Application Note, V1.0, May 2007

AP08059

XC886/888 CM/CLM

8-Bit Flash Microcontroller

Sensorless Field Oriented Control for PMSM Motors

Microcontrollers

Never stop thinking

Edition 2007-05

Published by
Infineon Technologies AG
81726 München, Germany

LEGAL DISCLAIMER

THE INFORMATION GIVEN IN THIS APPLICATION NOTE IS GIVEN AS A HINT FOR THE IMPLEMENTATION OF THE INFINEON TECHNOLOGIES COMPONENT ONLY AND SHALL NOT BE REGARDED AS ANY DESCRIPTION OR WARRANTY OF A CERTAIN FUNCTIONALITY, CONDITION OR QUALITY OF THE INFINEON TECHNOLOGIES COMPONENT. THE RECIPIENT OF THIS APPLICATION NOTE MUST VERIFY ANY FUNCTION DESCRIBED HEREIN IN THE REAL APPLICATION. INFINEON TECHNOLOGIES HEREBY DISCLAIMS ANY AND ALL WARRANTIES AND LIABILITIES OF ANY KIND (INCLUDING WITHOUT LIMITATION WARRANTIES OF NON-INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS OF ANY THIRD PARTY) WITH RESPECT TO ANY AND ALL INFORMATION GIVEN IN THIS APPLICATION NOTE.

Information

For further information on technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies Office (**www.infineon.com**).

Warnings

Due to technical requirements components may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies Office.

Infineon Technologies Components may only be used in life-support devices or systems with the express written approval of Infineon Technologies, if a failure of such components can reasonably be expected to cause the failure of that life-support device or system, or to affect the safety or effectiveness of that device or system. Life support devices or systems are intended to be implanted in the human body, or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

XC886/888 CM/CLM

Revision History: V1.0, 2007-05

Previous Version(s):
none

Page	Subjects (major changes since last revision)

We Listen to Your Comments
Any information within this document that you feel is wrong, unclear or missing at all?
Your feedback will help us to continuously improve the quality of this document.
Please send your proposal (including a reference to this document) to:
mcdocu.comments@infineon.com

1 Introduction

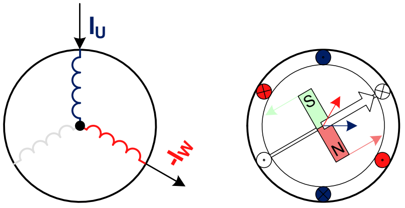
Field Oriented Control (FOC) is increasingly being used in consumer and industrial motor control owing to the higher efficiency, better dynamic response and less torque ripple. Typically a 16 or 32-bit MCU or DSP is required to handle the complex trigonometric equations. Infineon's XC886/888 offers the full benefits of FOC at a fraction of the cost compared to a DSP/DSC solution.

Compared to today's solution on the market, where FOC execution engages the whole central processing unit's (CPU) load, FOC execution on Infineon's XC886 and XC888 microcontrollers only requires 58% of the CPU's performance providing ample headroom for other applications. Unlike most hard-coded competitive FOC implementations, Infineon provides the additional benefit of software reprogrammability adding flexibility and potential for differentiation to the application. Field Oriented Control implemented on an 8-bit MCU is the perfect answer to energy related regulations and pricing pressure for appliance manufacturers.

1.1 Motor Theory

Motors rotate due to the torque produced by two interacting magnetic fields. On the one hand, there is the magnetic field from the permanent magnets mounted to the rotor. On the other hand, there is the magnetic field generated by the coils of the stator. The force $\vec{F}$, exerted on a current I carrying conductor with length $\vec{l}$ in the magnetic field $\vec{B}$, which is called Lorentz force, is defined as $\vec{F} = -(\vec{B} \times \vec{l}) \cdot I$.

When a force is given to a rotating system, it is called torque. It is defined by a linear force multiplied by a radius: $\vec{\tau} = \vec{r} \times \vec{F}$. The maximum torque is produced when the magnetic vector of the rotor is at 90° to the magnetic vector of the stator, because the poles of the rotor are forced to rotate in the direction of the stator field. The mechanical power P_{mech} is proportional to the angular speed ω. It is the torque which defines the ratio: $P_{mech} = \vec{\tau} \cdot \vec{\omega}$. On the other hand, the mechanical force is proportional to the current though the coils and the back EMF (electromotive force): $P_{elec} = I * V_{bemf}$.



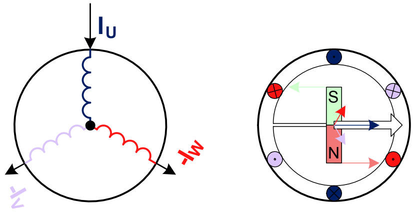
Torque_BC.emf

Figure 1 Magnetic Vectors in Block Commutation

See **Figure 1** for an illustration of the different vectors. The blue and red arrows represent the magnetic field caused by the stator coils. The big white arrow is the resulting stator field. The magnetic dipole represents the rotor field and the arrows at the poles (light red and green) represent the torque at the rotor.

In block commutation mode, the stator field is generated by a current flow through alternately two of the tree coils. The remaining third coil can be used to monitor the voltage of the connection point of the coils. By this information, the back EMF, a sensorless control can be achieved by moving the magnetic field blockwise sector by sector. As a result the stator magnetic vector is almost always misaligned with the rotor, causing an inconsistency in the amount of applied torque (torque ripple). Torque ripple can cause many problems, including mechanical wear, vibration, noise, and deficiencies in drive performance.

With sinusoidal commutation all three coils are used. As a result, the stator field will be stronger than in block commutation. See **Figure 2** for an illustration of the different vectors at sinusoidal commutation.



Torque_SC.emf

Figure 2 **Magnetic Vectors at Sinusoidal Commutation**

With sinusoidal commutation, the currents in the three coils follow a sine-wave which is phase shifted by an angle of 120°. As a result, the force created by the electromagnetic space vector is smoothly rotated along, so that the stator vector remains at a 90-degree angle from the rotor vector. An ideal motor with a sinusoidal drive does not show any torque ripple.

1.2 Three Phase Inverter

An inverter is an electronic circuit for converting direct current (DC) to alternating current (AC). A basic three-phase inverter consists of three single-phase inverter switches each connected to one of the three load terminals.

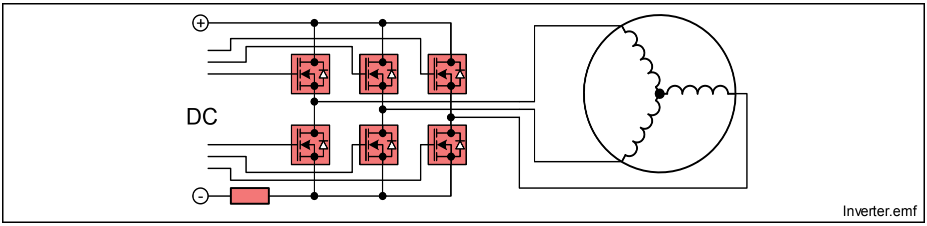


Figure 3 Three Phase Inverter with Motor

Figure 3 shows an inverter realized with MOSFETs which are preferred for low voltage drives. At high voltages, IGBTs are the common choice. In the current path of the DC terminals (+ and -) a shunt is available to measure the DC link current I_{DClink}.

The inputs of the power devices are connected via a driver circuit to the PWM outputs of the microcontroller. In order to avoid destruction of the devices, the switching and delay times of the system (driver and power switch) have to be considered. The time which is needed to delay the activation of one side after deactivation of the other side is called the deadtime of the inverter. Please refer to Figure 4 for details.

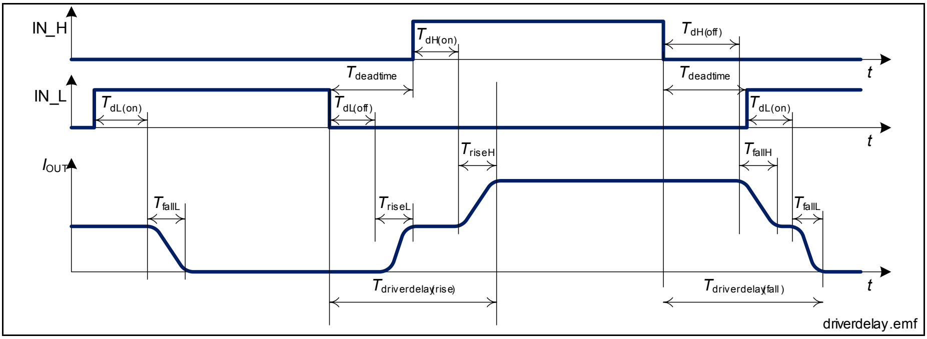


Figure 4 Timings During Switching

The timings can vary for the high-side and low-side switches as well as for switching on and off the power stages. When the dead time of the system is undercut, the high-side and the low-side switch is conducting current at the same time and a cross current begins to flow. The cross current bypasses the load and short-circuits the DC terminals.

The PWM unit of XC886/888 CM/CLM provides an integrated deadtime control which ensures proper control of the power stages at minimum software effort.

2 Sensorless Field Oriented Control (FOC)

A sensorless field oriented control (FOC) satisfies the advantages of a sinusoidal commutation by a minimum of system cost. Next to the inverter, there is just one shunt in the DC link necessary to acquire the phase currents.

2.1 Block Diagram

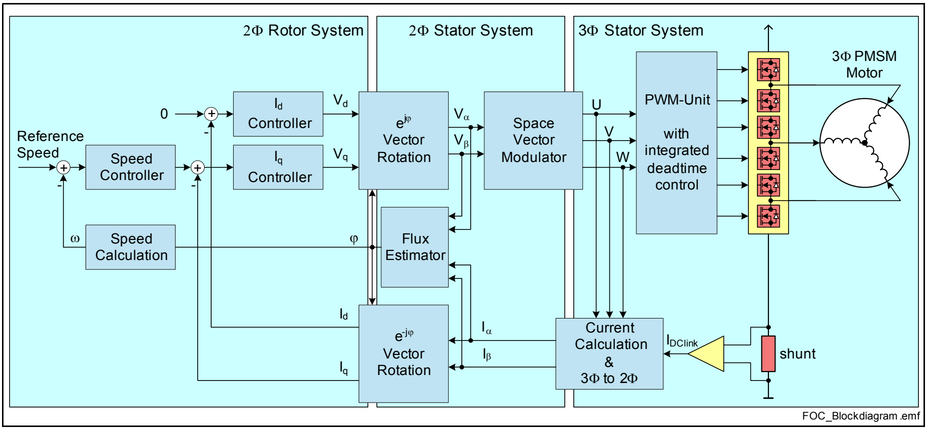


Figure 5 Block Diagram of Sensorless FOC

Figure 5 shows the block diagram of the sensorless speed control with field oriented control (FOC) of a permanent magnet synchronous motor (PMSM). From a control point of view, the FOC is comparable with that of a dc motor. The basic concept is a cascade control with the important difference that the electrical variables (V_d, I_d, V_q and I_q) are turning with the rotor. Thus the currents measured at the stator (I_α and I_β) have to be transformed into the rotor coordinates (I_d and I_q). The PI-controller for the currents is realized in the rotating system, whereas the field exciting d-component and the torque exciting q-component is controlled separately.

The speed controller adjusts - as for a dc motor - the reference value for the torque exciting current I_q. Due to the permanent magnets at the rotor, the reference value for the field exciting current I_d is set to zero.

The output of the current controllers represents the reference voltages (V_d and V_q) in the rotor coordinates. These values are transformed into the stator coordinates (V_α and V_β) in order to calculate the polar coordinates (norm and angle).

Using space vector pulse width modulation, the norm and angle values are converted to three phase currents by modulating the high-side and low-side switches of the power inverter.

2.2 Space Vector Pulse Width Modulation

In order to drive a three phase EC-motor with sinusoidal phase currents using a power inverter, the phase angle and norm of the torque vector can be generated by space vector pulse with modulation.

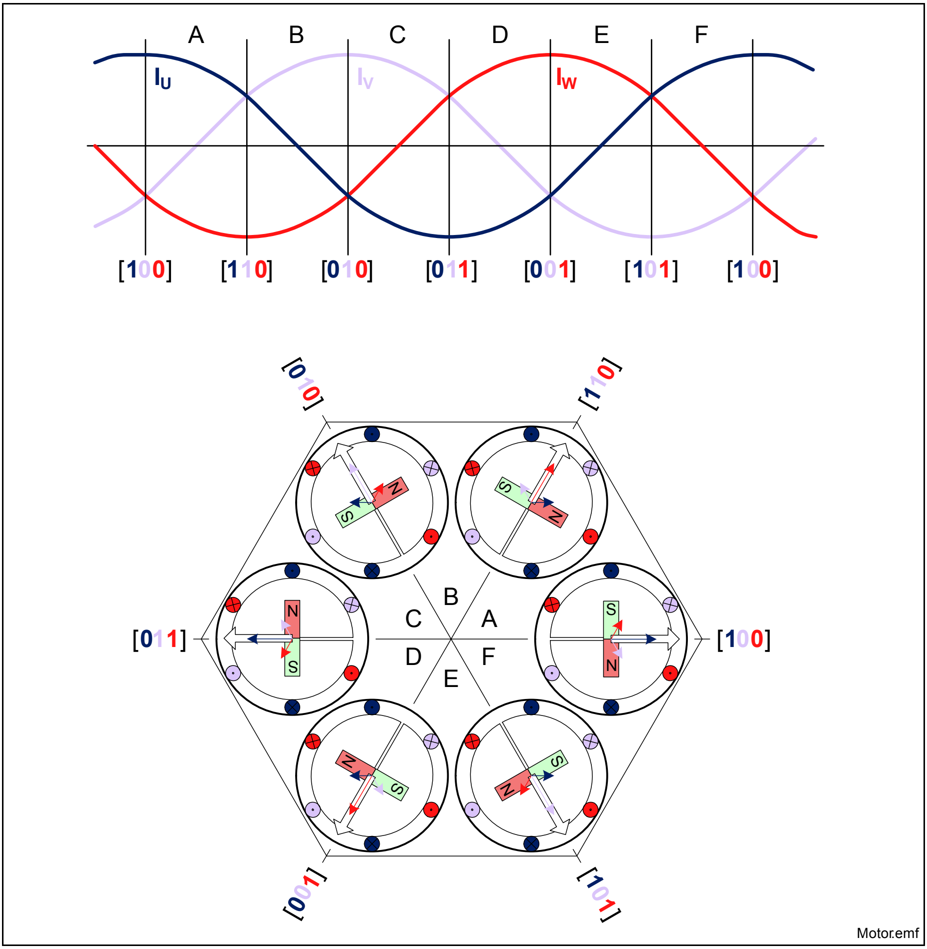


Figure 6 Space Vector Pulse With Modulation - Sectors

In a three-phase inverter, the three outputs can supply voltage at the motor to create one of six active states (100, 110, 010, 011, 001, 101). In addition, two inactive states that produce no voltage to the motor are possible (000, 111).

A space vector is a sinusoid whose center is able to "float" in space. The inactive states are used as an off time during the switching period when creating the space vector.

A three-phase space vector are represented by a hexagon which can be divided in six sectors. Any desired voltage-space vector will consist of a "real" voltage from one of the phases and an "imaginary" right-angle voltage created from the other two phases.

The space-vector algorithm will determine the time required at a first active state, a second active state, and an inactive state to produce the desired magnitude and angle of the space vector. See Figure 7 for an example in sector A. The first active state (b&f) is T_U, the second active state (c&e) is T_{-W}, the inactive state is T_0 which appears twice, first as (000) vector (a&f), second as (111) vector (d).

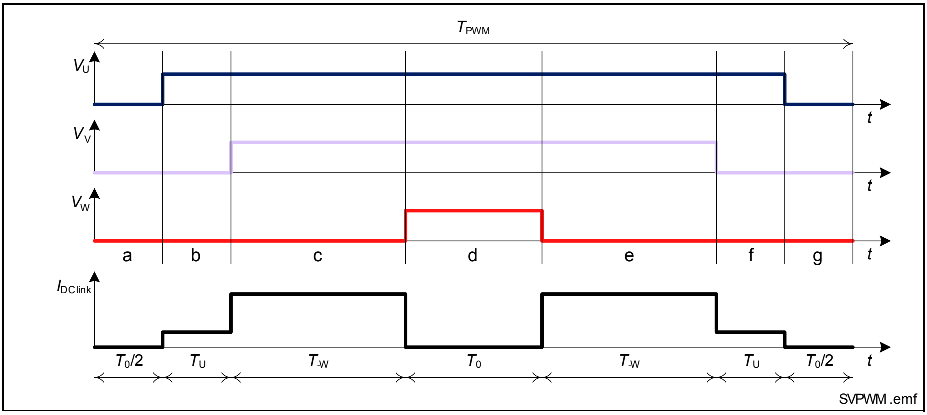


Figure 7 Voltage Space Vector - Sector A

If we translate the voltage waveform to the phasor diagram, we can see that the space-vector-modulation technique has a maximum predictable voltage of $V_{DC} \cdot \sqrt{3}$. A space-vector-modulation system does not constrain the phasor center, yielding a 15% increase in available motor voltage.

For a smooth rotation, it is the control of the sinusoidal current, not the shape of the voltage waveform that generates the magnetomotive force.

2.3 Acquisition of Actual Values

In order to estimate the rotor position by a single shunt measurement, the PWM pattern generation and the triggering of the ADC for current measurement must be very fast and accurate. Any jitter in the triggerpoint will influence the actual rotor’s angle estimation. As a result, the total harmonic distortion of the sinusoidal current signals will increase.

2.3.1 Current Measurement

The XC886/888 CM/CLM offers an event based hardware trigger from CapCom6E unit, used for generating the inverter signals, towards the ADC. The ADC provides in total four result registers, from which two are used to hold the appropriate DC-link current values I_{DClink}. The ADC sample time is as low as 250 ns. As the current is measured in time slot (b) and (e) of **Figure 7**, there is always enough time for conversion available.

2.3.2 Estimation of the Rotor Position

The voltage model is a simple model for rotating field motors which is based on dedicated differential equations. In order to determine the actual angle, the flux vector can be calculated by integrating the voltage. The integration can be simplified by replacing the integration by a lowpass filter with a very low cut-off frequency.

$$\vec{\Psi} = \int(\vec{v}_s - \vec{i}_s \cdot R)dt - (\vec{i}_s \cdot L)$$

A detailed description of the rotor position estimator can be found in **Section 3.2.5**.

2.3.3 Rotational Speed Calculation

The angular speed is calculated by the derivative of the angle with respect to time. The deviation by a difference quotient causes an amplification of high frequency signals. These high frequency signals are noise and other disturbances in the current measurement. As a result, the derivative needs to be lowpass filtered.

$$\omega = \frac{d\varphi}{dt}$$

2.4 Key Features of XC886/888 which enable Field Oriented Control

- High performance 16-bit vector computer (CORDIC + MDU)
  - Vector rotation and transformations like Park transform
  - Normalizing and scaling
  - Interrupt based operation with minimum CPU load
- PWM unit for advanced motor control (CapCom6E)
  - 16-bit resolution for high precision space vector PWM generation
  - Dead time control for minimum hardware effort (direct control of MOSFET/IGBT)
  - CTRAP provides hardware overload protection
- A fast 10-bit A/D Converter (sample time of 0.200 µs)
  - Hardware synchronization to PWM unit reduce CPU load
  - Two result registers to maximize sampling performance
  - Enables single shunt current measurement
  - Fast ADC reduces torque ripple due to minimized blind angle in sensorless FOC

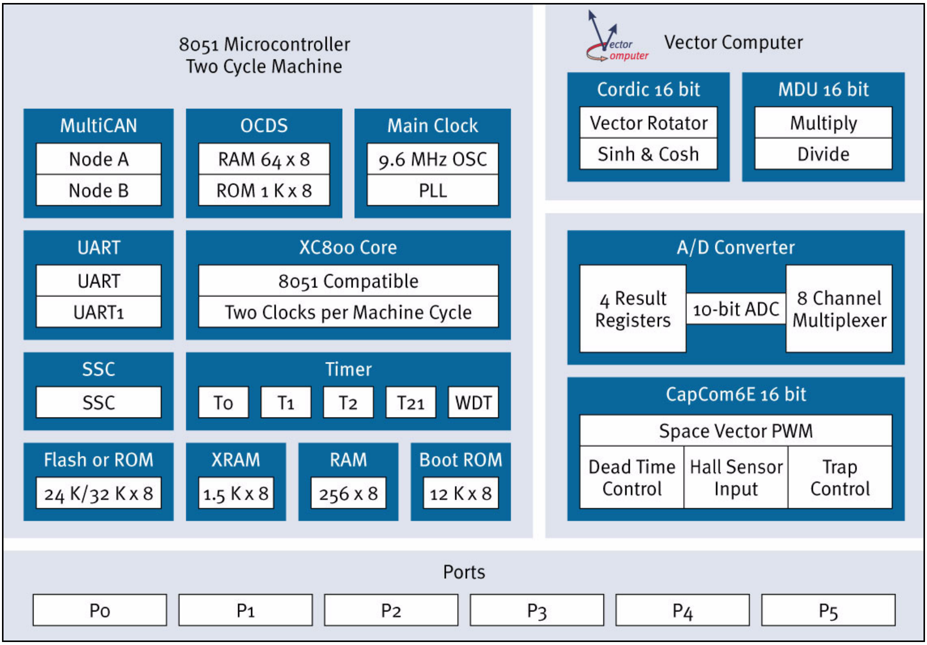


Figure 8 Block Diagram of XC886 / XC888

3 FOC Implementation

In this chapter, the field oriented control for PMSM motors is discussed in detail. The implementation of the FOC diagram can be seen in Figure 9. The yellow blocks implemented in hardware, the blue blocks are calculated by CORDIC, MDU and CPU.

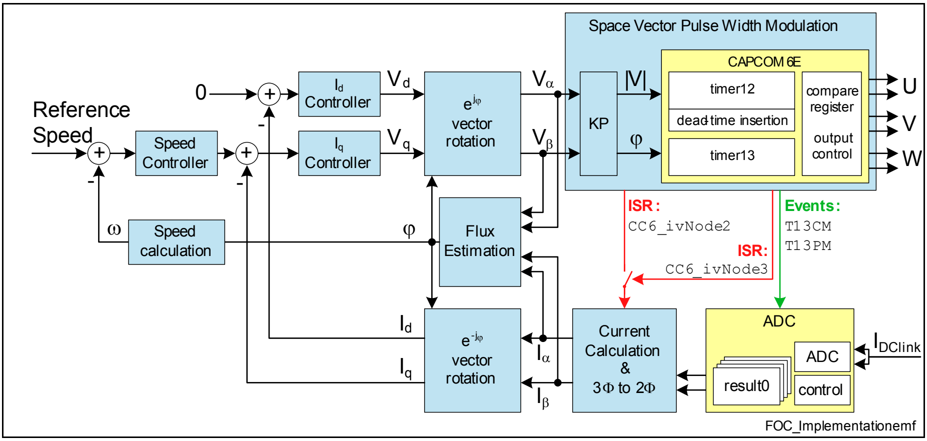


Figure 9 Implementation of FOC in XC886/888 CM/CLM

Figure 10 shows the timing diagram of timer12 and timer13 as well as the interaction of timer interrupts and the event trigger for start of ADC measurement.

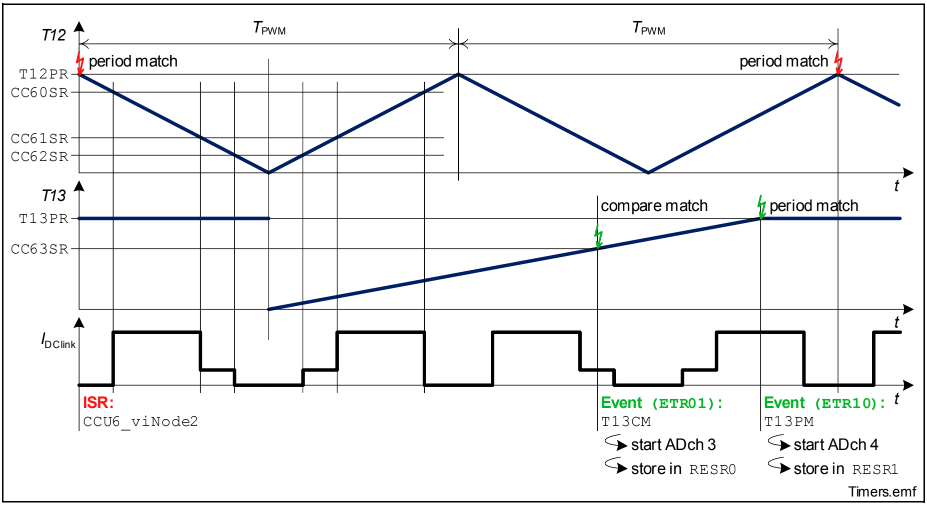


Figure 10 Timing Diagram: Timer12 and Timer13

Timer 12 is utilized to generate the control signals for the inverter. The compare registers CC60SR, CC61SR and CC62SR are responsible for the behavior of the output signals CC60, COUT60, CC61, COUT61, CC62 and COUT62.

The timer13 is configured in single shot mode. It starts synchronous with timer12 zero match. The period register of timer13 (CCU6_T13PR) is updated every second PWM period (T_{PWM}). The update value depends on the modulation, but is always greater than 1.5 * T_{PWM}.

The interrupt channel Node3 (timer12 period match interrupt) is locked in timer12 ISR and released in timer13 ISR. As a result of this setup, the timer 12 ISR is triggered every second PWM period.

The 10 bit ADC is configured in that way that the measurement result of channel 3 and 4 is stored in result register 2 and 3, respectively. This requires the DC-link current (I_{DClink}) to be available at both channels. As a result, the ADC conversion result is stored in separate result registers and can be used for calculations at any time independent of the modulation angle.

3.1 Timer2 Interrupt

The timer2 interrupt triggers every 1.6 ms. The interrupt service routine (ISR) handles the startup and the speed reference value for the speed control.

At startup of the motor, the space vector PWM angle is set to 0°. 100 interrupts after start, the voltage angle is changed stepwise with +/-90° and the control is enabled immediately.

After startup, any change in the user speed reference causes a linear ramp with +/-1430 rpm per second at the internal speed reference.

3.2 CCU6 Timer12 Period Match Interrupt (CCU6_viNode2)

This interrupt contains all calculations necessary for the field oriented control. It starts with the current calculation and transformation in the orthogonal system (V_a, I_a, V_b and I_b). After that all calculations are executed necessary for the motor control. In the end, the update for the modulation is prepared in the shadow registers.

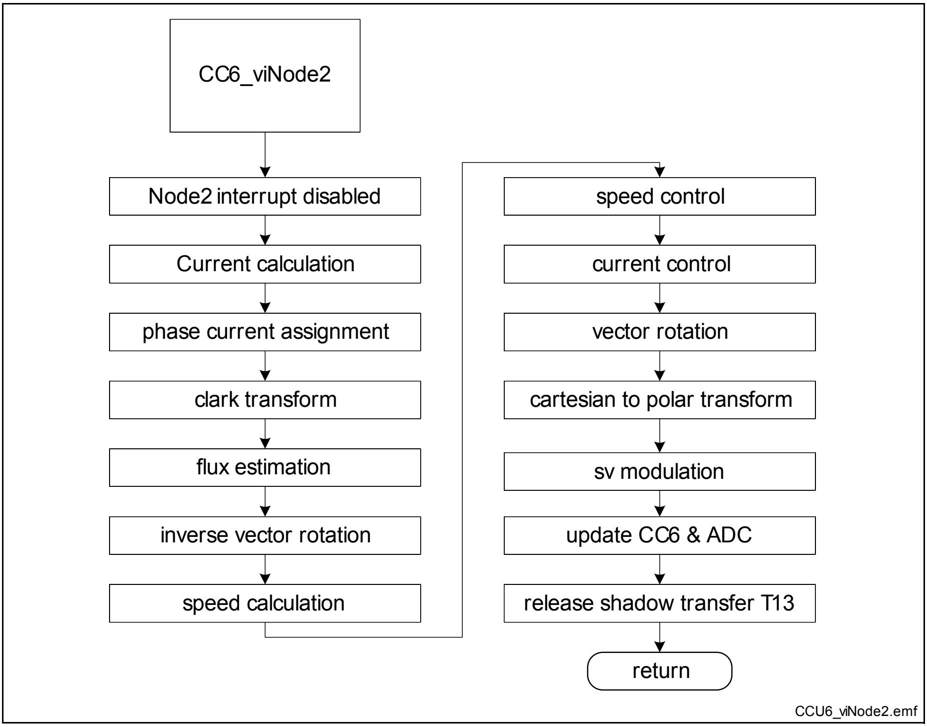


Figure 11 Flow Chart T12 Period Match Interrupt

In the following subsections all steps are discussed in detail.

3.2.1 Node2 Interrupt Disabled

The execution of all FOC calculations takes longer than one PWM period. As a result, the PWM period interrupt (T12 period match) has to be disabled during calculation. It is enabled after measuring the DC-link currents by node3 interrupt (T13 period match). As the interrupt priority of node2 is the highest possible, the enabling of the interrupt in node 3 is always after all steps of node2.

3.2.2 Current Measurement and Calculation

According to **Section 2.3.1**, there are four time slots where the DC-link current I_{DClink} is not zero and two different phase currents can be measured. In order to trigger the current measurement very accurately, two ADC channels are connected to the amplified voltage of the shunt.

The ADC conversions are consecutively started every second period of the modulation. The first measurement is triggered by the compare match of CC63 and sampled by ACD channel 3. The second measurement by ADC channel 4 is triggered by the period match of T13. ADC channel 3 is used to measure the positive currents (segment b of **Section 2.3.1**), channel 4 measures the negative ones (segment e). Sampling is triggered in the center of each segment. Please refer to **Figure 10** for detailed timing of the current measurement.

This method requires very fast switching and sampling times in order to ensure a proper current measurement. The time slot for the phase current measurement is limited by the sum of sampling time T_{sample} and deadtime T_{deadtime}.

Due to this restriction, the current can not be measured at the crossing of two sectors of the space vector modulation. Therefore the minimum time for the voltage space vector is limited to the minimum time for current measurement.

Here, the measurements of ADC channel 3 and 4 are evaluated by reading the result register 0 and 1. The values are right-shifted three bits by the MDU in order to remove the channel information and to multiply the 10-bit ADC value by 8.

Current Range and Scaling

The maximum current is defined as follows:

$$I_{\text{max}} = \frac{V_{\text{aref}}}{R_{\text{shunt}} \cdot G_{\text{OP}}}$$

R_{shunt} = Shunt Resistor, G_{OP} = Amplification in opamp, I_{max} = maximum current, V_{aref} = reference voltage for ADC

The scaling is the equivalent to the maximum physical value represented in a signed integer value.

$$N_{I_{(U,V,W)}} = \frac{I_{\text{max}} \cdot 2^{15}}{8 \cdot 2^{10}}$$

I_{max} = maximum measurable current

3.2.3 Phase Current Assignment

The phase currents are assigned according to the sectors of the space vector.

Sector		I_{DClink}(a)	I_{DClink}(b)	I_{DClink}(c)	I_{DClink}(d)	I_{DClink}(e)	I_{DClink}(f)	I_{DClink}(g)
A	0° - 60°	0	I_U	$-I_W$	0	$-I_W$	I_U	0
B	60° - 120°	0	I_V	$-I_W$	0	$-I_W$	I_V	0
C	120° - 180°	0	I_V	$-I_U$	0	$-I_U$	I_V	0
D	180° - 240°	0	I_W	$-I_U$	0	$-I_U$	I_W	0
E	240° - 300°	0	I_W	$-I_V$	0	$-I_V$	I_W	0
F	300° - 360°	0	I_U	$-I_V$	0	$-I_V$	I_U	0

3.2.4 Clark Transform

The transformation from three stator phase currents (I_U, I_V, I_W) into two orthogonal rotor phase currents (I_d, I_q) is realized in two steps. In the first step, the transformation into a two phase stator system (I_α, I_β) is calculated. The second step rotates the coordinates into the rotor system.

The two measured motor phase currents (I_a, I_b) are represented as vector in an orthogonal stator α/β-coordinate system. The vector is built by the currents (I_α, I_β). After startup, the α/β-coordinate system rotates synchronously with the rotor.

Following transformation, also known as Clarke Transform, is used to represent a three phase system by an orthogonal two phase system:

$$I_\alpha = I_a$$

$$I_\beta = \frac{1}{\sqrt{3}} \cdot (I_a + 2I_b)$$

Both components are calculated by the MDU. A factor of 0.5 simplifies both equations. Because of this, the scaling in the α/β-system is twice the scaling of the u/v/w-system.

$$N_{I_{(\alpha,\beta)}} = 2 \cdot N_{I_{(U,V,W)}}$$

3.2.5 Flux Estimation

The vector in the α/β-system is rotated by an angular transformation into d/q-system of the rotor. Here, the angle φ must be available, which is calculated from the rotor flux Ψ by following equations. During startup, the angle is not taken from the flux estimation.

$$\Psi_\alpha = \int (V_\alpha - I_\alpha \cdot R)dt - I_\alpha \cdot L$$

$$\Psi_\beta = \int (V_\beta - I_\beta \cdot R)dt - I_\beta \cdot L$$

$$\varphi = \text{atan}\left\{\frac{\Psi_\beta}{\Psi_\alpha}\right\}$$

R = stator resistance, L = stator inductance, $V_{\alpha,\beta}$ = stator voltage, $I_{\alpha,\beta}$ = stator current

The calculation of the induced voltage is realized with the CORDIC module.

CORDIC configuration	Linear rotation mode	
$X_{\text{final}} = \frac{X}{2^{\text{MPS}}}$	$Y_{\text{final}} = \frac{Y + X \cdot Z \cdot 16}{2^{\text{MPS}}}$	$Z_{\text{final}} = 0$
$X = R'$	$Y = V'_{\alpha,\beta}$	$Z = I'_{\alpha,\beta}$
MPS = 0	keep bit = 0	Interrupt disabled

The physical values $V_{\alpha,\beta}$ and $I_{\alpha,\beta}$ are represented in internal microcontroller values $V'_{\alpha,\beta}$ and $I'_{\alpha,\beta}$ as follows:

$$V'_{\alpha,\beta} = \frac{V_{\alpha,\beta} \cdot 2^{15}}{N_{V_{(\alpha,\beta)}}} \quad , \quad I'_{\alpha,\beta} = \frac{I_{\alpha,\beta} \cdot 2^{15}}{N_{I_{(\alpha,\beta)}}}$$

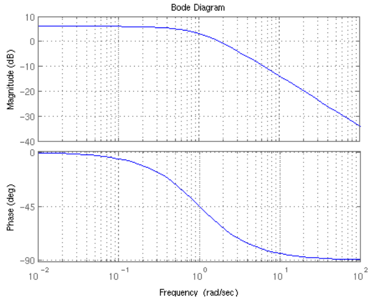
N_V and N_I are the scale factors for voltage and current. The resistance R is scaled by N_V and N_I as well.

$$R = \frac{V}{I} \quad , \quad R' = -\frac{V'}{I'} \cdot 2^{15} = R \cdot \frac{N_{I_{(\alpha,\beta)}} \cdot 2^{15}}{N_{V_{(\alpha,\beta)}} \cdot 16}$$

Finally, the two’s complement of the value R’ must be taken to get the correct calculation of the induced voltage.

Note: In the code, the resistance R' is defined by `STATOR_R`

The integration of the induced voltage during the calculation of the flux Ψ is calculated by a low pass filter of first order which is realized by a PT1 controller. See following figure for details:



$$G(p) = \frac{1}{1 + pT}$$

Figure 12 Transfer Function of a PT1 Controller

It is important for the accuracy at low speed that the cut-off frequency is low. It comes clear in **Figure 12** that the output value of a PT1 controller decreases at high frequencies. As a result, an additional gain is inserted in order to increase the resolution.

function call: `void PT1_MDU( void )`

Following equation is used for the PT1 algorithm:

$$Y_{(k)} = Y_{(k-1)} + Z_{(k)}(X_{(k)} - Y_{(k-1)})$$

$X_{(k)}$: Input Value

$Y_{(k-1)}$: stored value of $Y_{(k)}$

$$Z = \frac{\Delta t}{T} \cdot 2^{16} \cdot G_Z$$

The sampling time is Δt. The time constant for the filter is T.

The PT1 controller is calculated by the MDU. The implementation provides two gain factors G_Z selected by global flag `F0`.

- `F0` = 0: $Z_{(k)}$ = $Z_{(k)}$ / 256 , G_Z = 256
- `F0` = 1: $Z_{(k)}$ = $Z_{(k)}$, G_Z = 1

The interface to the PT1 algorithm is defined as follows:

Parameter	Format	Code
global flag F0	1 bit	`F0`
Input X	16 bit	`R7` = H, `R6` = L
time constant Z_k	16 bit	`MDU_MD5` = H, `MDU_MD4` = L

Parameter	Format	Code
storage for $Y_{(k)}$	32 bit	`@r0` = L
Output Y	16 bit	`@r0` = H, `@r1` = L

In the calculation of the rotor flux estimation, the PT1 algorithm is used with following parameters: T = 1 s, `F0` = 0 (G_Z = 256) , Δt = 2*T_{PWM} = 133.33 µs.

Note: In the code, the time constant Z_k *is defined by* `FLUX_PT1_TIME`

Stator Flux Calculation ($I_{\alpha,\beta}$ * L)

A signed multiplication is performed by the MDU in order to calculate the stator flux.

The inductance L has to be scaled similar to the resistance R.

$$L = \frac{\Psi}{I} \quad , \quad L' = -\frac{\Psi'}{I'} \cdot 2^{15} \cdot 2 \cdot G_Z = -L \cdot \frac{N_{I_{(\alpha,\beta)}} \cdot 2^{15} \cdot 2 \cdot G_Z}{N_{\Psi_{(\alpha,\beta)}}} \quad , \quad N_{\Psi_{(\alpha,\beta)}} = N_{V_{(\alpha,\beta)}}$$

The factor “2” in the calculation is added, because the result of the MDU is half of the expected value. The inductance L’ is negative, because all flux components are added afterwards. The calculation of the rotor flux was performed with a gain G_Z = 256. That’s why the stator flux must be scaled by 256 as well. This is done by scaling the inductance. The inductance value L’ has to be provided as two’s-complement

Note: In the code, the resistance L’ is defined by `STATOR_L`

Rotor Angle Calculation

The rotor angle is calculated from the flux-components Ψ_α and Ψ_β. The calculation is performed by the CORDIC.

CORDIC configuration	Circular vectoring mode	K ≈ 1,64676
$X_{\text{final}} = \frac{K \cdot \sqrt{X^2 + Y^2}}{2^{\text{MPS}}}$	$Y_{\text{final}} = 0$	$Z_{\text{final}} = Z + \text{atan}\left(\frac{Y}{X}\right)$
$X = \Psi_\alpha$	$Y = \Psi_\beta$	$Z = 0$
MPS = 0	keep bit = 0	Interrupt disabled

For speed calculation, the angle is stored in the variable `Winkel_mem`.

3.2.6 Vector Rotation $e^{-j\varphi}$

The space vector is rotated by the angle φ from the stator-fix α/β coordinates into the rotor-fix d/q coordinates by an angular transformation, also known as Park Transform. In the rotor-fix system, the current vector I_d represents the flux inducing component of the motor current. The current vector I_q represents the torque component.

The angular transformation is calculated in the CORDIC:

CORDIC configuration	**Circular rotation mode**	**K ≈ 1,64676**
$X_{\text{final}} = K \cdot \frac{-I_\alpha \sin\varphi + I_\beta \cos\varphi}{2^{\text{MPS}}}$ $I_q = X_{\text{final}}$	$Y_{\text{final}} = K \cdot \frac{I_\alpha \cos\varphi + I_\beta \sin\varphi}{2^{\text{MPS}}}$ $I_d = Y_{\text{final}}$	$Z_{\text{final}} = 0$
$X = I_\beta$	$Y = I_\alpha$	$Z = \varphi$
MPS = 0	keep bit = 0	Interrupt disabled

The scaling in the stator and rotor systems differs as follows.

$$N_{I_{(d,q)}} = N_{I_{(\alpha,\beta)}} \cdot \frac{2}{K} \qquad , \qquad K = 1{,}64676$$

3.2.7 Speed Calculation

The speed is calculated every third call of timer12 interrupt. At a PWM frequency of 15 kHz, the speed calculation and control is performed every 400 µs. The speed control can be adjusted ti the application’s needs by changing the parameter `SPEED_CONTROL_RATE`.

This function calculates the speed from the derivative of the angle. The calculation is performed by using the difference quotient and smoothing of the result by a PT1 filter.

$$\omega = \frac{\Delta\varphi}{\Delta t}$$

In the algorithm, just the difference between the current angle and the stored angle is calculated. The division by the time is not performed. The result is filtered by the PT1 algorithm without additional gain (`F0` = 1) with a time constant of 25.6 ms.

The scaling for the mechanical rotational speed is calculated as follows:

$$N_\omega = \frac{2^{15} \cdot 60}{\Delta t \cdot 2^{16} \cdot P_p} [\text{rpm}]$$

P_p is the number of pole pairs.

3.2.8 Speed Control

The PI controller for speed control is called in the same timing with the speed calculation. The output of the speed controller provides the torque building current component I_q.

$$y(t) = K_P\left(x(t) + \frac{1}{T_N} \cdot \int x(t)dt\right) \qquad \text{or} \qquad y(t) = [K_P \cdot x(t)] + [K_I \cdot \int x(t)dt]$$

function call: `void PI_controller( void )`

The implemented PI controller represents following transfer function.

$$G_R = K_P \cdot \left(1 + \frac{1}{pT_I}\right) = K_P + \frac{K_I}{p} \qquad , \qquad K_I = \frac{K_P}{T_I}$$

K_P and K_I are the control parameters. K_P corresponds to the proportional gain of the controller and K_I influences the reset time T_I together with the gain K_P.

The implementation for the microcontroller is as follows:

$$Y_{(k)} = [K'_P \cdot X_{(k)}] + \left[Y_{(k-1)} + K'_P \cdot \frac{\Delta t}{T_I} \cdot X_{(k)}\right] = [K'_P \cdot X_{(k)}] + [Y_{(k-1)} + K'_I \cdot X_{(k)}]$$

$X_{(k)}$:	control deviation (reference value - actual value)
$Y_{(k-1)}$:	stored value of $Y_{(k)}$
Δt:	sample time
T_I:	Time constant for integral component

$$K'_I = \frac{\Delta t}{T} \cdot K'_P$$

The PI algorithm is calculated by the CORDIC and MDU. The function provides following interface:

Parameter	Format	Code
reference value	16 bit	`R7` = H, `R6` = L
actual value	16 bit	`R5` = H, `R4` = L
struct {		
stored value $Y_{(k)}$	24 bit	
K_P (scale: N_{Kp} = 16)	16 bit	
K_I (scale: N_{KI} = 128)	16 bit	`@R0` = L
}		
output$Y$	16 bit	`R7` = L, `R6` = H

In this algorithm following calculations are executed:

1. The control deviation e (reference value - actual value) is calculated and limited to 16 bit
2. The control deviation is multiplied by the MDU with the K_I value and integrated. After calculation, the result is limited to 16 bit
3. The control deviation is multiplied by the CORDIC with the K_I value
4. The sum of proportional and integral component is taken and limited to 16 bit

CORDIC configuration	Linear rotation mode	
$X_{\text{final}} = \frac{X}{2^{\text{MPS}}}$	$Y_{\text{final}} = \frac{Y + X \cdot Z \cdot 16}{2^{\text{MPS}}}$	$Z_{\text{final}} = 0$
$X = K_P$	$Y = 0$	$Z = e$
MPS = 0	keep bit = 0	Interrupt disabled

3.2.9 Current Control

Both current components I_d and I_q are controlled independently by PI controller. For PMSM motors, the reference value for the field building component I_d is zero, because the field is available due to the permanent magnets. However, a variation of I_d can be used for field weakening techniques. The reference value for the torque building component is given by the output of the speed controller.

function call: `void PI_controller( void )`

The same function is used as for the speed controller. The outputs of the controller are the set values V_d and V_q in rotor-fix coordinates. This voltage space vector is used by the space vector modulator.

3.2.10 Vector Rotation $e^{+j\varphi}$

This angular transformation is performed with the CORDIC.

CORDIC configuration	Circular rotation mode	K ≈ 1,64676
$X_{\text{final}} = K \cdot \frac{-V_q \sin\varphi + V_d \cos\varphi}{2^{\text{MPS}}}$ $V_\alpha = X_{\text{final}}$	$Y_{\text{final}} = K \cdot \frac{V_q \cos\varphi + V_d \sin\varphi}{2^{\text{MPS}}}$ $V_\beta = Y_{\text{final}}$	$Z_{\text{final}} = 0$
$X = V_d$	$Y = V_q$	$Z = \varphi$
MPS = 2	keep bit = 0	Interrupt disabled

3.2.11 Transformation Carthesian to Polar Coordinates

Now, the voltage space vector (V_α, V_β) is transformed into a polar representation atan (φ,|V|). The transformation is calculated in the CORDIC:

CORDIC configuration	Circular vectoring mode	K ≈ 1,64676
$X_{\text{final}} = \frac{K \cdot \sqrt{X^2 + Y^2}}{2^{\text{MPS}}}$ $\lvert V \rvert = X_{\text{final}}$	$Y_{\text{final}} = 0$	$Z_{\text{final}} = Z + \operatorname{atan}\left(\frac{Y}{X}\right)$ $\varphi = Z_{\text{final}}$
$X = V_\alpha$	$Y = V_\beta$	$Z = 0$
MPS = 2	keep bit = 0	Interrupt disabled

The scaling is determined by the dead time of the driver and the sample time of the ADC:

$$N_{V_{(d,q)}} = \frac{V_{z,\,\mathrm{max}} \cdot 2^{16}}{2^{15} \cdot \left(1 - \frac{T_{\mathrm{sample}} + T_{\mathrm{deadtime}}}{T_{\mathrm{PWM}}}\right)}$$

Due to use of CORDIC and MDU for the transformations, the scaling for the stator-fix coordinates is as follows:

$$N_{V_{(\alpha,\beta)}} = \frac{N_{V_{(d,q)}} \cdot 2}{K} \qquad , \qquad K = 1.64676$$

This voltage scale factor is used in the flux estimation module.

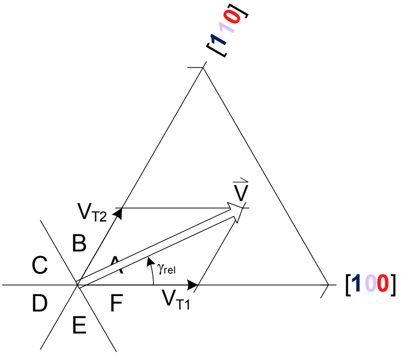
3.2.12 Space Vector Modulator

The inverter is driven by pulse with modulated signals. By this method, at frequencies higher than the electrical time constant of the motor, a smooth and sinusoidal current can be generated that features very little harmonic content. As a result, the motor is driven by constant rotating fields.

Calculation of the PWM Duty Cycles from the Space Vector

In the subroutine for the calculation of the PWM duty cycle, first the sector is acquired. Afterwards, the angle within in the sector (φ_{rel}) is calculated. By using the angle φ_{rel} and the norm $|\vec{V}|$, the components V_{T1} and V_{T2} are calculated.

$$V_{\mathrm{T1}} = \frac{2}{\sqrt{3}}(|\vec{V}| \cdot \sin\varphi_{\mathrm{rel}}) \quad , \quad V_{\mathrm{T2}} = \frac{2}{\sqrt{3}}(|\vec{V}| \cdot \sin(60° - \varphi_{\mathrm{rel}})) \qquad 0° < \varphi_{\mathrm{rel}} < 60°$$



PWM_DC.emf

Figure 13 Space Vector V in Sector A

Now, the duty cycle for the inverter have to be calculated. This is done by calculating the corresponding compare values within a PWM-period (see **Figure 10**).

Following intermediate calculation is performed:

$$V_{TA} = \frac{T_{T12}}{2} + \frac{V_{T1} + V_{T2}}{2} = \frac{T_{T12} + V_{T1} + V_{T2}}{2}$$

$$V_{TB} = \frac{T_{T12}}{2} - \frac{V_{T1} + V_{T2}}{2} = \frac{T_{T12} - V_{T1} - V_{T2}}{2}$$

$$V_{TC} = V_{TB} + V_{T1} = \frac{T_{T12} + V_{T1} - V_{T2}}{2}$$

The calculation V_{T2} / 2 and (V_{T1} + V_{T2}) /2 is performed by the CORDIC.

CORDIC configuration	**Linear rotation mode**	
$X_{final} = \frac{X}{2^{MPS}}$	$Y_{final} = \frac{Y + X \cdot Z \cdot 16}{2^{MPS}}$	$Z_{final} = 0$
$X = V_{T2}$	$Y = V_{T1}$	$Z = \frac{1}{16}$
MPS = 1	keep bit = 0	Interrupt disabled

The norm of vector V is scaled according to the maximum duty cycle of the pulse width modulation which is defined by the time T_{PWM} in timer12 representation and a combination of ADC sample time and driver delay time.

Note: In the code, the time T_{T12} is defined by `T12PERIODE`

$$V_{(\mathrm{T1,T2})\max} = T_{\mathrm{T12}} \cdot \left(1 - \frac{T_{\mathrm{sample}} + T_{\mathrm{driverdelay}}}{T_{\mathrm{T12}}}\right) = K_S \cdot |\vec{V}|_{\max}$$

In consideration of the previous calculations with the CORDIC and MDU, following scaling is used:

$$K_S = \frac{2 \cdot V_{(\mathrm{T1,T2})\max}}{|\vec{V}|_{\max}} \quad , \qquad |\vec{V}|_{\max} = 2^{11} \cdot K^2 = 2^{11} \cdot (1.64676)^2 = 5553.8$$

$$K_S = \frac{(T_{\mathrm{T12}} - (T_{\mathrm{sample}} + T_{\mathrm{driverdelay}}))}{2776.9}$$

The scale factor $1/K_{\mathrm{S}}$ is used for the calculation of the sin(φ) table:

$$\sin(30°) = \frac{1}{2} \cdot 2^{15} \cdot K_S$$

Note: In the code, the sin(φ) table is stored in file "modultab.h"

Additionally to the scaling of $V_{(\mathrm{T1,T2})}$ under consideration of the maximum duty cycle, it is necessary to limit $V_{(\mathrm{T1,T2})}$ to the minimum time required for current measurement T_{sample}.

Note: In the code, the time $T_{\mathrm{driverdelay}}$ is defined as `DRIVER_DELAY` *and (T_{sample} + $T_{\mathrm{driverdelay}}$) by* `T_MIN`.

3.2.13 Update of CapCom6 and ADC Configuration

Depending on the direction of the space vector, different combinations of the intermediate values V_{TA} to V_{TD} have to be moved to the shadow registers of CapCom6-Unit. See following table for details.

The time D corresponds to the delay of the ADC trigger in order to take care about the delay time and switching time of the driver.

Note: In the code, the delay D is defined by `DRIVER_DELAY`.

Sector	PWM Generation			ADC Trigger	
	CC60SR	CC61SR	CC62SR	CC63SR	T13PR
A	V_{TB}	V_{TC}	V_{TA}	D + (3T_{T12}+ V_{T1}) / 2	D + (5T_{T12} + V_{T2}) / 2
B	V_{TC}	V_{TB}	V_{TA}	D + (3T_{T12} + V_{T2}) / 2	D + (5T_{T12} + V_{T1}) / 2
C	V_{TA}	V_{TB}	V_{TC}	D + (3T_{T12} + V_{T1}) / 2	D + (5T_{T12} + V_{T2}) / 2
D	V_{TA}	V_{TC}	V_{TB}	D + (3T_{T12} + V_{T2}) / 2	D + (5T_{T12} + V_{T1}) / 2
E	V_{TC}	V_{TA}	V_{TB}	D + (3T_{T12} + V_{T1}) / 2	D + (5T_{T12} + V_{T2}) / 2
F	V_{TB}	V_{TA}	V_{TC}	D + (3T_{T12} + V_{T2}) / 2	D + (5T_{T12} + V_{T1}) / 2

3.2.14 Release Shadow Transfer T13

For a glitch free current measurement the update of the PWM registers must be set at a given time. That's why in the next step the shadow registers for the current measurement triggers (timer13) are released. The new PWM values are transferred after current measurement.

3.3 CCU6 Timer13 Period Match Interrupt (CCU6_viNode3)

Although the timer13 is used to trigger the ADC, the timer13 ISR contains the sequence to enable the interrupt of Node2 (see **Figure 14**) and to release the shadow transfer for timer T12.

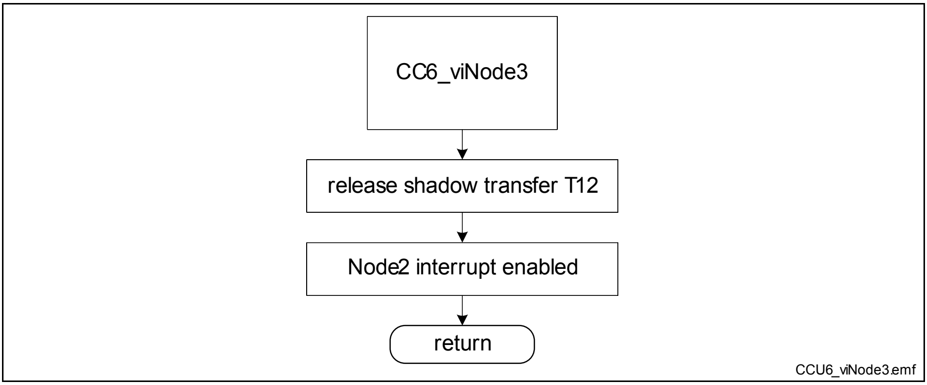


Figure 14 Flow Chart T13 Period Match Interrupt

The timer13 is configured to trigger the ADC measurement by compare match (T13CM) and period match (T13PM) hardware events.

www.infineon.com

Published by Infineon Technologies AG