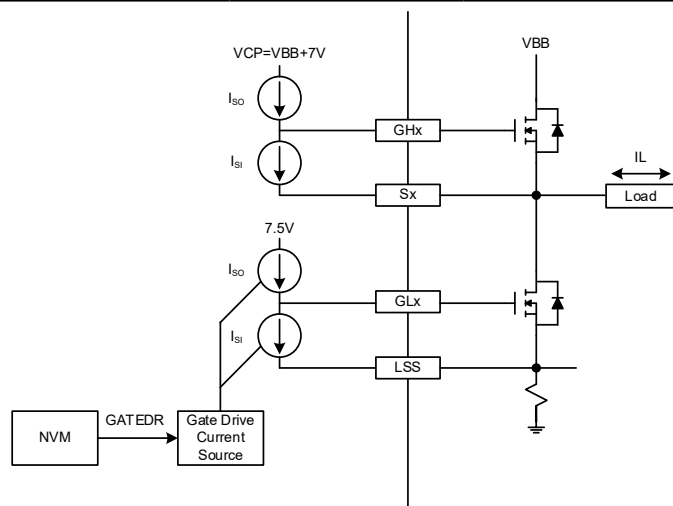


Figure 22: Gate Driver Diagram

FG/RD

The FG/RD terminal of the A89337 is configurable in the four modes described next. The FG and RD modes are mutually exclusive.

FG

When `enFGfunc` is set, a square wave representing the electrical rotation frequency is generated on the FG/RD terminal. The data from this square wave are used to obtain the motor speed according to:

Equation 21:

$$FG_{freq} [Hz] = \frac{0.00390625 \times (FG_{gain} + 1)}{FG_{pp} + 1} \times f_{elect} [Hz]$$

where `FGpp` and `FGgain` are NVM parameters and can assume integer values between 0 and 8, and 0 and 1023, respectively.

In a typical configuration, `FGgain` is set to 255 and `FGpp` is set to the number of motor pole pairs. In this configuration, the FG terminal toggles with electrical frequency normalized to $pp/2$.

After the FOC starts, during the ramp-up mode or after rotor synchronization (i.e., after the drive state), the FG signal can be generated according to the `FGStartType` parameter.

RD

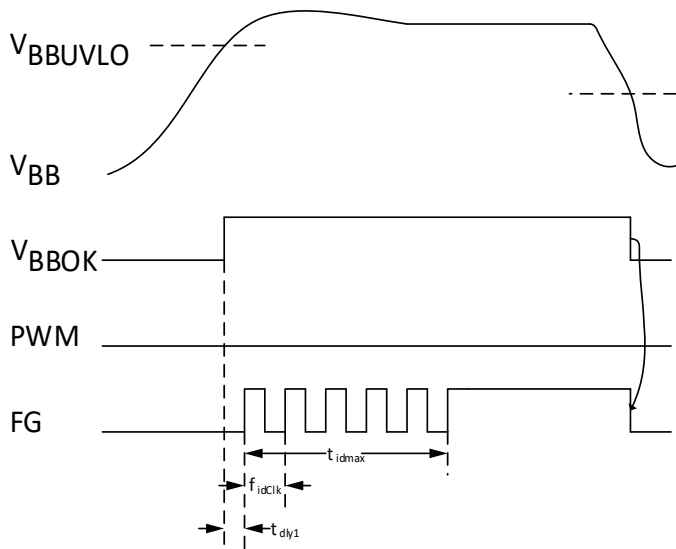
When the RD feature is enabled (`enRDfunc`), the stalled condition is reported on the FG/RD terminal according to the `RDActiveLevel` parameter in NVM:

- `RDActiveLevel = 0`: FG/RD terminal is set high in case of a stalled condition
- `RDActiveLevel = 1`: FG/RD is set low in case of a stalled condition

The terminal remains at an active level throughout the duration of the presence of the stalled latch

FG ID

If FG ID pattern generation (`enFGId`) is set, a programmable identification clock (ID clock) signal is provided on the FG/RD terminal upon startup. When the SPD/PWM terminal is low upon power-up of the A89337, this ID clock signal enables identification of a particular fan system. The ID clock (Figure 18) is available for a programmable time from 1 to 8 seconds, with resolution of 1 second or until the SPD/PWM terminal remains low.



The ID clock frequency (FGIdFreq) and duration (FGIdDur) are configurable in NVM.

Frequency is calculated as:

Equation 22:

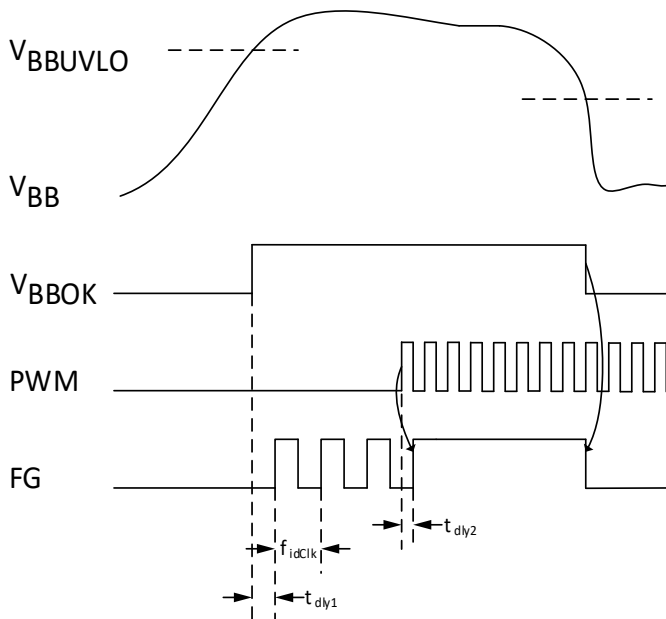
$$f_{(FG,ID)} = 300 + FGIdFreq \times 10 \text{ [Hz]}$$

Duration is calculated as:

Equation 23:

$$t_{(FG,ID,duration)} = FGIdDur + 1 \text{ [s]}$$

If configured, FG ID pattern generation occurs once during a power cycle after POR. If the duration of the command-above-threshold (cmdOnTh) parameter elapses, or if an undervoltage fault is present, FG ID pattern generation stops. If the controller has already entered the sleep state, this function is disabled.



$t_{dly1} = 100 \mu\text{s}$ maximum; $t_{dly2} = 100 \mu\text{s}$ maximum

Figure 23: ID Clock

FAKE FG

The fake FG feature extends the normal FG signal generation to compensate for manufacturing tolerances between motors of the same series. Such tolerances might cause variations in the FG output, even when the motors operate with identical settings. If two similar motors exhibit a speed difference of up to $\pm 25\%$, the fake FG function can be used to align the FG signals.

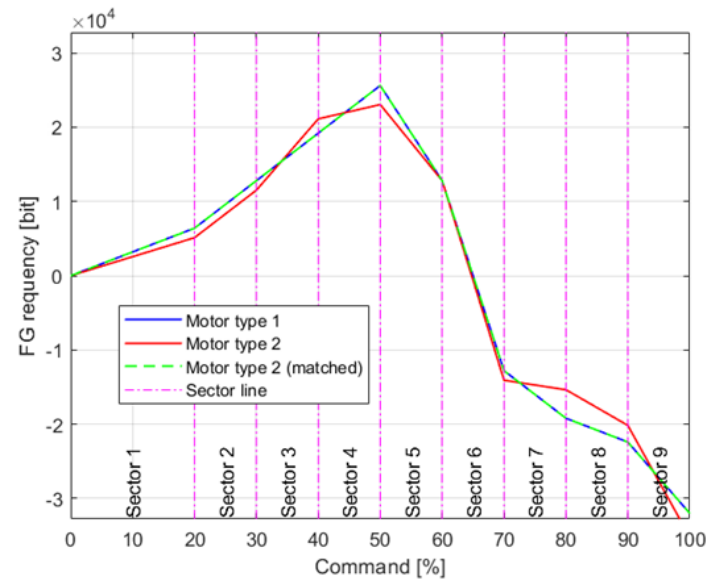


Figure 24: Fake FG

The feature modifies the FG signal in up to 10 sectors defined by the UCC curve configuration in use. To set the feature properly, the FG frequency difference between the two motors that are to match must be measured at the intersection line of the sector (i.e., the sector line). Gain can then be calculated as:

Equation 24:

$$Gain_1 = FG_{M1}/FG_{M2} = 6400/5120 = 1.25$$

Equation 25:

$$FakeFG1 = FG_{M2} \times Gain_1 \text{ (real-world units)}$$

The maximum allowable FG signal scaling is approximately $\pm 25\%$ (where gain_n is implemented as a signed 8-bit integer $[-128:127]$; as gain_n approaches the step of the gain parameter, scaling approaches $25.6/128 = 0.2\%$), according to:

Equation 26:

$$FakeFG1 = FG_{M2} \times (1000 + 2 \times FgFakeGain1)/1000$$

where gain is scaled to the range $[0.744:1.254]$.

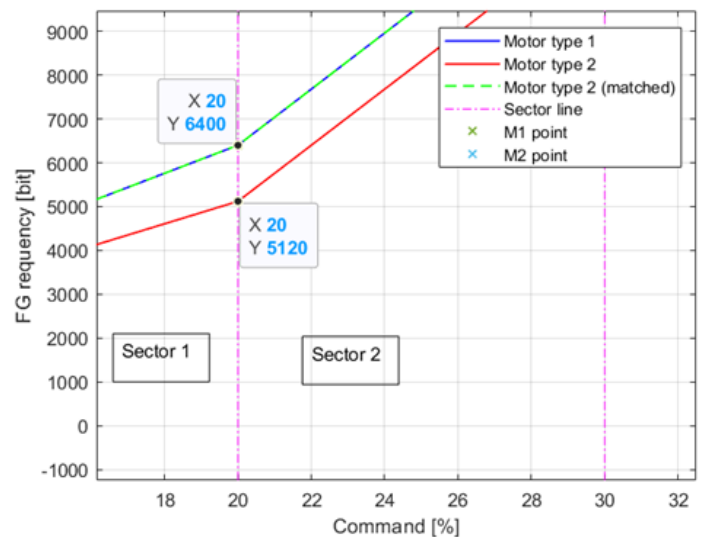


Figure 25: Fake FG Gain Measurement

FCOL

DESCRIPTION

Fan communication over legacy (FCOL) is an advanced, highly configurable operational mode implemented on the FG/RD terminal. This mode functions as a sophisticated, unidirectional serial interface from the controller to the master device. Data transmission occurs through the precise toggling of the FG/RD terminal, enabling the controller to send encoded data packets to the master. Data transfer is initiated by a predefined change in the external command of the master. The composition of these data packets is highly configurable, capable of containing both static data (e.g., preconfigured information such as manufacturer name or fan part number) and dynamic data (e.g., telemetry, including die temperature and I_{BB} current)

CONTROLLER OPERATION DURING FCOL TRANSACTION

During the FCOL initiation sequence, the controller maintains the previously established valid external command. Subsequently, upon the FCOL trigger event, the controller follows the external command change.

The `fcolExtCmdResponse` NVM setting governs the response of the controller to reference command changes during FCOL operation. If `fcolExtCmdResponse` is set to 1, the command change is applied immediately following the FCOL trigger. Conversely, if set to 0, the command change is deferred until the completion of the FCOL transaction.

The `fcolTrigTimeout` NVM setting defines the behavior of the controller concerning reference command changes in scenarios where an invalid FCOL trigger occurs (i.e., when the applied trigger duty cycle exceeds the configured timing threshold).

- If `fcolTrigTimeout` is set to 0, the trigger duty cycle is recognized as a valid command only after the configured total FCOL transaction time, which is calculated as:
 $(\text{fcolClk} \times \text{fcolTslice} \times 4 \times 8 \times \text{Nbytes} + \text{fcolAck} + \text{fcolTrigHighTh} + \text{fcolTrailer})$.
- If `fcolTrigTimeout` is set to 1, the trigger duty cycle is considered a valid command after the `fcolTrigHighTh` duration.

INITIATION OF DATA TRANSFER

The master device can initiate an FCOL data transfer by executing a preconfigured initiation action, referred to as a trigger condition. This trigger condition offers three selectable options, determined by the `fcolTrigDC` setting: a 95% to 100% duty-cycle range, a 10% to 15% duty-cycle range, or a precise 100% duty cycle without margin. The `fcolTrigLowTh` and `fcolTrigHighTh` NVM settings are utilized to define the timing margins for the selected trigger condition. FCOL initiation is inhibited if the trigger event concludes outside these defined timing margins. The trigger condition for the 95% to 100% duty-cycle range is shown in Figure 26.

ACKNOWLEDGE

Following the FCOL initiation sequence, the controller asserts a low state on the FG/RD terminal to provide an acknowledgment signal prior to the FCOL data transfer. The duration of this acknowledgment is programmable via the `fcolAckType` setting, allowing for a configurable duration of 750 milliseconds (when `fcolAckType` = 0) or 2 seconds (when `fcolAckType` = 1). An optional preamble, consisting of two 720-microsecond pulses, can be enabled by the `fcolPreambleEn` setting and is transmitted after 500 milliseconds (refer to Figure 27).

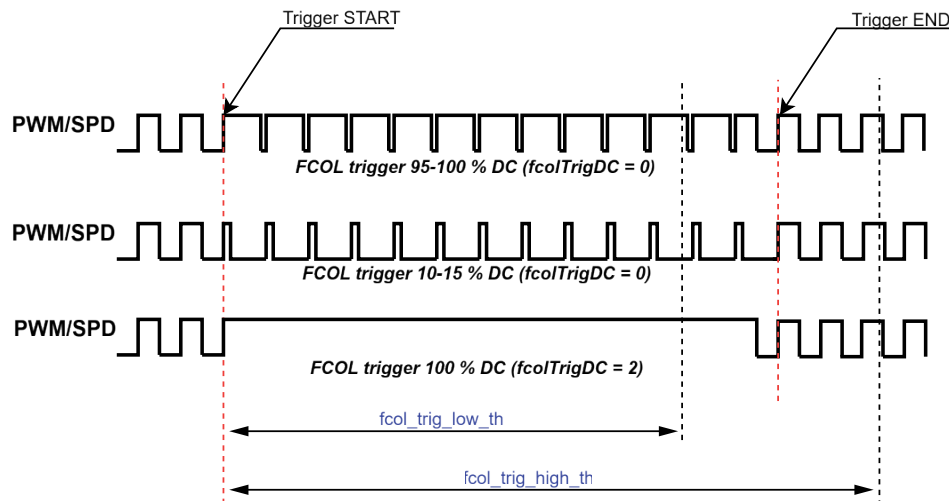


Figure 26: FCOL Trigger Condition Timing Diagram (95% to 100% Duty Cycle)

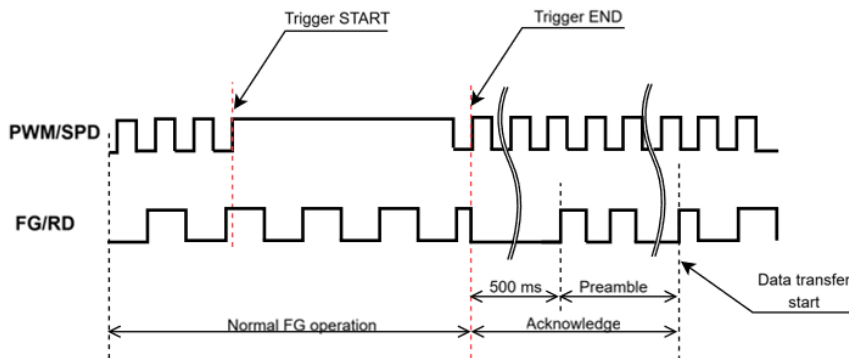


Figure 27: FCOL Preamble Timing Diagram

DATA TRANSFER AND BIT ENCODING

When the controller begins to transmit data, each data bit is encoded using four time periods of equal and configurable duration (T0, T1, T2, and T3; see Figure 28). The duration of time periods is controlled by fcolClk (rough setting) and fcolTslic (fine-grain setting). The byte-shift-out order can be set to msb first (fcoMSB = 1) or lsb first (fcolLSB = 0).

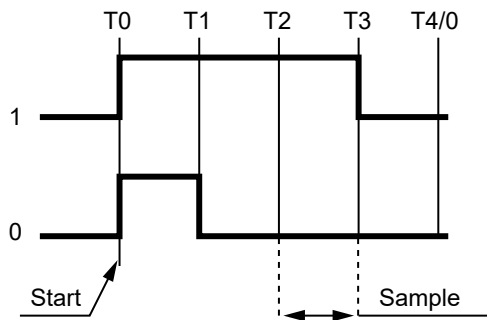


Figure 28: FCOL Bit-Encoding Timing Diagram

CONTENT OF DATA PACKAGE

The FCOL data content is configurable via the fcolDataType parameter, from A to E. The FCOL data type consists of flash and telemetry data in the order shown in Table 22 through Figure 25.

Table 22: FCOL Data Type A

Byte number	Data	Type	Description
0	Flash data	uint8	FCOL static data
1			
2			
3			
4			
5			
⋮			
58			
59			
60			
61			
62			
63			

Table 23: FCOL Data Type B

Byte number	Data	Type	Description
0	Flash data	uint8	FCOL static data
1			
2			
3			
4			
5			
⋮			
1494			
1495			
1496			
1497			
1498			
1499			

Table 24: FCOL Data Type C

Byte Number	Data	Type	Description
0	Flash data	uint8	FCOL static data
1			
2			
⋮			
45			
46			
47			
48	rpm	uint16	Rotation speed, rotations per minute
49			
50	P _{motor}	int16	P _{motor} signed, 14-bit $P_{motor} = ([value]/2^{14}) \times ((3/2) \times (FS \text{ Current } [A] \times 1.1547 \times (nominal \text{ voltage } [V])/2)) [W]$
51			
52	I _{BB}	int16	I _{BB} current, signed, 11-bit resolution: $I_{BB} = 1.5 \times ([value]/2^{11}) \times (0.5 \times 1.1547) \times (FS \text{ current } [A]) [A]$
53			
54	V _{BB}	uint16	V _{BB} voltage: $V_{BB} = ([value]/2^{14}) \times (nominal \text{ voltage } [V]) [V]$
55			
56	Temperature	uint16	Code, proportional to IC temperature (internal): $T = 1.12 \times [value]/(2^7) - 6.2, ^\circ C$
57			
58:61	Reserved		
62	CRC	uint16	CRC can also be configured as uint8, located in address 63 only
63			

Table 25: FCOL Data Type D

Byte Number	Data	Type	Description
1	Flash data	uint8	FCOL static data
2			
3			
⋮			
46			
47			
48			
49	rpm	uint16	Rotation speed, rotations per minute
50			
51	P _{motor}	int16	P _{motor} signed, 14-bit $P_{motor} = ([value]/2^{14}) \times ((3/2) \times (FS \text{ Current } [A] \times 1.1547 \times (nominal \text{ voltage } [V])/2)) [W]$
52			
53	I _{BB}	int16	I _{BB} current, signed, 11-bit resolution: $I_{BB} = 1.5 \times ([value]/2^{11}) \times (0.5 \times 1.1547) \times (FS \text{ current } [A]) [A]$
54			
55	V _{BB}	uint16	V _{BB} voltage: $V_{BB} = ([value]/2^{14}) \times (nominal \text{ voltage } [V]) [V]$
56			
57	Temperature	uint16	Code, proportional to IC temperature (internal): $T = 1.12 \times [value]/(2^7) - 6.2, ^\circ C$
58			
59	Reserved		
60			
61			
62	CRC	uint16	CRC can also be configured as uint8, located in address 63 only
63			

Table 26: FCOL Data Type E

Byte Number	Data	Type	Description
1	Flash data	uint8	FCOL static data
2			
3			
⋮			
45			
46			
47			
48	rpm	uint16	Rotation speed, rotations per minute
49			
50	Reserved	uint8	
51	Fan ID	uint8	Fan ID read from flash (static data)
52			
53			
54	Reserved	uint8	
55	IBB	uint8	I_{BB} current, signed, 11-bit resolution: $I_{BB} = 1.5 \times ([value]/2^{11}) \times (0.5 \times 1.1547) \times (FS \text{ current } [A]) [A]$
56	Reserved	uint8	
57			
58			
59			
60			
61			
62	CRC	uint16	CRC can also be configured as uint8, located in address 63 only
63			

TRAILER

The trailer delay occurs after FCOL data transfer ends and before the normal FG operation and optional trailer can be configured. the parameter fcolTrailer has a 100 ms step, and a range from 0 through 1500 ms.

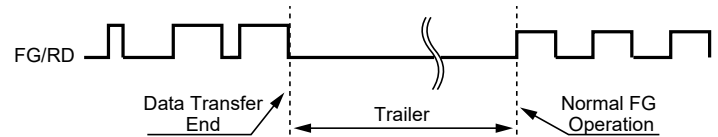


Figure 29: FCOL Trailer Delay Diagram

FCOL FLASH DATA PROGRAMMING

Data for a corresponding FCOL type needs to be loaded into nonvolatile e-flash memory. The flash address region is 0x0001100:0x000016DC. To be able to load data, user mode must be enabled.

AC Power Loss

The AC power loss function is intended for operation when the system switches to battery power due to the loss of the AC mains supply. In this condition, the motor continues to operate with a reduced I_{BB} level while maintaining the required speed.

When the AC power loss feature is enabled by `ACPwrlossEn`, a trigger on the PWM input signal is required. The expected trigger is the PWM input signal pulled low for the time configured by `ACPwrlossStartDur`. In case of a PWM signal that is stuck low for a duration longer than `ACPwrlossOffDur`, the AC power loss operation is exited. After the PWM trigger is applied, PWM toggling (duty cycle) must resume.

The current-limiting of I_{BB} is achieved by the Q-axis current limitation configurable via the `ACPwrlossQCurrLmt` parameter.

The device remains in the AC power loss mode for the duration configured by the `ACPwrlossReleaseSel` (if it is greater than 0). In the case of `ACPwrlossReleaseSel = 0`, the device remains in the mode upon receipt of an external command that is greater than the current FOC frequency.

This feature is functional only if the external command source is the PWM duty cycle.

Power Fan

The power fan feature allows the motor speed to increase from 100% through 170% of the nominal speed when required, while maintaining the current rotation direction. The A89337 variant of the IC includes the power fan function (power fan is always enabled), and the A89337-1 variant does not have the power fan function. The power fan function uses the `nBRAKE` (active low) input terminal as the control input. When the input is not active, the motor response to the UCC curve is normal.

When the trigger input is active, the motor speed changes from the speed curve level to a speed determined by:

Equation 27:

$$\text{Speed} = \text{MaxCloseLoopSpeed} \times \text{Gainfactor}$$

where:

- `MaxCloseLoopSpeed` is the maximum speed found on the UCC curve
- `Gainfactor` is $(\text{powerFanGainFact}+10)/10$; gain factor range is 100% to 170%
- `PowerFan` is active only when turn-on of the motor is commanded.

NOTE:

- If the power fan input terminal (`nBRAKE`) is low, the PWM input is ignored.
- Because the power fan function uses the `nBRAKE` and `nFLT` terminals, it is not possible to configure the A89337 variant to use those terminals for I^2C .

Forced OL

The forced OL feature allows the A89337 to switch from speed control to power or torque Control when the external command exceeds a configurable threshold. At greater than this threshold, the reference can either be set to a predefined power or torque value, or the reference can be derived from the applied PWM duty cycle or I²C command.

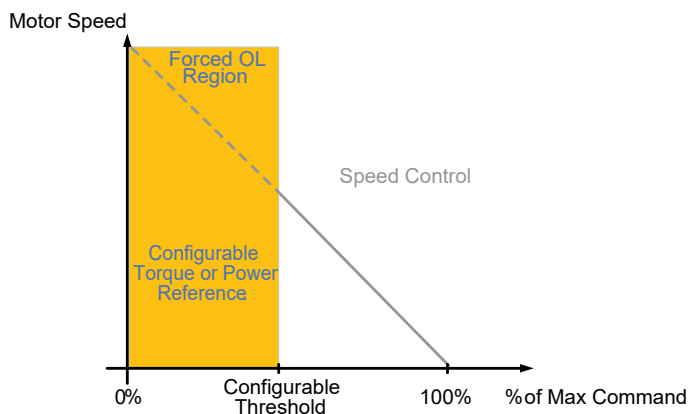
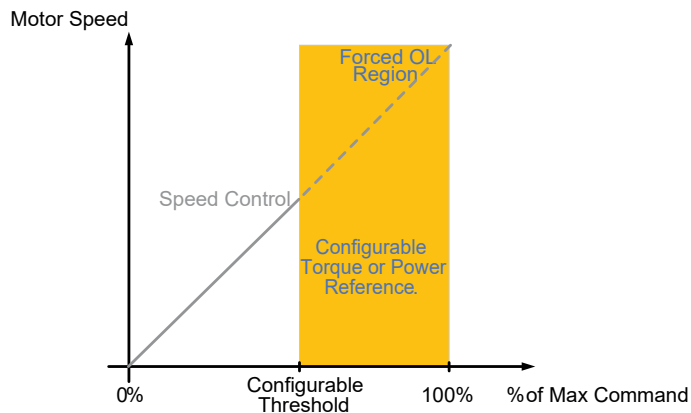


Figure 30: Forced OL

Brake on Fault

A brake-on-fault feature allows application of the brake sequence when the listed configurable fault is present:

- Overtemperature protection (OVT): brakeIfFaultOvtEn
- V_{CP} undervoltage lockout (UVLO): brakeIfFaultVcpUvloEn
- Loss of synchronization (lock detect): brakeIfFaultLockDetEn
- Overcurrent protection (OCP): brakeIfFaultOcpEn
- No-motor start: brakeIfFaultNoMotorStartEn
- V_{BB} overvoltage protection (V_{BB} OVP): brakeIfFaultVbbOvpEn

In the fail state, low-side FET braking is applied for a duration configured in NVM:

- brakeFETPulseAmount: Brake by FET pulses amount, 2^(N+1)
- brakeFETonDuration: Brake by duration of on-state of FET, where the duration equals N updates of V_{CL}.
- brakeFEToffDuration - Brake by FET Off duration; $N \times \text{brakeFETonDuration}$

Braking is applied for a duration configured in NVM and continues until the fault is cleared. If the braking sequence completes while the fault condition is still present, the braking sequence is repeated.

For faults that support a retry mechanism (such as no-motor start or lock detection), braking temporarily stops during the retry attempt.

Quiet Start

To achieve a quieter OL-CL transition, the A89337 implements a quiet-start feature. Functionally, this feature is enabled by setting the sfocQuietStartEn bit to 1 in NVM.

In the A89337, the phase current limit at the OL-CL transition is configurable. It is also possible to gradually reduce the current during the ramp-up stage (sfocRupDcurrRef), toward the end of the open-loop ramp-up period, by properly configuring the OL-CL transition duration. When the current level is closer to the required level for feedback control, the transition noise can be minimized.

For more information about this advanced feature, refer to the A89337 Evaluation Board User Guide.

BEMF Compensation

The A89337 implements a bemf compensation function to improve acoustic performance and control efficiency. This feature is particularly useful for motors with strong cogging and non-sinusoidal bemf characteristics, which can generate harmonic components when the pure sinusoidal voltages produced by the FOC algorithm are applied.

If compensation for these bemf harmonics is not implemented, these bemf harmonics can introduce significant current ripple. This ripple leads to inefficient use of energy, where the controller attempts to demagnetize the permanent magnets and results in torque ripple that causes rotor oscillations and vibrations. These vibrations can generate acoustic noise and might lead to premature wear or damage to the ball bearings.

For more information about this advanced feature, refer to the A89337 Evaluation Board User Guide.

Power-Loss Brake (PLB)

The A89337 detects power loss by monitoring the VIN terminal, and it can apply a braking function to slow the speed of the motor.

When the motor is spinning, the generated bemf voltage is rectified through the body diodes of the output DMOS devices and charges the V_{BB} supply capacitor. If sufficient voltage is stored on the V_{BB} capacitor, a turn-on of the low-side MOSFETs can be used to generate a braking force on the motor.

As the motor speed decreases, the generated bemf voltage also decreases. Eventually, the voltage on the V_{BB} line becomes insuff-

ficient to power the low-side MOSFETs, at which point continued application of the braking force is not possible, and the motor continues to coast.

The A89337 PLB circuitry boosts the voltage on the V_{BB} line while the motor is spinning by periodically pulsing the motor winding to the off state. During braking, if the motor is still rotating, current flows through the motor winding. This current can be used to charge the V_{BB} capacitor and to increase the V_{BB} voltage by briefly disabling the brake mode, operating similarly to a hysteretic boost converter. As a result, the IC is able to continue to brake the motor at a much lower speed than would otherwise be possible.

Table 27: PLB Events

VIN [1]	VBB	Mode of Operation	Notes
Low	< V _{BBTH}	Coast: No braking	
Low	> V _{BBTH} and < V _{BOOST}	Brake: PWM	If the motor spins fast enough, V _{BB} ramps up to V _{boost}
Low	= V _{BOOST}	Brake: Hysteretic boost	V _{BB} is regulated by V _{boost}
Low	> V _{BOOST}	Brake	V _{BB} decays depending on I _{BB2} and V _{BB} capacitance
High	< V _{BBUVLO}	Coast: No braking	Power-up or power-down
High	> V _{BBUVLO}	Run mode	V _{DIG} powers up logic; if PWM signal valid, motor starts

[1] V_{IN} < V_{INTH} = Low, V_{IN} > V_{INTH} = High

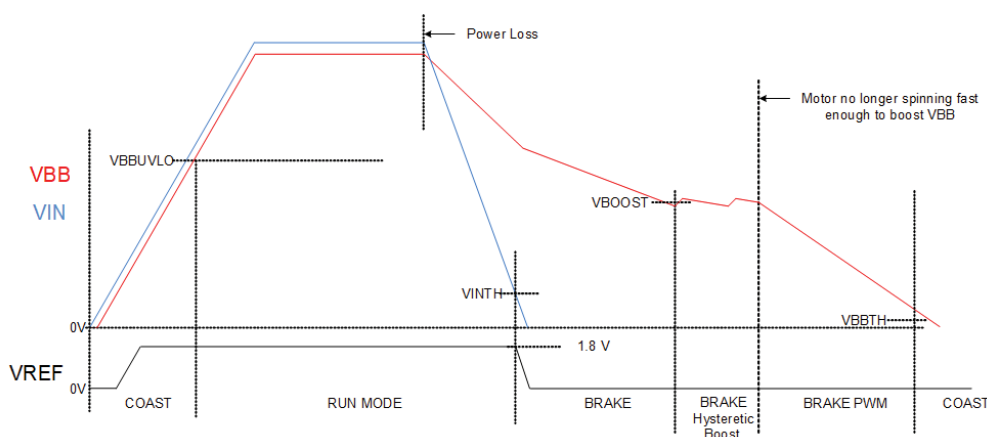


Figure 31: PLB Processing Diagram

No-Power Brake (NPB)

Braking can also be applied when external power is not supplied to the module and the motor is driven by an external force (wind-mill condition), as shown in Figure 32. In this case, the bEMF voltage generated by the rotating motor charges the V_{BB} line. Once the V_{BB} voltage exceeds the V_{BBTH} threshold, activation of the braking function is possible.

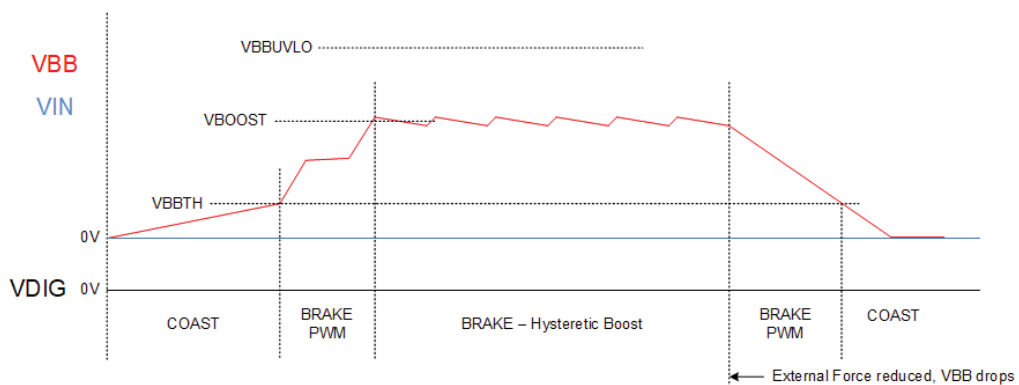


Figure 32: NPB Processing Diagram

PART CONFIGURATION**Registers and Nonvolatile Memory (NVM)**

A89337 uses NVM to store configuration data. Configuration data can be adjusted for a particular application. Upon power up or wake-up from the sleep state, the contents of the NVM (e-flash) are copied into the shadow NVM registers (RAM) before they are accessed by logic circuits and used to control the motor operation. Direct access to the shadow registers is also allowed via the I²C interface. A write to the shadow registers typically changes the operation of the driver immediately (for some setting to take effects, the A89337 must be commanded to the standby state and the motor must be restarted; i.e., some settings require POR). Changes to NVM do not immediately affect device operation; rather, a power cycle of the device is required or the must wake from the sleep state before changes to the NVM affect device operation.

REGISTERS

User registers are shown in Table 28. Registers not shown in the table are not for user access. A change to a value in an undocumented registers might cause the device to malfunction or damage to the part.

Table 28: User Registers

I2C Address	Register Name	Bits	RO/RW [1]	Default Value	Description
0x0000	SW_STATUS	31:0	RO	0x0	Software-defined status bits
0x0004	WRITE_REF_CMD	16:0	RW	0x0	Motor reference command and direction when external command source is I ² C
0x0008	READ_REF_CMD	15:0	RO	0x0	Motor reference command (speed, torque, power)
0x000C	BRAKE_CTRL	0:0	RW	0x1	If read = 0, the brake is applied (via register or via the nBRAKE terminal) If read = 1, the brake is not applied If write = 0, the brake is applied (the nBRAKE terminal is bypassed) If write = 1: the brake is controlled by the nBRAKE terminal

[1] RO = read only; RW = read and write.

Table 29: SW Status Register Description

Address 0x0000, read-only			
Variable Name	Bits	Size	Description
CONTROLLER MODE			
MCA MODE	3:0	4 bits	Motor-control mode 0 = OFF_MC_MODE 1 = STANDBY_MC_MODE 2 = ROTATE_MC_MODE 3 = BRAKE_MC_MODE 4 = FAIL_MC_MODE
CONTROLLER STATE			
MCA STATE	7:4	4 bits	Motor-control finite state machine (FSM) state 0 = MCA_FSM_IDLE 1 = MCA_FSM_STANDBY 2 = MCA_FSM_WINDMILL 4 = MCA_FSM_FAIL 5 = MCA_FSM_IPD 6 = MCA_FSM_ALIGN 7 = MCA_FSM_RAMP 8 = MCA_FSM_DRIVE 9 = MCA_FSM_SLEEP 10 = MCA_FSM_BRAKE_FROM_DRIVE

Address 0x0000, read-only			
Variable Name	Bits	Size	Description
FAULT STATUS			
OV	8	1 bit	Overvoltage
OT	9	1 bit	Overtemperature
OC	10	1 bit	Overcurrent latch
EE	11	1 bit	PDSP ROM or e-flash error
RS	12	1 bit	Rotor stalled latch
NMS	13	1 bit	No-motor start detection
AV	14	1 bit	Analog voltage error
SYNC_LOST	15	1 bit	CLK compensation invalid (lost synchronization)
UVP	16	1 bit	V _{BB} undervoltage error
UVLO	17	1 bit	V _{CP} undervoltage lockout error
VREG	18	1 bit	V _{REG} error
OK	19	1 bit	System OK
RESV	20	1 bit	Reserved
ROM_BIST_FAIL	21	1 bit	ROM cyclic redundancy check (CRC) failure flag
RAM_BIST_FAIL	22	1 bit	RAM built-in self-test (BIST) failure flag
BIST_DONE	23	1 bit	BIST completed flag
MOTOR CONTROL STATUS			
VQ_LIM	24	1 bit	V _q limit
VD_LIM	25	1 bit	V _d limit
SPD_LIM	26	1 bit	Speed limit
PWR_LIM	27	1 bit	Power limit
VGEN_LIM	28	1 bit	Voltage (generator mode) limit
OVV_LIM	29	1 bit	Overvoltage variable limit
ROT_SYNC	30	1 bit	Rotor synchronized
RESV	31	1 bit	Reserved

Table 30: Write Reference Command Register

Address 0x0004, read-write			
Variable Name	Bits	Size	Description
REF_CMD_W	15:0	16 bits	Motor reference command when external command source is I ² C
DIR	16	1 bit	Motor spin direction

NVM

For a detailed explanation of the NVM registers, refer to the A89337 evaluation board user manual.

NVM Programming

SHADOW NVM PROGRAMMING

While temporary change in configuration settings is required, the shadow config NVM configuration in RAM can be changed. The value of the shadow NVM is a copy of the NVM values stored in flash memory (the copy was created after POR).

Table 31: Shadow Config NVM Address

Name	Start Address	End Address
Shadow Config NVM	0x0000 0EA4	0x0000 0F40

FLASH NVM PROGRAMMING

Before permanently programming NVM register values in the e-flash memory, the shadow configuration NVM must be properly configured. The MPC1 then allows the shadow configuration NVM to be copied into the e-flash configuration NVM. For more information, refer to the Multipurpose Control Interface (MPC1) section.

e-Flash Programming*User Register***Table 32: User Registers**

I2C/Address	Register Name	Bits	R/W	Default Value	Description
0x0104	HW_PASSWORD_CTRL	31:0	ro	0x0	register to enter user mode (UM)
0x2FF4	EFLASH_CFG	16:0	rw	0xE0	e-flash configuration register
0x2FF8	EFLASH_STATUS	15:0	ro	0x0	e-flash status register
2FFC	HW_PASSWORD_STATUS	0:0	rw	0x1	Mode status register: • 0 = application mode • 1 = user mode

*HW_PASSWORD_CTRL Register***Table 33: HW_PASSWORD_CTRL Register**

Address 0x0104, read-write			
Variable name	Bits	Size	Description
RESERVED	31:16	16	
HW_PASSWORD_CTRL	15:0	16	Write password to enter user mode

*EFLASH_CFG Register***Table 34: EFLASH_CFG Register**

Address 0x2FF4, read-write			
Variable Name	Bits	Size	Description
reserved	31:7		
CUSTOMMER_LOCK	6:6	1	Lock/Unlock of read operation to the [0x0000:0x1000] relative addresses within e-flash 1 = locked 0 = unlocked NOTE: The address range [0x0000:0x1000] is not the absolute address of the entire IC
CUSTOMMER_WLOCK	5:5	1	Lock/Unlock of program/erase operations to the [0x0000:0x1000] relative addresses within e-flash 1 = locked 0 = unlocked NOTE: The address range [0x0000:0x1000] is not the absolute address of the entire IC
INIT_REQ	4:4	1	Write 1 to start the initialization procedure
RESERVED	3:3	1	
ERASE_CFG	2:1	2	0 = program cycle 1 = erase page 2 = erase sector 3 = erase MAT
ELASH_EN	0:0	1	Enable for e-flash memory 0 = power-down 1 = enabled

*EFLASH_STATUS Register***Table 35: EFLASH_STATUS Register**

Address 0x2FF8, read-only			
Variable Name	Bits	Size	Description
RESERVED	31:2		
STATUS	1:0	1	0 = power-down mode 1 = ready/standby 2 = busy

*HW_PASSWORD_STATUS Register***Table 36: HW_PASSWORD_STATUS register**

Address 0x2FFC, read-write			
Variable Name	Bits	Size	Description
RESERVED	31:16	16	
HW_PASSWORD_STATUS	15:0	16	0 = application mode 1 = user mode

e-Flash CRC-32 Calculation

GENERATOR POLYNOMIAL

The 32-bit cyclic redundancy check (CRC) value is calculated using a linear-feedback shift register (LFSR) implementation based on the following generator polynomial:

Equation 28:

$$(x) = x^{32} + x^{31} + x^{29} + x^{28} + x^{26} + x^{25} + x^{24} + x^{23} + x^{21} + x^{20} + x^{19} + x^{18} + x^{16} + x^{15} + x^{11} + x^{10} + x^9 + x^6 + x^0$$

The corresponding hexadecimal representation of the polynomial is:

Equation 29:

$$\text{Polynomial (hex)} = 0x973AFB51$$

CRC COMPUTATION ALGORITHM

1. CRC register: $\text{CRC} \leftarrow 0$
2. Process each input byte b:
 - Align the byte into the most-significant 8 bits of the 32-bit CRC variable: $\text{CRC} \leftarrow \text{CRC} \oplus (b \ll 24)$
 - For each bit (8 iterations): If the most significant bit of $\text{CRC MSB}(\text{CRC}) = 1$: $\text{CRC} \leftarrow (\text{CRC} \ll 1) \oplus P(x)$; else, $\text{CRC} \leftarrow (\text{CRC} \ll 1)$
3. Final result: After processing of all input bytes completes, the remaining value in the CRC register is the final 32-bit CRC result.

MATHEMATICAL REPRESENTATION

For a message consisting of bytes $m_0, m_1, \dots, m_{(n-1)}$, the CRC variable evolves as:

Equation 30:

$$\text{CRC} = (\oplus_{j=0}^{n-1} (m_j \times x^{24j})) \times x^{8j} \bmod P(x)$$

where:

- $\oplus$ denotes the bitwise XOR operation
- $\times$ represents a left-shift operation
- the modulus operation is performed with respect to the generator polynomial $P(x)$, where $\bmod P(x)$ denotes the polynomial modulus operation with respect to the generator polynomial $P(x)$.

Flash Operations

Access to a flash operation requires specific protection bits to be unlocked:

- A write or erase operation requires bit 5 (customer_wlock) to be cleared.
- A read operation requires bit 6 (customer_lock) to be cleared.
- All operations must begin by enabling and initializing the flash.
- After completion, the flash should be locked and disabled to ensure protection and reduce power consumption.

The typical sequence for accessing e-flash memory is illustrated in Figure 33. The flash must first be enabled and initialized before unlocking the read or write protection. After the required erase, program, or read operation completes, the flash should be locked again and disabled.

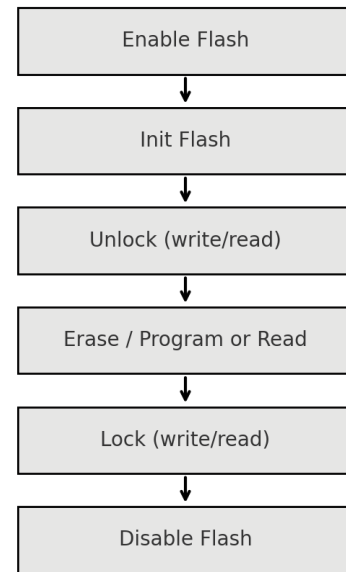


Figure 33: e-Flash Operation Flow

FLASH READ PROCEDURE

1. Enable flash: Set EFLASH_EN = 1 (bit 0).
2. Initialize flash: Set INIT_REQ = 1 (bit 4).
3. Unlock read access: Clear CUSTOMER_LOCK (bit 6 = 0).
4. Read data: Sequentially read flash memory starting from 0x1100 (relative start address).
5. Lock read access: Set CUSTOMER_LOCK = 1 (bit 6).
6. Disable flash: Clear EFLASH_EN = 0 (bit 0).

FLASH WRITE PROCEDURE

1. Enable flash: Set EFLASH_EN = 1 (bit 0).
2. Initialize flash: Set INIT_REQ = 1 (bit 4); wait for the initialization sequence to complete.
3. Unlock program/erase operations: Clear CUSTOMER_WLOCK (bit 5 = 0).
4. Erase page: Set ERASE_CFG = 01 (bits [2:1]); issue erase by writing 0x00000000 to each word in the target page address range.
5. Program cycle: Set ERASE_CFG = 00 (bits [2:1]); write data words to the desired flash addresses; write a CRC word following the programmed data.
6. Lock program/erase: Set CUSTOMER_WLOCK = 1 (bit 5).
7. Disable flash: Clear EFLASH_EN = 0 (bit 0).

Flash Address Map

Table 37: Flash Address Map

Region	Start Address	End Address	Description
Config NVM	0x0000 1000	0x0000 109B	Configuration NVM parameters
Reserved	0x0000 109C	0x0000 10FF	Reserved (do not write)
FCOL Data	0x0000 1100	0x0000 16FF	FCOL static data

PACKAGE OUTLINE DRAWING

For Reference Only – Not for Tooling Use

(Reference Allegro DWG-0000222 Rev. 4 or JEDEC MO-220WGGD.)

Dimensions in millimeters – NOT TO SCALE.

Exact case and lead configuration at supplier discretion within limits shown.

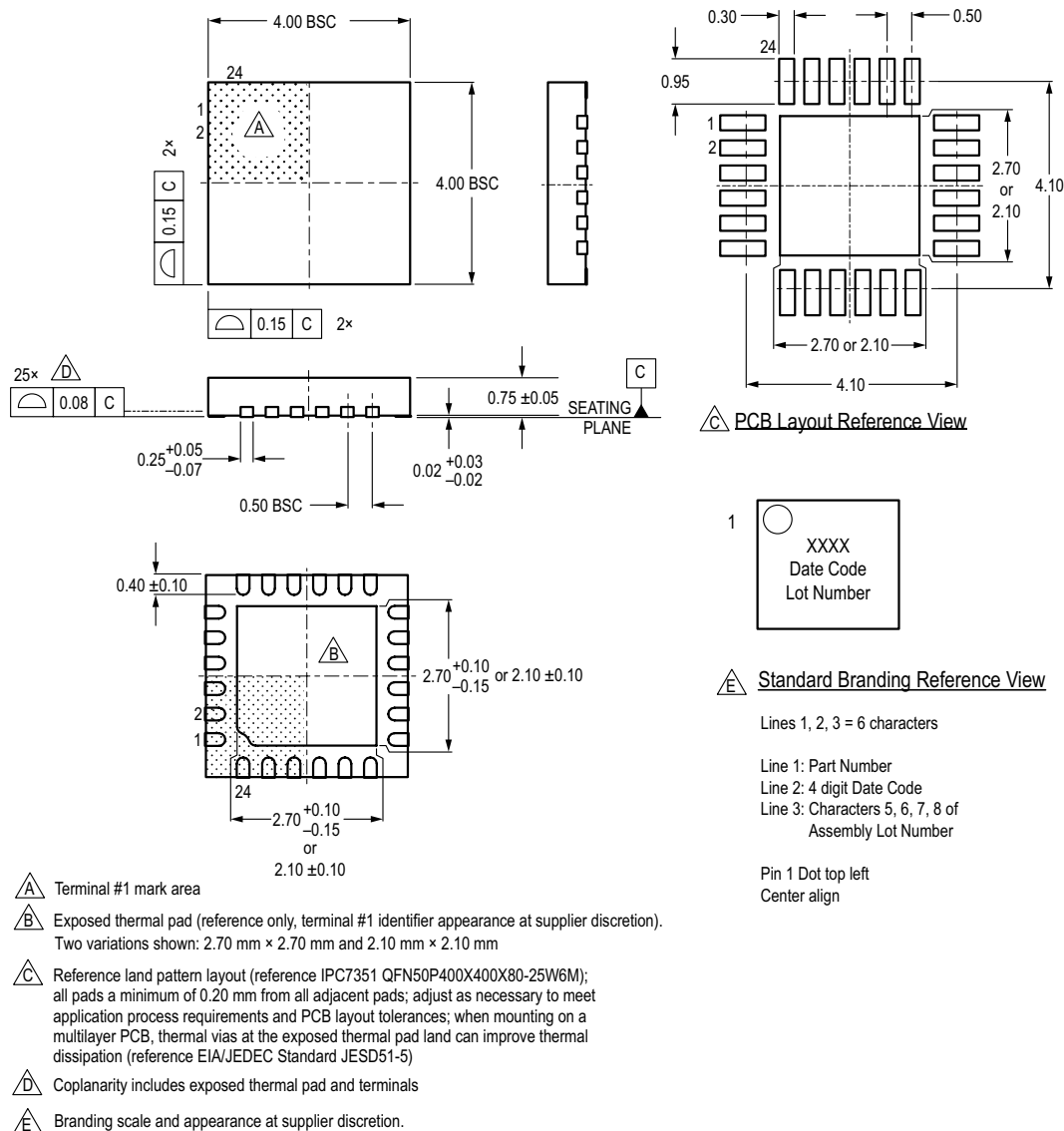


Figure 34: 24-Contact QFN, 4 mm x 4 mm, 0.75 mm Thickness, with Exposed Thermal Pad (Suffix ES)

REVISION HISTORY

Number	Date	Description
–	March 25, 2026	Initial release to web of short-form datasheet and initial release of limited distribution long-form datasheet.
1	March 26, 2026	Added A89337-1 without power fan function (headings and pages 2 and 40)
2	August 13, 2026	Initial release to web of long-form datasheet.

Copyright 2026, Allegro MicroSystems.

Allegro MicroSystems reserves the right to make, from time to time, such departures from the detail specifications as may be required to permit improvements in the performance, reliability, or manufacturability of its products. Before placing an order, the user is cautioned to verify that the information being relied upon is current.

Allegro's products are not to be used in any devices or systems, including but not limited to life support devices or systems, in which a failure of Allegro's product can reasonably be expected to cause bodily harm.

The information included herein is believed to be accurate and reliable. However, Allegro MicroSystems assumes no responsibility for its use; nor for any infringement of patents or other rights of third parties which may result from its use.

Copies of this document are considered uncontrolled documents.

For the latest version of this document, visit our website:

www.allegromicro.com