Design Guide: TIDA-010250 **1kW BLDC Motor Inverter Reference Design**



Description

This reference design illustrates a motor inverter with MSPM0G1507, an Arm[®] Cortex[®]-M0+ core microcontroller. The design not only supports a sensorless Field Orientation Control (FOC) algorithm with 1–3 shunt resistors, but also a sensored motor driving algorithm with a Hall-effect Sensor or Quadrature Encoder Interface (QEI) sensor. The hardware and software available with this reference design are tested and ready-to-use to help accelerate development time to market.

Resources

Design Folder
Product Folder

Features

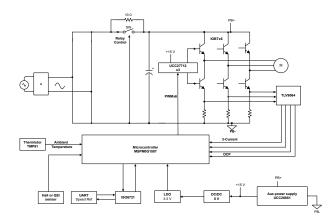
- Sensorless FOC algorithm motor control with 1, 2, or 3 shunt resistors
- · Sensored motor control with Hall-effect or QEI
- Up to 1kW inverter stage
- Robust design with overcurrent and overvoltage protections

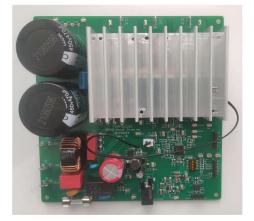
Applications

- Refrigerator and freezer
- Washer and dryer
- Mixer, blender, and food processor
- Cooker hood
- Air conditioner indoor unit



Ask our TI E2E[™] support experts





1

1 System Description

Today's motor control for major appliance applications must meet a growing list of demands on lower cost, smaller size, more power and higher energy efficiency. Brushless Direct Current (BLDC) motor and Permanent Magnet Synchronous Motor (PMSM) is becoming increasingly popular in major appliance applications. With the help of high-speed Arm Cortex-M0+ core and rich analog peripherals, the MSPM0G1507 microcontroller provides a true single-chip solution for sensored or sensorless motor control, which helps designers to reduce the number of components in the bill of materials and to design a compact, cost-effective motor inverter.

1.1 Key System Specifications

PARAMETERS	TEST CONDITION	MIN	NOM	MAX	UNIT	
Input Voltage		165	230	265	VAC	
Input Frequency		47	50	63	HZ	
PWM Switching Frequency (f_{SW})			8		kHz	
Rated Power Output (P _{OUT})			1		kW	
Output Current (I _{RMS})			3		А	
Inverter Efficiency			97		%	
Motor Electrical Frequency		30	100	200	Hz	
Fault Protections	Overcurrent, overvoltage, overte	mperature, und	ervoltage	1	1	
Drive Control Method	Sensored motor control with Hall-effect or QEI; Sensorless motor control with 1, 2, or 3 shunt resistors					
Operating Ambient	Open Frame	-10 25		55	°C	
Built-in Auxiliary Power Supply	V _{INAC} = MIN to MAX	15V ±10%, 200mA				
Board Size	140mm × 140mm × 40mm	1				

WARNING

TI intends this reference design to be operated in a lab environment only and does not consider the reference design to be a finished product for general consumer use.

TI Intends this reference design to be used only by qualified engineers and technicians familiar with risks associated with handling high-voltage electrical and mechanical components, systems, and subsystems.

High voltage! There are accessible high voltages present on the board. The board operates at voltages and currents that can cause shock, fire, or injury if not properly handled or applied. Use the equipment with necessary caution and appropriate safeguards to avoid injuring yourself or damaging property.

Hot surface! Contact can cause burns. **Do not touch!** Some components can reach high temperatures > 55°C when the board is powered on. The user must not touch the board at any point during operation or immediately after operating, as high temperatures can be present.

CAUTION

Do not leave the design powered when unattended.



2 System Overview

2.1 Block Diagram

Figure 2-1 shows the reference design block diagram.

The entire system is represented in the following blocks:

- Auxiliary power supply
- MCU controller
- 3-phase inverter
- · Phase current sensing with shunt resistors
- Hall-effect or QEI interface
- Isolated universal synchronous or asynchronous, receiver or transmitter (USART) serial communication

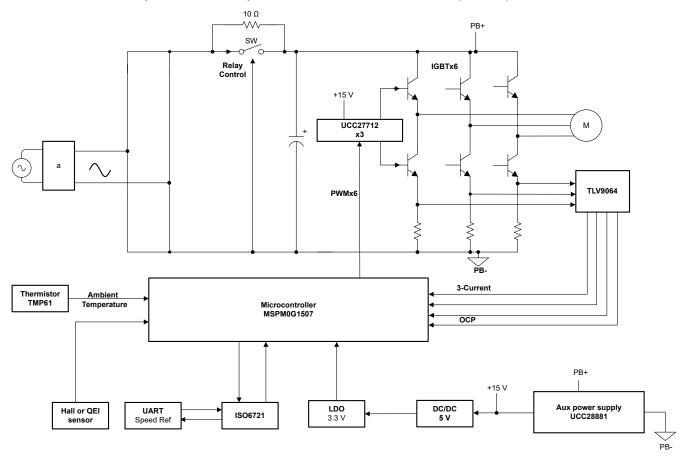


Figure 2-1. Block Diagram of the Reference Design



2.2 Design Considerations

This system starts from a diode bridge with electrolytic capacitors to create a stable DC bus. A simple UCC28881-based high-voltage buck supply provides non-isolated 15V to the gate driver UCC27712 to drive insulated-gate bipolar transistors (IGBT).

The TPS54202 DC/DC controller steps down 15V to 5V, and the TLV74033 provides a low-ripple 3.3V to power on the microcontroller as well as to be a voltage reference for an analog-to-digital converter (ADC).

The MSPM0G1507 microcontroller delivers six PWMs to gate drivers for IGBTs to control the current flow to the motor, the device also senses the DC bus voltage, ambient temperature, Hall-effect or QEI signals, and phase currents to implement the motor control algorithm.

2.3 Highlighted Products

The following highlighted products are used in this reference design. Key features for selecting the devices for use in this reference design are revealed in the following sections. Find more details of the highlighted devices in the respective product data sheet.

2.3.1 MSPM0G1507

The MSPM0G1507 microcontrollers (MCUs) are part of the highly-integrated, ultra-low-power, 32-bit MCU family from mixed-signal processing (MSP) based on the enhanced Arm® Cortex®-M0+ 32-bit core platform operating at up to 80MHz frequency. These cost-optimized MCUs offer high-performance analog peripheral integration, support extended temperature ranges from -40°C to 125°C, and operate with supply voltages ranging from 1.62V to 3.6V. The MSPM0G1507 devices provide up to 128KB embedded Flash program memory with built-in error correction code (ECC) and up to 32KB SRAM with hardware parity option. The devices also incorporate a memory protection unit, seven-channel DMA, math accelerator, and a variety of high-performance analog peripherals such as two 12-bit 4-MSPS ADCs, configurable internal shared voltage reference, one 12-bit 1MSPS DAC, three high-speed comparators with built-in reference digital-to-analog converters (DACs), two zero-drift, zero-crossover op amps with programmable gain, and one general-purpose amplifier. These devices also offer intelligent digital peripherals such as three 16-bit advanced control timers, three 16-bit general purpose timers, one 24-bit high-resolution timer, two windowed-watchdog timers, and one RTC with alarm and calendar mode. These devices provide data integrity and encryption peripherals and enhanced communication interfaces (four UARTs, two I2Cs, two serial-peripheral interfaces (SPIs)).

2.3.2 UCC28881

The UCC28881 integrates the controller and a 14 Ω , 700V power MOSFET into one monolithic device. The device also integrates a high-voltage current source, enabling start-up and operation directly from the rectified mains voltage. The UCC28881 is the same family device of the UCC28880, with higher current.

The low quiescent current of the device enables excellent efficiency. With the UCC28881, the most common converter topologies, such as buck, buck- boost, and flyback can be built using a minimum number of external components.

2.3.3 UCC27712

The UCC27712 is a 620V, high-side and low-side gate driver with 1.8A source, 2.8A sink current, targeted to drive power MOSFETs or IGBTs. The recommended VDD operating voltage is 10V to 20V for IGBTs and 10V to 17V for power MOSFETs. The UCC27712 includes protection features where the outputs are held low when the inputs are left open or when the minimum input pulse width specification is not met. Interlock and dead-time functions prevent both outputs from being turned on simultaneously.

2.3.4 TLV9064

4

The TLV9064 is a quad-low-voltage (1.8V to 5.5V) op amp with rail-to-rail input- and output-swing capabilities. These devices are highly cost-effective designs for applications where low-voltage operation, a small footprint, and high capacitive load drive are required. Although the capacitive load drive of the TLV906x is 100pF, the resistive open-loop output impedance makes stabilizing with higher capacitive loads simpler. The TLV906xS devices include a shutdown mode that allow the amplifiers to switch into standby mode with typical current

consumption less than 1µA. The TLV906xS family helps simplify system design, because the family is unity-gain stable, integrates the RFI and EMI rejection filter, and provides no phase reversal in the overdrive condition.

2.3.5 TPS54202

The TPS54202 is a 4.5V to 28V input voltage range, 2A synchronous buck converter. The device includes two integrated switching FETs, internal loop compensation and 5ms internal soft start to reduce component count.

Advanced Eco-mode implementation maximizes the light load efficiency and reduces the power loss.

Cycle-by-cycle current limit in both high-side MOSFETs protects the converter in an overload condition and is enhanced by a low-side MOSFET freewheeling current limit which prevents current runaway.

3 System Design Theory

The main focus of this reference design is a motor control system with a MSPM0G1507 microcontroller to support both sensored and sensorless FOC algorithms for appliance applications.

3.1 High-Voltage Buck Auxiliary Power Supply

A simplified UCC28881-based high-voltage buck supply provides an auxiliary power supply for this reference design to delivery up to 200mA for 15VDC.

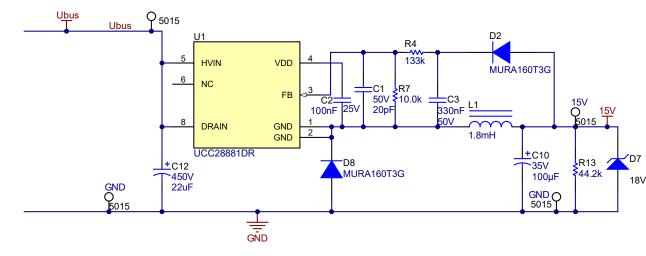


Figure 3-1 shows the UCC28881 high-voltage buck supply circuit.

Figure 3-1. UCC28881 High-Voltage Buck Supply Circuit

5



3.2 DC Bus Voltage Sensing

The DC voltage sensing circuit is used to convert the rectified voltage signal into a low-voltage signal, a low-cost resistor divider network as shown in Figure 3-2. The DC bus voltage can also be used to estimate AC input voltage.

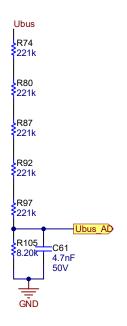


Figure 3-2. DC Bus Sensing Circuit

3.3 Motor Drive Stage

The three-phase motor inverter is realized with a six piece, 10A IGBT. Figure 3-3 shows a one-half bridge IGBT and the driver circuit. The IGBT gate resistor can be changed to control IGBT rising or falling time to make a tradeoff between power loss and EMI noise level.

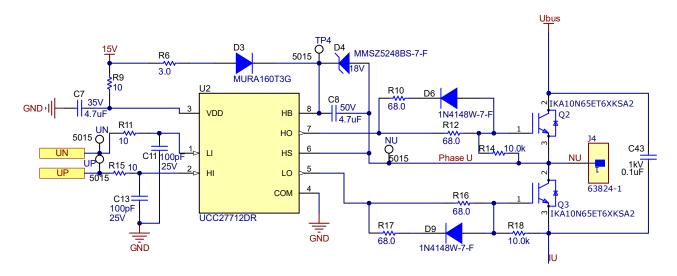


Figure 3-3. Half-Bridge IGBTs and Gate Driver

3.4 Bypass Capacitors

6

Each half-bridge has a high-voltage (1kV) ceramic bypass capacitor (C43, C48, and C62 on the schematic) providing a capacitance of 0.1μ F per half-bridge, see Figure 3-3. These bypass capacitors handle high-



frequency current in each half-bridge during switching, which cannot be provided by electrolytic DC bulk capacitors. Those X7R capacitors can also help to reduce EMI noise level.

3.5 Phase Current Sensing With Two or Three Shunt Resistors

This reference design supports 1, 2, or 3 shunt resistors. Three phase currents are sensed by an external TLV9064 in this reference design; however, the MSPM0G1507 microcontroller has two built-in zero-drift, zero-crossover operational amplifiers. These op amps support up to 2 phases of current sampling, so external op amps are not necessary, thus reducing the cost for the system. Figure 3-4 shows the shunt resistor configuration.

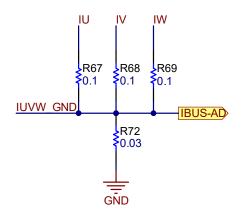
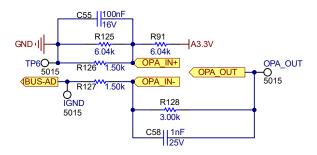


Figure 3-4. Shunt Resistor Configuration

3.6 Phase Current Sensing With a Single Shunt Resistor

This reference design also supports single-phase current sensing with a built-in op amp. R72 in Figure 3-4 is the shunt resistor for single-phase current sensing, R67, R68, and R69 can be shorted for such an application. The internal op amp OPA0 is selected to support single-phase current sensing. Figure 3-5 shows the circuit for internal OPA0, gain, and bandwidth can be set by those resistors and capacitors, the default gain is 2.





3.7 Hall-Effect Sensor or QEI Interface for Sensored Motor Control

This reference design also supports Hall or Quadrature Encoder Interface (QEI) mode-based sensored motor control algorithms. The MSPM0G1507 has a TIMG timer which supports 2- or 3-signal QEI. When configured in QEI mode, TIMG can collect the quadrature encoded signals to provide the information on the relative positioning and movement of a linear or rotary motion.

The QEI consists of two gray coded quadrature input signals PHA and PHB, and an index input signal IDX. All input signals go to the CCP inputs of a single counter, such that PHA and PHB are mapped to CCP0 and CCP1, and IDX is brought in as a separate input.

This reference design supports both a 3.3V or 5V level Hall-effect sensor for the following:

- Use the 3.3V Hall-effect or QEI sensor
- Switch on all Hall1V33, Hall2V33, Hall3V33



- Depopulate R64, R65, and R66
- Use the 5V Hall-effect or QEI sensor
- Switch off all Hall1V33, Hall2V33, Hall3V33, QEI1GND, QEI2GND, and QEI3GND of switcher S1
- Populate R64, R65, and R66

Figure 3-6 shows the Hall and QEI interface circuit

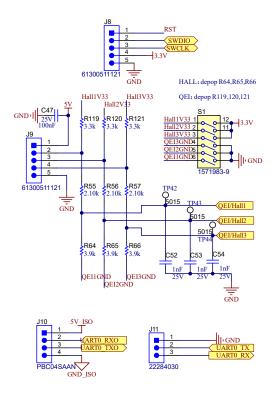


Figure 3-6. Hall and QEI Interface Circuit

3.8 DAC for Software Debug

To simplify software debugging, this reference design provides two DACs to export specified signals, such as rotator angle, simplified phase current, PWM duty, and so forth. As represented in Figure 3-7, DAC is an independent, internal, 12-bit digital-to-analog converter. DAC2 in Figure 3-7 is an indirect 8-bit digital to analog converter, which must be buffered through the internal op amp - OPA0. However; if OPA0 is used for phase current sensing, the op amp cannot output DAC2. Figure 3-7 shows the digital-to-analog converter circuit.

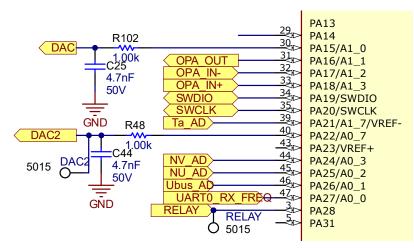


Figure 3-7. Digital-to-Analog Converter

8

(1)

9



3.9 Overcurrent Protection

This reference design implements overcurrent protection (OCP) with both external op amps and an internal comparator. Figure 3-8 shows the external op amp OCP, this circuit summarize three phases of current, then compares to a reference voltage (0.2V) to create an overcurrent fault signal. The other OCP is an internal comparator for a single shunt resistor current sensing. Internal op amp output can be connected directly to an internal comparator, to accomplish overcurrent protection. The designed external op amp OCP trigger point is 6A peak phase current, while the OCP trigger point for the internal comparator can be set by the internal reference.

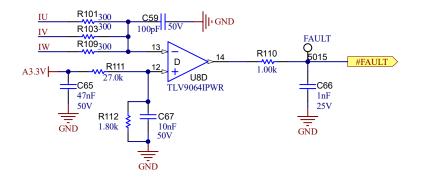


Figure 3-8. External Overcurrent Protection Circuit

3.10 Overtemperature Protection

An M3 screw mounting hole is designed to attach a thermistor on the heat sink to sense heat sink temperature. Software can sample this temperature signal to trigger overtemperature protection at a specified level.

3.11 Isolated UART port

An isolated UART port is provided to communicate with the host controller at J10. An external 5V or 3.3V rail is needed to power the isolated side of the U10 isolator.

3.12 Inverter Peak Power Capability

This reference design uses a 10A IGBT, which can deliver a continuous current of up to $6A_{RMS}$ at a junction temperature of 125°C. The $6A_{RMS}$ is an intermittent maximum current rating for the TIDA-010250 design. The continuous current capability of the inverter (which is $\leq 6 A_{RMS}$) is determined by the thermal design. The peak power capability (P_{PEAK}) of the GaN inverter, assuming a 3-phase output voltage (V_{OUT}) of 200V_{RMS} and unity power factor (PF) is given by Equation 1:

$$P_{PEAK} = \sqrt{3} \times V_{OUT} \times I_{OUT} \times PF = \sqrt{3} \times 200 \times 6 \times 1 = 2078 W$$

P_{PEAK} is the absolute maximum power the GaN inverter can handle - the continuous power rating of the inverter is decided by thermal design, peak ambient temperature, and overcurrent protection setting.



4 Hardware, Software, Testing Requirements, and Test Results

4.1 Hardware Requirements

This section details the necessary equipment, test setup, and procedure instructions for the design board and software testing and validation.

4.1.1 Hardware Board Overview

Figure 4-1 shows a system block diagram of this reference design.

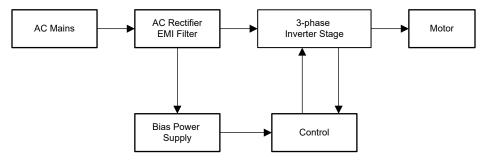


Figure 4-1. TIDA-010250 System Block Diagram

The motor control board has functional groups that enable a complete motor drive system. The following is a list of the blocks on the board and the functions of the blocks. Figure 4-2 shows the top view of the board and different blocks of the TIDA-010250 PCB.

- · AC input connectors and filter
- Auxiliary power supply
- MCU controller
- Motor inverter output connector
- · Phase current sensing with shunt resistors
- Hall or QEI interface
- USART serial communication

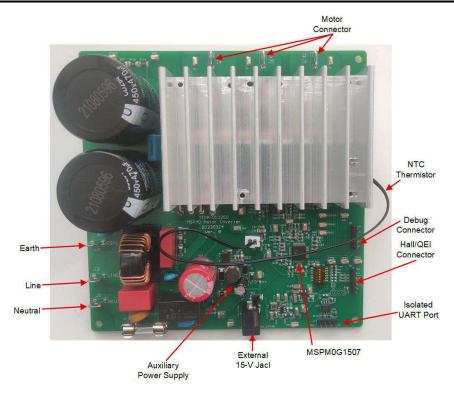


Figure 4-2. Layout of M0 Motor Inverter Board

TI recommends taking the following precautions when using the board:

- Do not touch any part of the board or components connected to the board when the board is energized.
- Use the AC Mains or wall power supply to power the kit. TI recommends an isolation AC source.
- Do not touch any part of the board, the kit, or the assembly when energized. Though the power module heat sink is isolated from the board, high-voltage switching generates some capacitive coupled voltages over the heat sink body.
- · Control Ground can be hot.

4.1.2 Test Equipment for Board Validation

The following equipment is recommended for board validation:

- Isolated AC source
- Single-phase power analyzer
- Digital oscilloscope
- Multimeters
- Three-phase PMSM motor
- Dynamo meters
- DC current source with up to 6A current
- 15V_{DC} power supply preferred
- Three-phase power analyzer



4.2 Software Requirements

4.2.1 Getting Started MSPM0 Firmware

Download and install MSPM0-SDK mspm0_sdk_2_01_00_03 or newer software from the link provided by TI. Install this MSPM0 SDK software in the default folder. The TIDA010250 software project then resides inside the MSPM0 SDK folder at

<install_location>\ti\mspm0_sdk_2_01_00_03\examples\nortos\LP_MSPM0G3507\motor_con trol_bldc_sensorless_foc\senslroess-foc_TIDA010250\

4.2.1.1 Download and Install Software Required for Board Test

- 1. Download and install MSPM0-SDK, mspm0_sdk_2_01_00_03 or newer software from the link provided by TI.
- 2. Install MSPM0-SDK in one of two ways:
 - Go to CCS, under Project → Import CCS Project, browse to choose the sens1roessfoc_TIDA010250 folder to import this project.
 - Go to CCS and under View → Resource Explorer. Under the TI Resource Explorer, go to Arm-based microcontroller → Embedded Software → MSPM0-SDK, and click Download and Install.
- 3. Once installation is complete, close CCS, and create a new workspace for importing the project

4.2.1.2 Import the Project Into CCS

Open CCS, and select *Project* → *Import CCS Project*, browse to following folder in MSPM0 SDK:

<install_location>\ti\mspm0_sdk_2_01_00_03\examples\nortos\LP_MSPM0G3507\motor_con trol_bldc_sensorless_foc\sensorless-foc_TIDA010250, Figure 4-3 shows the folder.

→ × ↑ 📙 « i	examples > nortos > LP_MSPM0G3507 > mot	tor_control_bldc_sensorless_foc	ٽ ~	Search motor_cont	trol_bldc_se
ganize • New fo	lder				-
199	Name	Date modified	Туре	Size	
	sensorless-foc_DRV8316	6/26/2024 9:29 PM	File folder		
1000	sensorless-foc_DRV8323RS	6/26/2024 9:29 PM	File folder		
1 m m	sensorless-foc DRV8329	6/26/2024 9:29 PM	File folder		_
This PC	sensorless-foc_TIDA010250	6/26/2024 9:29 PM	File folder		
3D Objects					
Desktop					
Desktop					
Desktop					
 Desktop Documents Downloads 					
Desktop Documents Downloads					
Desktop Documents Downloads Music					
 Desktop Documents Downloads Music Pictures 					
Desktop Documents Downloads Music Pictures	~				

Figure 4-3. Folder of sensorless-foc_TIDA010250



Figure 4-4 shows the project found in this folder. Select the project and click the *Finish* button to import this project.

😚 Import CCS Projects			×
mport CCS Projects Import existing CCS Proje	cts or example CCS Projects.		2
Select search-directory:	C\ti\mspm0_sdk_2_01_00_03\examples\nortos\LP_MSPM0G3507\motor_control_bldc_sensorless_foc\sensorless-foc_TIDA010250	Brow	se
) Select archive file: Discovered projects:		Brow	se
	DA010250_MSPM0G1507_nortos_tidang_(tidang/sensorless-foc_TIDA010250_LP_MSPM0G1507_nortos_tidang.projectspec]	Selec	t All
		Desele	ct All
		Refre	sh
	ferenced projects found in same search-directory		
Copy projects into work	rspace		
Open <u>Resource Explorer</u> to	browse a wide selection of example projects		
0	< Back Next > Finish	Cano	

Figure 4-4. Project in This Folder

4.2.1.3 Compile the Project

The Sensor less-foc project for TIDA-010250 is now active. Click the Build button or select the Project \rightarrow Build Project menu, the .out file is now generated after the compilation is finished. Figure 4-5 shows the resulting project build.

workspace_v12 - sensorless-foc_TIDA010250_MSPM0G	G1507_nortos_tic	clang/main.c - I	ide Composer Studio – O
ile Edit View Navigate Project Run Scripts			
*****	• • • • •		Q. (2)
Problems × 🖋 Search		A 1 - 0	@makex =
errors, 1 warning, 0 others			po -12. Entire egreement. Inits is the entire agreement between you and it and this agreement supersedes any prior agreement between the parties related to the subject matter of this 37 * -//OPPKIDHT*/
lescription	Resource	Path	38#include "main.h"
 Warnings (1 item) 			39#include "ISR.h"
			40Binlude "focPeriphTnit.h" 41Binlude "sort com.h"
			42#include "gateDriver.h"
			43
			44/1'#briefEnablegste driver read test */ 45_Boolgd#astrettm =0; /
			46
			47/*1 @brief Data read from gate driver */ 48volatile unitit_t regolara = 0;
			49
			50/11@briefRegister address to read from gate driver */ Sivolatile intig_ regider = 0x06;
			52 53 int main(void)
			Si { Si SYSCPG_DL_init();
			56 57 appInputReset(); /* Reset all the Application Input variables to Zero. */
Project Explorer ×	8	871-0	58 59 apReset(g_pMC_App); /* Reset all the Application Data variables to Zero */
pfc1phttpl_nonpowerSUITE_F28002x sensorless-foc_TIDA010250_MSPM0G1507_nortos	a distance i ta sele	Data at	00 1 userInputSInit(); /* Initialize the Default Motor Parameters */
> @ Generated Source	s_uciang (Activ	ve - Debugj	62 63 appConfig();
> it Binaries			<pre>states appcontag(); 5 footestathin(); 5 footestathin();</pre>
> 🕫 Includes			<pre>65 focPeriphInit();</pre>
> 🖕 Debug			66 57 while(1)
> 👄 modules			68 (
> 💩 targetConfigs			69 if(gdReadTestEn)
> 🔤 ticlang			<pre>20 { regData = gateDriverRegRead(regAddr); </pre>
> @ ISR.c			72 isguece = Beccol and ingeneral/secol/s
> 🗟 ISR.h			73 UART_checkForCommand(pUART);
> @ main.c			74 75 updateConfigs();
> la manuf > la mspm0g1507.cmd			76
6 MSPM0G1507_TIDA010250.syscfg			77
README.md			78) 79
tida_010265_gminv_280013x			
tida_010273_GaNInv_280013x			🛡 Console × 🗶 🖉 수 🔞 🖬 📰 = 🐘 🐖 한 🛡 👻 한 - 한 한 한
			Console A Console A Console A Console Sensorless-for TIDA010250 MSPM061507 nortos ticlanal
			Current control permission current control of prints comp. Circlification of the control of prints control of prints control of prints control of prints control of the co
			"C/TL/CSIZ//CSIZ/COS/COMPIN/TI-GT-ATMINW_3.Z.Z.LIS/DIN/TIATMEIANG.EXX - C -MARCH=TNUMDVOM -MECPU+COrtex-HMPIUS -MILOAT-ADI=SOT -MILTLA-EMDIAN -MTNUMD -UZ -ITC:/USEFS/AMH9ZOW/WORKSPA Finished building: "/main.c"
			uliding target: "sensorless-foc_TIDA010250_MSPM001507_nortos_ticlang.out"
			hvoking: Arm Linker C:/ti/ccst/cls/com/sicompiler/ti-cgt-armllvm 3.2.2.LTS/bi//tiarmclang.exe" -march=thumbv6m -mcpuscortex=m0plus -mfloat=abisoft -mlittle=endian -mthumb -02 -D_MSPM063507DGLOBAL_IO=27
			Control and the second s second second se
			···· Build Finished ····

Figure 4-5. Build Project

Now open the ISR.c file to check motor parameters, see the register map section in the *Sensorless FOC Motor Control User Guide*, this guide is found in the MSPM0 SDK folder:

<install_location>\ti\mspm0_sdk_2_01_00_03\docs\english\middleware\motor_control_b ldc_sensorless_foc



Figure 4-6 shows file location.

s (C:	(C:) > ti > mspm0_sdk_2_01_00_03 > docs > english > middleware > motor_control_bldc_sensorless_foc								
^		Name	Date modified	Туре	Size				
		doc_guide	6/26/2024 9:18 PM	File folder					
		🧵 doxygen	6/26/2024 9:12 PM	File folder					
	_ C	Sensorless_FOC_Motor_Control_User_Guide.html	6/7/2024 10:44 PM	Chrome HTML Docu	1 KB				

Figure 4-6. Folder for Sensorless FOC Motor Control User Guide

This design guide introduces all the registers for the FOC algorithm, the designer can also add or remove some special functions of FOC using those registers. Figure 4-7 shows the register map.

	 pUserStatu 		
	6. Press Play		
			Refresh" in the Expressions window. : 🛞
s FOC Motor Control ser Guide	8. Refer to the	e Tuning I	Document for spinning and tuning the motor.
1.00.xx	2. Regist	er Ma	р
	1 UserCtrl	Regs	
orless FOC Software	Set of register:	s used to	control the motor.
setup	1.1 Speed	Ctrl	
ip.			
egs tri	Speed control	register	
Ctrl1	SpeedCtrl	Definit	ion
tri2	speedinput	Speed	Command in 15 bit value. Value to set = (Desired motor speed) x 32768/(I
Ctri3	<		,
5	1.2 algoD€	ebugCt	
arameters	Algorithm deb	ug registe	er 1
		-02130	** *
fg	algoDebugCt	ri1	Definition
rtUp1	IqRefSpeedLo	opDis	Iq reference used if speed loop is disabled
Jp2	forceAlignAng		
1	forceISDEn		Halts after ISD is complete
2	forceIPDEn		Halts after IPD is complete
	forceSlowCyc	ienrstcyci	
	forceAlignEn		Halts after align is complete
	closeLoopDis		Disable open to closed loop transition
și -	forcedAlignAl	ngle	Align angle used in forced align
egs	clearFlt		Clears all faults
ultStatus			
ate	1.3 algoDe	ebugCt	r12
ы	Algorithm deb	ug registe	er 2
1		-	Definition
	algoDebugCt		
	forceVQCurrL		Vq applied when current loop and speed loop are disabled
peed	forceVDCurrL	oopDis	Vd applied when current loop and speed loop are disabled
e t	currLoopDis		Disable current loop and speed loop
	statusUpdate		Enables continous update of the UserStatusRegs
aultStatus aultStatus	updateConfig	IS .	If set 1, firmware will set to zero when all the motor control configuration
ISTARUS	- C		,
5	NOTE: If user u	updates a	Il or many of the configurations in pUserInputRegs, to ensure
tation			e updated by the motor control application to the FOC variables
otor Control Library			pdateConfigs as 1 and wait till it is set 0 by the firmware before This done to ensure all the parameters set by the user gets
control controly	updated to the		
on User Guide			
tV8316 Gate	1.4 algoD€	ebugCt	rl3
DRV8323RS Gate	Algorithm deb	ug registe	er 3
DRV8329 Gate	algoDebugCt	rl3	Definition
TIDA010250 User	fluxModeRefe	erence	Id reference used when Flux Mode is Enabled
	1.5 dacCtr	1	
	DAC control re	gister	

Motor and control parameters are found in ISR.c, which can be updated per the motor and system of the user. The parameters for motor EMJ_04APB22, unit of each parameter are shown below. For more information, see the *MSPM0 Sensorless FOC Tuning Guide* application note.

```
pUserInputRegs->systemParams.mtrResist = 2627; // Rs(mR)
pUserInputRegs->systemParams.mtrInductance = 8608; // Ls(uH)
pUserInputRegs->systemParams.mtrBemfConst = 3779; // 10*BEMP, BEMF(mVpHz)
pUserInputRegs->systemParams.voltageBase = 258.6; // Vbase(V)/sqrt(3)
pUserInputRegs->systemParams.currentBase = 8.25; // Ibase(A)
pUserInputRegs->systemParams.maxMotorSpeed = 200; // Maximum Speed(Hz)
pUserInputRegs->systemParams.speedLoopKp = 0.0539; // Speed Loop Kp
pUserInputRegs->systemParams.currLoopKi = 0.036; // Speed Loop Ki
pUserInputRegs->systemParams.currLoopKp = 3.42; // Current Loop Kp
```

After update motor and control parameters, click the *Compile* button, no error information and the .out file are created in the *Console* window, see Figure 4-8.

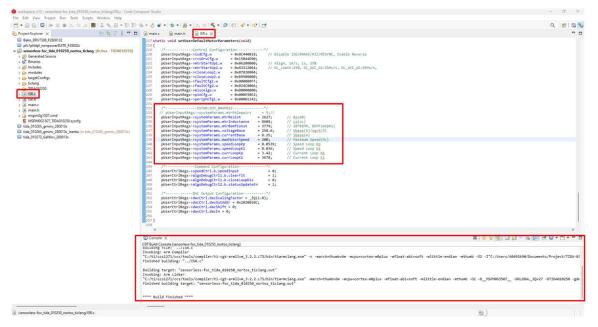


Figure 4-8. EMJ_04APB22 Motor Parameters and Compile Results

4.2.1.4 Download Image and Spin Motor

Click the Run → Debug menu options to download and debug software in CCS, then add pUserCtrlRegs variables into *Expressions* windows, Select *pUserCtrlRegs* \rightarrow *speedCtrl* \rightarrow *b* \rightarrow *speedInput*, right click, select Q-Values \rightarrow Q-Value(15). Figure 4-9 shows where and how to change value type.

•• Variables 🚿 Expre	ssions × Registers				B •	× % #	1004			
Expression > • pUserInputReg		Type struct USER_INPUT_INTER	RFACE T *	Value 0x20200000 (systemParams={mtrResist=2627	Address 0x20201100					
 pUserCtrlReqs 	12 12	struct USER_CTRL_INTER	-	0x20200400 (speedCtrl={b={speedInput=0,res						
✓ Ø *(pUserCtrlF	(eqs)	struct USER_CTRL_INTER		{speedCtrl={b={speedInput=0,reserved=0},w=0_						
✓	1	union RAM_SPEED_CTRL	т	{b={speedInput=0,reserved=0],w=0}	0x20200400					
× @ b		struct ramSpeedCtrl	-	(speedInput=0,reserved=0)	0x20200400					
10° S			1	0.0 (Q-Value(15))	0x20200400 bit 0-14					
** re	Select All	Ctrl+A		0	0x20200400 bit 15-3	1				
Copy Expressions Remove Sendopel Remove Remove		Ctrl+C		0	0x20200400					
	algoDe Remove All AlgoDe Number Format dacCtrl Add Expression Group	 Remove All Number Format Add Expression Group 		1.T	{b={iqRefSpeedLoopDis=0,forceAlignAngleSrc	0x20200404				
> 🥔 algoDe			oDe Number Format		_2_T	{b={reserved=0,forceVQCurrLoopDis=0,forceV	0x20200408			
> 🥔 algoDe				2	_3_T	{b={fluxModeReference=0,reserved1=0},w=0}	0x2020040C			
> 🥔 dacCtrl			>		{dacEn=1,dacShift=0,dacScalingFactor=134217	0x20200410				
Add new expr	View Memory									
	View Memory at Value									
	Find	Ctrl+F								
	Add Watch Expression.									
	Disable									
	Enable									
	Edit Watch Expression									
	Q-Values	>	Q-Value(31)							
	Cast To Type		Q-Value(24)							
	Add Global Variables		✓ Q-Value(15)							
	Export		Clear Q-Value							
	Import		Select Q-Value							
	Breakpoint (Code Com	operar Chudio)	Select Q-Value							
	Breakpoint (Code Com) In Graph	poser studio)								
	Watch									

Figure 4-9. Download And Debug

Run code by clicking the Run button in CCS, and change the speedInput to a non-zero value to spin the motor, for example, 0.6, this number is a normalization number of the maximum speed, so if speedInput = 1, the motor runs at the maximum speed. To stop the motor, set speedInput to zero.

Figure 4-10 shows where to change the speedInput value.

Variables & Expressions × Register			
Expression	Туре	Value	Address
> • pUserInputRegs	struct USER_INPUT_INTERFACE_T *	0x20200000 {systemParams={mtrResist=2627,	0x20201100
 pUserCtrlRegs 	struct USER_CTRL_INTERFACE_T *	0x20200400 {speedCtrl={b={speedInput=0,res	0x202010FC
 	struct USER_CTRL_INTERFACE_T	{speedCtrl={b={speedInput=0,reserved=0},w=0	0x20200400
✓ Ø speedCtrl	union RAM_SPEED_CTRL_T	{b={speedInput=0,reserved=0},w=0}	0x20200400
∽ <i>e</i> ∌ b	struct ramSpeedCtrl	{speedInput=0,reserved=0}	0x20200400
🕪 speedInput	unsigned int : 15	0.0 (Q-Value(15))	0x20200400 bit 0-14
reserved	unsigned int : 17	0	0x20200400 bit 15-31
00- W	unsigned int	0	0x20200400
> 🥔 algoDebugCtrl1	union RAM_ALGO_DEBUG_1_T	{b={iqRefSpeedLoopDis=0,forceAlignAngleSrc	0x20200404
> 🥔 algoDebugCtrl2	union RAM_ALGO_DEBUG_2_T	{b={reserved=0,forceVQCurrLoopDis=0,forceV	0x20200408
> is algoDebugCtrl3	union RAM_ALGO_DEBUG_3_T	{b={fluxModeReference=0,reserved1=0},w=0}	0x2020040C
> 🥔 dacCtrl	struct RAM_DAC_CNTRL_T	{dacEn=1,dacShift=0,dacScalingFactor=134217	0x20200410
Add new expression			

Figure 4-10. Set Motor Speed Percentage by speedInput



4.3 Test Results

The following sections show the test data. The test results are divided into multiple sections that cover the steady state performance and data, functional performance waveforms, and transient performance waveforms of a PMSM motor.

4.3.1 Test Setup

Figure 4-2 shows the position of these blocks and the connectors on the board. Use the following steps to set up the test.

- 1. Connect a serial wire debug (SWD) emulator to connector J8 to debug or program the MSPM01507. Isolate the host PC from TIDA-010250 board.
- 2. Connect a motor cable to the terminals J4, J5, and J6.
- 3. Apply a DC bus power, AC power supply or AC mains power to the inverter by connecting the power to J1, J2, and J3.
 - The maximum output of the DC power supply is 380VDC.
 - The maximum output of the AC power supply is 265VAC, 50/60Hz.
 - AC main power is 220VAC, 50/60Hz.

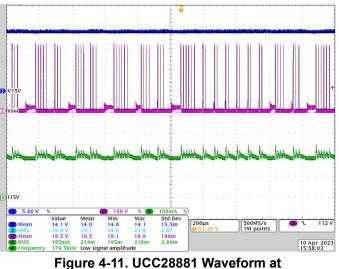


4.3.2 Auxiliary Power Supply Test

Figure 4-11 shows the buck power supply waveforms with 200mA load at 15V_{DC} output.

- CH1 (Blue): 15V voltage rail
- CH3 (Purple): switching node at pin 1 of UCC28881
- CH4 (Green): 15V output current

Figure 4-12 shows the buck inductor temperature at 15V, 100mA at 25°C ambient temperature.



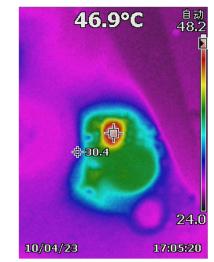


Figure 4-11. UCC28881 Waveform at 220V_{AC}, 200mA



4.3.3 Current Open Loop Test

This test shows software behavior to generate a sinusoidal phase current under open loop mode, Figure 4-13 shows the waveform.

- CH2 (Cyan): AC input voltage
- CH3 (Purple): DC bus voltage
- CH4 (Green): Current of phase U

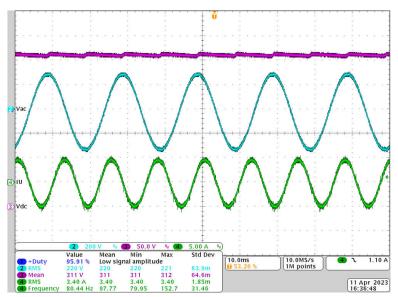


Figure 4-13. Phase Current at Open Loop Mode

4.3.4 Overcurrent Protection Test

This test can be done with an external 15V_{DC} power supply. Use the following steps for this test:

- 1. Power on the board with an external 15VDC power supply from DC Jack J7
- 2. Connect a DC power supply to two terminals of shunt resistor R67
- 3. Set the DC power supply at constant voltage mode, and maximum current
- 4. Enable output, then a surge output current flows through R67.
- 5. Adjust the power supply voltage and current output
- 6. Make sure the surge current is high enough (> 6.5ADC) to trigger OCP.

Figure 4-14 shows the FAULT signal falls once current is over 6A. The falling time is less than 2µs, which can be fast enough to protect most of IGBT.

- CH1 (Blue): Fault signal at test point of #FAULT
- CH4 (Green): Current on R67

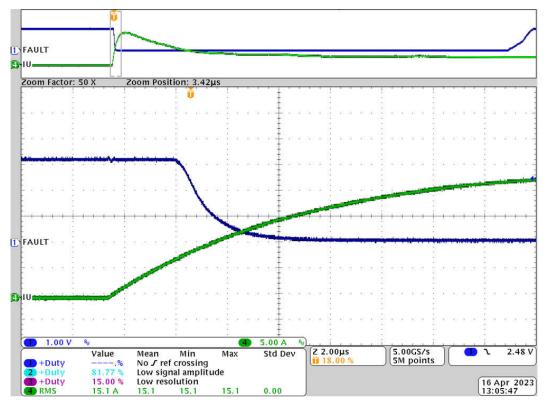


Figure 4-14. Overcurrent Protection Response Time



4.3.5 Motor Start-Up Sequence

Figure 4-15 shows the start-up sequence waveform for a PMSM motor with a sensorless FOC algorithm. The sequence usually consists of three phases: align, open loop ramp-up, speed closed loop. Complete the following steps during aligning:

- 1. Slowly increase aligning current to prevent any surge current
- 2. Move to open loop ramp-up
- 3. Ramp-up current can be set to meet the start load requirement
- 4. The rotor position observer starts to work during ramp-up time
- 5. Once the motor speed reaches the set point and the rotor angle error is small enough, then the software moves to closed speed and closed current loop

The following list of channels are presented in Figure 4-15.

- CH1(Blue): Observed rotor angle
- CH2 (Cyan): AC input voltage
- CH3 (Purple): DC bus voltage
- CH4 (Green): Current of phase U

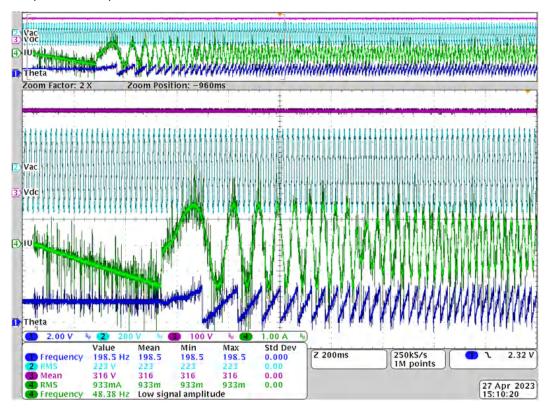


Figure 4-15. Motor Start-Up Sequence for PMSM With Senseless FOC Algorithm



4.3.6 Load Test

A load test was conducted to verify the overall thermal design. Figure 4-16 shows the board temperature rising with a 460W load. The heat sink temperature is only 32°C. Fuse and diode bridge are the hottest locations, but there is still enough margin, and the fuse cover can be removed to provide cooling.

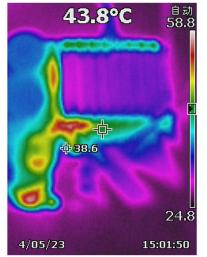


Figure 4-16. Board Temperature Rising With 460W Load

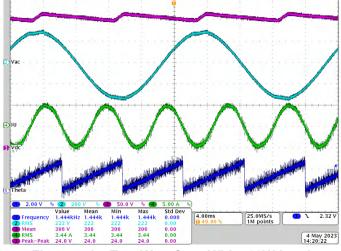


Figure 4-17. Test Waveform With 460W Load

Figure 4-17 shows the test waveform with a 460W load and the following channel assignments:

- CH1(Blue): Observed rotor angle
- CH2 (Cyan): AC input voltage
- CH3 (Purple): DC bus voltage
- CH4 (Green): Current of phase U

Figure 4-18 shows load test setup with a dynamometer.

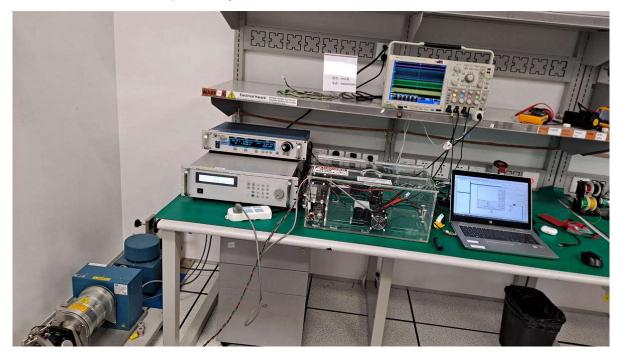


Figure 4-18. Load Test Setup



4.4 Migrate Firmware to a New Hardware Board

4.4.1 Configure the PWM, CMPSS, and ADC Modules

SYSCFG_DL_init() has initialization functions. Find SYSCFG_DL_init() in main.c, press CTRL and click the function, SYSCFG_DL_init() is opened, and functions to configure GPIO, ADC, PWM, and so forth can be found in SYSCFG_DL_init(). Figure 4-19 shows those functions.

lle Edit View Navigate Project Run So			
3 * 5 6 0 0 0 * 8 * 6 * 6 * 8 # 8 #	000000		9, 121
Problems × 🛷 Search		¥ I = 0	CREADMEnd @ISRc @ISRh @mainc @mainh @watcomme @itymsp.dl.config.x @itymsp.dl.config.h
errors, 4 warnings, 0 others			120
Ascription	Resource	Path	121 SYSCONFIG_MEAK void SYSCFG_DL_GPIO_INIT(void)
Warnings (4 items)			224
			224 DL_GPI0_initPeripheralOutputFunction(GPI0_M6W4_C0_IOHLX,GPI0_M6W4_C0_IOHLX,FUNC);
			125 D. GPID. enableCytput(GPID. Parka Ce. PCRT, GPID. Parka Ce. PIN);
			126 D_GPIO_initPeripheralOvtputFunction(GPIO_PWW@_C0_CPWL_OPUL_OPUL_OPUL_OPUL_OPUL_OPUL_OPUL_OPU
			127 DL_GPIO_enableOutput(GPIO_PH#H&_C0_CMPL_PDAT, GPIO_PH#H&_C0_CMPL_PIN);
			Di_0010_inite=ipheralOutputfunction(0010_phade_cl_IONU.0010_phade_il_IONUX_FUNK); Di_0010_enableOutput(0010_phade_cl_ION1_0010_phade_cl_ION1);
			10 OC OPD interplayed burget write or to make (1 OPL TORK, OPD RAVE (1 OPL TORK, FURC);
			131 D. GPIO enableCytoyt(GPIO PARAB (1 CMP, PORT, GPIO PARAB (1 CMP, PIN);
			132 DL_OPIO_initPeripheralOutputFunction(OPIO_PAGWa_C2_IONUX_GPIO_PAGWa_C2_IONUX_FUNC);
			133 DL_GPID_enableOutput(GPID_PARAB_22_PORT, GPID_PARAB_22_PIN);
			134 DL_0PTD_initVeripheraDoutputFunction(0PTD_PMHAB_C1_IONUX_PURC); 135 DL_0PTD_enableOutputF0FD_PMHAB_C3_POTA_PD_PMHAB_C3_IONUX_FURC);
			35 DL_0PTD_enableOutput(@PTD_NMAR_C1_PORT, @PTD_NMAR_C1_PTN); 35 DL_OPTD_interterportsore(DeptDeptMark_C1_PTN); 36 DL_OPTD_interterportsore(DeptMark_C1_PTN); 37 DL_OPTD_interterportsore(DeptMark_C1_PTN); 38 DL_OPTD_interterportsore(DeptMark_C1_PTN); 38 DL_OPTD_interterportsore(DeptMark_C1_PTN); 39 DL_OPTD_interterportsore(DeptMark_C1_PTN); 39 DL_OPTD_interterportsore(DeptMark_C1_PTN); 30 DL_OPTD_interportsore(DeptMark_C1_PTN); 30
			11 DL OPD enabledrupt(PDFLDDFLDDFLDDFLDDFLDDFLDDFLDDFLDDFLDDFL
			138 DL_OPID_initPeripheralIngutFunction(GPID_PWUAB_IONIX_FAULT_1_GPID_PWUAB_IONIX_FAULT_1_FUNC);
			139
			240 DL_OPID_initPeripheralOutputFunction(DL_OPID_initPeripheralOutputFunction(
			141 GHIO LART & IONA TA, GHIO LART & IONA TA, GHIO LART A, GHIO LART A
Project Explorer ×		8871-0	(a) GPTO UAT & DOWL RX, GPTO UAT _ DOWL RX, FUNC);
pfc1phttpl.nonpowerSUITE_F28002x			144
sensorless-foc TIDA010250 MSPM0G1507	nortos ticlang (Active - (Debug]	145 DL_GPIO_initDigitalOvtputFeatures(FOC_GPIO_OUT_WFAULT_IOMLX,
> Ø Generated Source			146 D. GPO DIWERSTON DISARLE, D. GPID. RESISTOR PULL (P. 147 D. GPID DIWESTON DISARLE J. DISARLEJ DISARLEJ)
> # Binaries			147 DK_GPIO_DRIVE_STREMGTH_LOW, DL_GPIO_MIZ_DISABLE); 148
> Ø Includes			<pre>149 DL_GPIO_initDigitalOutput(FOC_GPIO_OUT_RELAY_IONUX);</pre>
Debug			150
) in modules			151 DL_GPIO_initDigitalInput(FOC_GPIO_IN_DIR_IOMUX);
 w syscfq 			152 153 DL.0PIO_initDigitalInpvt(FCC_OPIO_IN_BRAKE_IOMUX);
a ti msp. dl. config.c			122 0 Contornationalizationalic (or output for output f
il 6 msp.dl.config.h			155 DL OPIO initDigitalOutput(TST PIN IONAX);
iii ti msp. dl. config.o - (ARM/le)			156
Event.dot			<pre>157 DL_GPID_initDigitalOutput(TST_FIN1_IOMAX);</pre>
ii ti msp. dl. config.d			158 DL OPID ClearPins(OPIDA, FOC OPID OUT NFAULT PIN D D D D D D D D D D D D D D D D D D
e tidang			100 FOC 6010 001 RELAY PN1):
B ISRo - (ARM/le)			161 DL_GPIO_enableOutput(GPIOA, FOC_GPIO_OUT_NFAULT_PIN
iii main.o - (ARM/e)			<pre>162 FOC_GPIO_OUT_RELAY_PIN);</pre>
O sensorless-foc TIDA010250 MSPM0G1	07 nortos ticlano.out - (Al	M/4	163 DL_OPID_configsubscriber(@PIOB, DL_OPID_SUBSCRIBER_INDEX_),
E ccsObis.opt			164 0., 070 SUBSCHIER, DV. POLY, TODEL, 165 0., 070 SUBSCHIER, DV. 161;
il ISR.d			<pre>14 DL OPD.stDuctiberCharle()(#10, DL OPTO_SUBSCRIBER_INDEX_1, OPICE_EVENT_SUBSCRIBER_1_CHANNEL);</pre>
il main.d			167 DL_GPID_enableSubscriber(GPID8, DL_GPID_SUBSCRIBER_INDEX_1);
la mainfile			168 DL_GPID_clearPins(GPIOB, TST_PIN_PIN
G objects.mk			4
iii sensorless-foc_TIDA010250_MSPM0G15	07 nortos ticlang linkinfo-	erri (Ø Console × X 4 4 %] □ □ = % Ø 2 0 • 0 • 0 • 0
il sensorless-foc TIDA010250 MSPM0G15			CDT Build Console (sensoriess-foc, TIDA010250, MSPM0G1507, nontos, ticlang)
G sources.mk	a second second second		"C:/ti/ccs1271/ccs/tools/compiler/ti-cgt-armllvm_3.2.2.UT5/bin/tiarmclang.exe" -march+thumbv6m -mcpu+cortex-m0plus -mfloat-abi+soft -mlittle-endian -mthumb -02 -D_H5PM003507D0LOB
la subdir rules.mk			<pre>Finished building target: "sensorless-foc_TIDA010250_MSPM001507_nortos_ticlang.out"</pre>
la subdir vars.mk			
> to modules			···· Build Finished ····
			BATA LTUTOA
> in taroetConfigs			

Figure 4-19. GPIO, ADC, and PWM Functions for Initialization

In *Debug* \rightarrow *syscfg* \rightarrow *ti_msp_dl_config.h*, all defines for GPIO, PWM, and ADC can be found and updated for the new hardware board. Figure 4-20 shows those defines.

	000.00				9.8
Problems × 🛷 Search		11-0	CREADMEnd @ISR.c RISR.h R main.c R main.h	il uart_comm.c il tijmspjdl_config.c il tijms	p_d(config.h ×
errors, 4 warnings, 0 others			70/* clang-format off */ 71		
escription	Resource	Path	72 Bdefine POWER_STARTUP_DELAY	(16)	
Warnings (4 items)			73	(11)	
			74		
			75 Bdefine CPUCLK_FREQ 76	5000000	
			76		
			78		
			79/* Defines for PWPU@ */		
			80 #define Publo_INST	TIMA	
			818define PWMA0_INST_IRQHandler 828define PWMA0_INST_INT_IRQN	TINA@_IRQHandler (TINA@_INT_IRQn)	
			83 #define PubWA@_INST_CLK_FREQ	5000000	
			84/* GPIO defines for channel 0 */		
			85 #define GPIO_PWPA8_C0_PORT	GPICA	
			86 Bdefine GPIO_PHAAB_C6_PIN 87 Bdefine GPIO_PHAAB_C6_IONUX	DL_GPIO_PIN_8 (IOPMX_PINCH19)	
			81 Bdefine GPIO_PAPAB_C0_IONUX_FUNC	IONUX_PINCH19_PF_TIMA8_CCP8	
			89 #define GPIO_PWHAB_C8_IDX	DL_TIMER_CC_0_INDEX	
			90/* GPIO defines for channel 0 */		
			918define GPIO_PWHAB_C0_CMPL_PORT 928define GPIO_PWHAB_C0_CMPL_PIN	GPICA DL GPIC PIN 9	
Project Explorer ×		8 9 7 1 - 0	93 Bdefine GPIO PANAB CO CMPL PIN	(ICHUK_PINCH20)	
pfc1phttpl.nonpowerSUITE_F28002x			94 #define GPIO PWHAB CB CHPL IONUX FUNC	IONUX PINCH20 PF_TIMA0_CCP0_CHPL	
Sensorless for TIDA010250 MSPM0G1507	nortes ticlana (Active - D	wheet I	95		
> @ Generated Source			96/* GPIO defines for channel 1 */		
i Sinaries			97 Bdefine GPIO_PWA0_C1_PORT 98 Bdefine GPIO_PWA0_C1_PIN	GPICA DL_GPIC_PIN_3	
Ø Includes			99 Bdefine GPIO_PWMA@_C1_IONUX	(ICHUX PINCHE)	
Debug			100 #define GPIO_PWMA8_C1_IOMUX_FUNC	IOMUX_PINCH8_PF_TIMA8_CCP1	
) in modules			101#define GPIO_PWMA@_C1_IDX	DL_TIMER_CC_1_INDEX	
 w syscha 			102/* GPIO defines for channel 1 */ 103 #define GPIO Publike C1 CMPL PORT	GPICA	
i ti msp.dl.config.c			104 #define GPIO_PWMAB_C1_CMPL_PIN	DL_GPIO_PIN_4	
il ti msp. dl. config.h			105 #define GPIO_PWMAB_C1_CMPL_IOMUX	(IOHUX_PINCH9)	
iii ti msp. dl. config.o - (ARM/le)			106 #define GPIO_PWMA8_C1_CMPL_IOMUX_FUNC	IOMUX_PINCH9_PF_TIMA8_CCP1_CMPL	
iii Event.dot			107 108/* GPIO defines for channel 2 */		
ii ti msp. dl. config.d			109 Bdefine GPIO Detines for channel 2 "/	GPICA	
) 💩 ticlang			110 #define GPIO PWMAB C2 PIN	DL GPIO PIN 7	
lil ISR.o - (ARM/le)			111 #define GPIO_PuPAB_C2_IOMUX	(ICHUX_PINCM14)	
iii main.o - (ARM/le)			112 #define GPIO_PWMAB_C2_IOMUX_FUNC	IONUX_PINCH14_PF_TIMA8_CCP2	
o sensoriess-foc TIDA010250 MSPM0G15	07 nortos ticlang.out - (AR	M/el	113 Bdefine GPIO_PWHAB_C2_IDX 114/* GPIO defines for channel 3 */	DL_TIMER_CC_2_INDEX	
III ccsObjs.opt			115 Bdefine GPIO PWWAB C3 PORT	GPIOB	
ii ISR.d			116 #define GPIO_PWMA@_C3_PIN	DL_GPI0_PIN_2	
🖩 main.d			117 #define GPIO_PWMA@_C3_IOMUX	(IOHUX_PINCH15)	
G makefile			118 #define GPIO_PWMA@_C3_IOMUX_FUNC	IONUX_PINCH15_PF_TIMA8_CCP3	
la objects.mk					
sensorless-foc_TIDA010250_MSPM0G15	07_nortos_ticlang_linkinfo.x	and	Console ×		x ♦ ♦ % 0 Q = 10 € 0 = 10 = 10 = 1
sensorless-foc_TIDA010250_MSPM0G15	07_nortos_ticlang.map		CDT Build Console [sensorless-foc_TIDA010250_MSPM0G1507_n		
Gi sources.mk					*thumbv6m -mcpu+cortex-m0plus -mfloat-abi+soft -mlittle-endian -mthumb -02 -D_MSP9003507DGLOB
Ga subdir_rules.mk			Finished building target: "sensorless-foc_TIDA	elezse_MSPH061507_nortos_ticlang.out*	
la subdir vars.mk					
io modules			**** Build Finished ****		

Figure 4-20. GPIO, ADC, and PWM Defines

4.4.2 Motor and Control Parameters Tuning

For more motor and control parameters tuning, see the MSPM0 Sensorless FOC Tuning Guide application note.



5 Design and Documentation Support

5.1 Design Files

5.1.1 Schematics

To download the schematics, see the design files at TIDA-010250.

5.1.2 BOM

To download the bill of materials (BOM), see the design files at TIDA-010250.

5.2 Tools

Tools

Code Composer Code Composer Studio[®] (CCS) is an integrated development environment (IDE) for TI's microcontrollers and processors. CCS comprises a suite of tools used to develop and debug embedded applications. Code Composer Studio is available for download across Microsoft[®] Windows[®], Linux[®] and macOS[®] desktops. CCS can also be used in the cloud by visiting the TI developer zone.

MSPM0-SDK The MSPM0 SDK provides the ultimate collection of software, tools and documentation to accelerate the development of applications for the MSPM0 MCU platform under a single software package.

5.3 Documentation Support

- 1. Texas Instruments, MSPM0G150x Mixed-Signal Microcontrollers Data Sheet
- 2. Texas Instruments, UCC28881 700-V, 225-mA Low Quiescent Current Off-Line Converter Data Sheet
- 3. Texas Instruments, UCC27712 620-V, 1.8-A, 2.8-A High-Side Low-Side Gate Driver with Interlock Data Sheet
- 4. Texas Instruments, *TLV906xS 10-MHz, RRIO, CMOS Operational Amplifiers for Cost-Sensitive Systems Data Sheet*
- 5. Texas Instruments, TPS54202 4.5-V to 28-V Input, 2-A Output, EMI Friendly Synchronous Step Down Converter Data Sheet
- 6. Texas Instruments, *ISO672x General Purpose Basic Dual-Channel Digital Isolators with Robust EMC Data Sheet*

5.4 Support Resources

TI E2E[™] support forums are an engineer's go-to source for fast, verified answers and design help — straight from the experts. Search existing answers or ask your own question to get the quick design help you need.

Linked content is provided "AS IS" by the respective contributors. They do not constitute TI specifications and do not necessarily reflect TI's views; see TI's Terms of Use.

5.5 Trademarks

TI E2E[™] are trademarks of Texas Instruments.

Arm[®] and Cortex[®] are registered trademarks of Arm Limited.

Code Composer Studio[®], Microsoft[®], and Windows[®] are registered trademarks of Microsoft Corporation. macOS[®] is a registered trademark of Apple Inc.

All trademarks are the property of their respective owners.



6 About the Author

HELY ZHANG is a System Application Engineer at Texas Instruments, where he is responsible for developing home appliance related power delivery and motor inverters. Hely earned his masters degree from Anhui University of Science and Technology with Power electronics in 2002, and worked in SolarEdge and General Electric before joining TI.

7 Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

С	hanges from Revision A (April 2024) to Revision B (September 2024)	Page
•	Established MSPM0G1507 as device used in this reference design	1
•	Added content to Section 4.2	12
•	Added Section 4.4	22

CI	nanges from Revision * (May 2023) to Revision A (April 2024)	Page
•	Replaced UCC54202 with TPS54202 in Section 2.2	4

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to TI's Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265 Copyright © 2024, Texas Instruments Incorporated