Application Note

AN2520/D
Rev. 0, 5/2003

BLDC Motor version I TPU Function Set (BLDCm)

By Milan Brejl, Ph.D.

Functional Overview

BLDC Motor version I (BLDCm) TPU function drives a Brush less DC Motor by decoding the Hall sensor signals and generating 3-phase complementary-switched center-aligned PWM signals. The state of the incoming Hall sensor signals is decoded into a position of the motion control system. Based on this, two of three phases generate the PWM signal while the third phase is switched off (see **Figure 1**). This way the TPU drives the motor independently of CPU. The CPU just sets a *dc* parameter in range (–1,1), which determines both the speed and the direction.

The function set consists of 5 TPU functions:

- 3-phase Hall Sensor Decoder for BLDC Motor (BLDCm_3HD)
- BLDC Motor (BLDCm)
- Synchronization Signal for BLDC Motor (BLDCm_sync)
- Resolver Reference Signal for BLDC Motor (BLDCm_res)
- Fault Input for BLDC Motor (BLDCm_fault)

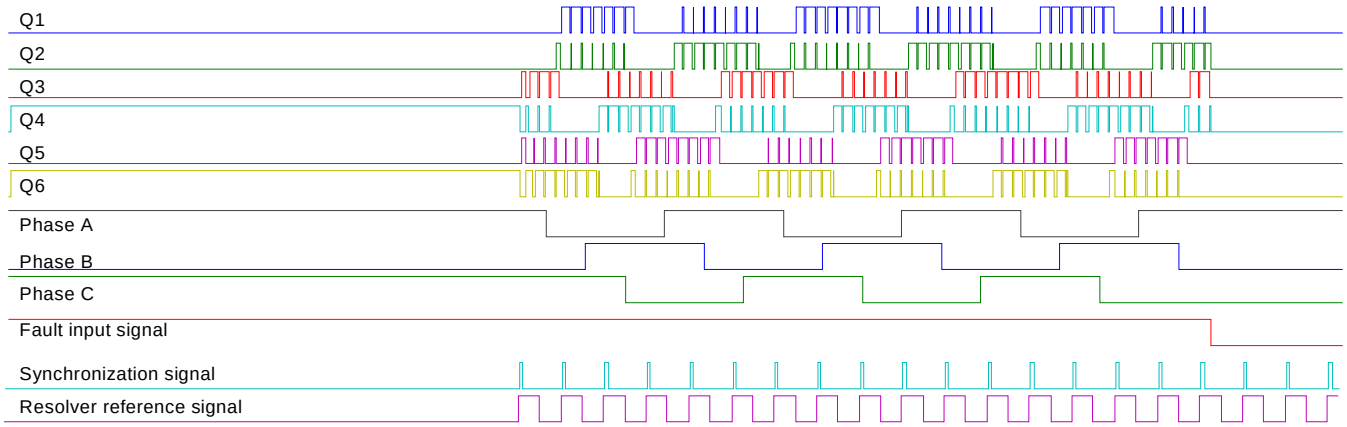


Figure 1. Signals processed by BLDCm TPU function set

The BLDCm_3HD function receives the Hall Sensor signals and controls the commutation of 3 PWM phases. The BLDCm TPU functions generate a 6-channel 3-phase center-aligned PWM signal with dead-time between the top and bottom channels. One of the phases is always switched off while the other two generate the PWM signal. The Synchronization Signal for BLDCm function can be used to generate one or more adjustable signals for a wide range of uses, which are synchronized to the PWM, and track changes in the PWM period. The Resolver Reference Signal for the BLDCm function can be used to generate one or more 50% duty-cycle adjustable signals that are also synchronized to the PWM.The Fault Input for BLDCm function is a TPU input function that sets all PWM outputs low when the input signal goes low. See **Figure 1**.

Function Set Configuration

None of the TPU functions in the BLDC Motor TPU function set can be used separately. The BLDCm_3HD and BLDCm functions have to be used together. The BLDCm_3HD is used on 3 input channels. The BLDCm on 6 output channels, and within each phase, the top channel has to be assigned on a lower TPU channel than the bottom channel. One or more channels running the Synchronization Signal for BLDCm as well as Resolver Reference Signals for BLDCm functions can be added. They can run with different settings on each channel. The function Fault Input for BLDCm can also be added. It is recommended to use it on channel 15, and to set the hardware option that disables all TPU output pins when the channel 15 input signal is low (DTPU bit = 1). This ensures that the hardware reacts quickly to a pin fault state. Note that it is not only the PWM channels, but all TPU output channels, including the synchronization signals, that are disabled in this configuration.

Table 1 shows the configuration options and restrictions.

Table 1. BLDCm TPU function set configuration options and restrictions

TPU function	Optional/ Mandatory	How many channels	Assignable channels
BLDCm_3HD	mandatory	3	any 3 channels
BLDCm	mandatory	6	any 6 channels, Q1 on a lower TPU channel than Q2, Q3 on a lower TPU channel than Q4, Q5 on a lower TPU channel than Q6
BLDCm_sync	optional	1 or more	any channels
BLDCm_res	optional	1 or more	any channels
BLDCm_fault	optional	1	any, recommended is 15 and DTPU bit set

Table 2 shows an example of configuration.

Table 2. Example of configuration

Channel	TPU function	Priority
0	BLDCm	high
1	BLDCm	high
2	BLDCm	high
3	BLDCm	high
4	BLDCm	high
5	BLDCm	high
6	BLDCm_3HD	high
7	BLDCm_3HD	high
8	BLDCm_3HD	high
10	BLDCm_sync	low
15	BLDCm_fault	high

Table 3 shows the TPU function code sizes.

Table 3. TPU function code sizes.

TPU function	Code size
BLDCm_3HD	61 μ instructions + 8 entries = 69 long words
BLDCm	213 μ instructions + 8 entries = 221 long words
BLDCm_sync	26 μ instructions + 8 entries = 34 long words
BLDCm_res	38 μ instructions + 8 entries = 46 long words
BLDCm_fault	9 μ instructions + 8 entries = 17 long words

Configuration Order

The CPU configures the TPU as follows.

1. Disables the channels by clearing the two channel priority bits on each channel used (not necessary after reset).
2. Selects the channel functions on all used channels by writing the function numbers to the channel function select bits.
3. Initializes function parameters. The parameters *PinAddrPrev_A*, *PinAddrNext_A, PinAddrPrev_B*, *PinAddrNext_B*, *T*, *DT*, *MPW* and *sync_presc_addr* must be set before initialization. If a BLDCm_sync channel or a BLDCm_res channel is used, then also its parameters must be set before initialization.
4. Issues an HSR (Host Service Request) type %10 to all BLDCm_3HD channels, to initialize them, and to one of the BLDCm_bottom channels to initialize all PWM channels. Issues an HSR type %10 to the BLDCm_sync channels, BLDCm_res channels and BLDCm_fault

channel, if used.

5. Enables servicing by assigning a high, middle or low priority to the channel priority bits. All Hall sensor channels, as well as all PWM channels, must be assigned the same priority to ensure correct operation. The CPU must ensure that BLDCm_bottom (which actually initializes all PWM channels) is initialized after the initialization of BLDCm_3HD, and that the BLDCm_sync or BLDCm_res function is initialized even after the initialization of BLDCm_bottom:
   - assign a priority to the Phase A, Phase B and Phase C Hall sensor channels to enable their initialization
   - wait until the HSR bits are cleared to indicate that initialization of these channels has completed
   - assign a priority to the PWM channels to enable their initialization
   - if a Synchronization Signal or a Resolver Reference Signal channel is used, wait until the HSR bits are cleared to indicate that initialization of the PWM channels has completed and
   - assign a priority to the BLDCm_sync or BLDCm_res channel to enable its initialization

NOTE: *A CPU routine that configures the TPU can be generated automatically using the MPC500_Quick_Start Graphical Configuration Tool.*

Detailed Function Description

3-phase Hall Sensor Decoder for BLDC Motor (BLDCm_3HD)

The BLDCm_3HD operates on three channels and processes the incoming Hall sensor signals. As a result of this processing, the Sector parameter gets a value that reflects the position of a motion system in one of six sectors. The state of the Hall sensor signals and the corresponding Sector value is listed in Table 4.

Table 4. Hall sensor signal states and corresponding Sector value

Phase A	Phase B	Phase C	Sector
1	0	0	4
1	1	0	6
0	1	0	2
0	1	1	3
0	0	1	1
1	0	1	5
0	0	0	0
1	1	1	7

A Sector value of 0 or 7 indicates an illegal state of the Hall sensor signals.

The Sector value history determines the direction of the motion system. The Direction parameter can be assigned a value of 0 or 1. See **Table 5**.

Table 5. Sector value sequence and corresponding Direction value

Sector value sequence	Direction
4, 6, 2, 3, 1, 5, 4, ...	0
4, 5, 1, 3, 2, 6, 4, ...	1

The Period value is calculated each time the sector is changed. The Period value is the TCR time of last revolution. It is measured from the last edge of similar type (low-high / high-low), on the same channel, to the current edge – see **Figure 2**. This method eliminates inaccuracies in the Hall sensor signals. The Period parameter does not contain a valid value during the first revolution after initialization, or after a change of direction.

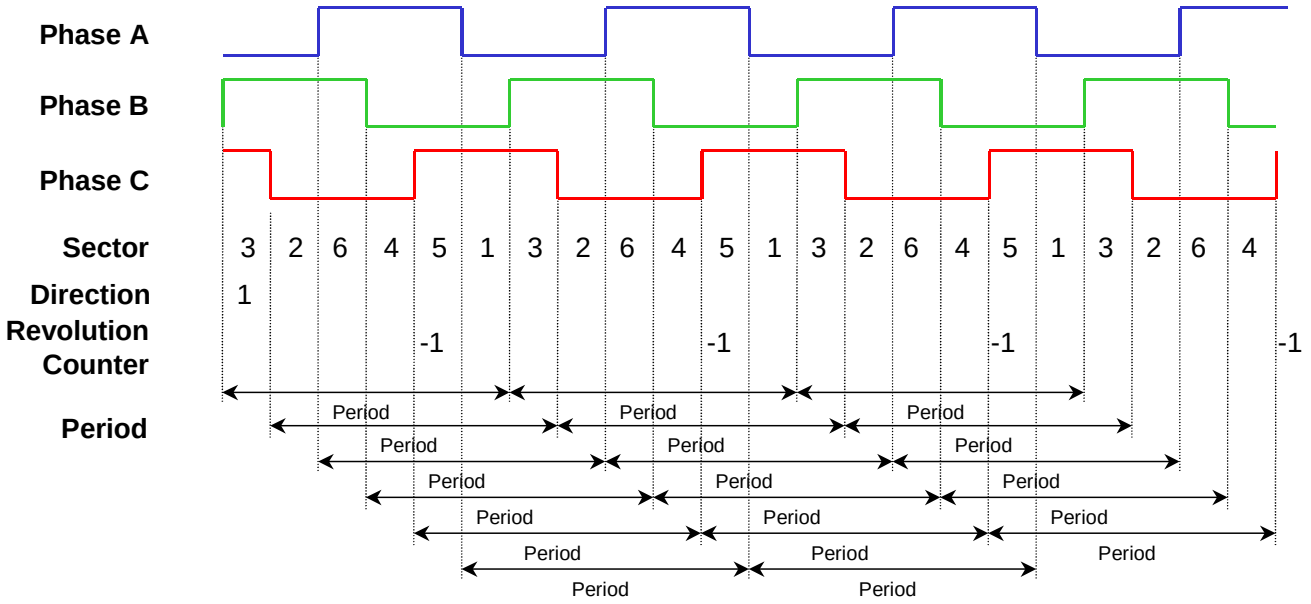


Figure 2. Hall sensor signals and corresponding values

2 function modes are provided:

- TCR1 clock selected
- TCR2 clock selected

The selected mode is determined by the HSQ bit 1. The user has to select the same mode on all channels.

The function provides interpolation support. The parameters LastEdgeT and ActualT are updated on a Host Service Request HSR = 11. LastEdgeT then

has the value of the last incoming edge time in TCR clocks and ActualT has the current value of the TCR clock.

The CPU program should use 32-bit reads to ensure coherency of the two parameters. This applies to coherent reads of LastEdgeT and ActualT as well as the Sector and TCR_VALUE, which is necessary for interpolation calculations.

Host Interface

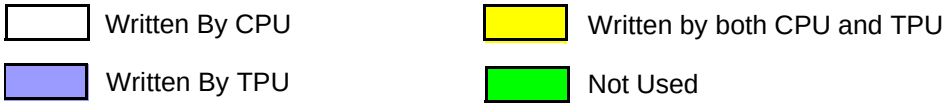


Table 6. BLDCm_3HD Control Bits

Name	Options
3 2 1 0 Channel Function Select	BLDCm_3HD function number (Assigned during assembly the DPTRAM code from library TPU functions)
1 0 Channel Priority	00 – Channel Disabled 01 – Low Priority 10 – Middle Priority 11 – High Priority
1 0 Host Service Bits (HSR)	00 – No Host Service Request 01 – Not used 10 – Initialization 11 – Get LastEdgeT and ActualT
1 0 Host Sequence Bits (HSQ)	x0 – Phase A or Phase B x1 – Phase C 0x – TCR1 clock selected 1x – TCR2 clock selected
0 Channel Interrupt Enable	0 – Channel Interrupt Disabled 1 – Channel Interrupt Enabled
0 Channel Interrupt Status	0 – Interrupt Not Asserted 1 – Interrupt Asserted

TPU function BLDCm_3HD generates an interrupt each time the Sector is changed.

Table 7. BLDCm_3HD Parameter RAM

Channel	Parameter	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Phase A	0																
	1																
	2																
	3	PinAdrPrev_A															
	4	PinAdrNext_A															
	5	PINSTATE_A															
	6	EdgeT_LH_A															
	7	EdgeT_HL_A															
Phase B	0	LastEdgeT															
	1	ActualT															
	2																
	3	PinAdrPrev_B															
	4	PinAdrNext_B															
	5	PINSTATE_B															
	6	EdgeT_LH_B															
	7	EdgeT_HL_B															
Phase C	0	Direction															
	1	RC															
	2	TCR_VALUE															
	3	Sector															
	4	Period															
	5	PINSTATE_C															
	6	EdgeT_LH_C															
	7	EdgeT_HL_C															

Table 8. BLDCm_3HD parameter description

Parameter	Format	Description
Parameters written by CPU		
PinAdrPrev_A	16-bit unsigned integer	$00XA, where X is a number of Phase C channel
PinAdrNext_A	16-bit unsigned integer	$00XA, where X is a number of Phase B channel
PinAdrPrev_B	16-bit unsigned integer	$00XA, where X is a number of Phase A channel
PinAdrNext_B	16-bit unsigned integer	$00XA, where X is a number of Phase C channel
Parameters written by both TPU and CPU		
RC	16-bit signed integer	Revolution Counter value
Parameters written by TPU		
LastEdgeT	16-bit unsigned integer	TCR time of last transition *

Table 8. BLDCm_3HD parameter description

Parameter	Format	Description
ActualT	16-bit unsigned integer	Actual TCR time *
Direction	0 or 1	Direction 0 – Sector sequence 4, 6, 2, 3, 1, 5, 4, ... 1 – Sector sequence 4, 5, 1, 3, 2, 6, 4, ...
TCR_VALUE	16-bit unsigned integer	TCR time of last transition
Sector	4, 6, 2, 3, 1 or 5	Sector: position in one of six sectors
Period	16-bit unsigned integer	Period: time of last revolution in TCR clocks.
PINSTATE_A PINSTATE_B PINSTATE_C	$0000 or $0001	The actual state of the pin is $0001 – high, $0000 – low
EdgeT_LH_A EdgeT_LH_B EdgeT_LH_C	16-bit unsigned integer	TCR time of last low-high transition
EdgeT_HL_A EdgeT_HL_B EdgeT_HL_C	16-bit unsigned integer	TCR time of last high-low transition
* The parameter values are entered by TPU on Host Service Request 11 (Get LastEdgeT and ActualT).		

Performance

Table 9. BLDCm_3HD State Statistics

State	Max IMB Clock Cycles	RAM Accesses by TPU
INIT	22	5
GET_TIME	8	3
LH1	28	12
HL1	28	12
LH2	32	13
HL2	32	13

NOTE: *Execution times do not include the time slot transition time (TST = 10 or 14 IMB clocks)*

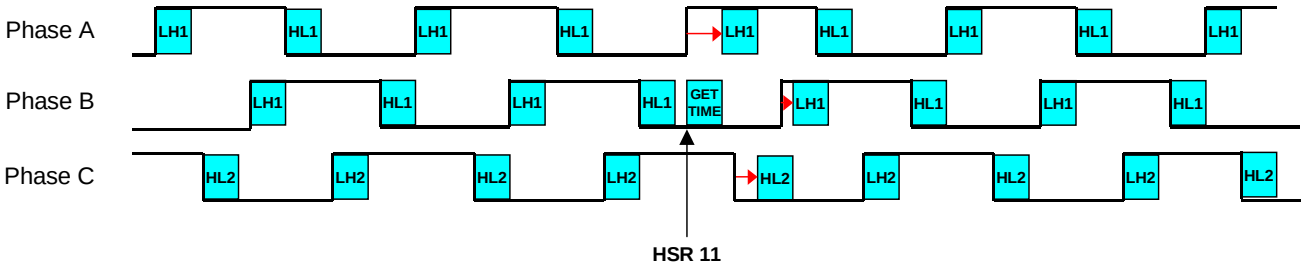


Figure 3. BLDCm_3HD timing

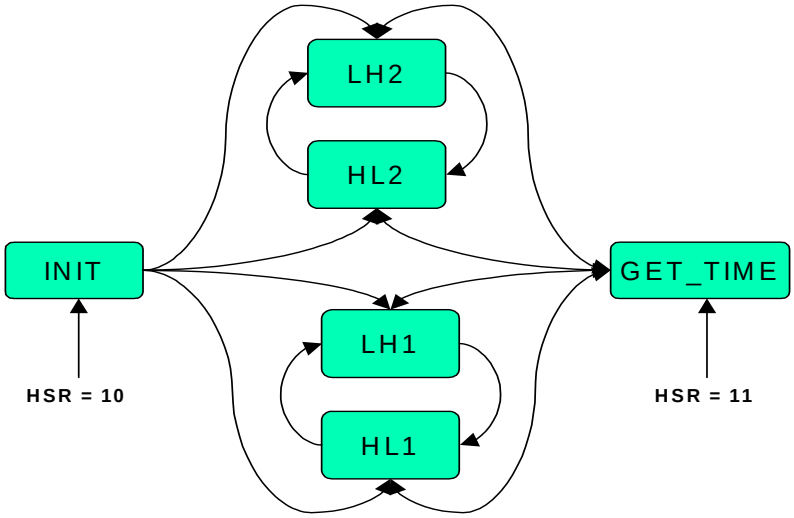


Figure 4. BLDCm_3HD state diagram

Noise Immunity

The input signals can be disturbed by an impulse noise. The TPU hardware rejects short input pulses of less than a configurable number of IMB clocks. Longer pulses are processed by the TPU. Furthermore the function itself uses a pin history to reject short error pulses that are long enough to get through the hardware filter, but not long enough to last from the actual transition time to the time that the TPU services the channel. Even longer error pulses are counted on both edges, resulting in a short-time error of the Sector value.

BLDC Motor (BLDCm)

The BLDCm TPU function generates a 6-channel, 3-phase PWM signal, with dead-time between the top and bottom channels. In order to charge the bootstrap transistors, the PWM signals start to run 1.6ms after their initialization (at 20MHz TCR1 clock). The functions generate signals corresponding to value

of 0 for duty-cycle ratio *dc* until the first *dc* value is processed. This accures for at least one PWM period.

One of the phases is always switched off while the other two generate the PWM signal. Which phase is switched off, depends on which one of the six sectors the rotor is positioned in. The position is decoded from the Hall sensor signals.

The CPU controls the PWM output by setting the TPU parameters. The duty-cycle ratio *dc* and PWM period *T* can be adjusted during run time. Conversely, dead-time (*DT*) and minimum pulse width (*MPW*) are not supposed to be changed during run time. The duty-cycle ratio *dc* can gain a value in the range (–1, 1). The sign controls the motion system direction, while the absolute value controls the amplitude of the applied voltage.

The following figures show the input *dc* value and corresponding output PWM signals (valid for sector 4, direction 0):

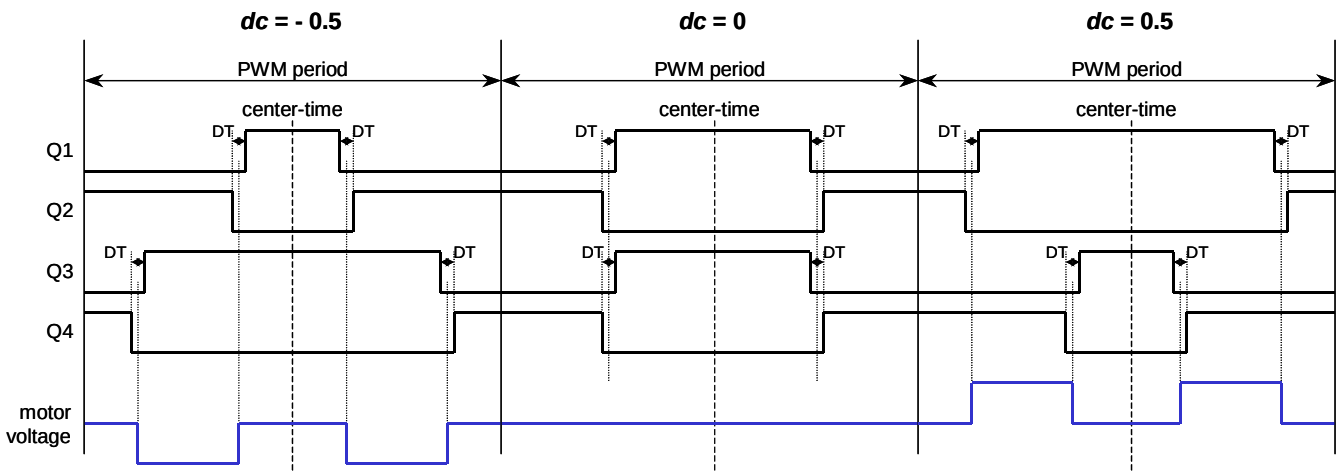


Figure 5. Unipolar switching

The following equations describe how the PWM signal transition times *T1p*, *T2p*, *B1p*, *B2p*, *T1m*, *T2m*, *B1m*, *B2m* are calculated, and how they are assigned to each channel transition time t_{LH} and t_{HL} based on BLDCm commutation rules:

$$Tdc_{\alpha} = T \cdot dc$$

$$X = \frac{T + Tdc}{2}$$

$$Y = \frac{T - Tdc}{2}$$

$$A = \frac{X - DT}{2} \qquad C = \frac{Y - DT}{2}$$

$$B = \frac{X + DT}{2} \qquad D = \frac{Y + DT}{2}$$

$$T1p = center\_time - A \qquad T2p = center\_time + A$$

$$B1p = center\_time - B \qquad B2p = center\_time + B$$

$$T1m = center\_time - C \qquad T2m = center\_time + C$$

$$B1m = center\_time - D \qquad B2m = center\_time + D$$

Table 10. Assignment of the calculated transition times to each channel transition times t_{LH} and t_{HL} based on the BLDCm commutation rules

Channel	Sector					
	4	6	2	3	1	5
Q1	$t_{LH} = T1p$ $t_{HL} = T2p$	$t_{LH} = T1p$ $t_{HL} = T2p$	OFF	$t_{LH} = T1m$ $t_{HL} = T2m$	$t_{LH} = T1m$ $t_{HL} = T2m$	OFF
Q2	$t_{LH} = B2p$ $t_{HL} = B1p$	$t_{LH} = B2p$ $t_{HL} = B1p$	OFF	$t_{LH} = B2p$ $t_{HL} = B1p$	$t_{LH} = B2p$ $t_{HL} = B1p$	OFF
Q3	$t_{LH} = T1m$ $t_{HL} = T2m$	OFF	$t_{LH} = T1p$ $t_{HL} = T2p$	$t_{LH} = T1p$ $t_{HL} = T2p$	OFF	$t_{LH} = T1m$ $t_{HL} = T2m$
Q4	$t_{LH} = B2p$ $t_{HL} = B1p$	OFF	$t_{LH} = B2p$ $t_{HL} = B1p$	$t_{LH} = B2p$ $t_{HL} = B1p$	OFF	$t_{LH} = B2p$ $t_{HL} = B1p$
Q5	OFF	$t_{LH} = T1m$ $t_{HL} = T2m$	$t_{LH} = T1m$ $t_{HL} = T2m$	OFF	$t_{LH} = T1p$ $t_{HL} = T2p$	$t_{LH} = T1p$ $t_{HL} = T2p$
Q6	OFF	$t_{LH} = B2p$ $t_{HL} = B1p$	$t_{LH} = B2p$ $t_{HL} = B1p$	OFF	$t_{LH} = B2p$ $t_{HL} = B1p$	$t_{LH} = B2p$ $t_{HL} = B1p$

Host Interface

Written By CPU

Written by both CPU and TPU

Written By TPU

Not Used

Table 11. BLDCm Control Bits

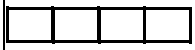
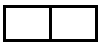
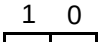
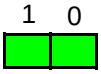
Name	Options
3 2 1 0  Channel Function Select	BLDCm function number (Assigned during assembly the DPTRAM code from library TPU functions)
1 0  Channel Priority	00 – Channel Disabled 01 – Low Priority 10 – Middle Priority 11 – High Priority
1 0  Host Service Bits (HSR)	00 – No Host Service Request 01 – Not used 10 – Initialization 11 – Stop
1 0  Host Sequence Bits (HSQ)	xx – Not used
0  Channel Interrupt Enable	x – Not used
0  Channel Interrupt Status	x – Not used

Table 12. BLDCm Parameter RAM

Channel	Parameter	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Q1	0	LHtime_Q1															
	1	HLtime_Q1															
	2	other_ch_Q1															
	3	C															
	4	Q120_chans															
	5	dc															
	6																
	7	fault_pinstate															

Table 12. BLDCm Parameter RAM

Channel	Parameter	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Q2	0	LHtime_Q2															
	1	HLtime_Q2															
	2	Tdc															
	3	center_time															
	4	old_idle_ch															
	5	DT															
	6																
	7																
Q3	0	LHtime_Q3															
	1	HLtime_Q3															
	2	other_ch_Q3															
	3	D															
	4	Q340_chans															
	5	T															
	6																
	7																
Q4	0	LHtime_Q4															
	1	HLtime_Q4															
	2	T_copy															
	3	plus_ch															
	4																
	5	MPW															
	6																
	7																
Q5	0	LHtime_Q5															
	1	HLtime_Q5															
	2	other_ch_Q5															
	3	minus_ch															
	4	Q560_chans															
	5	sync_presc_addr															
	6																
	7																
Q6	0	LHtime_Q6															
	1	HLtime_Q6															
	2	L															
	3	idle_ch															
	4																
	5																
	6																
	7																

Table 13. BLDCm parameter description

Parameter	Format	Description
Parameters written by CPU		
dc	16-bit fractional	duty-cycle ratio in the range <–1,1)
T	16-bit unsigned integer	PWM period in number of TCR1 TPU cycles
DT	16-bit unsigned integer	Dead-time in number of TCR1 TPU cycles
MPW	16-bit unsigned integer	Minimum pulse width in number of TCR1 TPU cycles. See **Performance** for details.
sync_presc_addr	8-bit unsigned integer	address of synchronization channel *prescaler* parameter: $X4, where X is synchronization channel number. $0 if no synchronization channel is used.
Parameters written by TPU		
fault_pinstate	0 or 1	If fault channel is used, state of fault pin: 0 ... low 1 ... high
Other parameters are just for TPU function inner use.		

Performance

Table 14. BLDCm State Statistics

State	Max IMB Clock Cycles	RAM Accesses by TPU
INIT	168	47
STOP	38	0
C10	6	1
C1	84	14
C1ACH	50	11
C20	16	2
C2	58	22
C2STC	58	10
LH	2	1
HL	12	2

Execution times do not include the time slot transition time (TST = 10 or 14 IMB clocks)

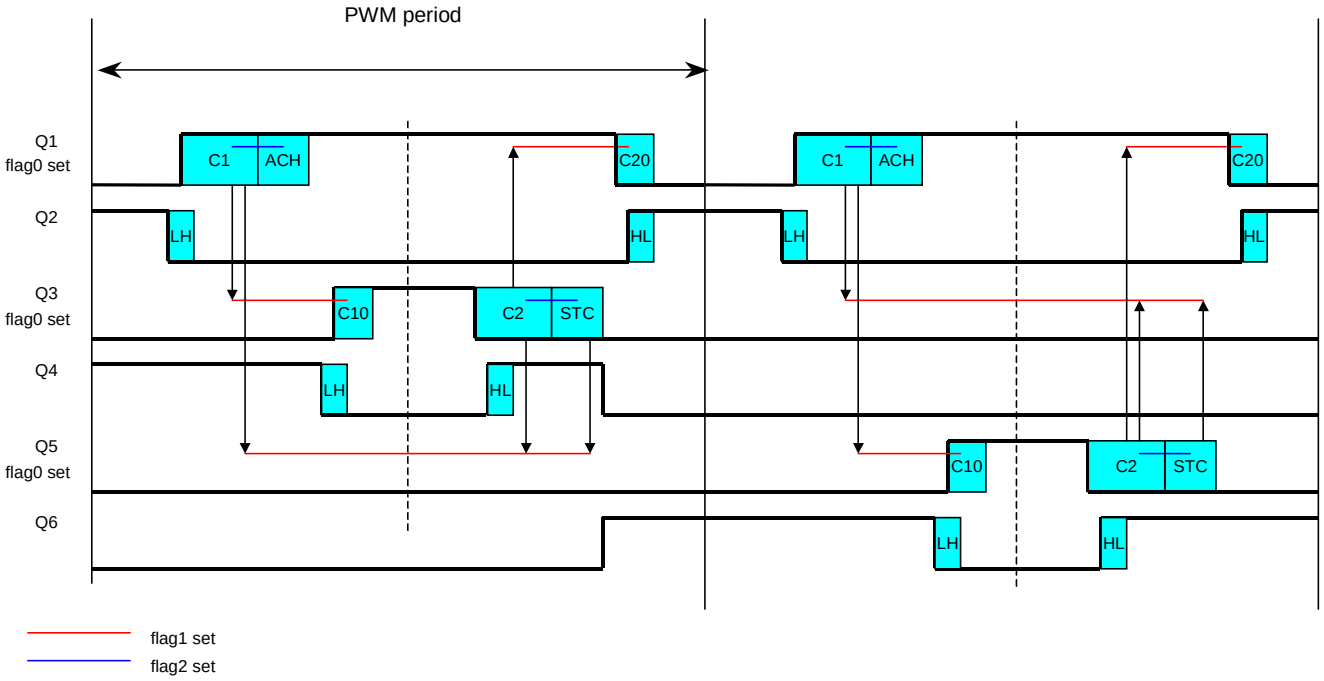


Figure 6. BLDCm timing

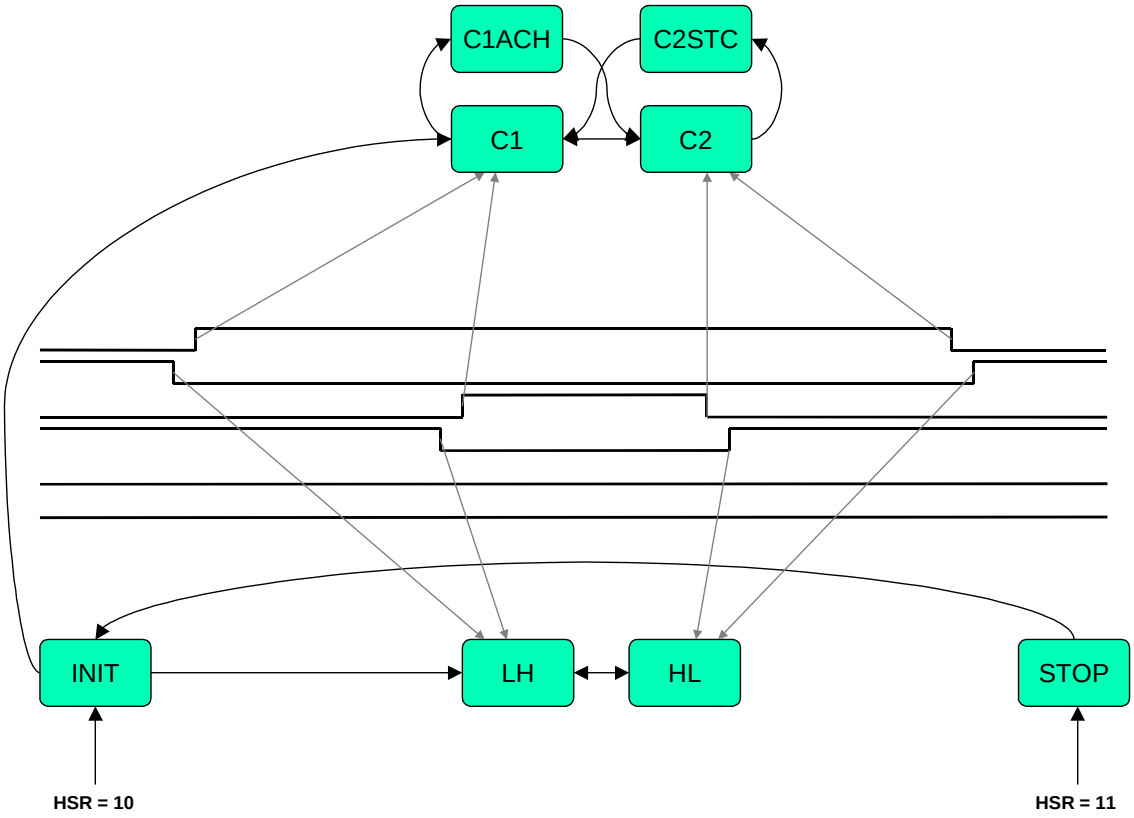


Figure 7. BLDCm state diagram

Minimum Pulse Width

The TPU cannot generate PWM signals with duty cycle ratios very close to 0% or 100%. It is the case when the dc value is close to 1 or –1. The minimum pulse width that the TPU can be guaranteed to correctly generate is determined by the TPU function itself and by the activity on the other channels. When the TPU function is requested to generate a narrower pulse a collision can occur. To prevent this, the parameter *MPW* (minimum pulse width) is introduced. The TPU function BLDCm limits the narrowest generated pulse widths to *MPW*. The CPU program should check the maximum absolute value of *dc* to avoid this limitation, or take account of the non-linear performance when *dc* moves towards the boundary values and the limitation is exhibited by the TPU. The maximum absolute value of *dc* should satisfy:

$$|dc| \leq 1 - \frac{2(MPW + DT)}{T}$$

The *MPW* is written by the CPU. The *MPW* depends on the whole TPU unit configuration, especially the lengths of the longest states of other functions, and their priorities, running on the same TPU. The *MPW* has to be correctly calculated at the time the whole TPU unit is configured.

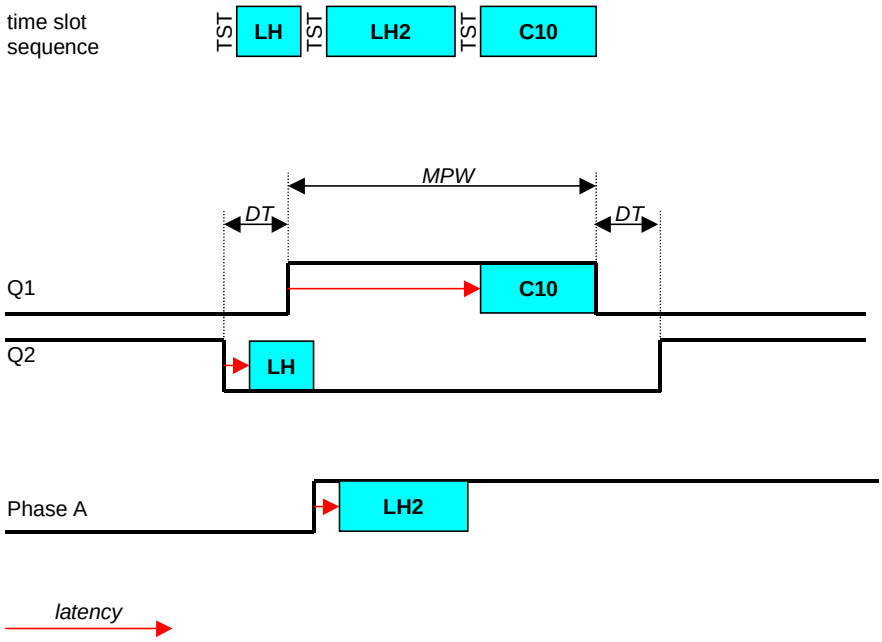


Figure 8. Worst case timing – case one

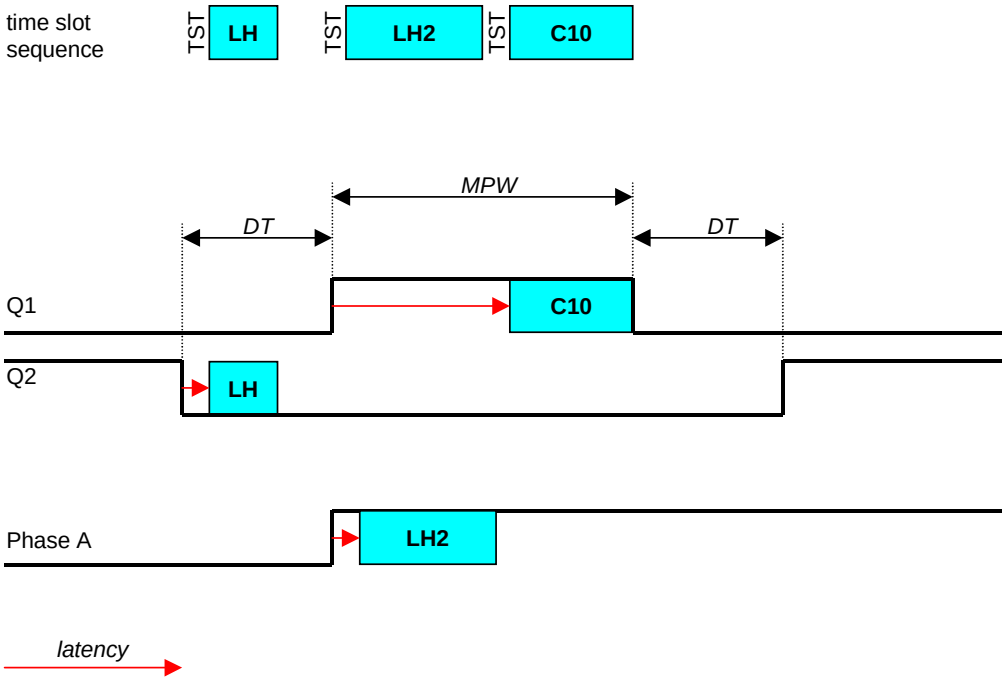


Figure 9. Worst case timing – case two

The minimum pulse width can be calculated according to **Figure 8** or **Figure 9**. These illustrate two possible worst cases of timing in the case when only compulsory BLDCm functions are running on one TPU (BLDCm and BLDCm_3HD). The worst case happens when a transition on one of the Hall sensor signals is detected and processed during the shortest pulse on a PWM channel. Each state is preceded by the Time Slot Transition (TST), which takes 10 or 14 IMB clock cycles.

According to the **Figure 8** the *MPW* is 70 IMB clock cycles – *DT*. According to **Figure 9** the *MPW* is 58 IMB clock cycles. In summary the MPW parameter value is equal to 70 IMB clock cycles – *DT*, and has a minimum value of at least 58 IMB clock cycles.

Note that the *MPW*, as well as the *DT*, are not entered into the parameter RAM in IMB clock cycles, but in TCR1 clock cycles. It is recommended for the BLDCm function that the TCR1 clock is configured for its maximum speed, which is the IMB clock divided by 2. In this case the *MPW* = 35 – *DT*, with a minimum value of 29.

When other functions are running concurrently on the same TPU, the longest state of each function with its time-slot transition can increase the calculated *MPW* value. The BLDCm_fault function does not affect the *MPW*. The BLDCm_sync, if used, increase the *MPW* value by 22 (44 IMB clock cycles). The BLDCm_res, if used, increase the *MPW* value by 20 (40 IMB clock cycles).

If a lower value than the one calculated, is set for the *MPW* parameter, the motion system can run with a higher motor voltage amplitude, but with a risk, that the dead-time is not maintained.

It is also possible to use the Worst-Case Latency (WCL), which is automatically calculated by the MPC500_Quick_Start Graphical Configuration Tool. It can serve as a good approximation of *MPW*. The calculated WCL is always longer than the real-case is. Let the WCL be calculated after the configuration of TPU channels and then find the longest WCL value within all BLDCm PWM channels. Convert the number, from IMB clock cycles to TCR1 clock cycles, to get the *MPW*.

Synchronization signal for BLDC Motor (BLDCm_sync)

The BLDCm_sync TPU function uses information obtained from BLDCm PWM functions, the actual PWM center times and the PWM periods. This allows a signal to be generated, which tracks the changes in the PWM period and is always synchronized with the PWM. The synchronization signal is a positive pulse generated repeatedly after the *prescaler* or *presc_copy* PWM periods (see next paragraph). The low to high transition of the pulse can be adjusted by a parameter, either negative or positive, to go a number of TCR1 TPU cycles before or after the PWM period center time. The pulse width *pw* is another synchronization signal parameter.

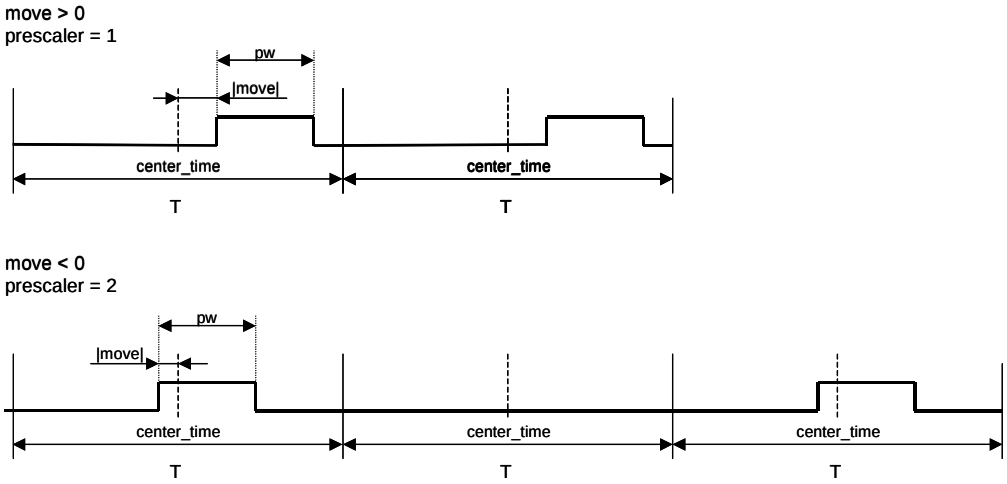


Figure 10. Synchronization signal adjustment examples

Synchronized Change of PWM Prescaler And Synchronization Signal Prescaler

The BLDCm_sync TPU function actually uses the *presc_copy* parameter instead of the *prescaler* parameter. The *prescaler* parameter holds the prescaler value that is copied to the *presc_copy* by the BLDCm_bottom function at the time the PWM parameters are reloaded. This ensures that new prescaler values for the PWM signals, as well as the synchronization signal, are applied at the same time. Write the synchronization signal *prescaler* parameter address to the *sync_presc_addr* parameter to enable this mechanism. Write 0 to disable it, and remember to set the synchronization signal *presc_copy* parameter instead of the *prescaler* parameter in this case.

Host Interface

Written By CPU

Written by both CPU and TPU

Written By TPU

Not Used

Table 15. BLDCm_sync Control Bits

Name	Options
3 2 1 0 Channel Function Select	BLDCm_sync function number (Assigned during assembly the DPTRAM code from library TPU functions)
1 0 Channel Priority	00 – Channel Disabled 01 – Low Priority 10 – Middle Priority 11 – High Priority

Table 15. BLDCm_sync Control Bits

Name	Options
1 0 Host Service Bits (HSR)	00 – No Host Service Request 01 – Not used 10 – Initialization 11 – Not used
1 0 Host Sequence Bits (HSQ)	xx – Not used
0 Channel Interrupt Enable	0 – Channel Interrupt Disabled 1 – Channel Interrupt Enabled
0 Channel Interrupt Status	0 – Interrupt Not Asserted 1 – Interrupt Asserted

TPU function BLDCm_sync generates an interrupt after each low to high transition.

Table 16. BLDCm_sync Parameter RAM

Channel	Parameter	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Synchronization channel	0	move															
	1	pw															
	2	prescaler															
	3	presc_copy															
	4	time															
	5	dec															
	6	T_copy															
	7																

Table 17. BLDCm_sync parameter description

Parameter	Format	Description
Parameters written by CPU		
move	16-bit signed integer	The number of TCR1 TPU cycles to forego (negative) or come after (positive) the PWM period center time
pw	16-bit unsigned integer	Synchronization pulse width in number of TCR1 TPU cycles.

Table 17. BLDCm_sync parameter description

Parameter	Format	Description
prescaler	16-bit unsigned integer	The number of PWM periods per synchronization pulse – use in case of synchronized prescalers change
presc_copy	16-bit unsigned integer	The number of PWM periods per synchronization pulse – use in case of asynchronized prescalers change
Parameters written by TPU		
Other parameters are just for TPU function inner use.		

Performance

There is one limitation. The absolute value of parameter *move* has to be less than a quarter of the PWM period *T*.

$$|move| < \frac{T}{4}$$

Table 18. BLDCm_sync State Statistics

State	Max IMB Clock Cycles	RAM Accesses by TPU
INIT	12	5
S1	12	6
S2	8	3
S3	16	7

NOTE: *Execution times do not include the time slot transition time (TST = 10 or 14 IMB clocks)*

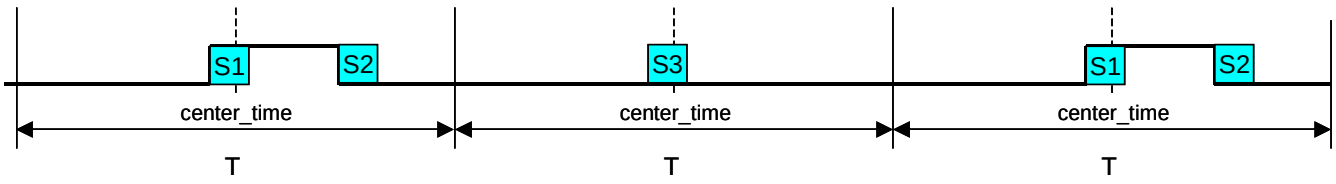


Figure 11. BLDCm_sync timing

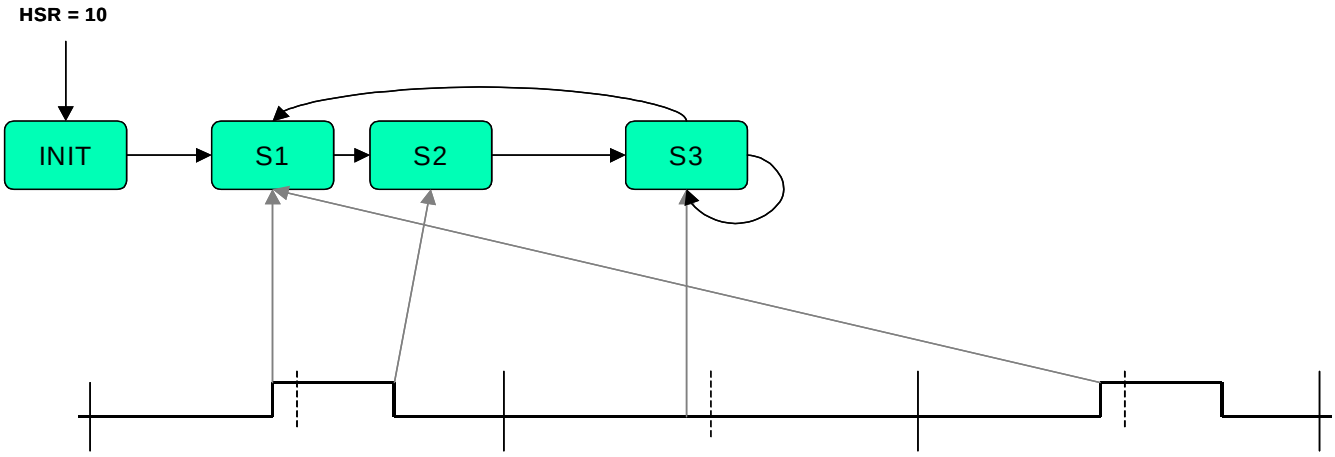


Figure 12. BLDCm_sync state diagram

Resolver Reference Signal for BLDC Motor (BLDCm_res)

The BLDCm_res TPU function uses information read from the BLDCm PWM functions, the actual PWM center times and the PWM periods. This allows a signal to be generated, which tracks the changes of the PWM period and is always synchronized with the PWM. The resolver reference signal is a 50% duty-cycle signal with a period equal to *prescaler* or synchronization channel *presc_copy* PWM periods (see next paragraph). The low to high transition of the pulse can be adjusted by a parameter, either negative or positive, to go a number of TCR1 TPU cycles before or after the PWM period center time.

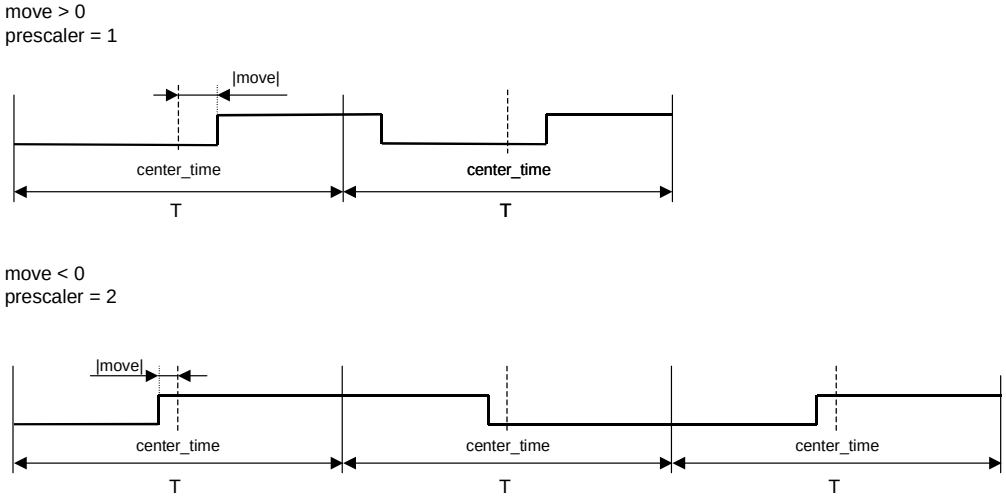


Figure 13. Resolver reference signal adjustment examples

Synchronized Change of PWM Prescaler And Resolver Reference Signals Prescaler

The BLDCm_res TPU function can inherit the Synchronization Signal prescaler that is synchronously changed with the PWM prescaler. Write the synchronization signals *presc_copy* parameter address to the *presc_addr* parameter to enable this mechanism. Write 0 to disable it, and in this case set the *prescaler* parameter to directly specify prescaler value.

Host Interface

Written By CPU

Written by both CPU and TPU

Written By TPU

Not Used

Table 19. BLDCm_res Control Bits

Name	Options
3 2 1 0 Channel Function Select	BLDCm_res function number (Assigned during assembly the DPTRAM code from library TPU functions)
1 0 Channel Priority	00 – Channel Disabled 01 – Low Priority 10 – Middle Priority 11 – High Priority
1 0 Host Service Bits (HSR)	00 – No Host Service Request 01 – Not used 10 – Initialization 11 – Not used
1 0 Host Sequence Bits (HSQ)	xx – Not used
0 Channel Interrupt Enable	x – Not used
0 Channel Interrupt Status	x – Not used

Table 20. BLDCm_res Parameter RAM

Channel	Parameter	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Resolver	0	move															
	1																
	2	presc_addr															
	3	prescaler															
	4	time															
	5	dec															
	6	T_copy															
	7																

Table 21. BLDCm_res parameter description

Parameter	Format	Description
Parameters written by CPU		
move	16-bit signed integer	The number of TCR1 TPU cycles to forego (negative) or come after (positive) the PWM period center time
presc_addr	16-bit unsigned integer	$00X6, where X is a number of Synchronization Signal channel, to inherit Sync. channel prescaler or $0000 to enable direct specification of prescaler value in prescaler parameter
prescaler	1, 2, 4, 6, 8, 10, 12, 14, ...	The number of PWM periods per synchronization pulse – use when apresc_addr = 0
Parameters written by TPU		
Other parameters are just for TPU function inner use.		

Performance

There is one limitation. The absolute value of parameter *move* has to be less than a quarter of the PWM period *T*.

$$|move| < \frac{T}{4}$$

Table 22. BLDCm_res State Statistics

State	Max IMB Clock Cycles	RAM Accesses by TPU
INIT	12	5
S1	26	9
S3	18	7

NOTE: *Execution times do not include the time slot transition time (TST = 10 or 14 IMB clocks)*

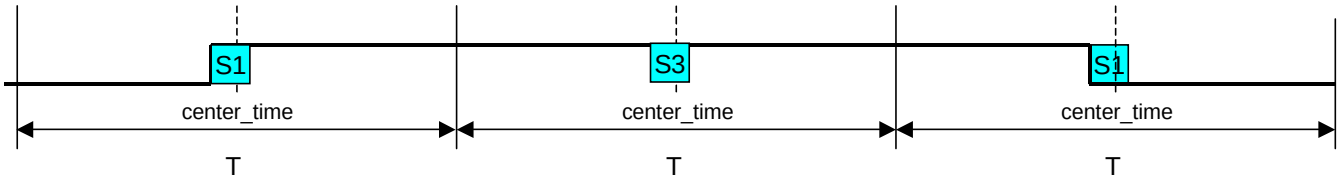


Figure 14. BLDCm_res timing

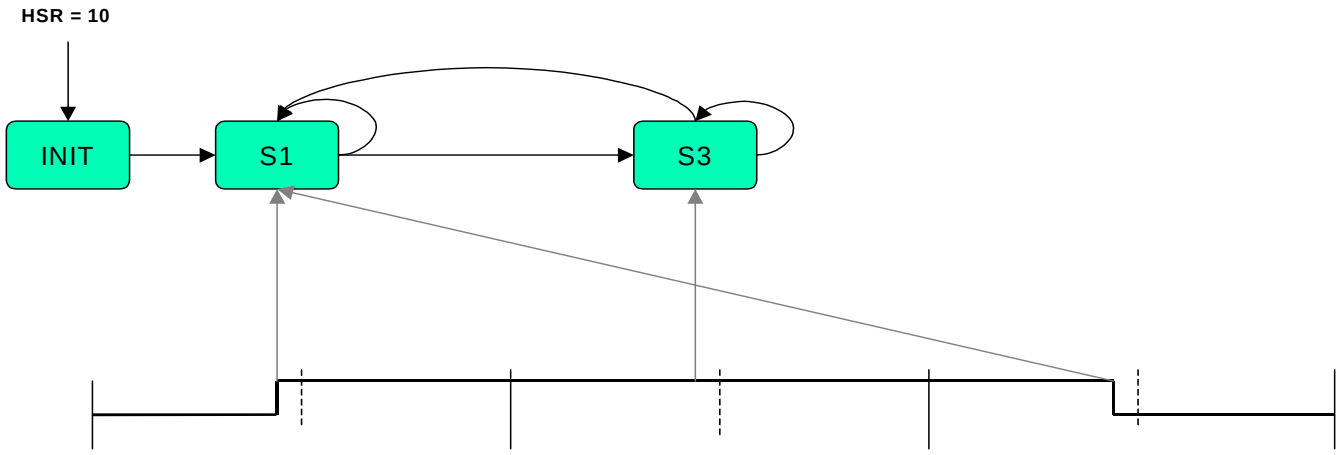


Figure 15. BLDCm_res state diagram

Fault Input for BLDC Motor (BLDCm_fault)

The BLDCm_fault is an input TPU function that monitors the pin, and if a high to low transition occurs, immediately sets all PWM channels low and cancels all further transitions on them. The PWM channels, as well as the synchronization and resolver reference signal channels (if used), have to be initialized again to start them running.

The function returns the actual pinstate as a value of 0 (low) or 1 (high) in the parameter *fault_pinstate*. The parameter is placed on the Q1 channel to keep the fault channel parameter space free.

Host Interface

Written By CPU
Written by both CPU and TPU
Written By TPU
Not Used

Table 23. BLDCm_fault Control Bits

Name	Options
3 2 1 0 Channel Function Select	BLDCm_fault function number (Assigned during assembly the DPTRAM code from library TPU functions)
1 0 Channel Priority	00 – Channel Disabled 01 – Low Priority 10 – Middle Priority 11 – High Priority

Table 23. BLDCm_fault Control Bits

Name	Options
1 0 Host Service Bits (HSR)	00 – No Host Service Request 01 – Not used 10 – Initialization 11 – Not used
1 0 Host Sequence Bits (HSQ)	xx – Not used
0 Channel Interrupt Enable	0 – Channel Interrupt Disabled 1 – Channel Interrupt Enabled
0 Channel Interrupt Status	0 – Interrupt Not Asserted 1 – Interrupt Asserted

TPU function BLDCm_fault generates an interrupt when a high to low transition appears.

Table 24. BLDCm_fault Parameter RAM

Channel	Parameter	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Fault input	0																
	1																
	2																
	3																
	4																
	5																
	6																
	7																

Table 25. BLDCm_fault parameter description

Parameter	Format	Description
Parameters written by TPU		
fault_pinstate	0 or 1	State of fault pin: 0 ... low 1 ... high

Performance

Table 26. BLDCm_fault State Statistics

State	Max IMB Clock Cycles	RAM Accesses by TPU
INIT	8	2
FAULT	44	1
NO_FAULT	4	1

NOTE: *Execution times do not include the time slot transition time (TST = 10 or 14 IMB clocks)*

Figure 16. BLDCm_fault timing

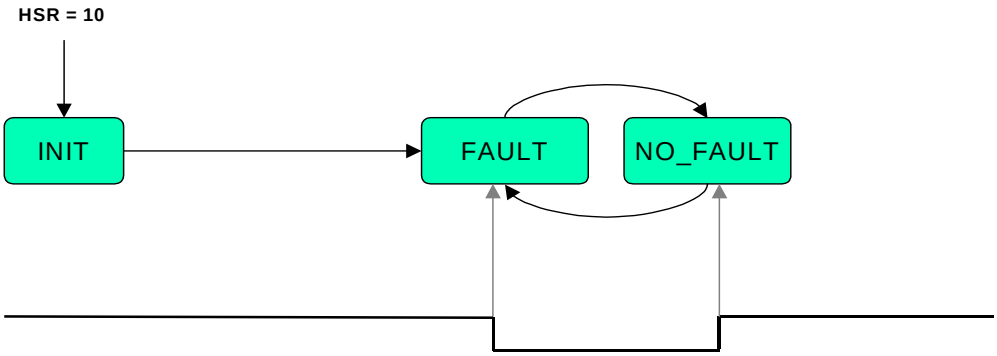


Figure 17. BLDCm_fault state diagram

[redacted]

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

freescale™
semiconductor

AN2520/D
Rev. 0
5/2003