بسم الله الرحمن الرحيم

سرشناسه	: قربانعلی‌افجه، محسن، ۱۳۶۳ -
عنوان و نام پدیدآور	: کنترل‌برداری موتور القایی به روش جهت‌دهی (foc) / محسن قربانعلی‌افجه.
مشخصات نشر	: مشهد:مینوفر، ۱۳۹۵.
مشخصات ظاهری	: ۱۵۲ص.: مصور، جدول، نمودار.
شابک	: ۱۵۰۰۰۰ ریال: ۳-۹۶-۸۰۶۰-۶۰۰-۹۷۸
وضعیت فهرست نویسی	: فیپا
موضوع	: موتورهای برقی القایی
موضوع	: Electric motors, Induction
موضوع	: موتورهای برقی القایی -- طرح و ساختمان
موضوع	: Electric motors, Induction -- Design and construction
رده بندی کنگره	: TK۲۷۸۵/ق۴ک۹ ۱۳۹۵
رده بندی دیویی	: ۶۲۱/۴۶
شماره کتابشناسی ملی	: ۴۳۴۰۱۶۵

کنترل برداری موتور القایی به روش

جهت دهی میدان (FOC)

محسن قربانعلی افجه

انتشارات مینوفر
Minufar.ir

مشهد مقدّس - ۱۳۹۵

کنترل برداری موتور القایی به روش جهت‌دهی میدان (foc)

محسن قربانعلی‌افجه

طراح جلد: مینوفر

نوبت چاپ: اول - ۱۳۹۵

شمارگان: ۱۰۰۰ نسخه

قطع: وزیری

چاپ و صحافی: دقت

قیمت: ۱۵،۰۰۰ تومان

شابک: ۹۷۸-۶۰۰-۸۰۶۰-۹۶-۳

تمام حقوق چاپ و نشر این اثر محفوظ است.

انتشارات مینوفر
Minufar Publications

مشهد مقدس، بلوار جلال‌آل‌احمد ۶۹ پ ۱۱۶، ص.پ: ۱۷۵۵-۹۱۸۹۵

همراه: ۰۹۳۵۲۱۶۲۷۵۵ - **تلفن و دورنگار**: ۳۸۳۲۳۵۵۳-۰۵۱

سایت: www.minufar.ir - **ایمیل**: minufar@yahoo.com

فهرست مطالب

فصل ۱:-پیاده سازی روش کنترل برداری با روش کلیدزنیSVPWM در SP.....۷

کنترل برداری موتور القایی ۷

پیاده سازی ۱۲

فلوچارت برنامه ی کنترل دور موتور به روش FOC ۱۳

بلوک تنظیمات اولیه ۱۵

ماژول InitSysCtrl ۱۵

ماژول InitPeripheralClocks ۱۶

تنظیمات واحد Event Manage ۱۷

تنظیم تایمر شماره ۱ به صورت (Underflow interrupt)..... ۱۹

ماژولهای نرم افزاری روش کنترل پیشنهادی ۲۳

ماژول RMP_CNTL..... ۲۴

ماژول RAMP_GEN ۲۸

ماژول F281XILEG_VI ۳۰

ماژول کلارک ۳۵

ماژول PARK..... ۳۷

ماژول IPARK..... ۴۰

ماژول PID_REG3 ..... ۴۳

ماژول SVGEN_DQ ۴۸

ماژول SPEED_PRD ۵۳

ماژول QEP_NO_INDEX_DRV ۵۷

ماژول FC_PWM_DRV ۶۱

ماژول CUR_MOD..... ۶۶

ماژول ADC04b_DRV ۷۱

ماژول ReadADCAverage..... ۷۵

ماژول های شبیه سازی در MATLAB ۷۸
ماژول volt_calc ۷۸
ماژول ESTIMATEDSPEED ۸۲
ماژول aci ۸۶
فصل ۲-:ارتقاء مکانیزم کنترل موتور ۹۱
دستورات مربوط به شارژ و فعال شدن کانورتر ۹۲
دستور جهت حرکت ۹۳
محدود سازی مقدار مرجع ترمز یا رانش ۹۴
مقدار رفرنس نیروی ترمز و رانش ۹۵
متغیر تصحیح مقدار وزن قطار ۹۶
محدود سازهای رفرنس نیروی ترمز و رانش ۹۶
شیب رفرنس نیروی ترمز و رانش ۹۶
محدود سازی لغزش و سرخوردن ۹۶
کنترل ترمز الکترو دینامیکی ۹۷
کنترل لغزش و سرخوردن ۹۸
فصل ۳-:حفاظت، نظارت و سیستم عیب یابی ۱۰۱
مقدمه ۱۰۱
مدل پیشنهادی برای حفاظت کانورتر موتور ۱۰۲
گروه بندی خطاهای منجر به برروز مسدود سازی جهت حفاظت ۱۱۶
گروه بندی خطاهای منجر به برروز مسدود سازی نرم ۱۱۷
گروه بندی خطاهای منجر به برروز خاموش سازی نرم ۱۱۸
گروه بندی خطاهای منجر به برروز ایزوله سازی کانورتر: ۱۲۱
گروه بندی خطاهای منجر به برروز مسدود سازی جهت حفاظت ۱۳۲
گروه بندی خطاهای منجر به برروز خاموش سازی جهت حفاظت ۱۳۷

فصل ۱- پیاده سازی روش کنترل برداری با روش کلیدزنی SVPWM در DSP

کنترل برداری موتور القایی

در روش کنترل برداری یافتن موقعیت شار روتور و بدست آوردن مؤلفه‌های جریان استاتور بر روی شار و گشتاور بر روی محور مختصاتی که منطبق بر موقعیت مکانی شار روتور است امکان کنترل مستقیم شار و گشتاور را میسر می سازد.

عموماً، درایوهای سرعت متغیر موتورهای القائی، بدون توجه به تغییرات بار، به رنج‌کاری وسیعی از سرعت و پاسخ سریع گشتاور نیاز دارند. روش کنترل جهت میدان، موفق‌ترین روش در برآورده ساختن نیازهای فوق بوده است.

به دلیل پیشرفت‌های چشم‌گیر در الکترونیک قدرت و میکروپروسسورها. استفاده از کنترل جهت میدان در درایوهای موتورهای القائی به طور وسیعی در کاربردهای مختلف نظیر، سروو موتورهای ac، سیستم درایو اتومبیل‌های برقی و ... مورد استفاده قرار گرفته است. با استفاده از روش کنترل جهت

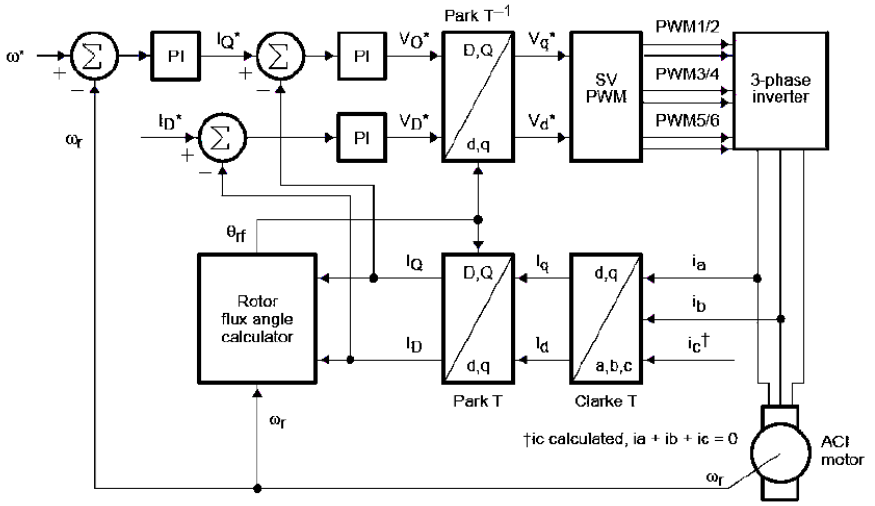
میدان، یک موتور القائی غیرخطی چند متغیره را به روش کنترل خطی، جدا شده و مستقل فلو و گشتاور، مشابه موتورهای DC تحریک مستقل، کنترل کرد.

با استفاده از روش کنترل با جهت دهی میدان (ield Oriented Control)، موتور القایی سه فاز، پاسخ سریع گشتاوری همانند موتور DC می‌تواند داشته باشد. برای تبدیل مختصات مولفه‌های جریان به یک سنسور موقعیت و برای تزریق جریان مورد نظر بدون موتور نیاز به اینورتر با جریان کنترل‌کننده وجود دارد. در این سیستم، مقادیر لحظه‌ای شار و گشتاور با استفاده از متغیرهای اولیه محاسبه می‌شوند. از مقایسه این مقادیر با دو سیگنال مقادیر مرجع و با استفاده از اینورتر، شار و گشتاور بطور مستقیم کنترل می‌شود.

معادلات دیفرانسیلی حاکم بر مدل ریاضی موتور به دلیل وابستگی پارامترها به موقعیت روتور و کوپلاژ بین فازهای روتور و استاتور دارای ضرایب متغیر با زمان هستند .برای رفع این مشکل و حذف پارامترهای متغیر با زمان، از تئوری محورهای dqاستفاده می‌شود .معادلات موتور القایی در سیستم محورهای سه فاز با استفاده از تئوری دو محوری پارک، به محورهای dq در مختصات مرجع دلخواه انتقال می یابد.

در این روش، معادلات و متغیرها بر روی دو محور مستقیم Direct Axis،d و محور عمودی q، Quadrature Axis بیان می‌شوند. این مدل را می توان نسبت به هر مختصات مرجع دلخواه بیان کرد .فرض می‌شود که سیستم سه فاز متعادل است، لذا مؤ لفه ای بر روی محور صفر وجود ندارد .با بررسی معادلات موتور روی محورهای dp اثر اندوکتانس متقابل بین فازهای روتور و استاتور که به صورت پارامترهای تابع زمان می‌باشند، حذف می‌گردد.

به منظور مشخص کردن دامنه و زاویه فازور فضایی شار روتور، از اندازه‌گیری مقادیر موقعیت روتور یا سرعت روتور و جریانهای استاتور، استفاده شده است .بلوک دیاگرام کلی کنترل سرعت موتور القایی به روش FOC در شکل زیر آمده است.



این سیستم دارای سه حلقه کنترل سرعت، جریان محور D و جریان محور Q می‌باشد. ابتدا با نمونه برداری از جریانهای مقدار واقعی مؤلفه های طولی و عرضی جریان استاتور در مختصات شار روتور، محاسبه می‌شود. بدین ترتیب که جریانهای سه فاز ابتدا توسط تبدیل کلارک به حوزه متعامد ساکن برده می‌شوند و سپس با استفاده از تبدیل پارک، مؤلفه های جریان در قاب مرجع dpبدست می آید. روابط تبدیلات کلارک و پارک به صورت زیر می‌باشد:

$$\begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} = \frac{2}{3}\begin{bmatrix} 1 & \frac{-1}{2} & \frac{-1}{2} \\ 0 & \frac{\sqrt{3}}{2} & \frac{-\sqrt{3}}{2} \end{bmatrix}\begin{bmatrix} i_{sA} \\ i_{sB} \\ i_{sC} \end{bmatrix}$$

$$\begin{bmatrix} i_{sD} \\ i_{sQ} \end{bmatrix} = \begin{bmatrix} Cos\,\theta_{rf} & Sin\,\theta_{rf} \\ -Sin\,\theta_{rf} & Cos\,\theta_{rf} \end{bmatrix}\begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} \quad (۱-۱)$$

که در آن i_{sd}وi_{sq} مؤلفه های جریان در قاب مرجع ساکن i_{sD} و i_{sQ} مؤلفه های جریان در قاب مرجع دوار و θ_{rf} زاویه فازور فضایی شار پیوندی روتور می‌باشد. با استفاده از این جریانها و نمونه برداری از سرعت روتور و با استفاده از مدل ارائه شده برای فلو، سرعت زاویه ای شار روتور (wmr) جریان مغناطیس کنندگی $|Imr|$ و زاویه θ_{rf} به دست می آید.

برخلاف موتورهای سنکرون، در موتور القایی موقعیت روتور نمی‌تواند معرف موقعیت شار روتور باشد و علاوه بر اندازه‌گیری مستقیم مکان روتور به کمک سنسورهای مکانیکی، لغزش نیز باید مد نظر قرار گیرد.

به منظور تخمین موقعیت شار روتور از مدل جریانی موتور استفاده می‌شود . در این مدل با استفاده از زاویه الکتریکی روتور و جریانهای استاتور، i_{sQ} وi_{sD} موقعیت دقیق شار روتور محاسبه می‌گردد. با محاسبه معادلات ولتاژ روتور در مختصات شار روتور، سرعت زاویه ای شا ر روتور به صورت زیر محاسبه می‌گردد:

$$w_{mr} = w_r + \frac{i_{sQ}}{T_r\left|\bar{i}_{mr}\right|} \quad (۲-۱)$$

که w_rسرعت زاویه ای روتور T_r ثابت زمانی مدار روتور و $\frac{i_{sQ}}{T_r|\bar{i}_{mr}|}$ بیانگر فرکانس زاویه ای لغزش میدان روتور می‌باشد $|Imr|$ نیز جریان مغناطیس کنندگی روتور در مختصات ویژه شار روتور است و مطابق روابط زیر محاسبه می‌گردد:

(۳-۱)

$$T_r\frac{d|\bar{i}_{mr}|}{d_t}+|\bar{i}_{mr}|=i_{sD}\Rightarrow|\bar{i}_{mr}|=\frac{1}{1+T_rP}i_{sD}$$

P عملگر مشتق را نشان می دهد. اگر$|Imr|$ ثابت باشد نتیجه می‌شود $|imr| = isD|$ است و دامنه فازور فضایی شار پیوندی روتور بوسیله مؤلفه طولی جریان استاتور کنترل می‌گردد. در زیر سرعت نامی نیز گشتاور الکتریکی متناسب با مؤلفه عرضی جریان استاتور خواهد شد.

زاویه θ_{rf} که در واقع موقعیت شار روتور را نشان می دهد نیز با استفاده از مدل ارائه شده برای فلو، به صورت زیر بدست می آید:

(۴-۱)

$$w_{mr}=\frac{d_{\theta_{rf}}}{d_t}\Rightarrow\theta_{rf}(t)=\int w_{mr}d_t+\theta_{rf}(0)$$

جریانهای i_{sD} وi_{sQ} به ترتیب با مقادیر مرجع $i_{sQ}ref$ مرجع مولد گشتاور $i_{sD}ref$ مرجع مولد شار مقایسه می‌شوند. فرمان شار مقدار ثابتی است که از روی جریان مغناطیس کنندگی تعیین می‌شود. مرجع گشتاور نیز توسط رگولاتور سرعت تولید می‌گردد. خروجی رگولاتورهای جریان، مطابق مدل

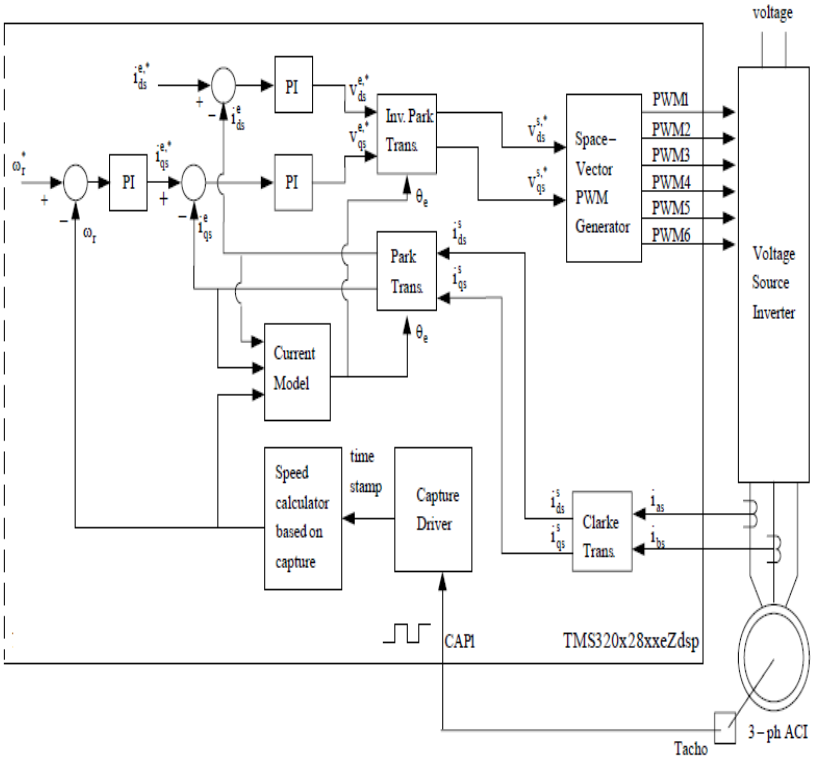
موتور و تابع تبدیل آن، ولتاژهای استاتور در قاب مرجع دوار هستند که با اعمال تبدیل معکوس پارک، مطابق زیر به قاب مرجع ساکن انتقال می یابند:

(۵-۱)

$$\begin{bmatrix} v_{sd} \\ v_{sq} \end{bmatrix} = \begin{bmatrix} Cos\,\theta_{rf} & -Sin\,\theta_{rf} \\ Sin\,\theta_{rf} & Cos\,\theta_{rf} \end{bmatrix} \begin{bmatrix} v_{sD} \\ v_{sQ} \end{bmatrix}$$

پیاده سازی

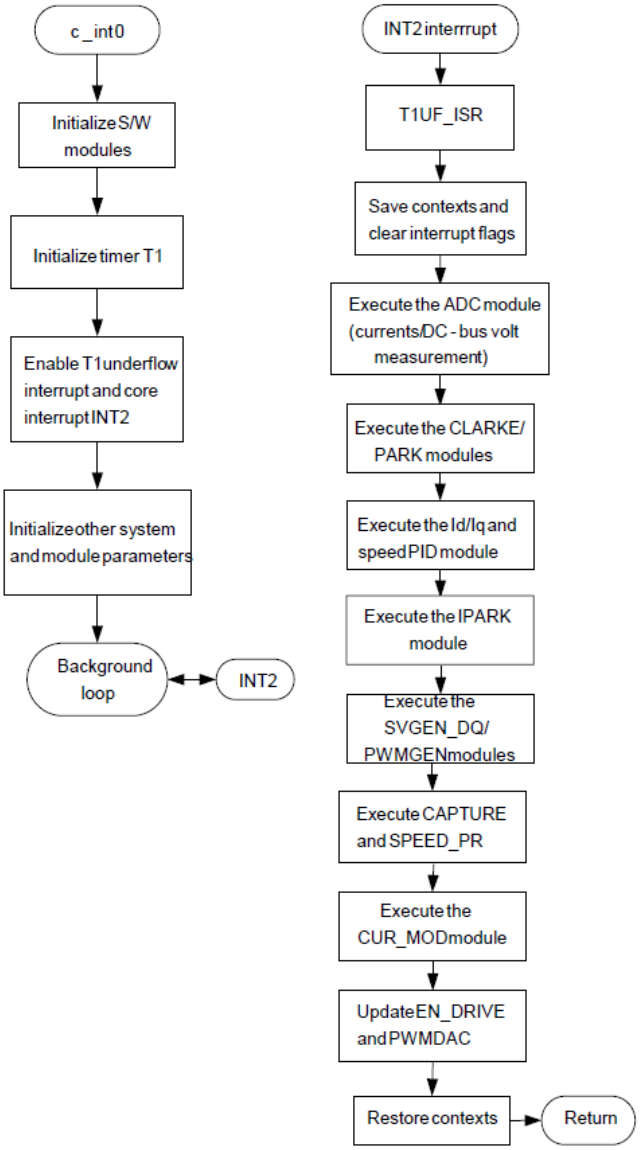
طبق مباحث مطرح شده، با توجه به قدرت پردازش بیشتر در پردازنده‌های سیگنال دیجیتال ساخت شرکت Texas Instruments بهتر است برای پیاده سازی روش کنترل اسکالر SVPWM از این پردازنده ها استفاده شود. شمای کلی سیستم کنترل مدنظر به روش SVPWM برای پیاده‌سازی دیجیتال در DSP را به صورت زیر است. در این روش سرعت با استفاده از سنسور اندازه‌گیری مانند تاکومتر یا انکدور اندازه‌گیری می‌شود، در نهایت در داخل بلوک Space Vector PWM Generator با استفاده از ولتاژ و فرکانس مرجع سیگنالهای کلیدزنی اینورتر با استفاده از روش SVPWM تولید می‌شود.



در شکل زیر اتصالات سخت افزاری بین بخش های متفاوت این سیستم نشان داده شده است. از PWM های ۱ تا ۶ جهت تولید پالس های مورد نیاز اینورتر و از پایه CAP1 جهت قرائت پالس های تاکومتر یا شفت انکودر استفاده شده است.

فلوچارت برنامه ی کنترل دور موتور به روش FOC

فلوچارت برنامه ی کنترل دور موتور به صورت زیر است:



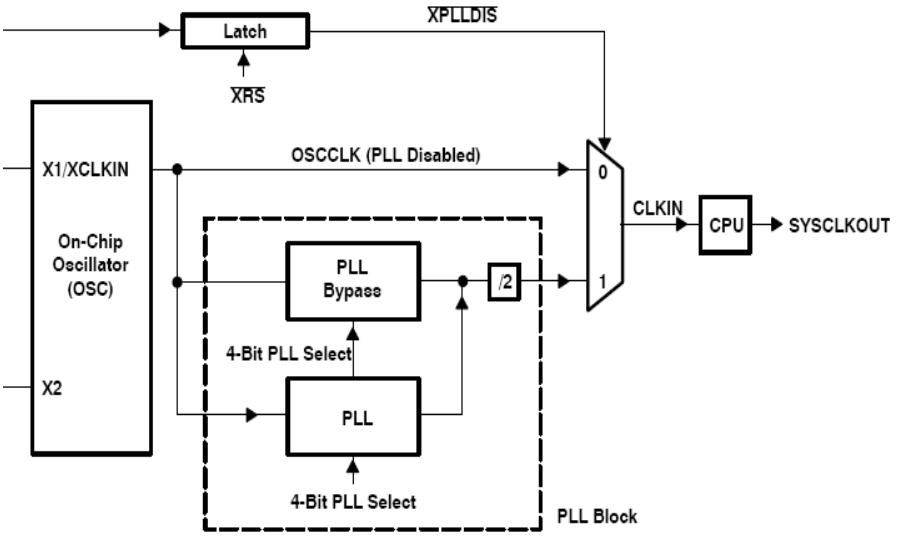
طبق این فلوچارت، برنامه از دو بلوک کلی تشکیل شده است. در بلوک سمت راست مقدار دهی اولیه و اصطلاحاً تنظیمات اولیه (Initialization) تمامی ماژول ها و تنظیمات اولیه سخت افزار DSP از جمله تنظیمات تایمر ۱

و وقفه شماره ۲، انجام می‌شود و در بلوک سمت راست حلقه کنترلی پیاده سازی می گردد.

بلوک تنظیمات اولیه

ماژول InitSysCtrl

کلاک توسط یک اسیلاتور ۳۰ مگاهرتز تعبیه شده بر روی برد پردازنده تأمین می‌گردد. خروجی این اسیلاتور نهایتاً به پین X1/XCLKIN متصل و پین X2 آزاد می‌باشد. با استفاده از یک مدار phase-locked loop یا PLL این فرکانس در داخل پردازنده ((SYSCLK به پنج برابر به مقدار ۱۵۰ مگاهرتز تبدیل می‌گردد. این عمل با قرار دادن مقدار 0x0A در رجیستر PLLCR در برنامه my_DSP281x_SysCtrl.c انجام می‌گیرد. بدین ترتیب سیکل دستورالعمل و یا Instruction Cycle پردازنده ۶,۶۷ نانوثانیه می‌شود. شکل زیر مدار داخلی پردازنده را نشان می‌دهد.



برنامه اصلی	void InitSysCtrl(void)
برنامه های فراخواننده	main
توضیحات برنامه	عددهی اولیه PLL

```
// Disable the watchdog
   DisableDog();
   // Initialize the PLLCR to 0xA
   InitPll(0xA);
   // Initialize the peripheral clocks
   InitPeripheralClocks();
```

ماژول InitPeripheralClocks

با اجرای این ماژول کلاک‌های بخش Peripheral مانند Event Manager فعال می‌گردد. طبق ماژول زیر، کلاک بخش peripherals با prescale یک یعنی کلاک ۱۵۰ مگاهرتز فعال گردیده است.

برنامه اصلی	void InitPeripheralClocks (void)
برنامه های فراخواننده	main
توضیحات برنامه	عددهی اولیه ساعت‌های بخش Peripheral

```
// HISPCP peripheral prescale register settings, normally it will be set to default values
   EALLOW;   // This is needed to write to EALLOW protected registers
   SysCtrlRegs.HISPCP.all = 0x0000;     // SYSCLKOUT/1
   EDIS;   // This is needed to disable write to EALLOW protected registers
```

تنظیمات واحد Event Manage

پردازنده TMS320F2812 دارای دو سامانه مدیریت رویداد (EVA) Event Manager A و (EVB) Event Manager Bمی‌باشد. مشخصات هر یک از این سامانه ها به طور خلاصه عبارتند از:

- دو تایمر عمومی General Purpose Timers (GP)

- سه واحد مقایسه (به منظور تولید سیگنال PWM)

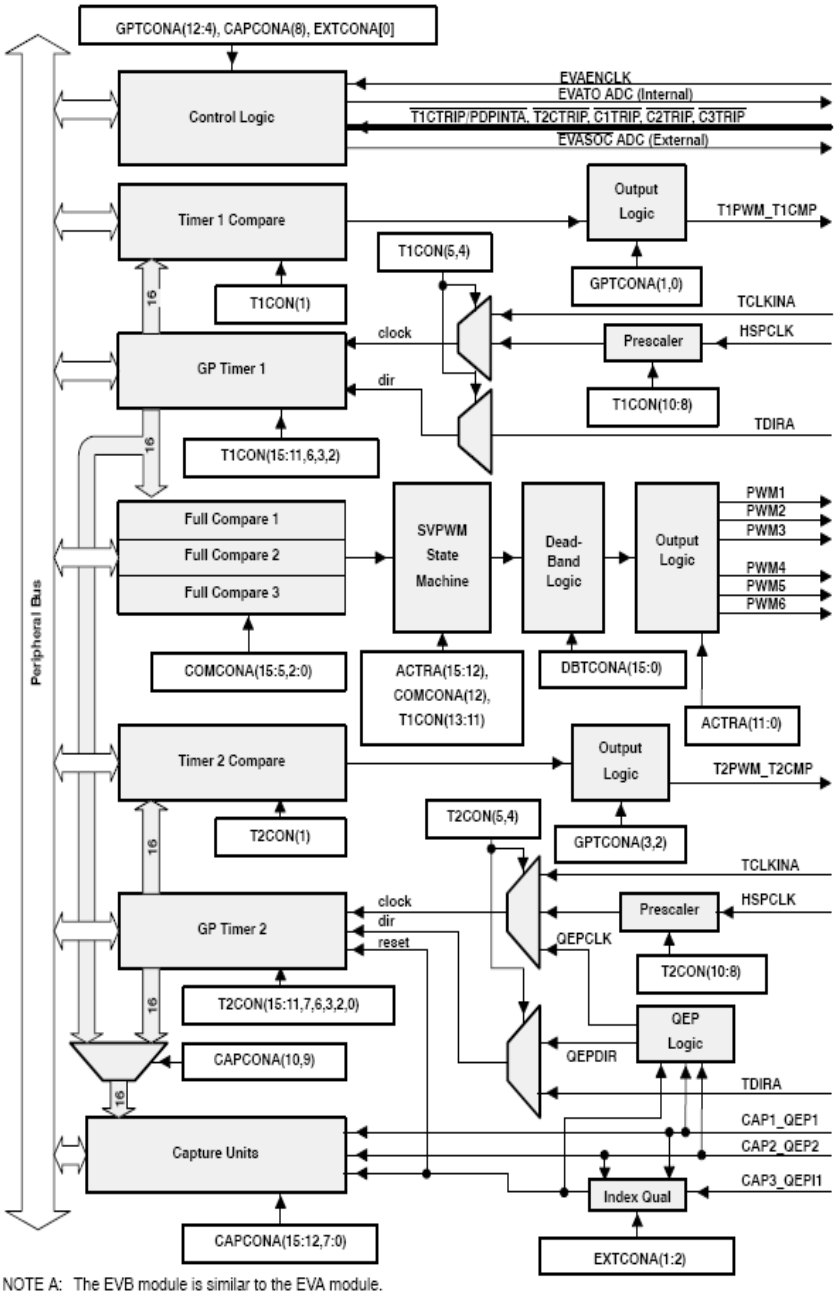
- مدار تولید سیگنال PWM و SVPWM (کلیدزنی فضای برداری) به همراه مدار تأخیر dead band

- سه واحد Capture

- مدار فیدبک Quadrature Encoder Pulse

- مدارهای وقفه

بلوک دیاگرام یکی از این سامانه ها (EVA) در شکل زیر نشان داده شده است. از این واحد برای تولید PWM مورد نیاز برای سوئیچ زنی و از واحد دیگر مدیریت رویداد به نام (EVB) برای تولید سه سیگنال آنالوگ استفاده شده است. در ادامه این زیربخش تنظیمات مدیریتهای رویداد تشریح خواهد شد.



NOTE A: The EVB module is similar to the EVA module.

تنظیم تایمر شماره ۱ به صورت (Underflow interrupt)

برای این منظور در دو ریجیستر EVAIMRA و EVAIFRA که به ترتیب ریجستر MASK تمامی وقفه های EVA و ریجستر پرچم وقفه EVA است، دو بیت مربوط به T1UFINT را یک می کنیم تا EVA به Underflow تایمر ۱ حساس باشد.

```
// Enable Underflow interrupt bits for GP timer 1
   EvaRegs.EVAIMRA.bit.T1UFINT = 1;
   EvaRegs.EVAIFRA.bit.T1UFINT = 1;
```

همان طور که در فلوچارت بیان شد بلوک سیستم کنترل در وقفه تایمر ۱ فرآخوانی می‌شود لذا در بردار وقفه به ازاء Underflow تایمر ۱ تابعی به نام MainISR که در آن الگورتیم کنترل پیاده خواهد شد، فراخوانی می‌شود.

```
EALLOW; // This is needed to write to EALLOW protected registers
PieVectTable.T1UFINT = &MainISR;
EDIS;   // This is needed to disable write to EALLOW protected registers
```

در جدول برخی از بردارهای وقفه های پردازنده ۲۸۱۲ نشان داده شده است، وقفه شماره ۲ با اولویت ۶ برای CPU تعریف شده و این وقفه متعلق به گروه ۲ است. این شرکت وقفه تایمر ۱ را به عنوان ورودی وقفه شماره ۲، CPU معرفی کرده است.

281x PIE Vector Table

Name	Vector ID	Address	Size (x16)	Description	CPU Priority	PIE Group Priority
Reset	0	0x0000 0D00	2	Reset is always fetched from location 0x003F FFC0 in Boot ROM or XINTF Zone 7	1 (highest)	–
INT1	1	0x0000 0D02	2	Not used. See PIE Group 1	5	–
INT2	2	0x0000 0D04	2	Not used. See PIE Group 2	6	–
INT3	3	0x0000 0D06	2	Not used. See PIE Group 3	7	–
INT4	4	0x0000 0D08	2	Not used. See PIE Group 4	8	–

در این پردازنده هرگروه شامل ۸ ورودی برای وقفه می‌باشد. به عنوان مثال در وقفه شماره ۲، ورودی ششم از گروه ۲، به عنوان وقفه Underflow تایمر ۱ تعیین شده است. لذا در کد نویسی گروه ۲ و ورودی ۶ ام فعال و وقفه شماره ۲ CPU نیز فعال شده است.

Name	Address	Size (x16)	Description		CPU Priority	PIE Group Priority
PIE Group 2 Vectors – MUXed into CPU INT2						
INT2.1	40	0x0000 0D50	2	CMP1INT (EV-A)	6	1 (highest)
INT2.2	41	0x0000 0D52	2	CMP2INT (EV-A)	6	2
INT2.3	42	0x0000 0D54	2	CMP3INT (EV-A)	6	3
INT2.4	43	0x0000 0D56	2	T1PINT (EV-A)	6	4
INT2.5	44	0x0000 0D58	2	T1CINT (EV-A)	6	5
INT2.6	45	0x0000 0D5A	2	T1UFINT (EV-A)	6	6
INT2.7	46	0x0000 0D5C	2	T1OFINT (EV-A)	6	7

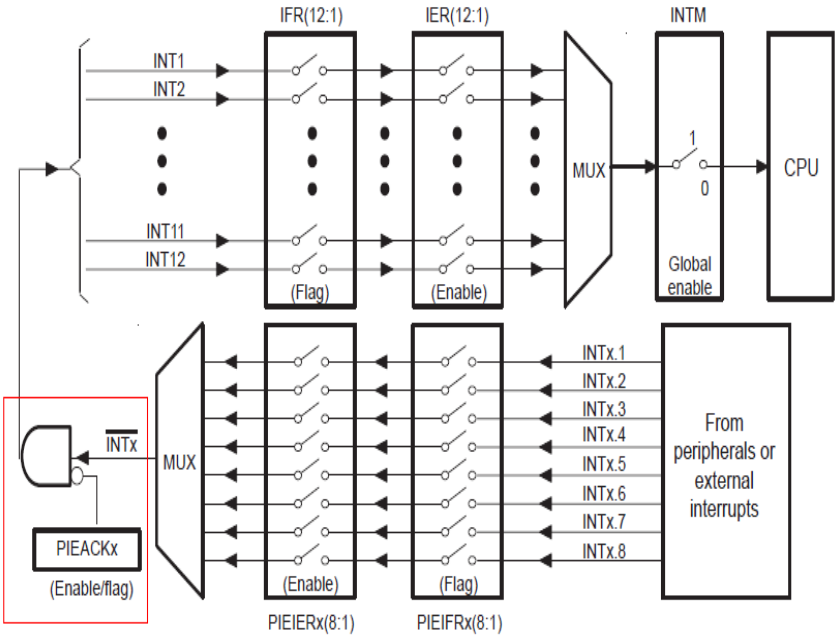
```
// Enable PIE group 2 interrupt 6 for T1UFINT
   PieCtrlRegs.PIEIER2.all = M_INT6;

// Enable CPU INT2 for T1UFINT:
         IER |= M_INT2;
```

بعد از اتمام تنظیمات اولیه و قبل از پایان حلقه بی نهایت، در آخرین خط تمامی وقفه ها را روشن می‌کنیم و منتظر فرآخوانی تابع MainISR خواهیم ماند.

```
// Enable global Interrupts and higher priority real-time debug events:
         EINT;   // Enable Global interrupt INTM
         ERTM;   // Enable Global realtime interrupt DBGM
```

پس از اعمال وقفه تایمر ۱ این مقدار مشروط به تایید (Acknowledge) توسط ریجیستر PIEACK اجازه انتقال به CPU را دارد.

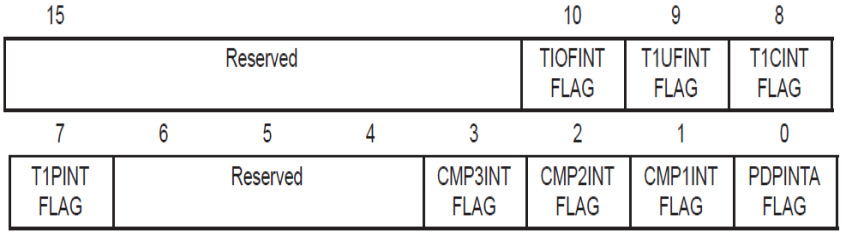
```
// Acknowledge interrupt to recieve more interrupts from PIE group 2
      PieCtrlRegs.PIEACK.all = PIEACK_GROUP2;
```



برای آنکه امکان وقفه دوباره تایمر ۱ بعد از وقفه قبلی ایجاد شود باید در ریجستر MASK بیت مربوط به Underflow تایمر ۱ را دوباره SET کرد این کار در انتهای برنامه MainISR انجام می‌شود.

```
// Enable more interrupts from this timer
        EvaRegs.EVAIMRA.bit.T1UFINT = 1;
```

از آنجا که ریجسیتر EVAIFRA در واحد مدیریت رویداد اول امکان ایجاد وقفه به تایمر ۱ را در حالت overflow و غیره را هم فراهم می کند، به جهت حصول اطمینان تاکید می کنیم که فقط Underflow تایمر ۱ تنها پرچم ریجستر اعلام وقفه باشد.



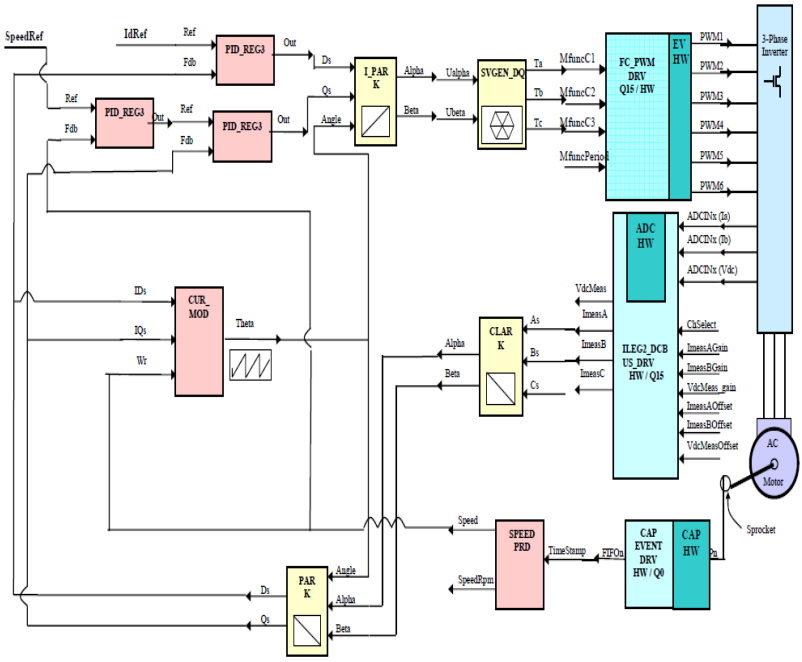

```
// Note: To be safe, use a mask value to write to the entire
        // EVAIFRA register.  Writing to one bit will cause a read-modify-write
        // operation that may have the result of writing 1's to clear
        // bits other then those intended.
  EvaRegs.EVAIFRA.all = BIT9;
```

ماژولهای نرم افزاری روش کنترل پیشنهادی

بعد از اینکه تنظیمات اولیه در برنامه صورت گرفت. برای اجرای روش کنترلی از چندین ماژول (زیربخش) نرم افزاری بهره گرفته می‌شود. سیستم کنترل پیشنهادی شامل ۱۱ ماژول با نام‌های زیر است:

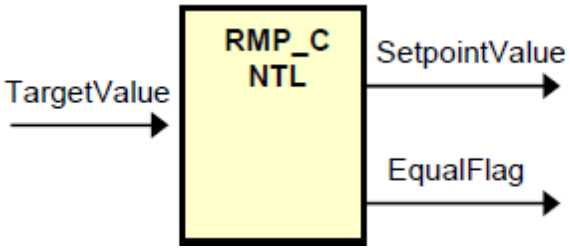
1. PWMDAC
2. PID_REG3
3. RAMP_CNTL
4. CUR_MOD
5. CLARKE
6. PARK/IPARK
7. SPEED_PRD
8. SVGEN_DQ
9. CUR_MOD_CONSTPWMGEN
10. QEP_NO_INDEX_DRV
11. ADC04b_DRV

در شکل زیر ارتباط این ۱۱ ماژول با یکدیگر جهت پیاده سازی سیستم کنترل SVPWM نشان داده شده است، IdRef مولد شار در موتور و IqRef مولد گشتاور در موتور است.



ماژول RMP_CNTL

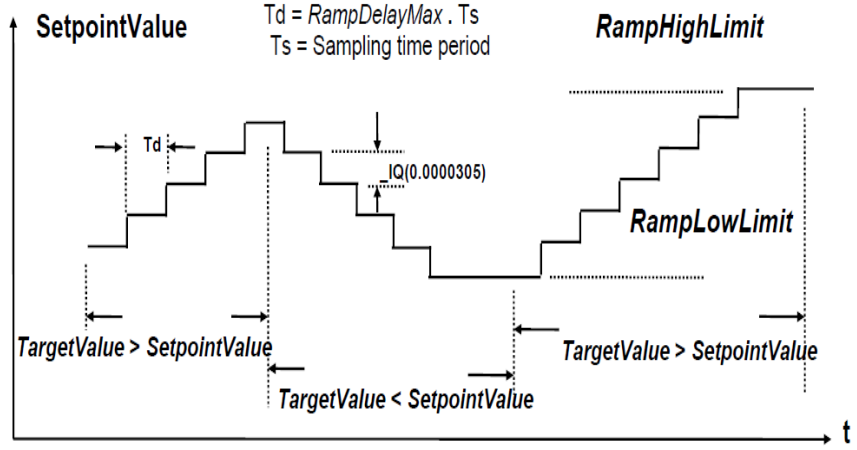
هنگامی که دستور سرعت‌گیری یا کاهش سرعت صادر می‌شود این ماژول با شیب مشخص و در زمانی معین سرعت را به صورت پله های معین (هر پله نسبت به پله قبل ۰,۰۰۰۰۳۰۵) کاهش یا افزایش می‌دهد و لحظه به لحظه این SetpointValue را اعمال می کند تا به TargetValue برسد.



برنامه		RMP_CNTL
برنامه های فراخواننده		interrupt void MainISR
فایل های توابع مورد نیاز		rmp_cntl.c, rmp_cntl.h
پارامترها	ورودی	۱-TargetValue:یا دستور سرعت مشخص
	خروجی	۱-SetpointValue:سرعت اعمالی در شیب کاهش و افزایش ۲-EqualFlag : یا پرچم برابری مقدار درخواستی و تنظیمی
	پارامتر های RMP_CNTL	1- RampDelayMax : بیشتری مقدار مجاز تاخیر در هر شیب 2- RampLowLimit: حد کمینه برای اشباع سرعت 3- RampHighLimit: حد بیشینه ای برای اشباع سرعت
	داخلی	RampDelayCount: تاخیر افزایشی

در شکل زیر یک روند کاهش و افزایش سرعت نشان داده شده است. متغیر Td مدت زمان تاخیر بین هر دو گام متوالی است.که از ضرب بیشتری مقدار مجاز تاخیر در هر شیب در پیود زمان نمونه گیری به دست می آید.

```
ramp time =(TargetValue-SetpointValue)x RampDelayMax x SamplingloopTimePeriod
```



مقدار دهی به پارامترهای اولیه:

```
// Instance a ramp controller to smoothly ramp the Frequency
RMPCNTL rc1 = RMPCNTL_DEFAULTS;
```

مقدار پیش فرض RampDelayMax درRMP_CNTL.H برابر ۴۰ در نظر گرفته شده است.

بررسی تابع: این تابع اگر تغییر سرعت را حس کند در حلقه ای به اندازه RampDelayMax یا تاخیر افزایشی که در اینجا ۴۰ سیکل است صبر کرده و دوباره نیاز تغییر سرعت را بررسی می کند. در صورت نیاز به کاهش سرعت ۰,۰۰۰۰۳۰۵ از رفرنس سرعت می کاهد و آن را اعمال می‌دارد و بعد ۴۰

سیکل صبر کرده نیاز تغییر سرعت را بررسی می کند و آنقدر این کار را ادامه می دهد تا نیاز برطرف شود. در کاهش و افزایش سرعت ممکن است نقطه سرعت تنظیمی غیر مجاز باشد لذا حد کمینه و بیشینه ای برای اشباع آن در نظر گرفته شده است. در زمان برابری و رفع کامل نیاز پرچم EqualFlag را با مقدار 7FFFFFFFh بالا می برد.

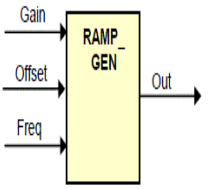
```
void rmp_cntl_calc(RMPCNTL *v)
{
    _iq tmp;

    tmp = v->TargetValue - v->SetpointValue;

     if (_IQabs(tmp) > _IQ(0.0000305))
     {
       v->RampDelayCount += 1;
       if (v->RampDelayCount >= v->RampDelayMax)
       {
        if (v->TargetValue >= v->SetpointValue)
        {
         v->SetpointValue += _IQ(0.0000305);       // 0.0000305 is resolution of Q15
         if (v->SetpointValue > v->RampHighLimit)
           v->SetpointValue = v->RampHighLimit;
         v->RampDelayCount = 0;
        }
        else
        {
         v->SetpointValue -= _IQ(0.0000305);       // 0.0000305 is resolution of Q15
         if (v->SetpointValue < v->RampLowLimit)
           v->SetpointValue = v->RampLowLimit;
         v->RampDelayCount = 0;
        }
       }
     }
    else  v->EqualFlag = 0x7FFFFFFF;
}
```

نمونه کد:

```
// ---------------------------------------------------------------------------
//   Connect inputs of the RMP_CNTL module and call the Ramp control
//   calculation function.
// ---------------------------------------------------------------------------
   rc1.TargetValue = _IQ(SpeedRef);
   rc1.calc(&rc1);

// ---------------------------------------------------------------------------
```

جهت شبیه سازی در محیط نرم افزار MATLAB این ماژول به صورت زیر برنامه نویسی و پیاده شده است و در تابع ای با نام rmp_cntl.M قرار گرفته است. جهت مشاهده این فایل می توانید به CD پیوست مراجعه فرمایید.

ماژول RAMP_GEN



برنامه		rampgen
برنامه های فراخواننده		interrupt void MainISR
فایل های توابع مورد نیاز		rampgen.c, rampgen.h
پارامترها	ورودی	Freq: فرکانس رمپ Gain: گین زوایه

		Offset: آفست زاویه
	خروجی	Out: خروجی زاویه
	داخلی	StepAngleMax Angle: مدل سازی زاویه بین قاب ساکن و قاب در حال حرکت

نمونه کد:

```
//      Instance a ramp generator to simulate an Anglele
RAMPGEN rg1 = RAMPGEN_DEFAULTS;
```

```
  // Initialize RAMPGEN module
     rg1.StepAngleMax = _IQ(BASE_FREQ*T);
```

مقادیر پارامترها هم به صورت زیر تعریف شده است.(مقادیر پریونیت هستند)، مقدار Gain وOffset هر دو برابر یک پریونیت و مابقی پارامتر ها صفر در نظر گرفته شده اند. مقدار BASE_FREQ در PARAMETER.H برابر فرکانس نامی موتور تعیین شده است. مقدار T نیز به صورت زیر تعریف شده است.

```
float32 T = 0.001/ISR_FREQUENCY;  // Samping period (sec), see parameter.h
```

```
// ---------------------------------------------------------------------------
//    Connect inputs of the RAMP GEN module and call the Ramp generator
//    calculation function.
// ---------------------------------------------------------------------------
    rg1.Freq = rc1.SetpointValue;
    rg1.calc(&rg1);
```

فایل rampgen.C به دنبال تولید نمودار ولتاژ فرکانس از روی پارامترها می‌باشد.

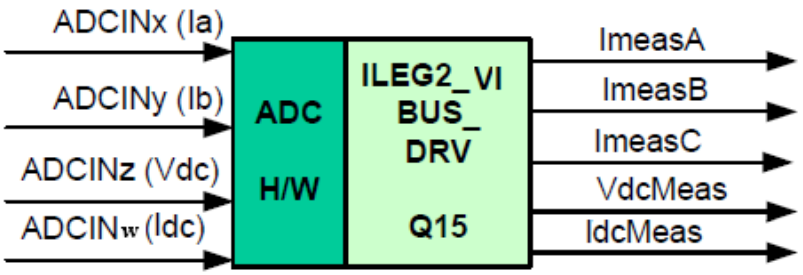
تشریح توابع:

```
void rampgen_calc(RAMPGEN *v)
{

// Compute the angle rate
       v->Angle += _IQmpy(v->StepAngleMax,v->Freq);

// Saturate the angle rate within (-1,1)
       if (v->Angle>_IQ(1.0))
         v->Angle -= _IQ(1.0);
       else if (v->Angle<_IQ(-1.0))
         v->Angle += _IQ(1.0);

// Compute the ramp output
      v->Out = _IQmpy(v->Angle,v->Gain) + v->Offset;

// Saturate the ramp output within (-1,1)
      if (v->Out>_IQ(1.0))
         v->Out -= _IQ(1.0);
       else if (v->Out<_IQ(-1.0))
         v->Out += _IQ(1.0);

}
```

جهت شبیه سازی در محیط نرم افزار MATLAB این ماژول به صورت زیر برنامه نویسی و پیاده‌سازی شده است و در تابع ای با نام rampgen.M قرار گرفته است.

ماژول F281XILEG_VI

این ماژول ۴ ورودی آنالوگ محاسبه ولتاژ لینک DC و جریان های ۲ فاز و جریان تفاضلی لینک DC را به ۵ خروجی دیجیتال با گین و آفست قابل تنظیم تبدیل می کند. این تبدیل با رخداد underflow تایمر ۱ رخ می‌دهد.



برنامه		F281XILEG_VI
برنامه های فراخواننده		interrupt void MainISR
فایل های توابع مورد نیاز		F281XILEG_VI.C, F281XILEG_VI.H
پارامترها	ورودی	ADCINx وADCINy وADCINz وADCINw که بر اساس پارامتر Ch_sel محل آن مشخص می شوند.
	خروجی	ImeasA: مقدار جریان فاز اول ImeasB: مقدار جریان فاز دوم ImeasC: مقدار جریان فاز سوم VdcMeas: ولتاژ لینک DC IdcMeas: جریان تفاضلی لینک DC
	پارامتر ماژول	Ch1Gain تا Ch4Gain: ۴ پارامتر تنظیم گین Ch1Offset تا Ch4Offset: ۴ پارامتر تنظیم آفست Ch_sel: یک عد هگزا دسیمال اشاره به پایه های ADC

در فایل F281XILEG_VI.C دو تابع F281X_ileg2 _acdcbus_drv _init و F281X_ileg2_acdcbus _drv_ read قرار دارد.

تنظیمات گین و آفست و کانال های فعال ADC، تعداد تبدیل حداکثر ۴ و دستور فعال شدن مبدل با عمل under flow تایمر یک همگی در F281X_ileg2_acdcbus_drv_init صورت می پذیرد. اگر چه از وقفه مبدل آنالوگ به دیجیتال استفاده نخواهد شد اما وقفه جهت بررسی اتمام عملکرد مبدل فعال می‌گردد.

```
void F281X_ileg2_acdcbus_drv_init(ILEG2DCBUSMEAS *p)
{
   DELAY_US(ADC_usDELAY);

   AdcRegs.ADCTRL1.all = ADC_RESET_FLAG;          // Reset the ADC Module
         asm(" NOP ");
         asm(" NOP ");

   AdcRegs.ADCTRL3.bit.ADCBGRFDN = 0x3;           // Power up bandgap/reference circuitry
         DELAY_US(ADC_usDELAY);                              // Delay before powering up rest of ADC

   AdcRegs.ADCTRL3.bit.ADCPWDN = 1;                     // Power up rest of ADC
   AdcRegs.ADCTRL3.bit.ADCCLKPS = 6;              // Set up ADCTRL3 register
         DELAY_US(ADC_usDELAY);

   AdcRegs.ADCTRL1.all = ADCTRL1_INIT_STATE_ILEG2;              // Set up ADCTRL1 register
   AdcRegs.ADCTRL2.all = ADCTRL2_INIT_STATE_ILEG2;        // Set up ADCTRL2 register
         AdcRegs.ADCMAXCONV.bit.MAX_CONV1 = 1;      // Specify 4 conversions
   AdcRegs.ADCCHSELSEQ1.all = p->ChSelect;     // Configure channel selection

         EvaRegs.GPTCONA.bit.T1TOADC = 1;               // Set up EV Trigger with Timer1 UF
}
```

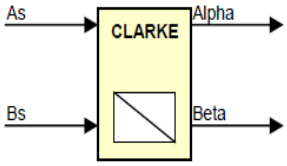
در تابع F281X_adc04b_drv_read ابتدا بیت وقفه مبدل جهت اطمینان از پایان عمل تبدیل خوانده می‌شود و سپس بیت مربوطه Clear می‌گردد. سپس نتایج اعداد ذخیره شده در رجیستر ADCRESULT0-15 خوانده می‌شود. در پایان جهت تبدیل بعدی مبدل باز نشانی می‌گردد.

اعداد خوانده شده از رجیستر ADCRESULT با عدد ۸۰۰۰، Exclusive OR میگردد تا ۱/۵ ولت به عنوان سطح صفر سیگنال ورودی در نظر گرفته شود. با توجه به یک بودن گین و صفر بودن آفست، عدد دو جریان AC و تفاضل آن دو برای فاز سوم و جریان تفاضلی و ولتاژ لینک DC به عنوان ۵ خروجی محاسبه خواهد شد.

```
void F281X_ileg2_acdcbus_drv_read(ILEG2DCBUSMEAS *p)
{
     int16 DatQ15;
     int32 Tmp;

      // Wait until ADC conversion is completed
      while (AdcRegs.ADCST.bit.SEQ1_BSY == 1)
      {};

      DatQ15 = AdcRegs.ADCRESULT0^0x8000;           // Convert raw result to Q15 (bipolar signal)
      Tmp = (int32)p->ImeasAGain*(int32)DatQ15; // Tmp = gain*dat => Q28 = Q13*Q15
      p->ImeasA = (int16)(Tmp>>13);           // Convert Q28 to Q15
      p->ImeasA += p->ImeasAOffset;           // Add offset
      p->ImeasA *= -1;                        // Positive direction, current flows to motor

      DatQ15 = AdcRegs.ADCRESULT1^0x8000;           // Convert raw result to Q15 (bipolar signal)
      Tmp = (int32)p->ImeasBGain*(int32)DatQ15; // Tmp = gain*dat => Q28 = Q13*Q15
      p->ImeasB = (int16)(Tmp>>13);           // Convert Q28 to Q15
      p->ImeasB += p->ImeasBOffset;           // Add offset
      p->ImeasB *= -1;                        // Positive direction, current flows to motor

      DatQ15 = (AdcRegs.ADCRESULT2>>1)&0x7FFF;   // Convert raw result to Q15 (unipolar signal)
      Tmp = (int32)p->VdcMeasGain*(int32)DatQ15; // Tmp = gain*dat => Q28 = Q13*Q15
      if (Tmp > 0x0FFFFFFF)                   // Limit Tmp to 1.0 in Q28
        Tmp = 0x0FFFFFFF;
      p->VdcMeas = (int16)(Tmp>>13);           // Convert Q28 to Q15
      p->VdcMeas += p->VdcMeasOffset;          // Add offset

      DatQ15 = (AdcRegs.ADCRESULT2>>1)&0x7FFF;   // Convert raw result to Q15 (unipolar signal)
      Tmp = (int32)p->VdcMeasGain*(int32)DatQ15; // Tmp = gain*dat => Q28 = Q13*Q15
      if (Tmp > 0x0FFFFFFF)                   // Limit Tmp to 1.0 in Q28
        Tmp = 0x0FFFFFFF;
      p->VdcMeas = (int16)(Tmp>>13);           // Convert Q28 to Q15
      p->VdcMeas += p->VdcMeasOffset;          // Add offset


      DatQ15 = (AdcRegs.ADCRESULT3>>1)&0x7FFF;   // Convert raw result to Q15 (unipolar signal)
      Tmp = (int32)p->IdcMeasGain*(int32)DatQ15; // Tmp = gain*dat => Q28 = Q13*Q15
      if (Tmp > 0x0FFFFFFF)                   // Limit Tmp to 1.0 in Q28
        Tmp = 0x0FFFFFFF;
      p->IdcMeas = (int16)(Tmp>>13);           // Convert Q28 to Q15
      p->IdcMeas += p->IdcMeasOffset;          // Add offset

      p->ImeasC = -(p->ImeasA + p->ImeasB);     // Compute phase-c current

      AdcRegs.ADCTRL2.all |= 0x4040;           // Reset the sequence

}
```

ماژول کلارک

تبدیل کلارک، تبدیل ولتاژ، جریان، شار یا بار الکتریکی سه فاز بالانس به یک قاب مرجع ساکن با ۲ متغیر فاز بالانس آلفا و بتا است.

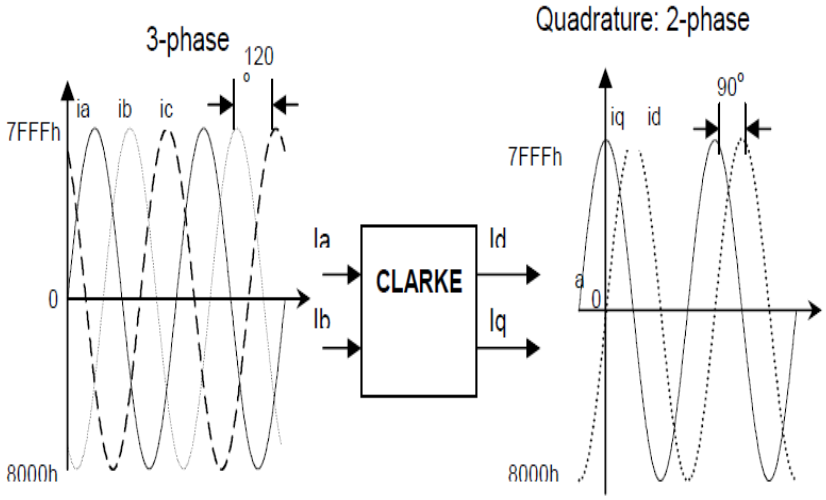


برنامه	CLARK	
برنامه های فراخواننده	interrupt void MainISR	
فایل های توابع مورد نیاز	CLARK.c, CLARK.h	
پارامترها	ورودی	As : فاز اول در سیستم سه فاز بالانس Bs : فاز دوم در سیستم سه فاز بالانس
	خروجی	Alpha : مولفه جریان تبدیل شده در محور مستقیم D Beta : مولفه جریان تبدیل شده در محور عمود Q

نمونه کد:

```
// Instance a few transform objects
CLARKE clarke1 = CLARKE_DEFAULTS;
```

```
// ---------------------------------------------------------------------------
//    Connect inputs of the CLARKE module and call the clarke transformation
//    calculation function.
// ---------------------------------------------------------------------------
      clarke1.As = _IQ15toIQ((int32)ilg2_vdc1.ImeasA);
      clarke1.Bs = _IQ15toIQ((int32)ilg2_vdc1.ImeasB);
      clarke1.calc(&clarke1);
```



$$V_{ds}^{s} = V_{an}$$

$$V_{qs}^{s} = \frac{1}{\sqrt{3}}\left(V_{an} + 2V_{bn}\right)$$

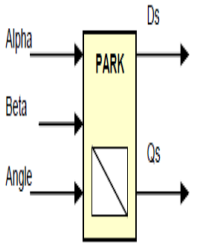
تشریح توابع:

```
void clarke_calc(CLARKE *v)
{
   v->Alpha = v->As;
   v->Beta = _IQmpy((v->As + _IQmpy(_IQ(2),v->Bs)),_IQ(0.57735026918963));
}
```

جهت شبیه سازی در محیط نرم افزار MATLAB این ماژول به صورت زیر برنامه نویسی و پیاده‌سازی شده است و در تابع ای با نام CLARK.M قرار گرفته است.

ماژول PARK

برای ساده سازی تحلیل مدار سه فاز به کار می‌رود. در حالتی که مدار سه فاز متعادل است استفاده از تبدیل dqo سه کمیت متناوب را را به دو کمیت DC کاهش می‌دهد. می‌توان محاسبات ساده شده را روی کمیت‌های DC موهومی انجام داد و سپس با استفاده از تبدیل معکوس، نتایج سه فاز AC را بدست آورد. از این تبدیل معمولاً برای ساده سازی تحلیل ماشین‌های سه فاز سنکرون یا برای ساده سازی محاسبات کنترل اینورتر استفاده می‌شود.



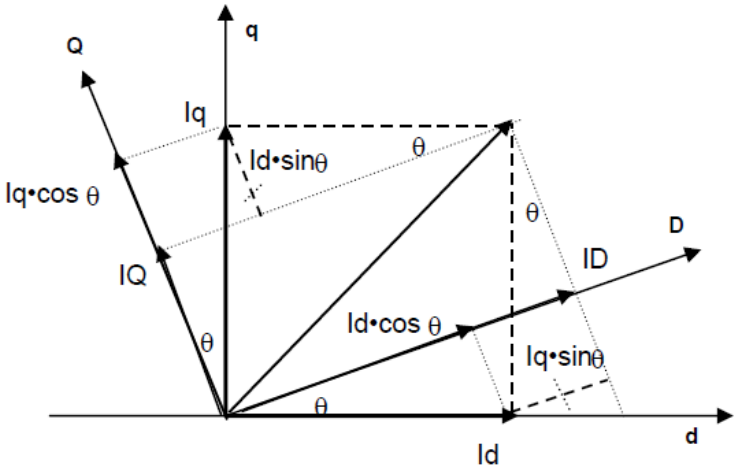
برنامه		PARK
برنامه های فراخواننده		interrupt void MainISR
فایل های توابع مورد نیاز		PARK.c, PARK.h
پارامترها	ورودی	Alpha :متغیر در قاب ثابت استاتور در محور مستقیم D Beta: متغیر در قاب ثابت استاتور در محور عمود Q Angle : زاویه بین قاب ساکن و قاب در حال حرکت
	خروجی	Ds :مولفه جریان تبدیل شده در محور مستقیم D QS : مولفه جریان تبدیل شده در محور عمود Q

نمونه کد:

```
    PARK park1 = PARK_DEFAULTS;

// ---------------------------------------------------------------------------
//    Connect inputs of the PARK module and call the park transformation
//    calculation function.
// ---------------------------------------------------------------------------
    park1.Alpha = clarke1.Alpha;
    park1.Beta = clarke1.Beta;
    park1.Angle = rg1.Out;
    park1.calc(&park1);
```

فایل PARK.C به دنبال تولید مولفه جریان تبدیل شده در محور مستقیم و عمود می‌باشد.

$$\begin{cases} ID = Id \times \cos\theta + Iq \times \sin\theta \\ IQ = -Id \times \sin\theta + Iq \times \cos\theta \end{cases}$$

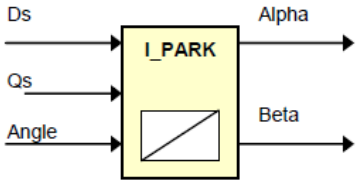


```
void park_calc(PARK *v)
{

    _iq Cosine,Sine;

// Using look-up IQ sine table
     Sine = _IQsinPU(v->Angle);
     Cosine = _IQcosPU(v->Angle);

     v->Ds = _IQmpy(v->Alpha,Cosine) + _IQmpy(v->Beta,Sine);
     v->Qs = _IQmpy(v->Beta,Cosine) - _IQmpy(v->Alpha,Sine);

}
```

جهت شبیه سازی در محیط نرم افزار MATLAB این ماژول به صورت زیر برنامه نویسی و پیاده‌سازی شده است و در تابع ای با نام PARK.M قرار گرفته است.

ماژول IPARK

این ماژول عکس ماژول پارک است.یعنی از به دو کمیت DC به معادل سه فاز دوباره برخواهیم گشت.



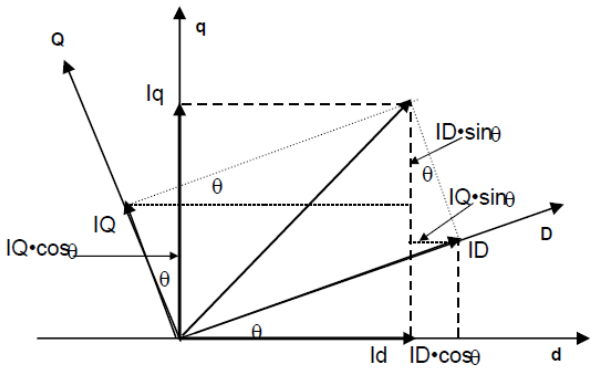
برنامه	IPARK	
برنامه های فراخواننده	interrupt void MainISR	
فایل های توابع مورد نیاز	IPARK.c, IPARK.h	
پارامترها	ورودی	Angle: Ds :مولفه جریان تبدیل شده در محور مستقیم D QS : مولفه جریان تبدیل شده در محور عمود Q زاویه بین قاب ساکن و قاب در حال حرکت
	خروجی	Alpha :متغیر در قاب ثابت استاتور در محور مستقیم D Beta: متغیر در قاب ثابت استاتور در محور عمود Q

نمونه کد:

```
IPARK ipark1 = IPARK_DEFAULTS;
```

```
// ---------------------------------------------------------------------------
//   Connect inputs of the INV_PARK module and call the inverse park transformation
//   calculation function.
// ---------------------------------------------------------------------------
    ipark1.Ds = pid1_id.Out;
    ipark1.Qs = pid1_iq.Out;
    ipark1.Angle = rg1.Out;
    ipark1.calc(&ipark1);
```

$$\begin{cases} Id = ID \times \cos\theta - IQ \times \sin\theta \\ Iq = ID \times \sin\theta + IQ \times \cos\theta \end{cases}$$



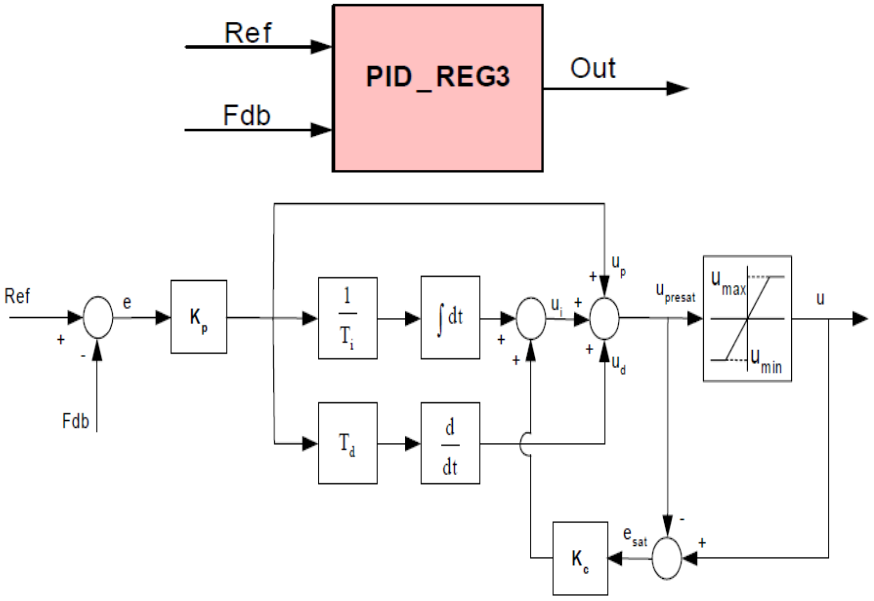
بررسی توابع:

```
void ipark_calc(IPARK *v)
{

   _iq Cosine,Sine;

// Using look-up IQ sine table
     Sine = _IQsinPU(v->Angle);
     Cosine = _IQcosPU(v->Angle);

     v->Alpha = _IQmpy(v->Ds,Cosine) - _IQmpy(v->Qs,Sine);
     v->Beta = _IQmpy(v->Qs,Cosine) + _IQmpy(v->Ds,Sine);
}
```

جهت شبیه سازی در محیط نرم افزار MATLAB این ماژول به صورت زیر برنامه نویسی و پیاده‌سازی شده است و در تابع ای با نام IPARK.M قرار گرفته است.

ماژول PID_REG3

این ماژول یک PID کنترلر ۳۲ بیتی است که می‌تواند به صورت PI یا PD نیز به کار گرفته شود. این ماژول دارای تمهیدات رفع خطای windup است. بلوک دیاگرام این PID کنترلر به صورت زیر است.



برنامه	PID_REG3	
برنامه های فراخواننده	interrupt void MainISR	
فایل های توابع مورد نیاز	pid_reg3.c, pid_reg3.h	
پارامترها	ورودی	Ref یا ورودی مرجع سرعت که از ماژول RAMP-CNTL Fdb یا فیدبک سرعت از ماژول SPEED_PRD

	خروجی	Out یا خروجی فرمان سرعت محاسبه شده
	پارامترهای PID	KP-1 یا ضریب کنترلر تناسبی KI-2 یا ضریب انتگرال گیر Kd-3 یا ضریب مشتق گیر Kc-4 یا ضریب انتگرال گیر رفع کننده خطا Umax-5 یا بیشنه مجاز مقدار خروجی سرعت جهت saturation Umin-6 یا کمینه مجاز مقدار خروجی سرعت جهت saturation
	داخلی	OutPreSat-1 یا خروجی فرمان سرعت PID قبل saturation Up-2 یا مقدار خروجی کنترلر تناسبی Up1-3 یا مقدار خروجی قبلی کنترلر تناسبی Ui-4 یا مقدار خروجی انتگرال گیر Ud-5 یا مقدار خروجی مشتق گیر Err-6 تفاضل ورودی مرجع و مقدار فیدبک

مقدار دهی به پارامترهای اولیه:

```
// Initialize the PID_REG3 module for speed
   pid1_spd.Kp = _IQ(0.5);
          pid1_spd.Ki = _IQ(T*SpeedLoopPrescaler/10);
          pid1_spd.Kd = _IQ(0/(T*SpeedLoopPrescaler));
          pid1_spd.Kc = _IQ(0.2);
   pid1_spd.OutMax = _IQ(0.05);
   pid1_spd.OutMin = _IQ(-0.05);
```

مقادیر کمینه و بیشینه مجاز مقدار خروجی سرعت Umax و Umin را این شرکت به عنوان متغیر ورودی PID تعریف کرده است ولی ما در اینجا آن را جزء پارامترهای PIDآورده ایم.

بررسی توابع:

تابع PID با استفاده از روابط زیر پیاده می‌شود، در هر لحظه خروجی PID تابع خروجی ناشی از سیگنال خطا به مجموع کنترل های تناسبی و انتگرال گیر و مشتق گیر باید باشد و این خروجی بعد از محاسبه باید در بین حدود تعریفی اشباع شود. لذا خواهیم داشت:

$$u_{presat}(t) = u_p(t) + u_i(t) + u_d(t) \quad (1)$$

$$u_p(t) = K_P e(t) \quad (2)$$

$$u_i(k) = u_i(k-1) + K_P \frac{T}{T_i} e(k) + K_c \left(u(k) - u_{presat}(k)\right) \quad (3)$$

$$u_d(k) = K_P \frac{T_d}{T} \left(e(k) - e(k-1)\right) \quad (4)$$

$$K_i = \frac{T}{T_i}, \text{ and } K_d = \frac{T_d}{T} \quad (5)$$

u(t) is the output of PID controller
$u_{presat}(t)$ is the output before saturation
e(t) is the error between the reference and feedback variables
K_p is the proportional gain of PID controller
T_i is the integral time (or reset time) of PID controller
T_d is the derivative time of PID controller
K_c is the integral correction gain of PID controller

```
void pid_reg3_calc(PIDREG3 *v)
{
   // Compute the error
   v->Err = v->Ref - v->Fdb;
   // Compute the proportional output
   v->Up = _IQmpy(v->Kp,v->Err);

   // Compute the integral output
   v->Ui = v->Ui + _IQmpy(v->Ki,v->Up) + _IQmpy(v->Kc,v->SatErr);

   // Compute the derivative output
   v->Ud = _IQmpy(v->Kd,(v->Up - v->Up1));

   // Compute the pre-saturated output
   v->OutPreSat = v->Up + v->Ui + v->Ud;
   // Saturate the output
   if (v->OutPreSat > v->OutMax)
     v->Out =  v->OutMax;
   else if (v->OutPreSat < v->OutMin)
     v->Out =  v->OutMin;
   else
     v->Out = v->OutPreSat;

   // Compute the saturate difference
   v->SatErr = v->Out - v->OutPreSat;

   // Update the previous proportional output
   v->Up1 = v->Up;
}
```

نمونه کد: جهت کاهش حجم محاسبات هر ۲۰ بار فقط یک بار محاسبه PID انجام شده است.

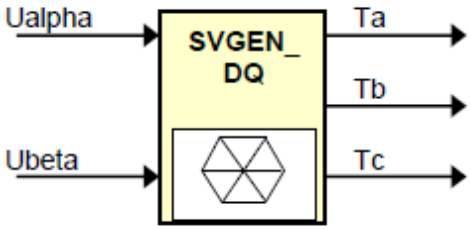
```
Uint16 SpeedLoopPrescaler = 20;     // Speed loop prescaler
Uint16 SpeedLoopCount = 1;          // Speed loop counter
```

```
// ---------------------------------------------------------------------------
//   Connect inputs of the PID_REG3 modules and compute the PID speed
//   controller.
// ---------------------------------------------------------------------------
   if (SpeedLoopCount == SpeedLoopPrescaler)
    {
     pid1_spd.Ref = rc1.SetpointValue;
            pid1_spd.Fdb = speed1.Speed;
            pid1_spd.calc(&pid1_spd);
     SpeedLoopCount = 1;
    }
   else SpeedLoopCount ++;

// ---------------------------------------------------------------------------
```

جهت شبیه سازی در محیط نرم افزار MATLAB این ماژول به صورت زیر برنامه نویسی و پیاده شده است و در تابع ای با نام pid_reg3.M قرار گرفته است. جهت مشاهده این فایل می توانید به CD پیوست مراجعه فرمایید.

ماژول SVGEN_DQ

این ماژول با اخذ دو مقدار ولتاژ آلفا و بتا به دنبال تولید ولتاژ رفرنس استاتور است که مورد نیاز در تکنیک space vector PWM می باشد و سه زمان Ta,Tb,Tc زمان اعمال بردارهای فعال برای PWM1,PWM3,PWM5 را تولید می‌کند.



برنامه		
SVGEN_DQ		
برنامه های فراخواننده		
interrupt void MainISR		
فایل های توابع مورد نیاز		
SVGEN_DQ.c, SVGEN_DQ.h		
پارامترها	ورودی	مقدار ولتاژ آلفا و بتا
	خروجی	زمان اعمال بردارهای فعال برای PWM1,PWM3,PWM5
	داخلی	1-Alpha یا زاویه سکتور ولتاژ 2-SectorPointer یا شماره سکتور

نمونه کد:

```
// ---------------------------------------------------------------------------
//    Connect inputs of the SVGEN_DQ module and call the space-vector gen.
//    calculation function.
// ---------------------------------------------------------------------------
    svgen_dq1.Ualpha = ipark1.Alpha;
        svgen_dq1.Ubeta = ipark1.Beta;
        svgen_dq1.calc(&svgen_dq1);
```

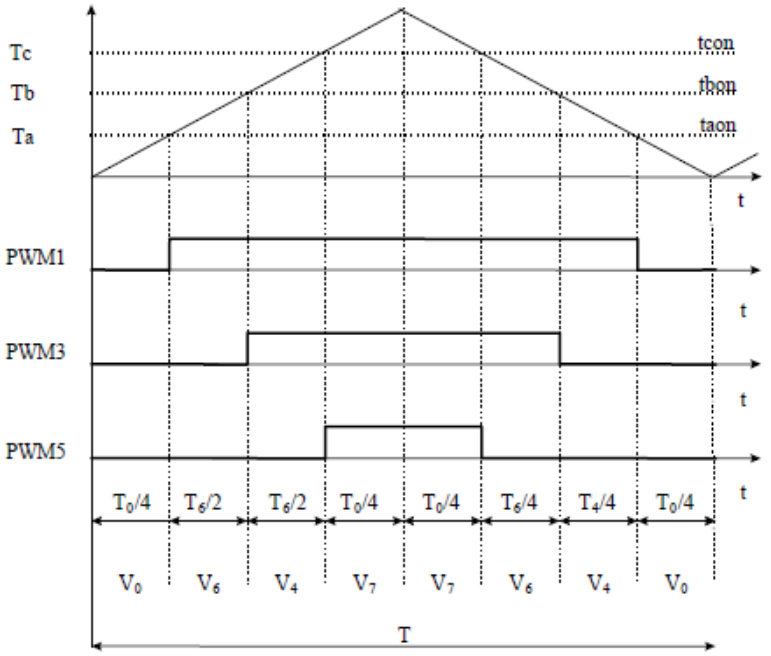
تابع svgendq_calc که SVGEN_dq.C موجود است قسمت اصلی محاسبات را در بر دارد.

ابتدا باید از مولفه های آلفا و بتا تبدیل اینورس کلارک گرفت تا به مدل سه فاز برگردیم. مطابق دایره معروف SVPWMدر هر سکتور که در اینجا هم ۶۰ درجه تعریف شده است، با استفاده از مقدار ولتاژ ها موقعیت سکتور را تعیین می کنیم.

کد مربوط به این قسمت در SVGEN_dq.C:

```
        _iq Va,Vb,Vc,t1,t2;
        Uint32 Sector = 0;  // Sector is treated as Q0 - independently with global Q

// Inverse clarke transformation
    Va = v->Ubeta;
    Vb = _IQmpy(_IQ(-0.5),v->Ubeta) + _IQmpy(_IQ(0.8660254),v->Ualpha);
    Vc = _IQmpy(_IQ(-0.5),v->Ubeta) - _IQmpy(_IQ(0.8660254),v->Ualpha);

// 60 degree Sector determination
    if (Va>_IQ(0))
       Sector = 1;
    if (Vb>_IQ(0))
       Sector = Sector + 2;
    if (Vc>_IQ(0))
       Sector = Sector + 4;

// X,Y,Z (Va,Vb,Vc) calculations
    Va = v->Ubeta;                                                    // X = Va
    Vb = _IQmpy(_IQ(0.5),v->Ubeta) + _IQmpy(_IQ(0.8660254),v->Ualpha);   // Y = Vb
    Vc = _IQmpy(_IQ(0.5),v->Ubeta) - _IQmpy(_IQ(0.8660254),v->Ualpha);   // Z = Vc
```

الگوی سوییچینگ نیز مطابق تعریف SVPWM به صورت زیر است:



لذا در سکتور های متفاوت مقدار سه زمان Ta,Tb,Tc متفاوت و به صورت زیر است:

$$X = U_{beta}$$
$$Y = \frac{1}{2}\left(\sqrt{3}U_{alfa} + U_{beta}\right)$$
$$Z = \frac{1}{2}\left(-\sqrt{3}U_{alfa} + U_{beta}\right)$$

$$\begin{cases} t_{aon} = \dfrac{PWMPRD - t_1 - t_2}{2} \\ t_{bon} = t_{aon} + t_1 \\ t_{con} = t_{bon} + t_2 \end{cases}$$

sectors	U_0, U_{60}	U_{60}, U_{120}	U_{120}, U_{180}	U_{180}, U_{240}	U_{240}, U_{300}	U_{300}, U_0
Ta	taon	tbon	tcon	tcon	tbon	taon
Tb	tbon	taon	taon	tbon	tcon	tcon
Tc	tcon	tcon	tbon	taon	taon	tbon

Sector	U_0, U_{60}	U_{60}, U_{120}	U_{120}, U_{180}	U_{180}, U_{240}	U_{240}, U_{300}	U_{300}, U_0
t1	-Z	Z	X	-X	-Y	Y
t2	X	Y	Y	Z	-Z	-X

کد مربوط به این قسمت در SVGEN_dq.C و در تابع svgendq_calc به صورت زیر است:

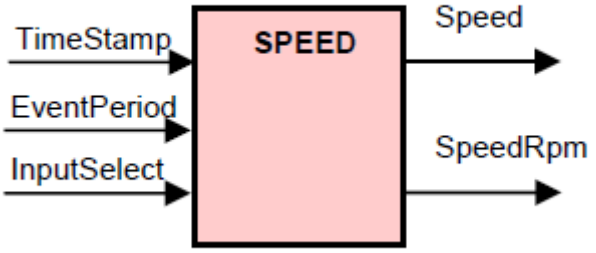
```
if (Sector==0)  // Sector 0: this is special case for (Ualpha,Ubeta) = (0,0)
{
   v->Ta = _IQ(0.5);
   v->Tb = _IQ(0.5);
   v->Tc = _IQ(0.5);
}
if (Sector==1)  // Sector 1: t1=Z and t2=Y (abc ---> Tb,Ta,Tc)
{
   t1 = Vc;
   t2 = Vb;
   v->Tb = _IQmpy(_IQ(0.5),(_IQ(1)-t1-t2));      // tbon = (1-t1-t2)/2
   v->Ta = v->Tb+t1;                             // taon = tbon+t1
   v->Tc = v->Ta+t2;                             // tcon = taon+t2
}
else if (Sector==2)  // Sector 2: t1=Y and t2=-X (abc ---> Ta,Tc,Tb)
{
   t1 = Vb;
   t2 = -Va;
   v->Ta = _IQmpy(_IQ(0.5),(_IQ(1)-t1-t2));      // taon = (1-t1-t2)/2
   v->Tc = v->Ta+t1;                             // tcon = taon+t1
   v->Tb = v->Tc+t2;                             // tbon = tcon+t2
}
else if (Sector==3)  // Sector 3: t1=-Z and t2=X (abc ---> Ta,Tb,Tc)
{
   t1 = -Vc;
   t2 = Va;
   v->Ta = _IQmpy(_IQ(0.5),(_IQ(1)-t1-t2));      // taon = (1-t1-t2)/2
   v->Tb = v->Ta+t1;                             // tbon = taon+t1
   v->Tc = v->Tb+t2;                             // tcon = tbon+t2
}
```

```
else if (Sector==4)  // Sector 4: t1=-X and t2=Z (abc ---> Tc,Tb,Ta)
{
   t1 = -Va;
   t2 = Vc;
   v->Tc = _IQmpy(_IQ(0.5),(_IQ(1)-t1-t2));      // tcon = (1-t1-t2)/2
   v->Tb = v->Tc+t1;                             // tbon = tcon+t1
   v->Ta = v->Tb+t2;                             // taon = tbon+t2
}
else if (Sector==5)  // Sector 5: t1=X and t2=-Y (abc ---> Tb,Tc,Ta)
{
   t1 = Va;
   t2 = -Vb;
   v->Tb = _IQmpy(_IQ(0.5),(_IQ(1)-t1-t2));      // tbon = (1-t1-t2)/2
   v->Tc = v->Tb+t1;                             // tcon = tbon+t1
   v->Ta = v->Tc+t2;                             // taon = tcon+t2
}
else if (Sector==6)  // Sector 6: t1=-Y and t2=-Z (abc ---> Tc,Ta,Tb)
{
   t1 = -Vb;
   t2 = -Vc;
   v->Tc = _IQmpy(_IQ(0.5),(_IQ(1)-t1-t2));      // tcon = (1-t1-t2)/2
   v->Ta = v->Tc+t1;                             // taon = tcon+t1
   v->Tb = v->Ta+t2;                             // tbon = taon+t2
}

Convert the unsigned GLOBAL_Q format (ranged (0,1)) -> signed GLOBAL_Q format
v->Ta = _IQmpy(_IQ(2.0),(v->Ta-_IQ(0.5)));
v->Tb = _IQmpy(_IQ(2.0),(v->Tb-_IQ(0.5)));
v->Tc = _IQmpy(_IQ(2.0),(v->Tc-_IQ(0.5)));
```

جهت شبیه سازی در محیط نرم افزار MATLAB این ماژول به صورت SVGEN_dq.C مجددا برنامه نویسی شده است و در تابع ای با نام SVGENdq.M قرار گرفته است. جهت مشاهده این فایل می توانید به CD پیوست مراجعه فرمایید.

ماژول SPEED_PRD

این ماژول سرعت موتور را را بر اساس تفاضل TimeStamp قدیم و جدید که آن را EventPeriod می نامند، محاسبه می‌کند و این مقدار را به صورت PU و به صورت RPM در خروجی ظاهر می‌کند.



برنامه		SPEED_PRD
برنامه های فراخواننده		interrupt void MainISR
فایل های توابع مورد نیاز		speed_pr.c, speed_pr.h
پارامترها	ورودی	۱-TimeStamp ماژول CAP_EVENT_DRV
	خروجی	۱-سرعت موتور به صورت PU ۲-سرعت موتور به صورت RPM
	داخلی	NewTimeStamp-1 OldTimeStamp-2

بیت InputSelect ماهیت ورودی به این ماژول ندارد و در هنگام به روز شدن مقدار TimeStamp فقط ماژول را فعال می کند. متغیر EventPeriod نیز ماهیت ورودی نداشته و همان طور که بیان شد فقط مقدار تفاضل TimeStamp قدیم و جدید را در خود نگه می‌دارد.

نمونه کد:

```
// ---------------------------------------------------------------------------
//   Connect inputs of the SPEED_PR module and call the speed calculation function
// ---------------------------------------------------------------------------
   if((cap1.read(&cap1))==0)           // Call the capture read function
   {
      speed1.TimeStamp=(int32)(cap1.TimeStamp);    // Read out new time stamp
      speed1.calc(&speed1);              // Call the speed calulator
   }

// ---------------------------------------------------------------------------
```

مقدار دهی به پارامترهای اولیه:

مقدار BaseRpm از رابطه تعیین RPM موتور و بر اساس پارامترهای موجود در BASE_FREQ PARAMETER.H برابر ۱۲۰ هرتز و تعداد قطب pبرابر ۴ محاسبه خواهد شد. عدد SpeedScaler نیز براساس SYSTEM_FREQUENCY برابر با ۱۵۰ مگاهرتز و BaseRpm و مقدار prescaler تایمر ۲ جهت capture یعنی ۱۲۸ و تعداد دندانه های انکدر یعنی ۱۰۰۰عدد، به صورت زیر محاسبه و این مقدار به عنوان مقدار مرجع سرعت جهت تولید مقادیر پریونیت سرعت به کار می رود.

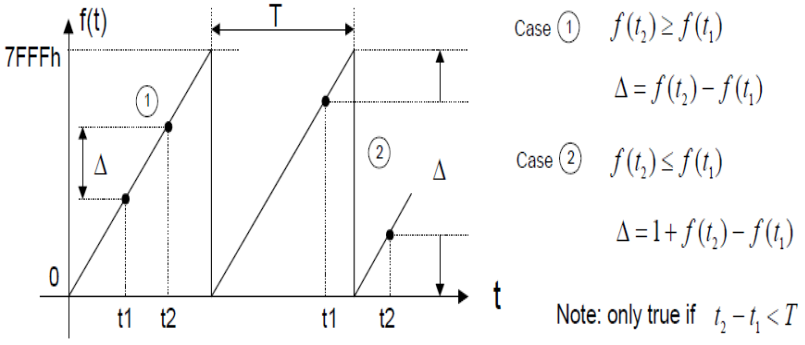
$$SpeedScaler = \frac{60}{T_{CLK} \times K_p \times n \times BaseRpm}$$

```
Initialize the SPEED_PR module
// x128-T2, 150MHz, 1000-teeth sprocket
speed1.InputSelect = 0;
      speed1.BaseRpm = 120*BASE_FREQ/P;
      speed1.SpeedScaler = 60*(SYSTEM_FREQUENCY*1000000/1000)*1/(128*speed1.BaseRpm);
```

تشریح توابع:

```
void speed_prd_calc(SPEED_MEAS_CAP *v)
{

  if (v->InputSelect == 0)
  {
    v->OldTimeStamp = v->NewTimeStamp;
    v->NewTimeStamp = v->TimeStamp;
    v->EventPeriod = v->NewTimeStamp - v->OldTimeStamp;

    if (v->EventPeriod < 0)
      v->EventPeriod += 32767;   // 0x7FFF = 32767
   }

    v->Speed = _IQdiv(v->SpeedScaler,v->EventPeriod);

// Q0 = Q0*GLOBAL_Q => _IQXmpy(), X = GLOBAL_Q
  v->SpeedRpm = _IQmpy(v->BaseRpm,v->Speed);

}
```

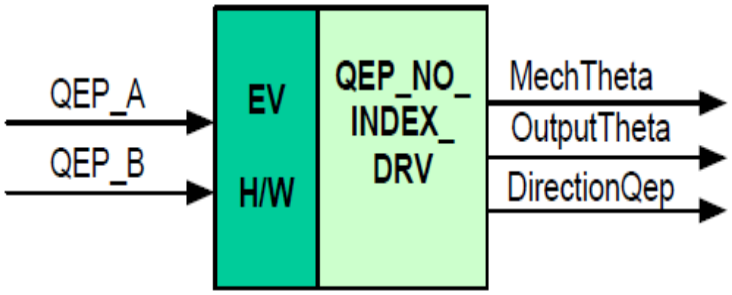
علت بررسی مثبت یا منفی بودن EventPeriod این است که مقدار تفاضل TimeStamp قدیم و جدید دوحالت زیر را دارد و در حالت دوم مقدار تفاضل از رابطه دوم مطلبق شکل محاسبه خواهد شد.

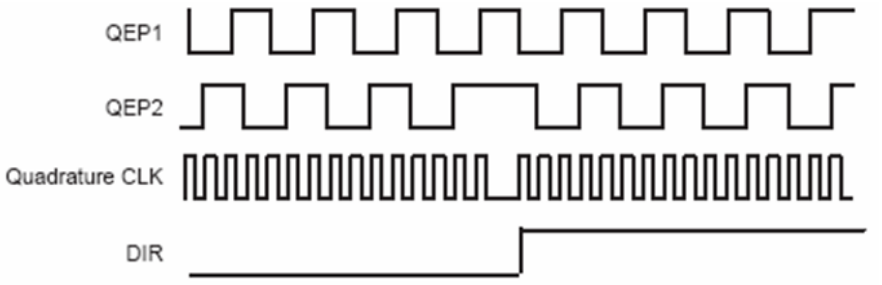


این ماژول در محیط نرم افزار MATLAB شبیه سازی نشده است و مدل موتور القایی جهت شبیه سازی سرعت پیش بینی شده است.

ماژول QEP_NO_INDEX_DRV

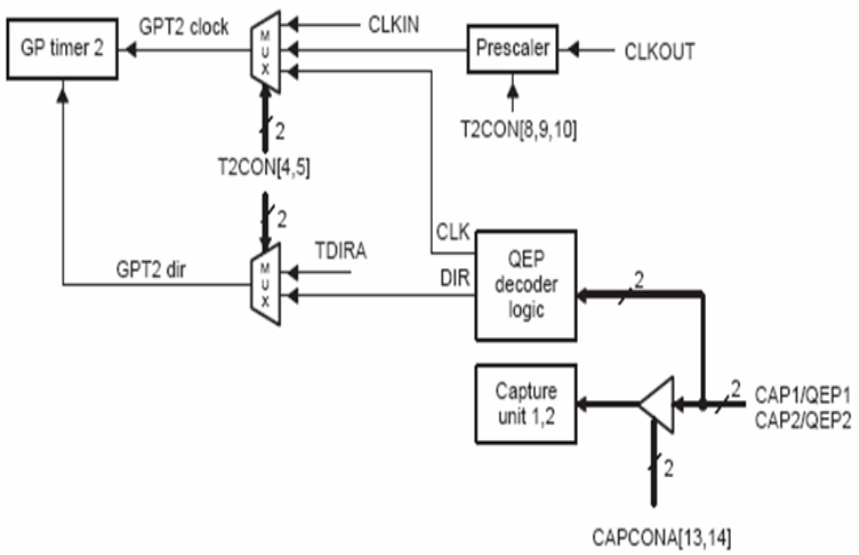
ماژول QEP_NO_INDEX_DRV با استفاده واحد CAPTURE و سنسور سرعت کار می کند. سنسور اندازه‌گیری زاویه بر روی محور موتور یا جعبه دنده در طراحی نصب خواهد شد. با چرخش موتور، سنسور پالسهائی با ۹۰ درجه اختلاف فاز ایجاد می‌نماید که با شمارش تعداد پالسها، زاویه و با در نظر گرفتن لبه بالا رونده و پائین رونده دو سیگنال می توان جهت حرکت موتور را مشخص نمود. سیگنال های تولید شده توسط سنسور به دو پین QEP1 و QEP2 از واحد CAPTURE مربوط به EVENT MANAGERA متصل خواهد شد. بدین‌صورت تایمر شماره ۲ وظیفه شمارش پالسهای تولید سنسور را برعهده دارد. جهت حرکت سنسور با واحد QEP decoder Logic مشخص می‌گردد. که جهت شمارش شمارنده (افزایشی و یا کاهشی) را مشخص می‌نماید. پردازنده در هر واحد نمونه پردازی با خواندن عدد درون شمارنده از زاویه محور موتور مطلع خواهد گردید. سنسور نصب گردیده بر روی محور موتور پایلوت در اصطلاح دارای ۲۵۰۰ خط (no_lines_encoder) یا ۱۰۰۰۰ پالسه می‌باشد. بدین ترتیب با عدد دهی رجیستر پریود پردازنده با عدد ۱۰۰۰۰، شمارنده با هر دور کامل موتور، از صفر تا ۱۰۰۰۰ افزایش و سپس دوباره صفر خواهد گردید.





برنامه		QEP_NO_INDEX_DRV
برنامه های فراخواننده		interrupt void MainISR
فایل های توابع مورد نیاز		f281xqep_no_index.c, f281xqep_no_index.h
پارامترها	ورودی	پالس به واحد input Captureشماره ۱ تا ۲
	خروجی	MechTheta: زاویه محور موتور DirectionQep: جهت حرکت موتور (۰ یا ۱)
	پارامتر QDP	Mech Scaler: متغیر پریونیت کننده زاویه وابسته تعداد پالس های سنسور OutPutMeahScaler: پارامتر وابسته به نسبت جعبه دنده

واحد شمارش پالس انکدر جهت اندازه‌گیری زاویه به صورت زیر است:



متغیری با نام Mech Scaler با مقدار ۰,۰۰۰۱ عدد دهی اولیه خواهد شد. در واقع وظیفه این عدد، پریونیت نمودن خروجی این واحد می‌باشد. بصورتیکه خروجی Mech Theta با یک دور کامل موتور بین صفر تا یک تغییر می‌نماید. با توجه عدم استفاده از جعبه دنده در گام طراحی اولیه، متغیر دوم که OutPutMeahScaler می‌باشد با MechScaler یکسان می‌باشد ولی در طراحی نهایی با عدد نسبت جعبه دنده بارگذاری خواهد شد.

$$\text{MechScaler} = \frac{1}{4 * \text{no_lines_encoder} * \text{gearbox ratio}}$$

$$\text{OutputMechScaler} = \frac{1}{4 * \text{no_lines_encoder}}$$

دراین برنامه ابتدا جهت حرکت موتور در متغیر DirectionQep ذخیره می‌گردد. سپس عدد شمارنده در متغیر RawTheta که عددی بین صفر تا ۱۰۰۰۰ می‌باشد ذخیره می‌گردد. نهایتاً این عدد به صورت پریونیت تبدیل

می‌شود که عددی بین صفر و یک است و در متغیر MechTheta ذخیره می‌گردد.

$$MechTheta = MechScaler \times RawTheta$$

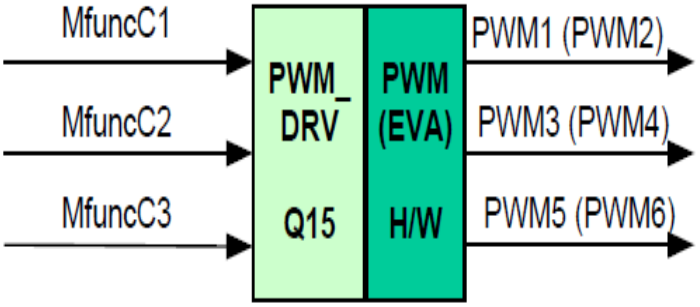
```
void F281X_EV1_QEP_NO_INDEX_Calc(QEP *p)
{

    int32 Tmp;
    Uint32 MechScalerTmp;

// Check the rotational direction
    p->DirectionQep = 0x4000&EvaRegs.GPTCONA.all;
    p->DirectionQep = p->DirectionQep>>14;

// Check the timer 2 counter for QEP
    p->RawTheta = EvaRegs.T2CNT;

// Compute the mechanical angle in Q15
    Tmp = __qmpy32by16(p->MechScaler,p->RawTheta,31);   // Q15 = Q30*Q0
    p->MechTheta = (int16)(Tmp);                        // Q15 -> Q15
    p->MechTheta &= 0x7FFF;                             // Wrap around 0x07FFF

// Compute the mechanical output angle in Q0
    p->OutputRawTheta = (int32)p->Counter*(int32)p->QepCountIndex + (int32)p->RawTheta;

// Re-scaling the mechanical scaler output angle (Q30)
    MechScalerTmp = p->OutputMechScaler<<16;

// Compute the mechanical output angle in Q15
    Tmp = __qmpy32(MechScalerTmp,p->OutputRawTheta,31);   // Q15 = Q30*Q0
    p->OutputTheta = (int16)(Tmp);                        // Q15 -> Q15
    p->OutputTheta &= 0x7FFF;                             // Wrap around 0x07FFF
 }
```

```
void F281X_EV1_QEP_NO_INDEX_Isr(QEP *p)
{

// Wrap around counter in both directions.
    if (p->DirectionQep==1) {
          if (p->Counter++ == p->PreScaler)
          p->Counter = 0;
    }
    else {
          if (p->Counter-- <= 0)
          p->Counter = p->PreScaler;
    }

}
```

این ماژول در محیط نرم افزار MATLAB شبیه سازی نشده است و مدل موتور القایی جهت شبیه سازی سرعت پیش بینی شده است.

ماژول FC_PWM_DRV

این ماژول اطلاعات زمان های Ta,Tb,Tc که توسط SVGEN_MF محاسبه شده است را به عنوان ورودی دریافت می‌دارد و توسط واحد مدیریت رویداد EVA مقدار مقایسه مورد نیاز را برای تولید ۶ پالس PWM 1-6 را محاسبه و اعمال می‌کند.



برنامه		FC_PWM_DRV
برنامه های فراخواننده		interrupt void MainISR
فایل های توابع مورد نیاز		f281xpwm.c, f281xpwm.h
پارامترها	ورودی	بردارهای Ta,Tb,Tc
	خروجی	تولید ۶ پالس PWM 1-6
	داخلی	PeriodMax بیشینه مقدار نصف پریود PWM برحسب کلاک سیستم

تشریح توابع: در فایل F281XPWM.C جهت مقدار دهی ریجیستر مولد PWM با نام CMPR1-3 دو تابع F281X_EV1_PWM_Init و F281X_EV1_PWM_Update تعریف شده اند.

همان طور در خط سوم تابع مقدار دهی اولیه مشاهده می‌شود ریجیستر DBTCONA با مقدار DBTCON_INIT_STATE که در F281XPWM.H

تعریف شده است و به معنای روشن بودن dead band برای مولد های pwm1-6، به میزان DBTPS_X8 یعنی یک هشتم کلاک CPU است.

ریجیستر T1CON نیز در تابع مقدار دهی اولیه با مقدار PWM_INIT_STATE در F281XPWM.H بارگذاری شده است،که به معنای تنظیم تایمر در حالت Continuous up-/down است در تصویر زیر شمای کلی از روش شمارش در این حالت نمایش داده شده است. تایمر ۱ فعال و prescaler آن با مقدار ۱ بارگذاری و در حالت FREE_RUN یعنی شمارش بدون توقف تنظیم شده است.

ریجیستر ACTRA نیز با مقدار ACTR_INIT_STATE که در محل که آن هم در F281XPWM.H تعریف شده بارگذاری شده است. این ریجیستر مشخص می‌کند که هر پین در هنگام برابری با مقدار COM خود چه رفتاری داشته باشد و به این صورت بارگذاری می‌شود که پین های خروجی فرد در هنگام برابری Active high و پین های زوج Active low گردند.

ریجیستر COMCONA با مقدار A200 بارگذاری شده است.

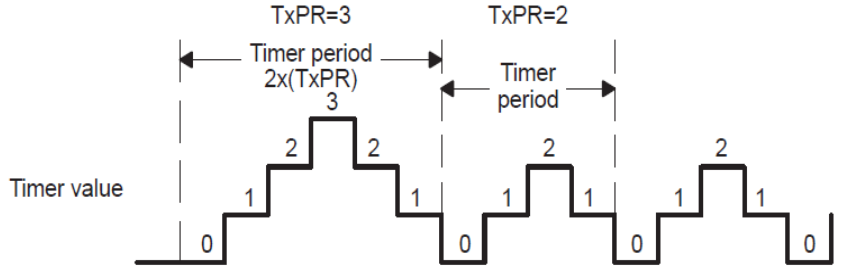
CENABLE	CLD1	CLD0	SVENABLE	ACTRLD1	ACTRLD0	FCMPOE	PDPINTA Status
1	0	1	0	0	0	1	0

این به این معنی است که روش کلیدزنی Space vector PWM سخت افزاری غیر فعال است. سخت افزار آماده تولید PWM می‌باشد و در عمل به صورت نرم‌افزاری ماژول مولد Space vector یک Space vector PWM را پیاده سازی می‌شود. ریجیستر فرمان می دهد که عمل مقایسه فعال باشد و

ریجستر مقایسه CMPR در هنگام برابری T1CNT = 0 به سرعت بارگذاری می‌شود و PWM1/2/3/4/5/6 متناظر با منطق مقایسه بار گذاری می‌شود.

```
// Initialize PWM module
 pwm1.PeriodMax = SYSTEM_FREQUENCY*1000000*T/2;  // Perscaler X1 (T1), ISR period = T x 1
        pwm1.init(&pwm1);

void F281X_EV1_PWM_Init(PWMGEN *p)
{
      EvaRegs.T1PR = p->PeriodMax;              // Init Timer 1 period Register
      EvaRegs.T1CON.all = PWM_INIT_STATE;       // Symmetrical Operation
      EvaRegs.DBTCONA.all = DBTCON_INIT_STATE;  // Init DBTCONA Register
      EvaRegs.ACTRA.all = ACTR_INIT_STATE;      // Init ACTRA Register

      EvaRegs.COMCONA.all = 0xA200;             // Init COMCONA Register

      EvaRegs.CMPR1 = p->PeriodMax;             // Init CMPR1 Register
      EvaRegs.CMPR2 = p->PeriodMax;             // Init CMPR2 Register
      EvaRegs.CMPR3 = p->PeriodMax;             // Init CMPR3 Register
      EALLOW;                    // Enable EALLOW
      GpioMuxRegs.GPAMUX.all |= 0x003F;   // Setting PWM1-6 as primary output pins
      EDIS;                      // Disable EALLOW
}
```

GP Timer Continuous Up-/Down-Counting Mode (TxPR = 3 or 2)



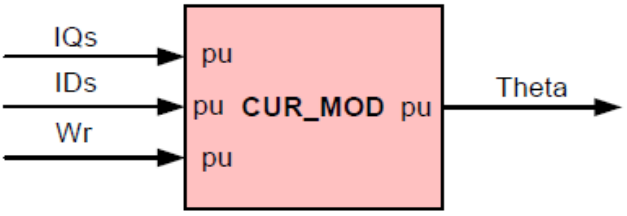
```
void F281X_EV1_PWM_Update(PWMGEN *p)
{
                    int16 MPeriod;
      int32 Tmp;

// Compute the timer period (Q0) from the period modulation input (Q15)
      Tmp = (int32)p->PeriodMax*(int32)p->MfuncPeriod;          // Q15 = Q0*Q15
      MPeriod = (int16)(Tmp>>16) + (int16)(p->PeriodMax>>1);    // Q0 = (Q15->Q0)/2 + (Q0/2)
      EvaRegs.T1PR = MPeriod;

// Compute the compare 1 (Q0) from the PWM 1&2 duty cycle ratio (Q15)
      Tmp = (int32)MPeriod*(int32)p->MfuncC1;                   // Q15 = Q0*Q15
      EvaRegs.CMPR1 = (int16)(Tmp>>16) + (int16)(MPeriod>>1);   // Q0 = (Q15->Q0)/2 + (Q0/2)

// Compute the compare 2 (Q0) from the PWM 3&4 duty cycle ratio (Q15)
      Tmp = (int32)MPeriod*(int32)p->MfuncC2;                   // Q15 = Q0*Q15
      EvaRegs.CMPR2 = (int16)(Tmp>>16) + (int16)(MPeriod>>1);   // Q0 = (Q15->Q0)/2 + (Q0/2)

// Compute the compare 3 (Q0) from the PWM 5&6 duty cycle ratio (Q15)
      Tmp = (int32)MPeriod*(int32)p->MfuncC3;                   // Q15 = Q0*Q15
      EvaRegs.CMPR3 = (int16)(Tmp>>16) + (int16)(MPeriod>>1);   // Q0 = (Q15->Q0)/2 + (Q0/2)

}
```

نمونه کد مقداردهی مقایسه‌گرها تولید PWM

```
// ---------------------------------------------------------------------------
//   Connect inputs of the PWM_DRV module and call the PWM signal generation
//   update function.
// ---------------------------------------------------------------------------
   pwm1.MfuncC1 = (int16)_IQtoIQ15(svgen_mf1.Ta); // MfuncC1 is in Q15
   pwm1.MfuncC2 = (int16)_IQtoIQ15(svgen_mf1.Tb); // MfuncC2 is in Q15
   pwm1.MfuncC3 = (int16)_IQtoIQ15(svgen_mf1.Tc); // MfuncC3 is in Q15
   pwm1.update(&pwm1);

// ---------------------------------------------------------------------------
```

ماژول CUR_MOD

الگوریتم foc به موقعیت شار در فضای موتور نیاز دارد این ماژول با داشتن دو ورودی خروجی پارک جریان موتور و مقدار سرعت شفت، موقعیت شار را تعیین می کند.



برنامه	CUR_MOD	
برنامه های فراخواننده	interrupt void MainISR	
فایل های توابع مورد نیاز	CUR_MOD.c, CUR_MOD.h	
پارامترها	ورودی	دو ورودی خروجی پارک جریان موتور مقدار سرعت شفت
	خروجی	Theta موقعیت شار
	پارامتر ماژول	Kr: ثابت ای جهت محاسبه جریان مغناطیس کنندگی Kt: ثابت ای جهت محاسبه لغزش K: ثابت ای جهت محاسبه زاویه شار در روتور

IMDs: متغیر سنکرون سازی محور dxجریان مغناطیس کنندگی Rr: مقدار مقاومت اهمی روتور Lr: مقدار اندوکتانس موتور (جمع اندوتانس ها) Fb: فرکانس نامی موتور Ts: فرکانس نمونه برداری		

این تابع به مقادیر زیر برای محاسبه شار نیاز دارد که در CURMOD_CONST قرار گرفته است.

```
// Instance a current model constant object
CURMOD_CONST cm1_const = CURMOD_CONST_DEFAULTS;
```

```
typedef struct  { float32  Rr;     // Input: Rotor resistance (ohm)
                  float32  Lr;     // Input: Rotor inductance (H)
                  float32  fb;     // Input: Base electrical frequency (Hz)
                  float32  Ts;     // Input: Sampling period (sec)
                  float32  Kr;     // Output: constant using in magnetizing current calculation
                  float32  Kt;     // Output: constant using in slip calculation
                  float32  K;      // Output: constant using in rotor flux angle calculation
                  void   (*calc)(); // Pointer to calculation function
                } CURMOD_CONST;

typedef CURMOD_CONST *CURMOD_CONST_handle;
```

مقدایر مقاومت اهمی روتور، اندوکتانس موتور، فرکانس نامی موتور، فرکانس نمونه برداری در parameter.h آمده است و ثابت های مورد نیاز از روی آنها محاسبه خواهد شد.

مشخصات موتور پیش فرض تست های اولیه به صورت زیر است.

V=400V, P=3kW, p=2(4pole), Rs=1.87Ω ,Rr=1.86Ω, Lls=Llr=7.54mH, Lm=210mH ,J=0.01kg-m2.

```
// Initialize the CUR_MOD constant module
        cm1_const.Rr = RR;
        cm1_const.Lr = LR;
        cm1_const.fb = BASE_FREQ;
        cm1_const.Ts = T;
        cm1_const.calc(&cm1_const);

// Initialize the CUR_MOD module
        cm1.Kr = _IQ(cm1_const.Kr);
        cm1.Kt = _IQ(cm1_const.Kt);
        cm1.K = _IQ(cm1_const.K);
```

در تابع cur_mod_const_calc که در فایل CUR_CONST.C موجود است روش محاسبه این ثابت ها آمده است.

```
void cur_mod_const_calc(CURMOD_CONST *v)
{
        float32 Tr;   // Rotor time constant (sec)

        Tr = v->Lr/v->Rr;

       v->Kr = v->Ts/Tr;
        v->Kt = 1/(Tr*2*PI*v->fb);
        v->K = v->Ts*v->fb;
}
```

به منظور تخمین موقعیت شار روتور از مدل جریانی موتور استفاده می‌شود . در این مدل با استفاده از زاویه الکتریکی روتور و جریانهای استاتور، i_{sQ} و i_{sD} موقعیت دقیق شار روتور محاسبه می‌گردد. با محاسبه معادلات ولتاژ روتور در مختصات شار روتور، سرعت زاویه ای شا ر روتور به صورت زیر محاسبه می‌گردد:

$$w_{mr} = w_r + \frac{i_{sQ}}{T_r |\bar{i}_{mr}|}$$

که w_rسرعت زاویه ای روتور T_r ثابت زمانی مدار روتور $T_R = \frac{L_R}{R_R}$ ت. $\frac{i_{sQ}}{T_r |\bar{i}_{mr}|}$ بیانگر فرکانس زاویه ای لغزش میدان روتور می‌باشد.

اندوکتانس موتور جمع دو اندوکتانس مغناطیس کنندگی و اندوکتانس نشتی روتور است.

$$L_R = L_H + L_{\sigma R}$$

|Imr| نیز جریان مغناطیس کنندگی روتور در مختصات ویژه شار روتور است و مطابق روابط زیر محاسبه می‌گردد:

$$T_r \frac{d|\bar{i}_{mr}|}{d_t} + |\bar{i}_{mr}| = i_{sD} \Rightarrow |\bar{i}_{mr}| = \frac{1}{1 + T_r P} i_{sD}$$

P عملگر مشتق را نشان می دهد. اگر|Imr| ثابت باشد نتیجه می‌شود |imr| = isD | است و دامنه فازور فضایی شار پیوندی روتور بوسیله مؤلفه طولی جریان استاتور کنترل می‌گردد. در زیر سرعت نامی نیز گشتاور الکتریکی متناسب با مؤلفه عرضی جریان استاتور خواهد شد.

با توجه به روابط مدل گسسته محاسبه جریان مغناطیس کنندگی روتور به صورت زیر خواهد بود.

$$i_{mR_{k+1}} = i_{mR_k} + \frac{T}{T_R}(i_{dS_k} - i_{mR_k})$$

اگر سرعت شار موتورFs، ωb سرعت نامی شار الکتریکی، n سرعت زاویه‌ای روتور، θ موقعیت شار روتور باشد داریم.

$$fs = \frac{1}{\omega_b}\frac{d\theta}{dt} = n + \frac{i_{qS}}{T_R i_{mR}\omega_b}$$

مدل گسسته این را بطه به صورت زیر است:

$$f_{S_{k+1}} = n_{k+1} + \frac{1}{T_R \omega \omega_b}\frac{i_{qS_k}}{i_{mR_{k+1}}}$$

مقادیر ثابت یادشده به صورت زیر به عنوان بخشی از دو رابطه گسسته بالا و به صورت زیر است.

$$K_t = \frac{T}{T_R}$$

$$K_R = \frac{1}{T_R \omega_b}$$

θ موقعیت شار روتور بر اساس سرعت شار موتورFs، به صورت زیر محاسبه می شود.

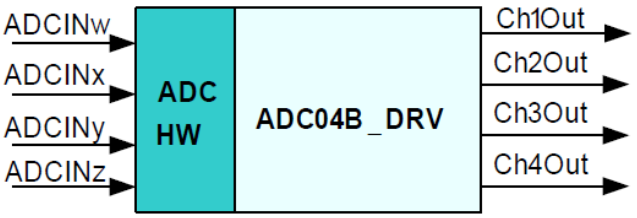
$$\theta = \theta_{k-1} + Kf_{s_k} \qquad K = Tf_b$$

```
void cur_mod_calc(CURMOD *v)
{

        _iq Wslip, We;

        v->IMDs +=  _IQmpy(v->Kr,(v->IDs - v->IMDs));
        Wslip = _IQdiv(_IQmpy(v->Kt,v->IQs),v->IMDs);
        We = v->Wr + Wslip;
        v->Theta +=  _IQmpy(v->K,We);

    if (v->Theta > _IQ(1))
       v->Theta -=  _IQ(1);
    else if (v->Theta < _IQ(0))
       v->Theta += _IQ(1);

}
```

ماژول ADC04b_DRV

مبدل آنالوگ به دیجیتال پردازنده ۲۸۱۲ دارای شانزده کانال می‌باشد. هدف این است که از این کانال ها جهت اندازه‌گیری ولتاژها و جریان ها استفاده شود. تمامی سیگنال ها از نوع bipolar می‌باشند بدین معنی که در صورتی که اندازه سیگنال متصل به پایه مبدل ۱/۵ ولت باشد، سیگنال مربوطه دارای مقدار صفر می‌باشد و درصورتیکه اندازه سیگنال ۳ ولت باشد، سیگنال دارای بیشترین مقدار وهنگامی که مقدار صفر ولت باشد، سیگنال دارای منفی ترین مقدار می‌باشد.

این ماژول ۴ ورودی آنالوگ را به ۴ خروجی دیجیتال به صورت سیگنال bipolar با گین و آفست قابل تنظیم تبدیل می کند. این تبدیل با رخداد underflow تایمر ۱ رخ می دهد.



برنامه		ADC04b_DRV
برنامه های فراخواننده		interrupt void MainISR
فایل های توابع مورد نیاز		f281xadc04b.c, f281xadc04b.h
پارامترها	ورودی	ADCINx وADCINy وADCINz و ADCINwکه بر اساس پارامتر Ch_sel محل آن مشخص می شوند.
	خروجی	Ch1Out تا Ch4Out: ۴ خروجی دیجیتال
	پارامتر ماژول	Ch1Gain تا Ch4Gain: ۴ پارامتر تنظیم گین Ch1Offset تا Ch4Offset: ۴ پارامتر تنظیم آفست Ch_sel: یک عد هگزا دسیمال اشاره به پایه های ADC

قبل از بررسی نحوه اندازه‌گیری سیگنال‌ها، به استراکچر تعریف شده در فایل f281xadc04b.h می پردازیم. این استراکچر برای هر شانزده کانال متغیر گین و آفست تعریف نموده است، مقادیر این دو پارامتر برای هر شانزده کانال

به ترتیب یک و صفر اختیار خواهد شد، با توجه به عدم استفاده از گین و آفست این مقادیر در برنامه تغییر نمی نمایند و در واقع تأثیری در مقادیر خوانده شده نخواهند داشت.

در فایل F281XADC04B.C دو تابع F281X_adc04b_drv_read و F281X_adc04b_drv_init قرار دارد.

تنظیمات گین و آفست و کانال های فعال ADC و دستور فعال شدن مبدل با عمل under flow تایمر یک همگی در F281X_adc04b_drv_init صورت می پذیرد. اگر چه از وقفه مبدل آنالوگ به دیجیتال استفاده نخواهد شد اما وقفه جهت بررسی اتمام عملکرد مبدل فعال می‌گردد.

```
void F281X_adc04b_drv_init(ADCVALSB *p)
{
  DELAY_US(ADC_usDELAY);

  AdcRegs.ADCTRL1.all = ADC_RESET_FLAG;          // Reset the ADC Module
       asm(" NOP ");
       asm(" NOP ");

  AdcRegs.ADCTRL3.bit.ADCBGRFDN = 0x3;           // Power up bandgap/reference circuitry
       DELAY_US(ADC_usDELAY);                                  // Delay before powering up rest of ADC

  AdcRegs.ADCTRL3.bit.ADCPWDN = 1;                    // Power up rest of ADC
  AdcRegs.ADCTRL3.bit.ADCCLKPS = 6;              // Set up ADCTRL3 register
       DELAY_US(ADC_usDELAY);

  AdcRegs.ADCTRL1.all = ADCTRL1_INIT_STATE_BIPOLAR;      // Set up ADCTRL1 register
  AdcRegs.ADCTRL2.all = ADCTRL2_INIT_STATE_BIPOLAR;      // Set up ADCTRL2 register
       AdcRegs.ADCMAXCONV.bit.MAX_CONV1 = 3;          // Specify four conversions
  AdcRegs.ADCCHSELSEQ1.all = p->ChSelect;          // Configure channel selection

       EvaRegs.GPTCONA.bit.T1TOADC = 1;                        // Set up EV Trigger with Timer1 UF
}
```

در تابع F281X_adc04b_drv_read ابتدا بیت وقفه مبدل جهت اطمینان از پایان عمل تبدیل خوانده می‌شود و سپس بیت مربوطه Clear می‌گردد. سپس نتایج اعداد ذخیره شده در رجیستر ADCRESULT0-15 خوانده می‌شود. در پایان جهت تبدیل بعدی مبدل باز نشانی می‌گردد.

اعداد خوانده شده از رجیستر ADCRESULT ابتدا با عدد ۸۰۰۰، Exclusive OR می‌گردد تا ۱/۵ ولت به عنوان سطح صفر سیگنال ورودی در نظر گرفته شود. سپس با توجه به یک بودن گین و صفر بودن آفست، این عدد در رجیستر CH1OUT تا CH16OUT ذخیره می‌گردد.

```
void F281X_adc04b_drv_read(ADCVALSB *p)
{
    int16 DatQ15;
    int32 Tmp;

    // Wait until ADC conversion is completed
    while (AdcRegs.ADCST.bit.SEQ1_BSY == 1)
    {};
    DatQ15 = AdcRegs.ADCRESULT0^0x8000;          // Convert raw result to Q15 (bipolar signal)
    Tmp = (int32)p->Ch1Gain*(int32)DatQ15;  // Tmp = gain*dat => Q28 = Q13*Q15
    p->Ch1Out = (int16)(Tmp>>13);          // Convert Q28 to Q15
    p->Ch1Out += p->Ch1Offset;             // Add offset

    DatQ15 = AdcRegs.ADCRESULT1^0x8000;          // Convert raw result to Q15 (bipolar signal)
    Tmp = (int32)p->Ch2Gain*(int32)DatQ15;  // Tmp = gain*dat => Q28 = Q13*Q15
    p->Ch2Out = (int16)(Tmp>>13);          // Convert Q28 to Q15
    p->Ch2Out += p->Ch2Offset;             // Add offset

    DatQ15 = AdcRegs.ADCRESULT2^0x8000;          // Convert raw result to Q15 (bipolar signal)
    Tmp = (int32)p->Ch3Gain*(int32)DatQ15;  // Tmp = gain*dat => Q28 = Q13*Q15
    p->Ch3Out = (int16)(Tmp>>13);          // Convert Q28 to Q15
    p->Ch3Out += p->Ch3Offset;             // Add offset

    DatQ15 = AdcRegs.ADCRESULT3^0x8000;          // Convert raw result to Q15 (bipolar signal)
    Tmp = (int32)p->Ch4Gain*(int32)DatQ15;  // Tmp = gain*dat => Q28 = Q13*Q15
    p->Ch4Out = (int16)(Tmp>>13);          // Convert Q28 to Q15
    p->Ch4Out += p->Ch4Offset;             // Add offset
    AdcRegs.ADCTRL2.all |= 0x4040;         // Reset the sequence
}
```

بر اساس ماژول ADC04b_DRV ماژولی دقیقاً با همان خصوصیات برای محاسبه ولتاژ لینک DC و جریان‌های ۲ فاز و جریان تفاضلی لینک DC می خواهیم بنویسیم. در حقیقت ماژول ADC04b_DRV در درون ماژولی جدید قرار خواهد گرفت این ماژول را F281XILEG_VI می‌نامیم.

ماژول ReadADCAverage

با توجه به حساس بودن برخی از این پارامترها، بطور مثال ولتاژ خازن یا جریان کلیدها، در صورتی که این پارامترها از حدی تجاوز نماید تابع

ShutDownCheck برنامه را متوقف خواهدنمود. برنامه کد خطای علت توقف را ارائه خواهد کرد. برنامه نهائی که جهت خواندن سیگنال های ورودی مبدل ها مورد استفاده می‌گردد ماژول ReadADCAverage می‌باشد. با استفاده از تابع F281X_adc04b_drv_read ۴ کانال و ۵ متغیر سه بار دیگر خوانده و متوسط گیری می‌گردد. بدین ترتیب فقط کانال های جریان ها و ولتاژ خازن ۴ بار خوانده و متوسط گیری می‌شوند تا اثر نویزهای احتمالی را کاهش دهند.

برنامه		void ReadADCAverage()
برنامه های فراخواننده		MainISR
پارامترها	خروجی	جریان فاز ها، ولتاژ خازن پریونیت
توضیحات برنامه		قرائت ADC و متوسط گیری از جریان و ولتاژ ها قرائت شده

```
void ReadADCAverage(void)
{
    int i;
    int32 TmpImeasA;
    int32 TmpImeasB;
    int32 TmpImeasC;
    int32 TmpVdcMeas;
    int32 TmpIdcMeas;

        for (i=0;i<= NUM_SAMPLE_ADC ;i++)
        {
        AdcRegs.ADCTRL2.bit.SOC_SEQ1=1; // software tigge to start convertion
        F281X_adc04b_drv_read();          //ADC conversion is completed
        TmpImeasA+=p->ImeasA;
        TmpImeasB+=p->ImeasB;
        TmpImeasC+=p->ImeasC;
        TmpVdcMeas+=p->VdcMeas;
        TmpIdcMeas+=p->IdcMeas;
        }
   p->ImeasA=TmpImeasA>>2;
   p->ImeasB=TmpImeasB>>2;
   p->ImeasC=TmpImeasC>>2;
   p->VdcMeas=TmpVdcMeas>>2;
   p->IdcMeas=TmpIdcMeas>>2;
}
```

به علت ایده آل نبودن المان های مورد استفاده در مدار، کانال های ورودی مبدل آنالوگ به دیجیتال عموماً دارای آفست می‌باشند. قبل از شروع برنامه اصلی، میزان آفست هر کانال هشت بار اندازه‌گیری می‌شود تا در عملکرد های بعدی مبدل، مقادیر آفست اندازه‌گیری شده استفاده گردد.

هر یک از کانالهای مبدل آنالوگ به دیجتال دارای گین و آفست متفاوتی می‌باشند که اضافه نمودن گین و آفست این کانال ها، در تابع SampellingADC در نظر گرفته خواهد شد.

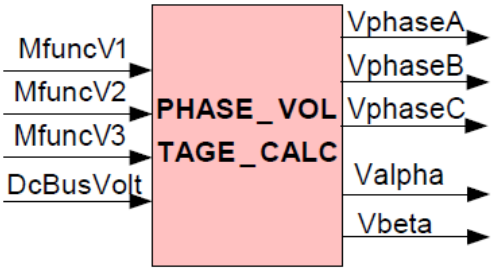
دراین تابع ابتدا از اعداد خام ذخیره شده در صورت نیاز آفست مربوطه کسر می‌گردد. این اعداد که در فرمت Q15 می‌باشند سپس به فرمت Q24 تبدیل می‌گردند و سپس گین مربوطه به آن ها اعمال می‌گردد.

برنامه		void SampellingADC2
برنامه های فراخواننده		SampleADCAverage
پارامترها	ورودی	اطلاعات خام مبدل آنالوگ به دیجیتال
	خروجی	جریان و ولتاژ
توضیحات برنامه		اعمال گین و آفست مبدل آنالوگ به دیجیتال و حفاظت جریان و ولتاژ

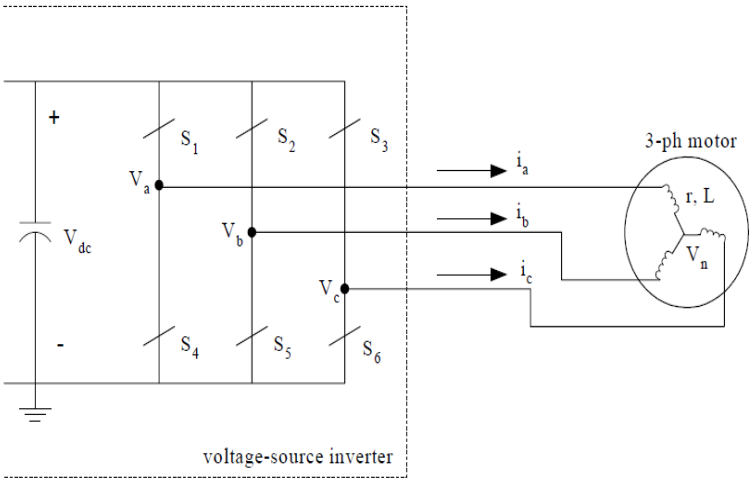
ماژول های شبیه سازی در Matlab

ماژول volt_calc

جهت شبیه سازی قسمت تایمر های سازنده PWM و یک اینورتر منبع ولتاژ به صورت یکپارچه در محیط نرم افزار MATLAB، ماژول زیر با ورودی اطلاعات زمان های Ta,Tb,Tc ماژول SVGEN_MF و ولتاژ لینک DC و خروجی ولتاژ های سه فاز A,B,C و خروجی ولتاژ دو فاز در دستگاه DQ به صورت زیر تعریف شده است.جهت تبدیل سه فاز به دو فاز از تبدیل کلارک در داخل این ماژول استفاده شده است.



برنامه		volt_calc
برنامه های فراخواننده		فقط جهت شبیه سازی در MATLAB
فایل های توابع مورد نیاز		volt_calc.c, volt_calc.h
پارامترها	ورودی	زمان های Ta,Tb,Tc ماژول SVGEN_MF ولتاژ لینک DC
	خروجی	ولتاژ های سه فاز A,B,C ولتاژ دو فاز در دستگاه DQ



طبق قانون KCL جمع جریان ها سه فاز در یک موتور القایی و ولتاژ نقطه نول که برآیند مجموع ولتاژ های سه فاز است برابر صفر است. لذا می توان گفت:

$$V_{an} = V_a - \frac{1}{3}(V_a + V_b + V_c) = \frac{2}{3}V_a - \frac{1}{3}V_b - \frac{1}{3}V_c$$

$$V_{bn} = V_b - \frac{1}{3}(V_a + V_b + V_c) = \frac{2}{3}V_b - \frac{1}{3}V_a - \frac{1}{3}V_c$$

$$V_{cn} = V_c - \frac{1}{3}(V_a + V_b + V_c) = \frac{2}{3}V_c - \frac{1}{3}V_a - \frac{1}{3}V_b$$

از آنجا که طبق زمان های Ta,Tb,Tc در هر ساق مدت زمانی خاص هر سوئیچ روشن است می توان ولتاژ هر فاز اینورتر را برابر با مقدار زیر دانست.

$$V_{an} = V_{dc}\left(\frac{2}{3}S_1 - \frac{1}{3}S_2 - \frac{1}{3}S_3\right)$$
$$V_{bn} = V_{dc}\left(\frac{2}{3}S_2 - \frac{1}{3}S_1 - \frac{1}{3}S_3\right)$$
$$V_{cn} = V_{dc}\left(\frac{2}{3}S_3 - \frac{1}{3}S_1 - \frac{1}{3}S_2\right)$$

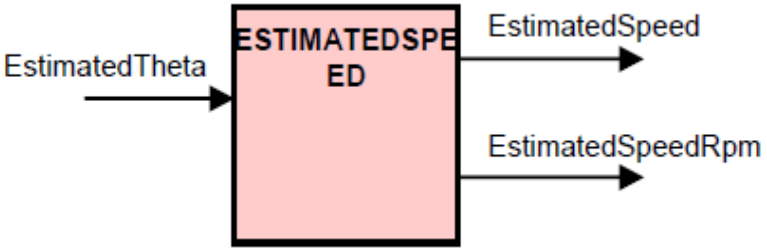
ماژول FC_PWM_DRV مولد سخت افزاری PWM است و جهت شبیه سازی در محیط نرم افزار MATLAB این ماژول، ماژولی به نام volt_calc برنامه نویسی شده و در تابعی با نام volt_calc.M قرار گرفته است. جهت مشاهده این فایل می توانید به CD پیوست مراجعه فرمایید.

```
function  [VphaseA,VphaseB,VphaseC,Valpha,Vbeta] = volt_calc(MfuncV1,MfuncV2,MfuncV3,DcBusVolt)

% scale the incomming Modulation functions with the DC bus voltage value
     MfuncV1 =(DcBusVolt*MfuncV1);
     MfuncV2 = (DcBusVolt*MfuncV2);
     MfuncV3 = (DcBusVolt*MfuncV3);

% calculate the 3 Phase voltages
     % Phase A
     VphaseA = (TWO_THIRD*MfuncV1);
     VphaseA =VphaseA- (ONE_THIRD*MfuncV2);
     VphaseA =VphaseA- (ONE_THIRD*MfuncV3);

     % Phase B
     VphaseB = (TWO_THIRD*MfuncV2);
     VphaseB =VphaseB-(ONE_THIRD*MfuncV1);
     VphaseB =VphaseB- (ONE_THIRD*MfuncV3);

     % Phase C
     VphaseC =(TWO_THIRD*MfuncV3);
     VphaseC =VphaseC-(ONE_THIRD*MfuncV1);
     VphaseC =VphaseC- (ONE_THIRD*MfuncV2);

% Voltage transformation (a,b,c)  ->  (Alpha,Beta)
     % Alpha-axis
     Valpha = VphaseA;

     % Beta-axis
     Vbeta = ((VphaseA + (2*VphaseB))*INV_SQRT3);
```

ماژول ESTIMATEDSPEED

این ماژول جهت اندازه‌گیری سرعت از محاسبه مشتق زاویه اندازه‌گیری شده نسبت به زمان استفاده می‌شود.

مقدار زاویه خروجی ماژول QEP_NO_INDEX_DRV است.



برنامه		ESTIMATEDSPEED
برنامه های فراخواننده		interrupt void MainISR
فایل های توابع مورد نیاز		speed_est.c, speed_est.h
پارامترها	ورودی	EstimatedTheta: خروجی ماژول QEP_NO_INDEX_DRV زاویه محور موتور
	خروجی	EstimatedSpeed: خروجی سرعت (پریونیت)
	پارامترهای فیلتر دیجیتال	K1 = 1/(fb*T) K2 = 1/(1+T*2*pi*fc) K3 = T*2*pi*fc/(1+T*2*pi*fc)
توضیحات برنامه		محاسبه سرعت از زاویه بصورت پریونیت

سرعت با استفاده از رابطه $W(K) = K_1(\theta(K) - \theta(K-1))$ محاسبه می‌شود. که در آن K_1 عکس زمان نمونه برداری می‌باشد. جهت محاسبه K_1 از روابط زیر استفاده می نمائیم.

$$W(K) = \frac{\theta(K) - \theta(K-1)}{\Delta T}$$

$$\frac{W(K)}{W_{base}} = \frac{\theta base}{Wbase} \qquad \frac{\frac{\theta(k)}{\theta base} - \frac{\theta(k-1)}{\theta\ base}}{\Delta T}$$

$$W_{pu}(k) = \frac{\theta base}{Wbase} \quad \frac{\theta_{pu}\ (k) - \theta_{pu}\ (k-1)}{\Delta T}$$

$$W_{pu}(k) = \frac{2\pi}{\left(\frac{2\pi f_{base}}{Pole/2}\right)} \quad \frac{\theta_{pu}\ (k) - \theta_{pu}\ (k-1)}{\Delta T}$$

بنابراین K_1 عبارتست از:

$$k_1 = \frac{2\pi}{\left(\frac{2\pi f_{base}}{Pole/2}\right)\Delta T} = \frac{1}{\left(\frac{f_{base}}{Pole/2}\right)\Delta T}$$

همچنین در تابع محاسبه سرعت، یک فیلتر Low pass دیجیتال تعبیه شده است، که وظیفه آن حذف نویزهای اندازه‌گیری سرعت می‌باشد. با در نظر گرفتن فرکانس قطع f_c برای فیلتر خواهیم داشت.

$$\frac{d\hat{w}}{dt} = \frac{1}{\tau_c}\ (w - \hat{w})$$

$$\tau_c = \frac{1}{2\lambda\Pi f_c}$$

که در آن w سرعت و $\hat{w}$سرعت با اعمال فیلتر می‌باشد. فیلتر فوق بصورت دیجیتال عبارت خواهد بود از:

$$\widehat{w_e}\ (k) = k_2\ \ \widehat{w_e}\ (k-1) + k_3\ \ w_e\ (k)$$

$$K_2 = \frac{\tau_c}{\tau_c + T}$$

$$K_3 = \frac{T}{\tau_c + T}$$

در ماژول EstimatedSpeed ابتدا اختلاف زاویه در متغیر Temp1 ذخیره می‌گردد. از آنجائیکه Theta بین صفر و یک تغییر می‌نماید، در صورتی که مقدار آن از یک به صفر و یا از صفر به یک تغییر نماید، محاسبات دچار خطا می‌گردد به همین دلیل متغیر Temp1 اگر از حدی بیشتر یا کمتر گردد، برنامه با اضافه و یا کم نمودن عدد یک به متغیر Temp1 این مسئله را جبران می‌نماید. سپس مشتق زاویه مجدداً در Temp1 ذخیره می‌گردد و پس از فیلتر شدن در متغیر Estimated Speed ذخیره می‌گردد و آن را برمی گرداند.

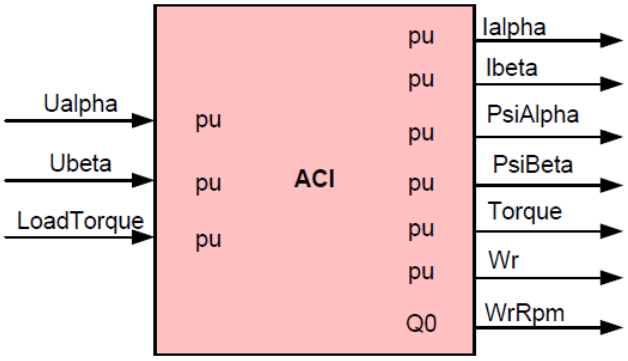
```
void speed_est_calc(SPEED_ESTIMATION *v)
{
  _iq Temp1;
// Synchronous speed computation

  if ((v->EstimatedTheta < DIFF_MAX_LIMIT)&(v->EstimatedTheta > DIFF_MIN_LIMIT))
                    // Q21 = Q21*(GLOBAL_Q-GLOBAL_Q)
                    Temp1 = _IQmpy(v->K1,(v->EstimatedTheta - v->OldEstimatedTheta));
  else Temp1 = _IQtoIQ21(v->EstimatedSpeed);

// Low-pass filter
  // Q21 = GLOBAL_Q*Q21 + GLOBAL_Q*Q21
  Temp1 = _IQmpy(v->K2,_IQtoIQ21(v->EstimatedSpeed))+_IQmpy(v->K3,Temp1);

  if (Temp1>_IQ21(1))
    v->EstimatedSpeed = _IQ(1);
  else if (Temp1<_IQ21(-1))
    v->EstimatedSpeed = _IQ(-1);
  else
    v->EstimatedSpeed = _IQ21toIQ(Temp1);

// Update the electrical angle
   v->OldEstimatedTheta = v->EstimatedTheta;

// Change motor speed from pu value to rpm value (GLOBAL_Q -> Q0)
// Q0 = Q0*GLOBAL_Q => _IQXmpy(), X = GLOBAL_Q
  v->EstimatedSpeedRpm = _IQmpy(v->BaseRpm,v->EstimatedSpeed);
}
```

این ماژول در محیط نرم افزار MATLAB شبیه سازی نشده است و مدل موتور القایی جهت شبیه سازی سرعت پیش بینی شده است.

ماژول aci

این ماژول جهت شبیه سازی رفتار موتور القایی در محیط نرم افزار MATLAB تعریف شده است و در DSP پیاده سازی نخواهد شد. ورودی این ماژول ولتاژ های خروجی اینورتر در دستگاه DQ و مقدار گشتاور بار است.

خروجی این مدل جریان دو فاز در دستگاه DQ، گشتاور الکترومغناطیسی، سرعت زاویه ای الکتریکی، RPM موتور و مقدار شار روتور در دستگاه DQ است.



برنامه		aci
برنامه های فراخوانده		جهت شبیه سازی در MATLAB
پارامترها	ورودی	ولتاژ های دوفاز خروجی اینورتر در دستگاه DQ مقدار گشتاور بار
	خروجی	جریان دو فاز در دستگاه DQ گشتاور الکترومغناطیسی سرعت زاویه ای الکتریکی RPM موتور مقدار شار روتور در دستگاه DQ

جهت مدل سازی موتور به پارامترهای زیر نیاز است:

مقاومت استاتور Rs، مقاومت روتور Rr، اندوکتانس استاتور Ls، اندوکتانس روتور Lr، تعداد قطب های موتور P، ضریب دمپینگB، ممان اینرسی ناشی از وزن روتورJ، ولتاژ نامی Vb، جریان نامی Ib، گشتاور نامی Tb، سرعت نامی Wb، شار نشتی نامی Lb مورد نیاز است.

معادله حالت ماشین القایی در دستگاه DQبه صورت زیر است.

$$\sigma = 1 - \frac{L_m^2}{L_s L_r},\ \gamma = \frac{\left(L_m^2 R_r + L_r^2 R_s\right)}{\sigma L_s L_r^2},\ \alpha = \frac{1}{\tau_r} = \frac{R_r}{L_r},\ \text{and}\ \beta = \frac{L_m}{\sigma L_s L_r}$$

$$\psi_{\beta r}(k) = \psi_{\beta r}(k-1) + 0.5\left((1+a)\Delta\psi_{\beta r,p}(k) + (1-a)\Delta\psi_{\beta r}(k)\right)$$
$$\psi_{\alpha r}(k) = \psi_{\alpha r}(k-1) + 0.5\left((1+a)\Delta\psi_{\alpha r,p}(k) + (1-a)\Delta\psi_{\alpha r}(k)\right)$$
$$i_{\beta s}(k) = i_{\beta s}(k-1) + 0.5\left((1+a)\Delta i_{\beta s,p}(k) + (1-a)\Delta i_{\beta s}(k)\right)$$
$$i_{\alpha s}(k) = i_{\alpha s}(k-1) + 0.5\left((1+a)\Delta i_{\alpha s,p}(k) + (1-a)\Delta i_{\alpha s}(k)\right)$$
$$\omega_r(k) = \omega_r(k-1) + 0.5\left((1+a)\Delta\omega_{r,p}(k) + (1-a)\Delta\omega_r(k)\right)$$
$$T_e = K_8\left(\psi_{\alpha r}(k) i_{\beta s}(k) - \psi_{\beta r}(k) i_{\alpha s}(k)\right)$$

که در آن K1,2,3ضرائب مربوط به شار و k4,5,6,7ضرائب مربوط به جریان استاتور و k8 ضریب مربوط به محاسبه گشتاور و k9,10ضرائب مربوط به سرعت روتور و به صورت زیراند:

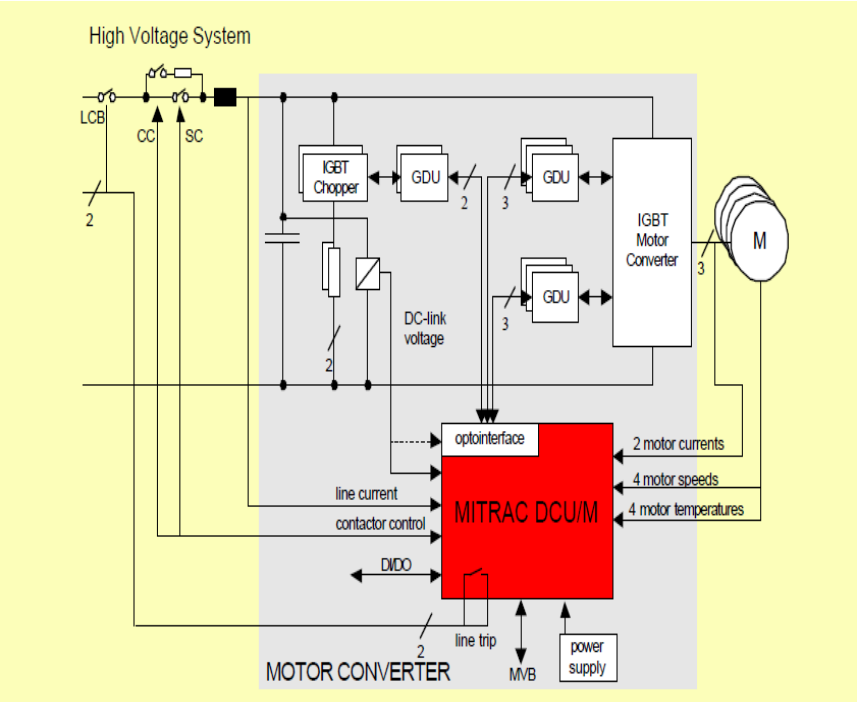
$$K_1 = T\alpha,\quad K_2 = T\omega_b,\quad K_3 = T\alpha L_m \frac{I_b}{\psi_b},\quad K_4 = T\alpha\beta\frac{\psi_b}{I_b},\quad K_5 = T\beta\frac{\psi_b\omega_b}{I_b},$$
$$K_6 = T\gamma,\ K_7 = T\frac{1}{\sigma L_s}\frac{V_b}{I_b},\ K_8 = 1.5n_p\frac{L_m}{L_r}\frac{\psi_b I_b}{T_b},\ K_9 = T\frac{B}{J},\ K_{10} = T\frac{n_p}{J}\frac{T_b}{\omega_b}$$

جهت شبیه سازی در محیط نرم افزار MATLAB این ماژول پیاده شده و در تابع ای با نام aci.M قرار گرفته است. جهت مشاهده این فایل می توانید به CD پیوست مراجعه فرمایید.

فصل ۲- ارتقاء مکانیزم کنترل موتور

تا کنون روش کنترل موتور FOC پیاده سازی شده است و به دنبال ارتقاء مدل کنترلی هستیم،فارق از این که الگوریتم کنترل یک موتور DTCیا FOC یا ISC است اقتضاعات ورودی و خروجی در یک مبدل کنترل دور موتور صنعتی همانند آنچه در مترو تهران یا MCM به صورت ذیل است:

فرمان شارژ لینک DC، فرمان فعال شدن، جهت حرکت و میزان قدرت رانش یا ترمز از اطلاعات ورودی به کنترلر درایو یا DCU/M است و مقدار قدرت رانش یا ترمز واقعی و حداکثر میزان ترمز الکتریکی قابل اعمال و سرعت محور چرخ و خطاهای رخداده، اطلاعات خروجی DCU/M می باشند.



شکل ۱- اطلاعات ورودی و خروجی در DCU/M

دستورات مربوط به شارژ و فعال شدن کانورتر

فرمان شارژ شدن لینک DCتوسط VCUصادر می شود. اما MCMها برای کنترل جریان هجومی با تاخیر ۲ ثانیه ای نسبت به هم در هر واگن شروع به شارژ می نمایند. با دریافت فرمان (cut-out) از MVB کانورتر قطع شده و لینک DC دشارژ خواهد شد.

به طور کلی دو دستور شارژ و شروع به کار کانورتر باعث چهار وضعیت زیر در کانورتر وجود خواهد داشت.

فعال سازی شارژ	فعال سازی کانورتر	وضعیت کانورتر
۰	۰	به سرعت موتور را متوقف کرده و لینک DC را دشارژ می کند.
۰	۱	اگر موتور در حال کار باشد اجازه داده می‌شود لینک DCهمچنان شارژ بماند و سرعت قطار اگر در حد ترمز پشتیبان است در همان سرعت قبل از فرمان خواهد بود. در غیر این صورت لینک DC دشارژ می‌شود.
۱	۰	اگر هیچ نوع خطایی وجود نداشته باشد یا کانورتر (cut-out)قطع نشده باشد لینک DC شارژ می‌شود. اگر موتور در حال کار است آنگاه مقدار مرجع گشتاور در شیب نزولی به سمت صفر قرار می گیرد و کانورتر متوقف می‌شود ولی لینک DC همچنان شارژ می ماند.
۱	۱	برای شارژ شدن خازن ها صبر می کند سپس موتور را روشن می کند.

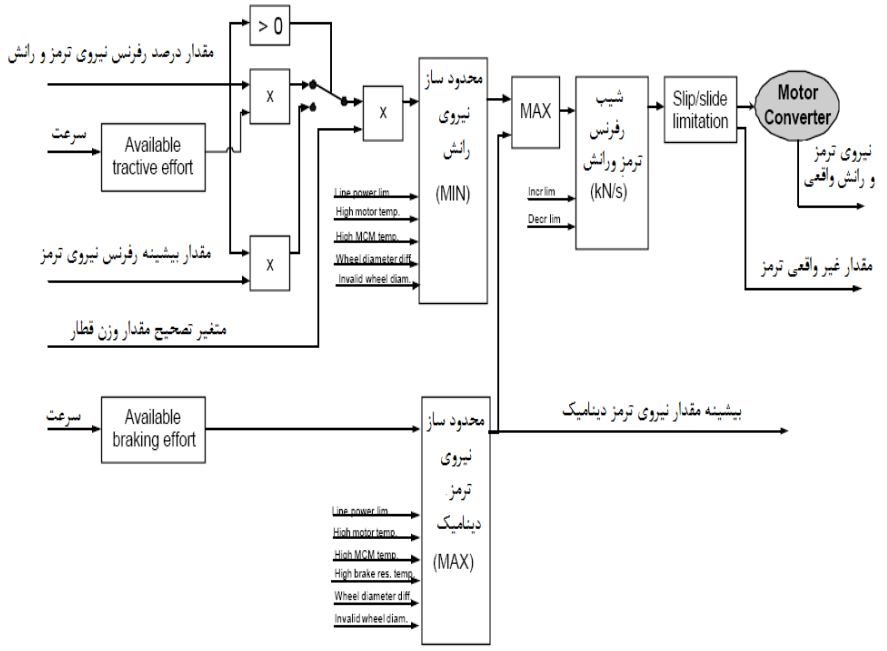
دستور جهت حرکت

دستور جهت حرکت توسط VCUصادر می‌شود. اگر دستور هر دو جهت همزمان انتخاب شده باشد بلوک حفاظت که ادامه به طور تفصیلی حواهد آمد اجرا شده و کانورتر متوقف می‌شود و پیغام خطا جهت حرکت صادر خواهد شد.دستور تغییر جهت فقط در سرعت های زیر ۵ کیلومتر بر ساعت قابل

قبول است. اگر برای MCM حرکت به سمت جلو یا عقب تعیین نگردد نیز کانورتر باید متوقف شود.

محدود سازی مقدار مرجع ترمز یا رانش

مقدار مرجع اعمالی رانش یا ترمز از طرف VCUدر MCMبا توجه به بعضی پارامترها محدود سازی خواهد شد. برای ترمز دو مقیاس تعریف شده است بدین دلیل که گاهی به علت کم شدن امکان ترمز دینامیکی نیاز به ترمز پنوماتیکی برای جبران آن می باشیم.

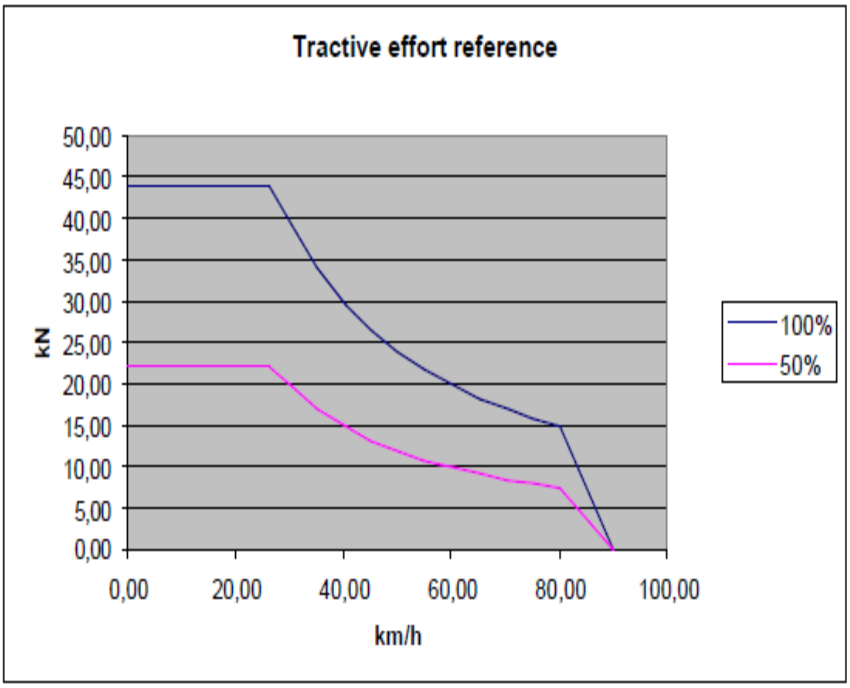


در زیر کمیت های تعریف شده در بالا تشریح شده اند:

مقدار رفرنس نیروی ترمز و رانش

مقدار بیشینه رفرنس نیروی ترمز مقداری ثابت است که به خلاف رفرنس نیروی رانش به سرعت بستگی ندارد، وقتی نیروی ترمز دینامیک محدود شده باشد این مقدار از مقدار بیشینه نیروی ترمز دینامیک کمتر است لذا ترمز پنوماتیکی به نیروی آن می افزاید تا به مقدار ثابت کیلو نیوتن قبل برسیم. مقدار بیشینه نیروی ترمز دینامیک به VCUبازگشت داده می شود تا برای ترمز پنوماتیک ارسال شود.

به صورت کلی مقدار رفرنس نیروی ترمز و رانش عددی بین ۰ تا ۱۰۰ درصد می باشند. مقدار رفرنس نیروی رانش تابع میزان سرعت قطار است. شکل زیر این رابطه را نشان می دهد.



متغیر تصحیح مقدار وزن قطار

این متغیر نیز عددی بین ۰ تا ۱۰۰ درصد است که میزان رفرنس نیروی ترمز و رانش را تصحیح می کند.

محدود سازهای رفرنس نیروی ترمز و رانش

محدودیت های خط تغذیه، افزایش دمای موتور و افزایش دمای کانورتر، مقدار قطر چرخ غیر مجاز، یا تفاوت مقدار قطر چرخ بیش از مقدار مجاز و افزایش دمای مقاومت های ترمزی

شیب رفرنس نیروی ترمز و رانش

برای شروع به حرکت نرم قطار و برای کاهش میزان اغتشاش EMI و کاهش میزان افت ولتاژ ناشی از تغییر توان به علت ترمز و یا حرکت از بلوک کنترل شیب استفاده می شود. این بلوک متغیر ورودی مجزایی دارد که بر اساس سرعت قطار و نیروی ترمز و رانش این شیب محدود سازی می شود.بیشینه نرخ افزایش نیروی ترمز و رانش 50%/s و بیشینه نرخ کاهش نیروی ترمز و رانش 100%/s است.

محدود سازی لغزش و سرخوردن

مقدار نیروی واقعی ترمز و رانش پس از اعمال محدودیت های سرخوردن و لغزش تعیین می گردد.مقدار غیر واقعی ترمز نیز می تواند برای ترمز ترکیبی ارسال شود تا از فشار بیش از حد ترمز پنوماتیک به محور در هنگام سرخوردن بکاهد.

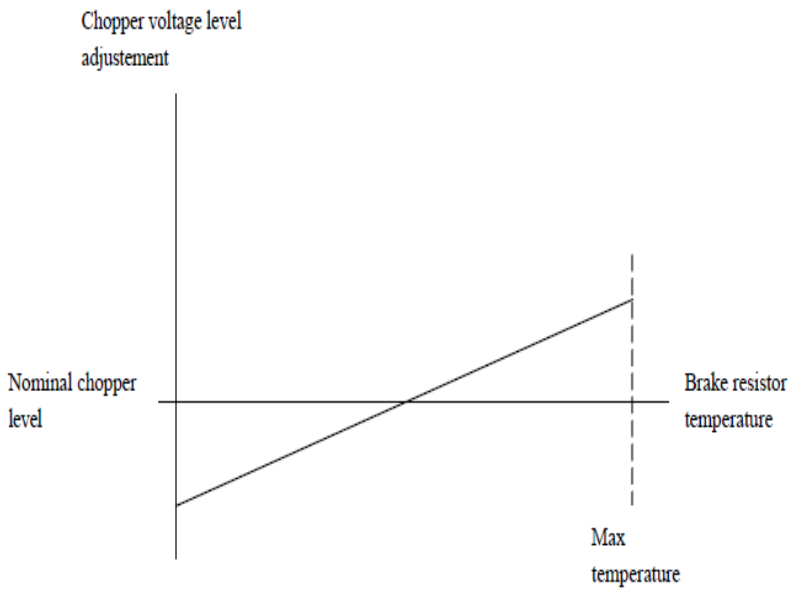
کنترل ترمز الکترو دینامیکی

مکانیزم ترمز سه دسته است:

بازتولید توان: با کاهش فرکانس استاتور نسبت به فرکانس روتور سبب افزایش ولتاژ لینک DC می شود.

ترمز مکانیکی: اگر بازگشت توان به خط در کانورتر محدود یا خاموش شده باشد ترمز مکانیکی برای کاهش سرعت جانشین می شود.

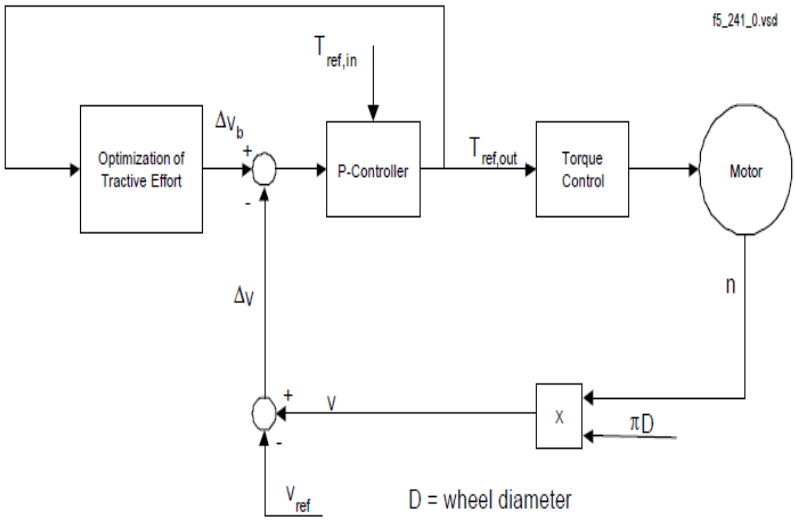
ترمز مقاومتی: فرکانس چاپر ترمز مقاومتی ثابت است، مقدار توانی که باید تلف شود و توان اتلافی در مقاومت دائما با هم مقایسه می شود تا توان دیگر واگن ها در اینجا تلف نشود. دمای مقاومت ها می تواند کمی میزان ولتاژ چاپر را تعدیل کند مثلا مقاومتی که داغ تر شده تلفاتش کاهش می یابد و حالت بازگشت توانش بیشتر می شود.مقدار ترمز الکتریکی قابل اعمال دائما در حال ارسال به VCUاست.



کنترل لغزش و سرخوردن

سرعت روی هر محور توسط یک تاکومتر اندازه گیری می شود متوسط سرعت این دو محور جهت کنترل گشتاور به کار می رود و مقدار آن نیز به VCUگزارش می شود و سرعت قطار از این سرعت ها ارسالی محاسبه می شود و VCU آن را دوباره به DCU/M جهت محاسبه لغزش یا سرخوردن بر می گرداند. لغزش در حالت فرمان رانش و سرخوردن در فرمان حالت ترمز توسط DCU/M بررسی می شود و در صورت رخداد آن گشتاور مرجع کم می شود تا شفت در اثر نیروهای پچشی تاب نخورد. مقدار غیر واقعی ترمز نیز محاسبه می شود و به عنوان یک گزینه برای ترمز ترکیبی ارسال می شود تا از فشار بیش از حد ترمز پنوماتیک به محور در هنگام سرخوردن بکاهد. مقدار غیر واقعی ترمز برای لحظات گذرای سرخوردن به کار نخواهد رفت.

در زیر دیاگرام سیستم کنترل سرخوردن نمایش داده شده است. میزان سرعت واقعی و سرعت چرخ مقایسه شده و مقدار لغزش به دست آمده و با در نظر گرفتن بلوک بهینه سازی نیرو میزان لغزش و گشتاور به بلوک کنترلر وارد شده و این بلوک با کاهش گشتاور به حل مشکل خواهد پرداخت.



اگر مقدار نیروی ترمزی به دست آمده خیلی کم شود (کمتر از ۲۵ درصد) و اگر سر خوردن هنوز ادامه دارد برای تامین ترمز ۲ ثانیه مقدار نیروی غیر واقعی ترمز برابر میزان نیروی ترمزی به دست آمده خواهد شد.

فصل ۳- حفاظت، نظارت و سیستم عیب یابی

مقدمه

در برنامه اولیه جهت حفاظت از مدار قدرت در مقابل افزایش ناخواسته ولتاژ و یا جریان، پس از اندازه‌گیری تمامی این پارامترها، در صورتیکه هر یک از آنها از مقدار مقرر تخطی می کرد، برنامه متوقف می‌شد و تمامی سوئیچ های خاموش می شدند. جهت توقف برنامه وخاموش کردن IGBT ها از تابع Shut Down استفاده شد. برنامه‌ای که این تابع را فراخوانی می‌نمود، کد (iCode) به آن ارسال می‌نمودکه استفاده کننده (user) با بررسی آن می توانست منشأ درخواست توقف را مشخص نماید. به منظور جلوگیری از توقف برنامه به جهت ایجاد نویزهای ناخواسته و یا حالتهای گذرا، در صورتیکه هر یک از پارامترهای برای یک بار از حد خود تجاوز می‌نمود، برنامه متوقف نمی شد. برنامه ShutDownCheck بدین منظور آماده شد تا برای هر پارامتر، در صورتیکه افزایش بیش از حد به تعداد معینی اتفاق افتاد سیستم متوقف گردد. پس از وقوع هر بار اضافه ولتاژ ویا جریان، این شماره یکی افزایش می یافت. آدرس این شمارنده (iCounter) به همراه اندازه پارامتر حد آن تعداد ماکزیمم

دفعات مجاز به همراه کد مربوطه که معرف پارامتر در حال بررسی می‌باشد به تابع shutDownCheck ارسال می‌گردد. درصورت بیشتر بودن میزان پارامتر از حد مجاز شمارنده، برنامه متوقف و سوئیچ ها خاموش می شدند.

برنامه		int ShutDownCheck
برنامه های فراخواننده		SampellingADC
پارامترها	ورودی	اندازه سیگنال، حد مجاز سیگنال، تعداد مجاز وقوع خطا، آدرس شمارنده تعداد خطا
	خروجی	تعداد خطا تا بحال
توضیحات برنامه		بررسی حد مجاز سیگنال و توقف برنامه در صورت نیاز

مدل پیشنهادی برای حفاظت کانورتر موتور

همان طور که مشخص است این بلوک حفاظت اولیه برای حفاظت از کانورتر کافی نمی باشد و نیازمند یک سیستم حفاظت و نظارت و عیب یابی هستیم، در ادامه با توجه به سناریو های احتمالی خطا در قسمت های مختلف تجهیز MCM به دنبال طراحی سیستم نظارت و حفاظت و عیب یابی هستیم، لذا در گام اول برای حفاظت از کانورتر ۵ سطح حفاظت زیر تعریف شده تا سطوح عکس العمل به خطا ها متنوع گردند.

نوع حفاظت	توضیحات
مسدود سازی نرم	با یک شیب گشتاور رفرنس کم می‌شود و کانورتر مسدود می‌شود و HSB هنوز بسته است.
مسدود سازی جهت حفاظت	گشتاور رفرنس به صورت آنی صفر می‌شود و کانورتر مسدود می‌شود، کنتاکتور جداساز مقاومت پری شارژ و HSB هنوز بسته است.
خاموش سازی نرم	با یک شیب گشتاور رفرنس کم می‌شود و کانورتر مسدود می‌شود، کنتاکتور جداساز مقاومت پری شارژ باز می‌شود و HSB هنوز بسته است.
خاموش سازی جهت حفاظت	HSB از طریق رله روی برد DCU/Mباز می‌شود، کنتاکتور جداساز مقاومت پری شارژ باز می‌شود و خازن دشارژ می‌شود. این رده حفاظت در خطاهای جدی مثل اضافه ولتاژ لینک DC اجرا می‌شود. این فرمان از DCU/M به VCU فرستاده شده و VCU به HSB دستور قطع داده و DCU/M های دیگر نیز را خبر دار کرده و HSBآنها را نیز قطع می نماید.این HSB می توانند بعد از جداسازی کانورتر معیوب، دوباره وصل شوند.
ایزوله شدن	اگر خطا رخ داده رفع نشده باشد یا فرکانس رویداد خطا زیاد باشد کانورتر ایزوله شده و کانورتر دیگر نمی‌تواند دوباره فعال شود.بعد از رفع عیب در تعمیرگاه قابل ریست شدن توسط VCU می باشد.

کلیه رویدادهای سیستم که در ادامه بیان شده اند از طریق لینک MVB به سیستم عیب یابی قطار به عنوان پیام ارسال خواهند شد. شماره پیام و مقدار آن در این سیستم ذخیره می‌شود. کانورتر در هنگام بروز خطاهای

بزرگ توسط Line Circuit Breaker حفاظت می‌شود. تعداد بارهای بروز خطا در یک پریود زمانی محدود خواهد بود و در این صورت کانورتر ایزوله شده و در تعمیرگاه باید به صورت دستی بعد از رفع عیب باید ریست شود.

به طور ۴ گروه اطلاعات محیطی وجود خواهد داشت، در این طبقه بندی گروه اول متغییر هایی با حساسیت بالا هستند که باید به صورت متوالی نظارت شوند و در گروه چهارم لَخت ترین و کم اهمیت ترین متغییر ها جهت حفاظت حضور دارند.

کل رویدادهای اطلاعات محیطی اول که هر میلی ثانیه ثبت می‌شود به صورت زیر می باشد:

Word	Bit	Type	Unit	Signal
1	0	Analogue	V	DC-link voltage
2	0	Analogue	V	Brake chopper voltage reference
3	0	Analogue	A	DC-link current
4	0	Analogue	Hz	Average mechanical motor frequency
5	0	Analogue	kN	Reference torque
6	0	Analogue	kN	Reference torque after pull out prot./ volt stab.
7	0	Analogue	kN	Torque
8	0	Analogue	Hz	Slip frequency reference
9	0	Analogue	Vs	Ramped reference flux
10	0	Analogue	Vs	Absolute value estimated stator flux
11	0	Analogue	1/1	Modulation index pulse pattern generator
12	0	Analogue	1/1	Sum of mode and number of vectors per section
13	0	Analogue	A	Filtered failure current (rotor flux orient/imag part)
14	0	Analogue	V	Brake chopper voltage reference from MCU
15	0	Analogue	%	Brake chopper duty cycle
16	0	Analogue	1/1	GDU feedback status word
18	1	Boolean		Command start converter
18	2	Boolean		Phase current calibration enabled
18	3	Boolean		Line current calibration enabled
18	4	Boolean		DC-link voltage calibration enabled
18	5	Boolean		Low voltage operation enabled

Word	Bit	Type	Unit	Signal
18	6	Boolean		Fault reset
18	7	Boolean		Drive direction forward
18	8	Boolean		DC-link voltage controlled by MCU
18	9	Boolean		Undefined
18	10	Boolean		Status DC-link charged
19	1	Boolean		Converter operation enabled
19	2	Boolean		Status converter in operation
19	3	Boolean		Status magnetization ready
19	4	Boolean		Status field weakening
19	5	Boolean		Status braking
19	6	Boolean		Status pantograph bouncing/line interruption
19	7	Boolean		Torque ramp has been reset
19	8	Boolean		OVP is active
19	9	Boolean		Pull Out-Protection and -Limitation
19	10	Boolean		Stator current is limited
19	11	Boolean		Modulation index is limited
19	12	Boolean		Status mode change
19	13	Boolean		Converter block request
19	15	Boolean		Converter blocked
19	16	Boolean		Asymmetrical stabilization at ice operation
20	1	Boolean		Converter externally blocked

20	2	Boolean		MCU crystal frequency failure
20	3	Boolean		MCU - DSP communication failure
20	4	Boolean		DSP overload
20	5	Boolean		FPGA failure
20	6	Boolean		Status opto enabled
20	7	Boolean		DC-link overvoltage
20	8	Boolean		Phase overcurrent
20	9	Boolean		Status Converter switching failure
20	10	Boolean		Earth fault
20	11	Boolean		Brake chopper switching failure
20	12	Boolean		Capacitor failure
20	13	Boolean		Overspeed
20	14	Boolean		Capacitor supervision failure
20	15	Boolean		No valid speed sensor

کل رویدادهای گروه دوم ۱۰۰ اطلاعات محیطی است که هر ۶۴ میلی ثانیه ثبت می‌شود.

Word	Bit	Type	Unit	Signal
1	0	Analogue	V	DC link voltage
2	0	Analogue	A	DC link current
3	0	Analogue	km/h	Average speed
4	0	Analogue	1/1	Converter state of operation
5	1	Boolean		Digital Input 1
5	2	Boolean		Digital Input 2
5	3	Boolean		Digital Input 3
5	4	Boolean		Digital Input 4
5	5	Boolean		Digital Input 5
5	6	Boolean		Pantograph is up
5	7	Boolean		Line circuit breaker is closed
5	8	Boolean		Converter 1 in operation
5	9	Boolean		Converter 2 in operation
5	10	Boolean		DC link is charged
5	11	Boolean		DSP is running
5	12	Boolean		FPGA is running
5	13	Boolean		DCU initialized
5	14	Boolean		Line trip feedback
5	15	Boolean		Maintenance mode
5	16	Boolean		Slip/slide indication
6	1	Boolean		Order charging contactor
6	2	Boolean		Order separation contactor
6	3	Boolean		Charging failure
6	4	Boolean		Protective shutdown due to charging failure
6	5	Boolean		DC link voltage failure during charging

6	6	Boolean		DC link current failure during discharging
6	7	Boolean		DC link voltage failure during discharging
6	8	Boolean		Forward drive direction
6	9	Boolean		Backward drive direction
6	10	Boolean		Undefined
6	11	Boolean		Undefined
6	12	Boolean		Undefined
6	13	Boolean		Undefined
6	14	Boolean		Undefined
6	15	Boolean		Undefined
6	16	Boolean		Undefined
7	0	Analogue	kN	Tractive/braking effort reference
8	0	Analogue	kN	Achieved tractive/braking effort
9	0	Analogue	℃	Estimated brake resistor temperature
10	0	Analogue	℃	Charging resistor temperature
11	0	Analogue		DSP summary event group 1
12	0	Analogue		DSP summary event group 2

کل رویدادهای گروه سوم اطلاعات محیطی است که هر ۶۴ میلی ثانیه ثبت می‌شود.

Word	Bit	Type	Unit	Signal
1	0	Analogue	V	DC link voltage
2	0	Analogue	A	DC link current
3	0	Analogue	km/h	Average speed
4	0	Analogue	1/1	Converter state of operation
5	1	Boolean		Digital Input 1
5	2	Boolean		Digital Input 2
5	3	Boolean		Digital Input 3
5	4	Boolean		Digital Input 4
5	5	Boolean		Digital Input 5
5	6	Boolean		Pantograph is up
5	7	Boolean		Line circuit breaker is closed
5	8	Boolean		Converter 1 in operation
5	9	Boolean		Converter 2 in operation
5	10	Boolean		DC link is charged
5	11	Boolean		DSP is running
5	12	Boolean		FPGA is running
5	13	Boolean		DCU initialized
5	14	Boolean		Line trip feedback
5	15	Boolean		Maintenance mode
5	16	Boolean		Slip/slide indication
6	1	Boolean		Power supply failure
6	2	Boolean		Power fail +15V
6	3	Boolean		Power fail -15V
6	4	Boolean		Power fail +3V
6	5	Boolean		Internal 2.5V failure
6	6	Boolean		Internal 15V failure
6	7	Boolean		A/D converter 2 failure
6	8	Boolean		DSP error
6	9	Boolean		DSP ok
6	10	Boolean		Watchdog failure
6	11	Boolean		DSP stall alarm
6	12	Boolean		WD test performed
6	13	Boolean		DSP failure
6	14	Boolean		FPGA error
6	15	Boolean		FPGA failure
6	16	Boolean		Battery voltage fault
7	1	Boolean		Digital output enable
7	2	Boolean		Digital output failure
7	3	Boolean		Digital output test performed

7	4	Boolean		SPI failure
7	5	Boolean		SPI failure
7	6	Boolean		SPI failure
7	7	Boolean		SPI failure
7	8	Boolean		Analogue output fault
7	9	Boolean		Analogue output test performed
7	10	Boolean		Activate analogue output test
7	11	Boolean		MVB watchdog error
7	12	Boolean		MVB initialisation error
7	13	Boolean		MVB read error
7	14	Boolean		MVB write error
7	15	Boolean		Line trip failure
7	16	Boolean		All speed sensors +15V supply failure
8	1	Boolean		MCU watchdog enabled
8	2	Boolean		Opto enable feedback
8	3	Boolean		Traction safe enable
8	4	Boolean		Traction safe/opto enable fault
8	5	Boolean		DSP <-> FPGA communication failure
8	6	Boolean		DSP overload
8	7	Boolean		MCU <-> DSP communication failure
8	8	Boolean		MCU crystal frequency failure
8	9	Boolean		FPGA revision number failure
8	10	Boolean		Overload task T1
8	11	Boolean		Overload task T2
8	12	Boolean		Overload task T3
8	13	Boolean		Overload task T4
8	14	Boolean		Overload task T5
9	0	Analogue	1/1	DSP error code
10	0	Analogue	1/1	FPGA error code
11	0	Analogue	℃	Control board temperature
12	0	Analogue	℃	Converter heatsink temperature

کل رویدادهای گروه چهارم اطلاعات محیطی است که هر ۱۰۲۴ میلی ثانیه ثبت می‌شود.

Word	Bit	Type	Unit	Signal
1	0	Analogue	V	DC link voltage
2	0	Analogue	A	DC link current
3	0	Analogue	km/h	Average speed
4	0	Analogue	1/1	Converter state of operation
5	1	Boolean		Digital Input 1
5	2	Boolean		Digital Input 2
5	3	Boolean		Digital Input 3

5	4	Boolean		Digital Input 4
5	5	Boolean		Digital Input 5
5	6	Boolean		Pantograph is up
5	7	Boolean		Line circuit breaker is closed
5	8	Boolean		Converter 1 in operation
5	9	Boolean		Converter 2 in operation
5	10	Boolean		DC link is charged
5	11	Boolean		DSP is running
5	12	Boolean		FPGA is running
5	13	Boolean		DCU initialized
5	14	Boolean		Line trip feedback
5	15	Boolean		Maintenance mode
5	16	Boolean		Slip/slide indication
6	0	Analogue	℃	Motor temperature 1
7	0	Analogue	℃	Motor temperature 2
8	0	Analogue	℃	Motor temperature 3
9	0	Analogue	℃	Motor temperature 4
10	0	Analogue	℃	Converter air temperature
11	0	Analogue	℃	Control board temperature
12	0	Analogue	℃	Converter heatsink temperature

از میان اطلاعات تنها حداکثر ۸ سیگنال به مدت ۴ ثانیه پیشنهاد می شود توسط نرم افزاری خاص که باید نوشته شود به عنوان سیگنال های استاندارد گذرا (Standard Transient Recorder) ثبت شوند اگر خطایی مشاهده شود اطلاعات ۲,۴ ثانیه قبل و ۱,۶ ثانیه بعد از خطا با فرکانس نمونه گیری ۴ میلی ثانیه نگهداری شود. هر برد DCU/M باید دارای امکان ثبت سیگنال های

گذرا داخلی باشد که این اطلاعات با اتصال رایانه از طریق پورت های استاندارد قابل دریافت خواهد بود.

از میان اطلاعات تنها حداکثر ۸ سیگنال حسابی و ۱۶ سیگنال دیجیتال به مدت ۴ ثانیه پیشنهاد می شود به عنوان سیگنال های بدون تغیر گذرا (Fixed transient recorder) ثبت شوند اگر خطایی مشاهده شود اطلاعات ۲,۴ ثانیه قبل و ۱,۶ ثانیه بعد از خطا با فرکانس نمونه گیری ۴ میلی ثانیه نگهداری شود. این بلوک حساس به لبه مسدود سازی نرم، مسدود سازی جهت حفاظت، خاموش سازی نرم، خاموش سازی جهت حفاظت خواهد بود و اطلاعات زیر را ثبت می کند.

Word	Bit	Type	Unit	Signal
1	1..8	Analogue	1/1	Event number 1..255
1	9-16	Analogue	1/1	Actual state 1..31
2	0	Analogue	V	DC link voltage
3	0	Analogue	Nm	Actual torque
4	0	Analogue	Nm	Torque reference
5	0	Analogue	A	Stator current, RMS
6	0	Analogue	Hz el.	Average mechanical motor speed
7	0	Analogue	℃	Converter heatsink temperature
8	1	Boolean		Brake chopper active
8	2	Boolean		Separation contactor closed
8	3	Boolean		Charging contactor closed
8	4	Boolean		Overcurrent motor phase 1,2 or 3
8	5	Boolean		DC link overvoltage
8	6	Boolean		DC link undervoltage
8	7	Boolean		Backup braking active
8	8	Boolean		Protective shutdown request active
8	9	Boolean		Protective blocking request active
8	10	Boolean		Soft shutdown request active
8	11	Boolean		Soft blocking request active
8	12	Boolean		Order open line circuit breaker
8	13	Boolean		Line trip relay closed
8	15	Boolean		Slip indication
8	16	Boolean		3-phase auxiliary voltage bus active

گروه بندی خطاهای منجر به برروز مسدود سازی جهت حفاظت

۱. تشخیص خطای اتصال به زمین، پیام Earth fault

جریان های فاز +DC و -DCبررسی می شود تا خطای اتصال به زمین را در موتور، کابل های موتور، کابل های مقاومت ترمزی، یا مقاومت ترمزی و MCMرا سیستم کشف کند هرگونه نابربری این دو جریان به معنای اتصال به زمین یا خرابی سنسور DCتفاضلی یا خرابی DCU/M است، این رخداد سبب بروز خاموش سازی جهت حفاظت می شود، اگر این حالت ۳ بار در ۳۰ دقیقه قبل رخ دهد این رخداد سبب بروز مسدود سازی جهت حفاظت شده و پیغام خطایی به نامEarth fault در گروه اول اطلاعات محیطی صادر خواهد شد.

۲. خطای جهت حرکت، پیام Converter direction failure

وقتی مقدار عددی دو سنسور سرعت قابل قبول باشد ولی در سرعت بالای ۳ کیلومتر در ساعت جهت این دو متفاوت باشد پیغام خطایی با نام Converter direction failure در گروه دوم اطلاعات محیطی صادر خواهد شد و کانورتر مسدود می‌شود.به عبارت دیگر اگر فرمان جهت حرکت در سرعت کمتر از ۳ کیلومتر در ساعت صادر شود پذیرفته می شود

۳. نظارت بر جهت حرکت، پیام Direction selection failure

دستور جهت حرکت توسط VCUصادر می شود. اگر دستور هر دو جهت همزمان انتخاب شده باشد کانورتر جهت حفاظت مسدود می‌شود. پیغام خطایی با نام Direction selection failure در گروه دوم اطلاعات محیطی

صادر خواهد شد اگر هیچ جهت همزمان انتخاب نشده باشد رفرنس گشتاور صفر می‌شود ولی خطایی صادر نمی‌شود.

۴. نظارت بر رفرنس نیرو، پیام Tractive/brake effort reference plausibility failure

علامت رفرنس نیروی رانش و ترمز با حالت رانش و ترمز مقایسه و در صورت تناقض ظرف ۱ ثانیه کانورتر مسدود می‌شود و پیغام خطایی با نام Tractive/brake effort reference plausibility failure در گروه دوم اطلاعات محیطی صادر خواهد شد. این مشکل فقط می تواند به علت نرم افزار VCU باشد.

گروه بندی خطاهای منجر به برروز مسدود سازی نرم

۱. نظارت بر سنسور دمای داخلی کانورتر، پیام Converter internal air over temperature

رسیدن دمای به ۷۰ درجه یا رسیدن به ۶۵ درجه و طول کشیدن طولانی مدت ۱۵ دقیقه سبب می شود سه فاز موتور مسدود سازی نرم می‌شود از دمای ۶۰ درجه پس از آن نیروی رانش تا ۵۰ درصد در ۷۰ درجه کاهش می یابد و وقتی دما پایین آمد دوباره کانورتر فعال سازی انجام خواهد شد. اگر این حالت ۳ بار در ۳۰ دقیقه قبل رخ دهد این رخداد سبب بروز حفاظت ایزوله شدن خواهد شد و پیغام خطایی به نام Converter internal air over temperature در گروه چهار اطلاعات محیطی صادر خواهد شد. دلایل این مشکل قطع تغذیه ۲۴+، اشتباه در اندازه گیری دما، اضافه بار، اشکال در سیم کشی فن، و خرابی فن داخلی یا از دست رفتن خنک ساز می تواند باشد.

گروه بندی خطاهای منجر به برروز خاموش سازی نرم

۱. نظارت بر سنسور دمای روی برد، پیام DCU/M control board over temperature

اگر دمای برد به بالای ۸۵ درجه برسد کانورتر حفاظت خاموش سازی نرم می‌شود و منبع تغذیه برد ۱۰ ثانیه خاموش می‌شود و پیغام خطایی به نام DCU/M control board over temperature در گروه چهارم اطلاعات محیطی صادر خواهد شد.

۲. خطای خاموش سازی جهت حفاظت، پیام Protective shutdown failure

اگر طی خاموش سازی جهت حفاظت، مراحل خاموش سازی همچون باز شدن کنتاکتور های جداساز و متصل کننده مقاومت پری شارژ یا دشارژ لینک DC یا قرارگیری ولتاژ لینک DCبه زیر حد مجاز در یک زمان معین تکمیل نشود، پیغام خطایی به صورت Protective shutdown failure در گروه دوم اطلاعات محیطی صادر و خاموش سازی نرم در دستور کار قرار می گیرد. علت مشکل می‌تواند خرابی GDU یا خرابی چاپر یا خرابی مقاومت ترمزی، خرابی کنتاکتور شارژر یا کنتاکتور جدا ساز، خرابی DCU/Mیا خرابی کابل های مقاومت ترمزی باشد.

۳. نظارت بر سنسور جریان فاز، پیام Phase current calibration failure

این یک مشکل داخلی کانورتر است که به علت خطای سنسور جریان یا خطای DCU/M پیش می‌تواند بیاید این مشکل سبب خاموش سازی نرم

خواهد شد و با یک بار باز و بسته شدن LCB سنسور های جریان دوباره کالیبره خواهند شد و پیغام خطایی به همین نام در گروه اول اطلاعات محیطی صادر خواهد شد.

۴. نظارت بر سنسور جریان خط، پیام Line current calibration failure

این یک مشکل داخلی کانورتر است که به علت خطای سنسور جریان یا خطای DCU/M پیش می‌تواند بیاید این مشکل سبب خاموش سازی نرم خواهد شد و با یک بار باز و بسته شدن LCB سنسور های جریان دوباره کالیبره خواهند شد و پیغام خطایی به همین نام در گروه اول اطلاعات محیطی صادر خواهد شد.

۵. نظارت بر سنسور ولتاژ، پیام DC-link voltage calibration failure

این یک مشکل داخلی کانورتر است که به علت خطای سنسور ولتاژ یا خطای DCU/Mیا خطای توابع چاپر پیش می‌تواند بیاید این مشکل سبب خاموش سازی نرم خواهد شد و با یک بار باز و بسته شدن LCB سنسور دوباره کالیبره خواهند شد و پیغام خطایی به همین نام در گروه اول اطلاعات محیطی صادر خواهد شد.

۶. نظارت بر سنسور ولتاژ، پیام MCM DC-Link voltage measurement inaccuracy

اختلاف بیش از ۳۰ ولت در لینک DCدو موتور کانورتر در یک واگن سبب بروز این خطا می‌شود. علت این خطا خرابی سنسور سرعت یا خرابی DCU/Mاست. شناسایی سنسور خراب در تعمیر گاه و پس از بررسی مقدار

عددی سنسور دیگر واگن ها است، بدین صورت که در صورت افزایش درصد خطا به بالای ۳ درصد سنسور ولتاژ خراب است و باید تعویض گردد.

۷. نظارت بر دمای موتور، پیام Motor 1 over temperature

به علت اضافه بار موتور یا از دست رفتن کولینگ موتور یا خرابی سنسور دما اگر high temp را به مدت بیش از ۱۵ دقیقه ادامه داشته باشد یا به سر حد دمای اضافه دما برسیم سه فاز موتور مسدود سازی نرم می‌شود و در مواردی وقتی دما پایین آمد دوباره فعال سازی انجام می‌شود. اگر این حالت ۳ بار در ۳۰ دقیقه قبل رخ دهد این رخداد سبب بروز حفاظت ایزوله شدن شده یا پیغام خطایی به نام Motor 1,2 over temperatureدر گروه چهار اطلاعات محیطی صادر خواهد شد.

۸. نظارت بر لغزش و سرخوردن، پیام Slip speed limit exceeded

اگر برای ۱۰ ثانیه لغزش و سرخوردن از حدود مجاز تخطی کند و ادامه یابد مسدود سازی نرم اجرا می‌شود. اگر این حالت ۳ بار در ۳۰ دقیقه قبل رخ دهد این رخداد سبب بروز حفاظت ایزوله سازی شده و پیغام خطایی به نام Slip speed limit exceededگروه دوم اطلاعات محیطی صادر خواهد شد. می‌تواند به علت خرابی سیستم ترمز یا خرابی سنسور سرعت باشد.

گروه بندی خطاهای منجر به برروز ایزوله سازی کانورتر:

۱. ایزوله بودن، پیام MCM is isolated or does not start when ordered

این پیام یک پیام ثانویه خواهد بود و بدین معنی است که کانورتر ۳ خطای خاص را در ۳۰ دقیقه قبل مشاهده کرده است و به علت این خطا های متوالی کانورتر باید ایزوله شود.

۲. خطای مکانیکی سیستم، پیام Mechanical drive system fault

اگر فاز بندی موتور یا کابل موتور دچار اشکال باشد یا شفت و یا محور آن قفل شده باشد جریان واقعی موتور و جریان پیش بینی محاسبه شده با هم تفاوت خواهند داشت این خطا سبب بروز خاموش سازی جهت حفاظت شده و اگر این حالت ۳ بار در ۳۰ دقیقه قبل رخ دهد منجر به بروز ایزوله سازی کانورتر و پیغام خطایی به نام Mechanical drive system fault صادر خواهد شد.

۳. نظارت بر کنتاکتور شارژر، پیام Charging contactor opening / closing failure

فیدبک از این کنتاکتورها نظارت می‌شود اگر به علت خرابی رله یا خرابی کنتاکتور یا خرابی کابل ارتباطی یا خرابی DX، کنتاکتور درست بسته یا باز نشده باشد یا بر خلاف فرمان باز یا بسته بماند سبب بروز خاموش سازی جهت حفاظت شده و اگر این حالت ۳ بار در ۳۰ دقیقه قبل رخ دهد منجر به بروز ایزوله سازی کانورتر و پیغام خطاهایی به نام های Charging contactor opening/closing failure در گروه اول اطلاعات محیطی صادر خواهد شد.

۴. نظارت بر کنتاکتور جدا ساز، پیام Separation contactor opening/ closing failure

فیدبک از این کنتاکتور نظارت می‌شود اگر کنتاکتور به علت خرابی رله یا خرابی کنتاکتور یا خرابی کابل ارتباطی یا خرابی DX درست بسته یا باز نشده باشد یابر خلاف فرمان باز یا بسته بماند سبب بروز خاموش سازی جهت حفاظت شده و اگر این حالت ۳ بار در ۳۰ دقیقه قبل رخ دهد منجر به بروز ایزوله سازی کانورتر و پیغام خطاهایی به نام های Separation contactor opening/closing failure در گروه دوم اطلاعات محیطی صادر خواهد شد.

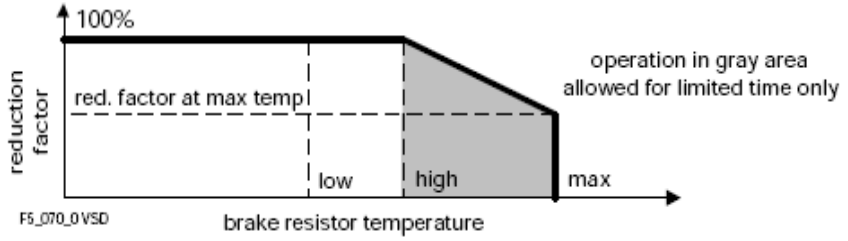
۵. نظارت بر شارژ لینک DC، پیام DC-link charging failure

اگر فرمان بسته شدن کنتاکتور شارژر آمد ولی ولتاژ لینک DC در طی مدت زمان مورد انتظار بالا نرفت سبب بروز خاموش سازی جهت حفاظت شده و کنتاکتور شارژر باز می‌شود و اگر این حالت ۳ بار در ۳۰ دقیقه قبل رخ دهد منجر به بروز ایزوله سازی کانورتر و پیغام خطایی به نام DC link charging failure در گروه دوم اطلاعات محیطی صادر خواهد شد.

به دنبال خطای در سنسور ولتاژ DCیا اتصال کوتاه لینک DC یا خرابی کنتاکتورهای جداساز مقاومت پری شارژ یا کنتاکتور مقاومت پری شارژ یا سیکل شارژ های پی در پی که سبب داغ شدن مقاومت پری شارژ شده یا نبودن خط تغذیه در برخی نقاط می‌تواند علت های این رخداد باشد. اتصال کوتاه لینک DC با بررسی با مولتی متر قابل بررسی است ولتاژ خازن در حالت دشارژ بودن لینک DCو تامین برق با باتری باید ۲۴ ولت باشد و ولتاژ زیر ۱۰ ولت به معنای اتصال کوتاه است همچنین بررسی سنسور لینک DCتفاضلی نیز می‌تواند در کشف این خطا موثر باشد.

۶. نظارت بر مقاومت ترمزی، پیام Brake resistor overtemperature

اگر دما بیش از حد افزایش یابد حفاظت خاموش سازی فعال و مشخصات جهت کاهش رفرنس گشتاور انتشار می یابد. اگر این حالت ۳ بار در ۳۰ دقیقه قبل رخ دهد این رخداد سبب بروز حفاظت ایزوله شدن می شود پیغام خطایی به نام Brake resistor over temperature در گروه دوم اطلاعات محیطی صادر خواهد شد. اشکال در سنسور ولتاژ DCیا افزایش تعداد رخداد اضافه ولتاژ ها و همزمان افزایش ترمز الکتریکی سبب بروز این خطا است. برای رفع مشکل مقدار دمای مقاومت را در سیستم ثبت اطلاعات نگاه کرده و به دنبال خطای در سنسور ولتاژ DC در سیستم عیب یابی بگردید.



۷. نظارت بر دشارژ لینک DC، پیام DC-link discharging failure

قبل از دشارژ باید کنتاکتور جدا کننده باید باز باشد و چاپر با پریود خاصی باید روشن شود در این صورت طی زمانی مشخص ولتاژ لینک DC باید صفر شود. حال به علت مشکل خرابی GDUیا خرابی مقاومت ترمزی، خرابی کنتاکتور شارژر یا کنتاکتور جدا ساز یا سیستم نظارت آن، یا خرابی کابل های مقاومت ترمزی یا خرابی DCU/M که ترتیب و توالی عمل دشارژ را درست پیاده نکرده ولتاژ غیر صفر خواهد شد گاهی نیز جریان به حدی زیاد است که

کنتاکتور نمی‌تواند آن را قطع کند لذا سبب بروز خاموش سازی جهت حفاظت می‌شود و اگر این حالت ۳ بار در ۳۰ دقیقه قبل رخ دهد منجر به بروز ایزوله سازی کانورتر و پیغام خطایی به نام DC link discharging failureدر گروه دوم اطلاعات محیطی صادر خواهد شد.

۸. حفاظت اضافه ولتاژ، پیام DC-link overvoltage

اگر با آمدن مدارات حفاظت از اضافه ولتاژ باز اضافه ولتاژ تا حد بیشینه حفاظت خاموش سازی 1050 V بالا رود این حالت سبب بروز خاموش سازی جهت حفاظت شده و اگر این حالت ۳ بار در ۳۰ دقیقه قبل رخ دهد منجر به بروز ایزوله سازی کانورتر و پیغام خطایی به نام DC link Overvoltage در گروه اول اطلاعات محیطی صادر خواهد شد. برای عیب یابی به دنبال خطای در سنسور ولتاژ DCو خطاهای مرتبط با اضافه ولتاژ در سیستم عیب یابی باید بود.

۹. نظارت بر سنسور های ولتاژ لینک DC، پیام DC-link voltage measurement failure

اعداد سنسور های ولتاژ روی لینک DC توسط VCUنظارت می شوند تا معقول باشند و از این راه سنسور ولتاژ معیوب یا DCU/Mیا کابل های خراب با نشان دادن عدد ولتاژ لینک DCخارج رنج شناسایی می‌شوند اگر این حالت ۳ بار در ۳۰ دقیقه قبل رخ دهد این رخداد سبب بروز حفاظت ایزوله سازی شده و پیغام خطایی به نام DC link voltage measurement failure در گروه اول اطلاعات محیطی صادر خواهد شد.

۱۰. حفاظت سنسور های جریان لینک DC، پیام DC-link input overcurrent

اعداد سنسور های جریان روی لینک DC اگر بیش از ۱۵۰۰ آمپر را حس کند می‌تواند به معنای اتصال کوتاه باشد این حالت سبب بروز خاموش سازی جهت حفاظت یا مسدود سازی جهت حفاظت شده و اگر این حالت ۳ بار در ۳۰ دقیقه قبل رخ دهد این رخداد سبب بروز حفاظت ایزوله سازی شده و پیغام خطایی به نام DC link input over current در گروه دوم اطلاعات محیطی صادر خواهد شد.

خرابی سنسور جریان یا خرابی IGBTیا GDU یا اتصال کوتاه کانورتر یا اتصال کوتاه لینک DC علت بروز مشکل است.

۱۱. نظارت بر مقاومت ترمزی، پیام Brake resistor high temperature

با استفاده از ولتاژ لینک DC و مشخصه مدولاسیون چاپر مقدار تلفات محاسبه می گردد. اگر دما مقاومت بالا رود نیروی ترمز الکتریکی کاهش می یابد و پیغام خطایی به نامBrake resistor high temperature در گروه دوم اطلاعات محیطی صادر خواهد شد.

۱۲. نظارت بر دمای موتور، پیام Motor 1 high temperature

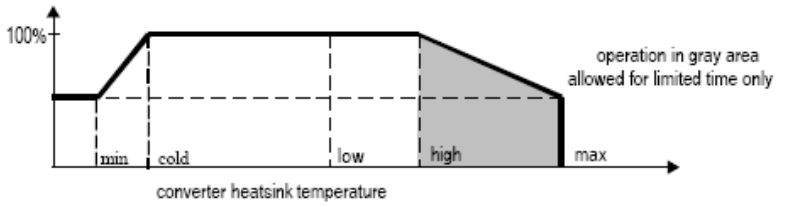
اگر توسط سنسور دمای داخلی موتور اضافه دما تشخیص داده شود رفرنس گشتاور تا ۵۰ درصد کاهش می یابد و پیغام خطایی به نام Motor 1,2 high temperature در گروه چهار اطلاعات محیطی صادر خواهد شد.

۱۳. نظارت بر دمای موتور، پیام Motor 1 temperature measurement failure

گاهی سنسور دمای داخلی موتور ها خراب می‌شود در این حالت اگر سنسوری دمای خارج از حد معقول ۵۰- تا ۵۳۰+ درجه را نشان دهد اطلاعات سنسورهای دیگر ملاک قرار می گیرد، اگر سنسوری سالم نباشد بر اساس شرایط کاری دما تقریب زده می‌شود پیغام خطایی به نام Motor 1,2 temperature measurement failure گروه چهار اطلاعات محیطی صادر خواهد شد.برای عیب یابی کابل کشی بین PT100وDCU/M بررسی شود.

۱۴. نظارت بر سنسور دمای هیت سینک، پیام Converter heatsink high temperature

جهت نظارت بر دمای کانورتر دو سنسور دمای PT100بر روی هیت سینک و در داخل قسمت داخلی کانورتر قرار دارد و یک NTC نیز روی برد DCU/M قرار گرفته است.اگر دمای هیت سینک بالا رود، رسیدن دمای هیت سینک به بالای ۷۰ درجه سبب کاهش نیروی رانش خواهد شد. این خطا سبب بروز واکنش سریع نمی‌شود و پیغام خطایی به نام Converter heat sink high temperature در گروه چهار اطلاعات محیطی صادر خواهد شد.



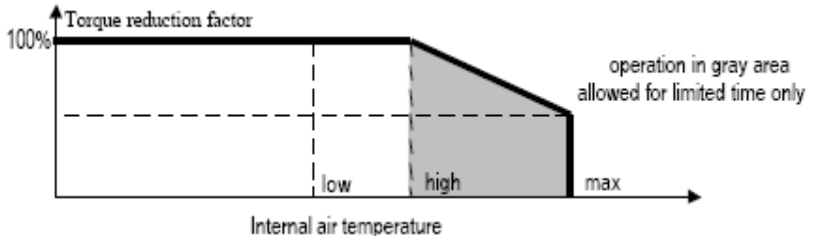
اگر دما کاهش به زیر ۲۰- برسد دوباره مقدار گشتاور محدود می‌شود ولی این بار پیغام خطایی صادر نمی‌شود.

۱۵. نظارت بر سنسور دمای هیت سینک، پیام Converter heatsink overtemperature

رسیدن دمای هیت سینک به ۷۵ درجه یا رسیدن به ۷۱ درجه و طول کشیدن ۱۵ دقیقه ای آن سبب بروز این خطا سه فاز موتور مسدود سازی نرم خواهد شد و پس از آن نیروی رانش تا ۵۰ درصد در ۷۵ درجه کاهش می یابد و وقتی دما پایین آمد دوباره فعال سازی انجام خواهد شد. اگر این حالت ۳ بار در ۳۰ دقیقه قبل رخ دهد این رخداد سبب بروز حفاظت ایزوله شدن خواهد شد و پیغام خطای ی به نام Converter heat sink over temperature در گروه چهار اطلاعات محیطی صادر خواهد شد.دلایل این اشکال افزایش دمای هوا، مسدود شدن فین های هیت سینک، از دست رفتن برق ACM، کثیفی پره های فن، خرابی کنتاکتور یا فن خارجی، اضافه بار یا اشکال در اندازه گیری دما می‌تواند باشد.

۱۶. نظارت بر سنسور دمای داخلی کانورتر، پیام Converter internal air high temperature

اگر سنسور هوای داخل فضای بسته افزایش دما را نشان داد مقدار گشتاور تا ۵۰ درصد محدود خواهد شد و پیغام خطایی به نام Converter internal air high temperature در گروه چهار اطلاعات محیطی صادر خواهد شد و تا هنگام بروز پیام over temp اقدام سریعی برای آن صورت نمی پذیرد.



۱۷. نظارت بر سنسور دمای داخلی کانورتر، پیام Converter internal air over temperature

رسیدن دمای به ۷۰ درجه یا رسیدن به ۶۵ درجه و طول کشیدن طولانی مدت ۱۵ دقیقه سبب می‌شود سه فاز موتور مسدود سازی نرم می‌شود از دمای ۶۰ درجه پس از آن نیروی رانش تا ۵۰ درصد در ۷۰ درجه کاهش می یابد و وقتی دما پایین آمد دوباره کانورتر فعال سازی انجام خواهد شد. اگر این حالت ۳ بار در ۳۰ دقیقه قبل رخ دهد این رخداد سبب بروز حفاظت ایزوله شدن خواهد شد و پیغام خطایی به نام Converter internal air over temperature در گروه چهار اطلاعات محیطی صادر خواهد شد. دلایل این مشکل قطع تغذیه ۲۴+، اشتباه در اندازه گیری دما، اضافه بار، اشکال در سیم کشی فن، و خرابی فن داخلی یا از دست رفتن خنک ساز می تواند باشد.

۱۸. نظارت بر سنسور دما، پیام Converter heatsink internal air-board temperature measurement failure

ارقام اعداد سنسور ها باید بین ۵۰- تا ۱۰۰+ باشد اگر دمای سنسور روی هیت سینک خارج از رنج باشد، سرعت فن خارجی به حداکثر انتقال یافته و پیغام های خطایی به نامهای Converter heat sink-internal air-board temperature measurement failure در گروه چهار اطلاعات محیطی صادر خواهد شد. علت خرابی سنسور یا خرابی کابل ارتباطی یا خرابی DCU/M است. اگر در هنگام تعمیر سنسور روی برد دمای محیط را با ۲۰ درجه بیشتر نشان می دهد نشان دهنده خطای گذرایی بوده است.

۱۹. نظارت بر سنسور دمای، پیام Multiple temperature sensor fault

اگر بیش از یک سنسور خراب باشد حفاظت مسدود سازی نرم اجرا می‌شود و اگر این حالت ۳ بار در ۳۰ دقیقه قبل رخ دهد این رخداد سبب بروز حفاظت ایزوله شدن شده و پیغام خطایی به نام Multiple temperature sensor fault در گروه چهار اطلاعات محیطی صادر خواهد شد.

۲۰. نظارت بر سنسور جریان لینکDC ، پیام DC-link input current measurement failure

اعداد سنسور های جریان روی لینک DC از نظر منطقی بودن نظارت می‌شود خرابی سنسور یا DCU/Mیا کابل های آن خود را با نشان دادن عدد جریان لینکDC خارج از رنج نشان می دهد که سبب بروز خاموش سازی جهت حفاظت شده و لذا اگر این حالت ۳ بار در ۳۰ دقیقه قبل رخ دهد این رخداد سبب بروز حفاظت ایزوله سازی شده و پیغام خطایی به نام DC link input current measurement failure در گروه دوم اطلاعات محیطی صادر خواهد شد.

۲۱. حفاظت سنسور های جریان ۲ فاز، پیام Overcurrent motor phase n

اعداد سنسور های جریان روی ۲ فاز و نتیجه فاز سوم اگر از حد مجاز عبور کند می‌تواند به معنای اتصال کوتاه باشد این حالت سبب بروز مسدود سازی جهت حفاظت می‌شود البته امکان تشخیص اتصال کوتاه توسط گیت درایور نیز وجود دارد که سبب بروز همان نتیجه می‌شود و اگر این حالت ۳ بار در ۳۰ دقیقه قبل رخ دهد این رخداد سبب بروز حفاظت ایزوله سازی شده و

پیغام خطایی به نام۱-۳ Over current motor phase در گروه اول اطلاعات محیطی صادر خواهد شد. بعضی اغتشاشات نذیر پریدن ترمز از روی اکسل چرخ می‌تواند سبب بروز این خطا گردد علاوه بر آن خرابی سنسور جریان، خرابی سنسور سرعت، اتصال کوتاه موتور یا کابل آن، خرابیGDU یا آرک در IGBT می توان سبب آن باشد.

۲۲. نظارت بر اعداد سنسور جریان ۲ فاز، پیام Phase N current measurement failure

سنسور های جریان روی دو فازU,V نصب اند اعداد سنسور های جریان روی ۲ فاز و نتیجه فاز سوم دائما از نظر معقول بودن یا قطع نشدن سیم های سنسور و برد DCU/M نظارت و با هم مقایسه می‌شود انحراف سبب بروز مسدود سازی جهت حفاظت خواهد شد و اگر این حالت ۳ بار در ۳۰ دقیقه قبل رخ دهد این رخداد سبب بروز حفاظت ایزوله سازی شده و پیغام خطایی به نام Phase 1-2 current measurement failure در گروه اول اطلاعات محیطی صادر خواهد شد.

۲۳. نظارت بر سوئیچ زنی کانورتر، پیام Switching time failure

اگر فرکانس سویچ زنی کمتر از ۵۰ هرتز یا بیشتر از ۳/۳ کیلوهرتز شود یا نوع مدولاسیون مجاز نباشد یا پلاریته ولتاژ اندازه گیری شده با فاز های سه گانه تطابق نداشته باشد و این صورت اگر این حالت ۳ بار در ۳۰ دقیقه قبل رخ دهد این رخداد سبب بروز حفاظت ایزوله سازی شده و پیغام خطایی به نام Switching time failure در گروه اول اطلاعات محیطی صادر خواهد شد. این خطا خطای غیر معمولی DCU/Mاست.

۲۴. نظارت بر لغزش و سرخوردن، پیام Slip speed limit exceeded

اگر برای ۱۰ ثانیه لغزش و سرخوردن از حدود مجاز تخطی کند و ادامه یابد مسدود سازی نرم اجرا می‌شود. اگر این حالت ۳ بار در ۳۰ دقیقه قبل رخ دهد این رخداد سبب بروز حفاظت ایزوله سازی شده و پیغام خطایی به نام Slip speed limit exceededگروه دوم اطلاعات محیطی صادر خواهد شد. می‌تواند به علت خرابی سیستم ترمز یا خرابی سنسور سرعت باشد.

۲۵. نظارت بر برد کنترل DCU/M، پیام DCU/M hardware failure

بردDCU/M خود بارگذاری اطلاعات از FPGA به DSP و سخت افزارهای مبدل آناگوگ به دیجیتال و دیجیتال به آنالوگ و منبع تغذیه داخلی را دائما تحت تست و نظارت دارد. این پیام یک پیام خطای گروهی و نشانه خطای جدی در DCU/M است.این خطا در عیب یابی محل بروز خطا در DCU/M موثر است اگر این حالت ۳ بار در ۳۰ دقیقه قبل رخ دهد این رخداد سبب بروز حفاظت ایزوله سازی شده و پیغام خطایی به نام DCU/M hardware failure در گروه سوم اطلاعات محیطی صادر خواهد شد.

۲۶. اضافه بار در پردازش DCU/M، پیام MCU overload

پروسسور روی DCU/M دارای چند وظیفه در real time در اجرای هر سیکل دارد. اگر DCU/M به علت خرابی یا قطع تغذیه یا نیاز به تعویض برنامه دچار خطای اجرایی شود و یکی از این وظایف قبل از پایان سیکل انجام نشود پیغام خطایی به نام MCU overload در گروه سوم اطلاعات محیطی صادر خواهد شد.ولی حفاظتی فعال نمی‌شود، اما برنامه نظارت بر برد کنترل

DCU/M، که سیستم را تحت نظارت دارد اگر بارگذاری دوباره سیکل را حس نکرد رله فرمان تریپ خط را فرمان می دهد.

گروه بندی خطاهای منجر به برروز مسدود سازی جهت حفاظت

۱. نظارت بر اعداد سنسور جریان ۲ فاز، پیام Phase N current measurement failure

سنسور های جریان روی دو فازU,V نصب اند اعداد سنسور های جریان روی ۲ فاز و نتیجه فاز سوم دائما از نظر معقول بودن یا قطع نشدن سیم های سنسور و برد DCU/M نظارت و با هم مقایسه می‌شود انحراف سبب بروز مسدود سازی جهت حفاظت خواهد شد و اگر این حالت ۳ بار در ۳۰ دقیقه قبل رخ دهد این رخداد سبب بروز حفاظت ایزوله سازی شده و پیغام خطایی به نام Phase 1-2 current measurement failure در گروه اول اطلاعات محیطی صادر خواهد شد.

۲. حفاظت کاهش ولتاژ لینک DC:

کاهش ولتاژ لینک DCبه زیر حد کاری در حین کار کانورتر سبب بروز این حفاظت پیغام خطایی با نام DC link under voltage در گروه اول اطلاعات محیطی صادر خواهد شد.

۳. نظارت بر سنسور سرعت، پیام Speed sensor N failure

سنسور سرعت هر موتور به طور مستقل از نظر معقول بودن سرعت بررسی می‌شود اگر عدد سنسور سرعت یا کابل های آن یا DCU/M یا تغذیه سنسور

سرعت قطع شده باشد عکس العمل سریعی نخواهیم داشت بلکه عدد این سنسور نادیده گرفته می‌شود و سبب بروز این حفاظت پیغام خطایی به نام های Speed sensor 1,2 failure در گروه سوم اطلاعات محیطی صادر خواهد شد بعضی مواقع در قدرت رانش بیشینه یا در زمان ترمز این خطا رخ میدهد که ناشی از خطای نویز است. بعضی مواقع هم دنده های گیربکس سنسور خراب شده است. ثبت اطلاعات سنسور ها در خطایابی راهگشا است لذا همان طور که اشاره شد دیتا لاگری باید طراحی که بعضی اطلاعات محیطی را برای عیب یابی ثبت نماید.

۴. نظارت بر سنسور سرعت، پیام All speed sensors failed

اگر سنسور دوم هم دچار خطا شود مسدود سازی جهت حفاظت فعال می‌شود پیغام خطایی به نام All speed sensors failed در گروه سوم اطلاعات محیطی صادر خواهد شد.

۵. نظارت بر Traction safe، پیام Traction safe failure

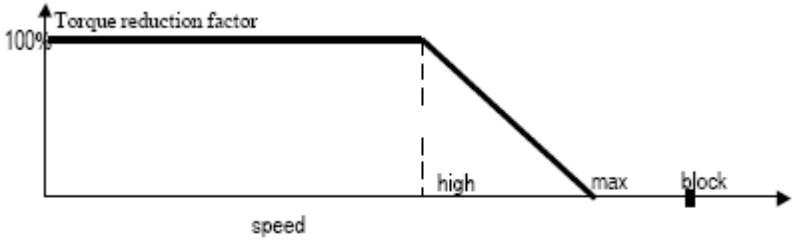
دو ورودی Traction safe در DCU/M وجود دارد، شرط ارسال سیگنال های نوری به گیت درایور ها صفر نبودن این دو فرمان با هم است. اگر VCU فرمان حرکت دهد و همزمان این دو فرمان صفر، یا همان low باشد حفاظت مسدود سازی جهت حفاظت فعال شده و پیغام خطایی به نام Traction safe failure در گروه یک اطلاعات محیطی صادر خواهد شد.

۶. نظارت بر Traction safe، پیام Traction safe loop failure

این حالت یک پیام هشدار است، اگر هر دو Traction safe در دو کانورتر فعال باشد Traction safe از کانورتر اول با Traction safe کانورتر دوم مقایسه شده و اگر بیش از ۳ ثانیه این دو با هم متفاوت باشند پیام بالا صادر خواهد شد.علت این خطا آن است که فرمان Traction safe از یکی از دو طرف به دلیل خرابی I/Oیا DCU/M یا کابل معنا دار نباشد یا ارتباط به DCU/M ها به این دلیل خطا دهد.

۷. اضافه سرعت موتور، پیام Motor overspeed

اگر سرعت قرائت شده از متوسط سرعت دو موتور بیش از مقدار خاصی اختلاف داشته باشد خطا صادر خواهد شد و نیروی رانش در شیب رو به صفر کاهش می یابد تا مقدار اختلاف به حد مجاز برسد. در هنگام رسیدن سرعت به حد (block) کانورتر سبب بروز حفاظت مسدود سازی شده و پیغام خطایی با نام Motor over speed در گروه اول اطلاعات محیطی صادر خواهد شد. این خطا، خطا ناشی از خرابی سنسور سرعت است.



۸. نظارت بر FPGA ، پیام FPGA Switching frequency failure

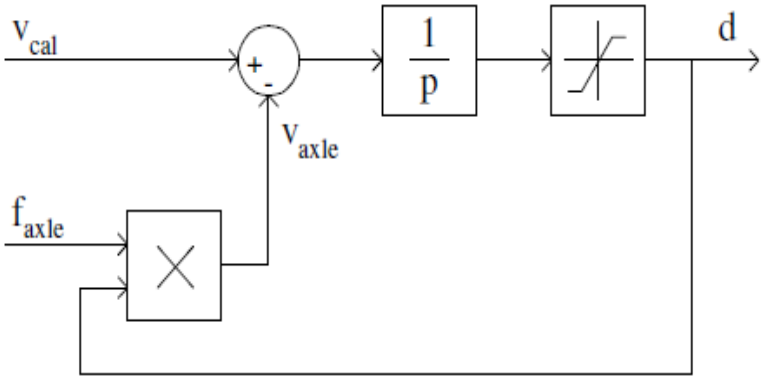
اگر فرکانس خروجی با فرکانس دستور داده شده برابر نباشد کانورتر مسدود سازی جهت حفاظت می‌شود. مشکل این بخش ناشی از خرابی DCU/M است.

۹. حفاظت سنسور های جریان ۲ فاز، پیام Overcurrent motor phase n

اعداد سنسور های جریان روی ۲ فاز و نتیجه فاز سوم اگر از حد مجاز عبور کند می‌تواند به معنای اتصال کوتاه باشد این حالت سبب بروز حفاظت مسدود سازی می‌شود البته امکان تشخیص اتصال کوتاه توسط گیت درایور نیز وجود دارد که سبب بروز همان نتیجه می‌شود و اگر این حالت ۳ بار در ۳۰ دقیقه قبل رخ دهد این رخداد سبب بروز حفاظت ایزوله سازی شده و پیغام خطایی به نام۱-۳ Over current motor phase در گروه اول اطلاعات محیطی صادر خواهد شد. بعضی اغتشاشات نذیر پریدن ترمز از روی اکسل چرخ می‌تواند سبب بروز این خطا گردد علاوه بر آن خرابی سنسور جریان، خرابی سنسور سرعت، اتصال کوتاه موتور یا کابل آن، خرابیGDU یا آرک در IGBT می توان سبب آن باشد.

۱۰. نظارت بر مقدار قطر چرخ، پیام MCM High wheel diameter difference

مقدار قطر چرخ دائما با مقایسه سرعت کالیبره شده قطار که VCU اعلام می‌دارد و سرعت محور به صورت زیر محاسبه خواهد شد قطر چرخ مجاز بین ۷۹۰ تا ۸۷۰ میلی متر است. مقدار قطر چرخ در حافظه دارای باتری پشتیبان ذخیره می‌شود.



اگر کلایبراسیون اندازه قطر چرخ در اکسل اول و اکسل دوم بیش از ۶ میلی متر اختلاف داشته باشد این پیام صادر خواهد شد.نتیجه آن کاهش ۱۰۰ به ۸۰ درصدی قدرت رانش است.

جهت رفع مشکل یک بار برای ۳۰ دقیقه و سرعت در حدود ۴۰ کیلومتر بر ساعت دوباره کالیبراسیون انجام شود و در صورت مشکل چرخ تعمیر یا تعویض گردد.

۱۱. نظارت بر مقدار قطر چرخ، پیام MCM Too high wheel diameter difference

اگر کلایبراسیون اندازه قطر چرخ در اکسل اول و اکسل دوم بیش از ۸ میلی متر اختلاف داشته باشد این پیام صادر خواهد شدکه نتیجه آن مسدود شدن کانورتر است. جهت رفع مشکل یک بار برای ۳۰ دقیقه و سرعت در حدود ۴۰ کیلومتر بر ساعت دوباره کالیبراسیون انجام شود و در صورت مشکل به چرخ تعمیر یا تعویض گردد.

۱۲. نظارت بر مقدار قطر چرخ، پیام MCM Wheel diameter axle 1 faulty

اگر اندازه چرخ خارج از رنج کالیبره ی ۸۷۰ تا ۷۹۰ میلی متر باشد این پیام صادر خواهد شد. نتیجه آن کاهش ۱۰۰ به ۵۰ درصدی قدرت رانش آن کانورتر است که احتمالا از اشکال سنسور سرعت یا کاهش زیاد قطر چرخ ناشی شده است. جهت عیب یابی سایر پیام های خطا درباره سنسور سرعت بررسی شود.

گروه بندی خطاهای منجر به برروز خاموش سازی جهت حفاظت

۱. نظارت بر عملکرد رله تریپ خط، پیام Line trip on-failure

اگر دستور بسته شدن رله صادر شود ولی فیدبک همچنان نشان دهنده باز بودن باشد پیغام خطایی به نام Line trip on-failure صادر می‌شود. و اقدام دیگری صورت نمی گیرد.

۲. نظارت بر عملکرد رله تریپ خط، پیام Line trip off-failure

اگر دستور باز شدن رله صادر شود ولی فیدبک همچنان نشان دهنده بسته بودن باشد سبب بروز خاموش سازی جهت حفاظت و پیغام خطایی به نام Line trip off-failure در گروه دوم اطلاعات محیطی صادر خواهد شد و اگر ۳ خطا در ۳۰ دقیقه قبل مشاهده کند کانورتر ایزوله می‌شود.علت خرابی DCU/M است.

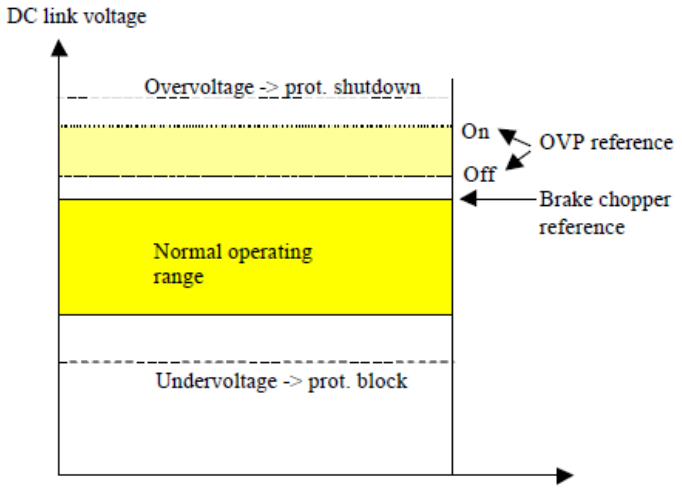
۳. وضعیت ماشین در نرم افزار DCU/M پیام Sequence fault state machine

این خطا، خطای نرم افزاری است وضعیت ماشین به صورت نرم افزاری بررسی می‌شود اگر بیشتر از یک حالت در هر لحظه برای ماشین تنظیم گردد، این خطا در هنگام برنامه ریزی تراشه ها امکان دارد رخ دهد و در حین کار اتفاق نباید بیفتند و سبب بروز خاموش سازی جهت حفاظت و پیغام خطایی به نام Sequence fault state machine در گروه سوم اطلاعات محیطی صادر خواهد شد.

۴. تست توابع حفاظت در برابر اضافه ولتاژ، پیام OVP self test failure

اگر ولتاژ لینک DCاز میزان بیشینه ۱۰۰۰ ولت افزایش یابد مقامت ها کامل در مدار می آیند تا ولتاژ به ۹۵۰ ولت برسد در حالت کنترل اضافه ولتاژ این چاپر به صورت PWMای کار نمی کند و حالت چاپری در هنگام ترمز رخ می دهد.

اگر اضافه ولتاژ لینک DCادامه یابد بلوک حفاظت فراخوانی و جهت حفاظت، کانورتر خاموش می شود.اگر در حین کار ولتاژ به زیر حد کمینه نیز برسد بلوک حفاظت فراخوانی می شود.



ولتاژ لینک DC	نوع رفتار حفاظت ولتاژ	وضعیت لینک DC
1050 V	protective shutdown	overvoltage level
1000 V		OVP on level
950 V		OVP off level
880 V		brake chopper activation level
450 V	protective blocking	under voltage level

هنگامی که ولتاژ لینک به ولتاژ از قبل تعیین شده ای رسید بلوک اضافه ولتاژ برای مدت زمان خاصی روشن می‌شود و انتظار آن است که ولتاژ به اندازه خاصی کاهش یابد اگر کاهش ولتاژ کمتر از مقدار مورد انتظار باشد این رخداد سبب بروز خاموش سازی جهت حفاظت شده و پیغام خطایی به نام OVP self test failure, branch 1 در گروه دوم اطلاعات محیطی صادر

خواهد شد. علت خطای مدارات حفاظت از GDU، مقاومت، کابل، فیبر نوری تا IGBT و DCU/M می‌تواند باشد.

۵. بالانس بودن توان، پیام Power balance fault

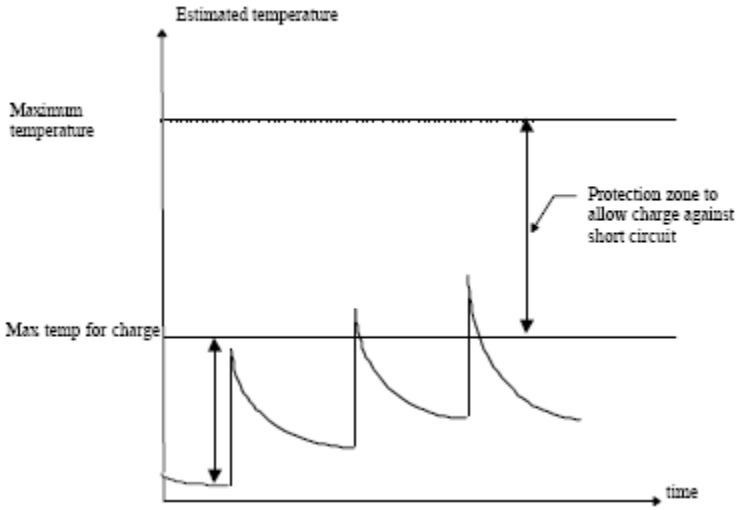
کانورتور همواره مقدار ضرب ولتاژ در جریان ارسالی به موتور و چاپر و خط تغذیه را محاسبه باید کند اگر توان ورودی و خروجی کانوتر بیشتر از حد مشخصی با هم تفاوت داشته باشد سبب بروز این رخداد در خاموش سازی جهت حفاظت شده و پیغام خطایی به نام Power balance fault در گروه دوم اطلاعات محیطی صادر خواهد شد.علت خطای سنسور جریان فاز یا سنسور ولتاژ لینک Dc یا خرابی کابل های مقاومت ترمزی و موتور یا خرابی مقاومت ترمزی است.

۶. نظارت بر اتصال MVB، پیام MVB communication failure

یک سگ نگهبان (watch dog) در VCUوجود دارد که اتصال MVB را دائما تحت نظارت دارد اگر به علت قطعی کابل، یا خرابی DCU/M در مسیر خواندن یا نوشتن پروتکل ارتباط قطعی حاصل شود یک شمارنده که مسئول شمارش داده های انتقالی است آن را حس می کند و اگر این شمارنده بالا نرود سگ نگهبان آن را تشخیص داده و خطای منجر به بروز خاموش سازی جهت حفاظت برای هر دو کانورتر در واگن می‌شود و پیغام خطایی با نام MVB communication failure در گروه سوم اطلاعات محیطی صادر خواهد شد.با این کار مشخص می شود اگر یکی از DCU/M ها خطا داده باشد یعنی آن DCU/M خراب است و اگر دو تا پیام داده باشد یعنی کابل دچار آسیب است.

۷. نظارت دمایی بر مقاومت شارژ

مقدار جریان شارژر تحت نظارت است اگر دمای مورد انتظار مقاومت از حد بیشینه دمای حین شارژ بالاتر رود (در محاسبات خنک سازی مقاومت ها هیچ فنی را فعال در نظر نمی گیریم) این رخداد سبب صدور خطایی به نام Charging resistor over temperature در گروه دوم اطلاعات محیطی می‌شود.



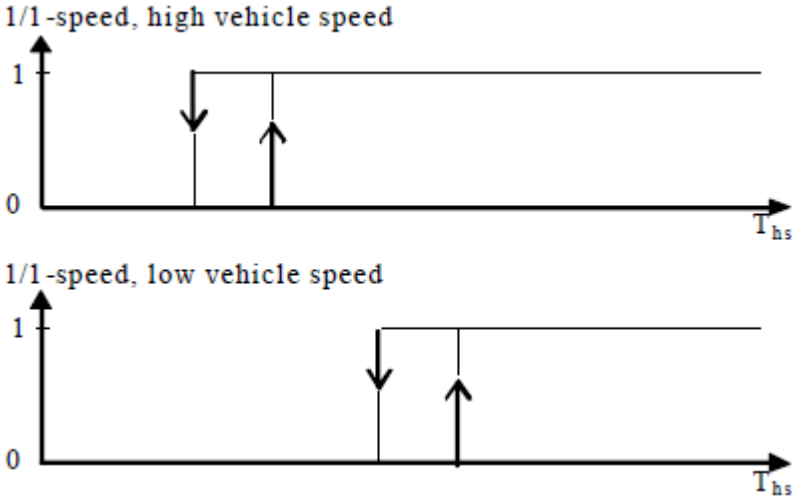
۸. نظارت بر فن داخلی

فن داخلی با تغذیه ۲۴ ولتی منبع تغذیه کار می کند و با شروع به کار کانورتر به کار می افتد.اگر تفاضل دمای سنسور دمای داخلی و سنسور دمای روی برد از مقدار مشخصی بیشتر شود فن خارجی به سرعت حداکثر انتقال

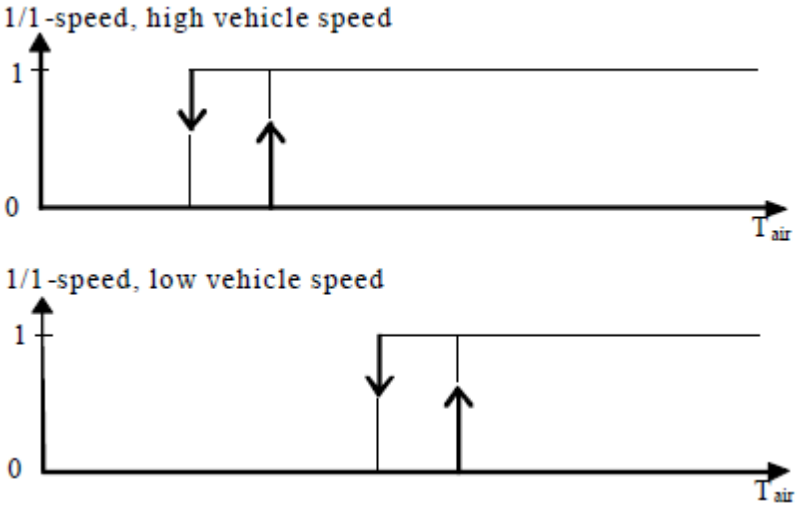
یافته و پیغام خطایی به نام Internal cooling problemدر گروه چهار اطلاعات محیطی صادر خواهد شد.

۹. نظارت بر فن خارجی

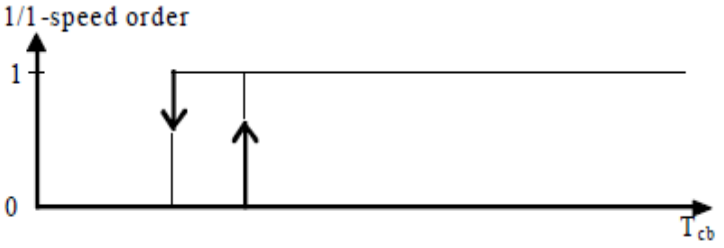
فن خارجی دو سرعته است و در حالت عادی در half speed کار می کند و تا ۶۰ ثانیه بعد از اتمام کار کانورتر نیز روشن می ماند و وقتی دمای هیت سینک بالا رود یا هوا داخلی داغ شود سرعت فن full speed می شود و اگر سرعت قطار از عددی خاصی کمتر شود بدین معنا است که به ایستگاه رسیده لذا سرعت را کم می شود ولی اگر سنسور دمای روی برد یا سنسور دمای هوای کانورتور یا سنسور دمای هیت سینک دچار خطا باشد سرعت همچنان full speed می ماند. رابطه بین دمای هیت سینک و سرعت قطار به صورت زیر است.



رابطه بین دمای هوای داخل و سرعت قطار به صورت زیر است.

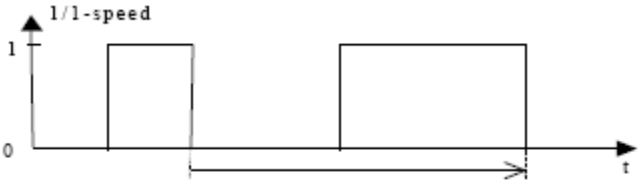


رابطه بین دمای سنسور روی برد DCU/M و سرعت قطار به صورت زیر است.



این دو سرعت از طریق ۲ فرمان خروجی دیجیتال اعمال شده و ۲ ورودی دیجیتال نیز بر عملکرد کنتاکتورهای این فن نظارت می کنند. این ۲ سرعت نمی تواند با هم اعمال گردد و در تغییر از یک سرعت به سرعت دیگر تاخیر عمل کنتاکتورها و کاهش سرعت فن در نظر گرفته شده است. پریود زمانی خاموش شدن های متوالی در یک سرعت خاص در فن ها مطابق شکل زیر در

هر دو سرعت محدود است و این شرط از شرایط دمایی دارای اولویت بالاتری است.



کنتاکتورهای هر دو سرعت تحت نظارت می باشند، فرمان سرعت زیاد یا کم با هم مقایسه شده و سیگنال پاسخ را می سازد. خطا وقتی رخ می دهد که فرمان و پاسخ مشابه هم نباشد.خطا با پیام پاک کردن خطا از طرف VCUرفع می‌شود. اگر سه فاز در دسترس نباشد یا هر دو سرعت فن خطا دهد رفرنس گشتاور کم می‌شود.

۱۰. خطای تمام سرعت فن، پیام External fan 1/1 speed fault

اگر فرمان روشن شدن تمام سرعت را داده باشیم و جواب ندهد فرمان نیم سرعت داده می‌شود. اگر آن هم جواب نداد تمام سرعت فرمان داده می‌شود.

اگر فرمان خاموش شدن تمام سرعت صادر شود و جواب ندهد فرمان تمام سرعت داده می‌شود.

علت می‌تواند کنتاکتورهای خراب، مشکلات سیم کشی، نبود تغذیه ACM، خرابی DCU/M و عملکردن حفاظت موتور فن باشد که سبب پیغام خطایی به نام External fan 1/1 speed fault در گروه دوم اطلاعات محیطی صادر خواهد شد.

۱۱. خطای هر دو سرعت فن، پیام External fan 1/2 and 1/1 speed fault

اگر هر دو سرعت فن کار نکند گشتاور کاهش و پیغام خطایی به نام External fan 1/2 and 1/1 speed fault در گروه دوم اطلاعات محیطی صادر خواهد شد.

۱۲. خطای نیم سرعت فن، پیام External fan 1/2 speed fault

اگر فرمان روشن شدن نیم سرعت باشد جواب ندهد فرمان تمام سرعت داده می‌شود. اگر فرمان خاموش شدن نیم سرعت باشد جواب ندهد فرمان نیم سرعت روشن داده می‌شود و پیغام خطایی به نام External fan 1/2 speed fault در گروه دوم اطلاعات محیطی صادر خواهد شد.

به طور خلاصه:

خطا	عملکرد
اگر دستور روشن شدن کم سرعت باشد و سیگنال پاسخ دریافت نشد	دستور سرعت زیاد داده می‌شود
اگر دستور روشن شدن تمام سرعت باشد و سیگنال پاسخ دریافت نشد	فرمان کم سرعت داده می‌شود اگر پاسخ آمد به صورت دائم کم سرعت فرمان داده می‌شود و گرنه دائما پر سرعت فرمان داده می‌شود.
اگر دستور خاموشی کم سرعت باشد و سیگنال پاسخ دریافت نشد	دائما پر سرعت فرمان داده می‌شود.

اگر دستور خاموشی سرعت زیاد باشد و سیگنال پاسخ دریافت نشد	دائما پر سرعت فرمان داده می‌شود.

۱۳. خطای مکانیکی سیستم، پیام Mechanical drive system fault

اگر فاز بندی موتور یا کابل موتور دچار اشکال باشد یا شفت و یا محور آن قفل شده باشد جریان واقعی موتور و جریان پیش بینی محاسبه شده با هم تفاوت خواهند داشت این خطا سبب بروز خاموش سازی جهت حفاظت شده و اگر این حالت ۳ بار در ۳۰ دقیقه قبل رخ دهد منجر به بروز ایزوله سازی کانورتر و پیغام خطایی به نام Mechanical drive system fault صادر خواهد شد.

۱۴. نظارت بر کنتاکتور شارژر، پیام Charging contactor opening / closing failure

فیدبک از این کنتاکتورها نظارت می‌شود اگر به علت خرابی رله یا خرابی کنتاکتور یا خرابی کابل ارتباطی یا خرابی DX، کنتاکتور درست بسته یا باز نشده باشد یا بر خلاف فرمان باز یا بسته بماند سبب خاموش سازی جهت حفاظت شده و اگر این حالت ۳ بار در ۳۰ دقیقه قبل رخ دهد منجر به بروز ایزوله سازی کانورتر و پیغام خطاهایی به نام Charging contactor opening/closing failure در گروه اول اطلاعات محیطی صادر خواهد شد.

۱۵. نظارت بر کنتاکتور جدا ساز، پیام Separation contactor opening/ closing failure

فیدبک از این کنتاکتور نظارت می‌شود اگر کنتاکتور به علت خرابی رله یا خرابی کنتاکتور یا خرابی کابل ارتباطی یا خرابی DX درست بسته یا باز نشده

باشد یابر خلاف فرمان باز یا بسته بماند سبب بروز خاموش سازی جهت حفاظت شده و اگر این حالت ۳ بار در ۳۰ دقیقه قبل رخ دهد منجر به بروز ایزوله سازی کانورتر و پیغام خطاهایی به نام های Separation contactor opening/closing failure در گروه دوم اطلاعات محیطی صادر خواهد شد.

۱۶. نظارت بر شارژ لینک DC، پیام DC-link charging failure

اگر فرمان بسته شدن کنتاکتور شارژر آمد ولی ولتاژ لینک DC در طی مدت زمان مورد انتظار بالا نرفت سبب بروز خاموش سازی جهت حفاظت شده و کنتاکتور شارژر باز می‌شود و اگر این حالت ۳ بار در ۳۰ دقیقه قبل رخ دهد منجر به بروز ایزوله سازی کانورتر و پیغام خطایی به نام DC link charging failure در گروه دوم اطلاعات محیطی صادر خواهد شد.

۱۷. نظارت بر دشارژ لینک DC، پیام DC-link discharging failure

قبل از دشارژ باید کنتاکتور جدا کننده باید باز باشد و چاپر با پریود خاصی باید روشن شود در این صورت طی زمانی مشخص ولتاژ لینک DC باید صفر شود. حال به علت مشکل خرابی GDUیا خرابی مقاومت ترمزی، خرابی کنتاکتور شارژر یا کنتاکتور جدا ساز یا سیستم نظارت آن، یا خرابی کابل های مقاومت ترمزی یا خرابی DCU/M که ترتیب و توالی عمل دشارژ را درست پیاده نکرده ولتاژ غیر صفر خواهد شد یا اگر جریان به حدی زیاد است که کنتاکتور نمی‌تواند آن را قطع کند لذا سبب بروز خاموش سازی جهت حفاظت می‌شود و اگر این حالت ۳ بار در ۳۰ دقیقه قبل رخ دهد منجر به بروز ایزوله سازی کانورتر و پیغام خطایی به نام DC link discharging failureدر گروه دوم اطلاعات محیطی صادر خواهد شد.

۱۸. حفاظت اضافه ولتاژ، پیام DC-link overvoltage

اگر با آمدن مدارات حفاظت از اضافه ولتاژ باز اضافه ولتاژ تا حد بیشینه حفاظت خاموش سازی 1050 V بالا رود این حالت سبب بروز خاموش سازی جهت حفاظت شده و اگر این حالت ۳ بار در ۳۰ دقیقه قبل رخ دهد منجر به بروز ایزوله سازی کانورتر و پیغام خطایی به نام DC link Overvoltage در گروه اول اطلاعات محیطی صادر خواهد شد. لذا باید به دنبال خطای در سنسور ولتاژ DC و خطاهای مرتبط با اضافه ولتاژ در سیستم عیب یابی بگردیم.

۱۹. حفاظت سنسور های جریان لینک DC، پیام DC-link input overcurrent

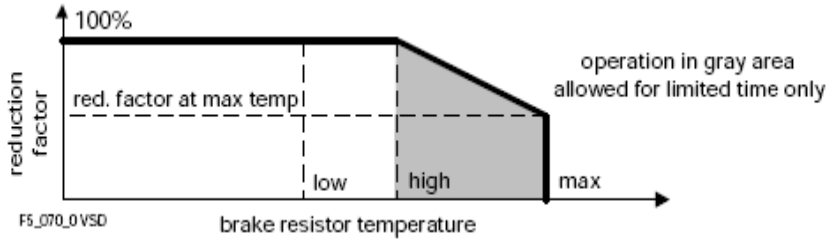
اعداد سنسور های جریان روی لینک DC اگر بیش از ۱۵۰۰ آمپر را حس کند می‌تواند به معنای اتصال کوتاه باشد این حالت سبب بروز خاموش سازی جهت حفاظت شده و اگر این حالت ۳ بار در ۳۰ دقیقه قبل رخ دهد این رخداد سبب بروز حفاظت ایزوله سازی شده و پیغام خطایی به نام DC link input over current در گروه دوم اطلاعات محیطی صادر خواهد شد.

خرابی سنسور جریان یا خرابی IGBTیا GDU یا اتصال کوتاه کانورتر یا اتصال کوتاه لینک DC علت بروز مشکل است.

۲۰. نظارت بر مقاومت ترمزی، پیام Brake resistor overtemperature

اگر دما بیش از حد افزایش یابد حفاظت خاموش سازی فعال و مشخصات جهت کاهش رفرنس گشتاور انتشار می یابد. اگر این حالت ۳ بار در ۳۰ دقیقه

قبل رخ دهد این رخداد سبب بروز حفاظت ایزوله شدن شده پیغام خطایی به نام Brake resistor over temperature در گروه دوم اطلاعات محیطی صادر خواهد شد. اشکال در سنسور ولتاژ DCیا افزایش تعداد رخداد اضافه ولتاژ ها و همزمان افزایش ترمز الکتریکی سبب بروز این خطا است. برای رفع مشکل مقدار دمای مقاومت را در سیستم ثبت اطلاعات نگاه کرده و به دنبال خطای در سنسور ولتاژ DC در سیستم عیب یابی بگردید.



۲۱. نظارت بر سنسور جریان لینکDC ، پیام DC-link input current measurement failure

اعداد سنسور های جریان روی لینک DC از نظر منطقی بودن نظارت می‌شود خرابی سنسور یا DCU/Mیا کابل های آن خود را با نشان دادن عدد جریان لینکDC خارج از رنج نشان می دهد که سبب بروز خاموش سازی جهت حفاظت شده و لذا اگر این حالت ۳ بار در ۳۰ دقیقه قبل رخ دهد این رخداد سبب بروز حفاظت ایزوله سازی شده و پیغام خطایی به نام DC link input current measurement failure در گروه دوم اطلاعات محیطی صادر خواهد شد.

۲۲. نظارت بر سیگنال های برگشتی از گیت درایور ها، پیام GDU feedback failure of IGBT

اگر سیگنال بازگشتی از GDU با فرمان ارسالی به علت خرابی فیبر نوری، خرابی کابل های بین GDU وDCU/M ، خرابیGDU، IGBT DCU/M تطابق نداشته باشد این رخداد سبب بروز خاموش سازی جهت حفاظت شده باشد در این صورت اگر این حالت ۳ بار در ۳۰ دقیقه قبل رخ دهد این رخداد سبب بروز مسدود سازی جهت حفاظت شده و پیغام خطای ۰۰ تا ۰۵ به نام GDU feedback failure of IGBT 1-6 در گروه اول اطلاعات محیطی صادر خواهد شد.

۲۳. نظارت بر سیگنال های برگشتی از گیت درایور های اضافه ولتاژ و ترمز

می‌تواند به علت خطا سیگنال برگشتی از گیت درایور با دیگر سیگنال ها مطابقت نداشته باشد. لذا این رخداد سبب بروز خاموش سازی جهت حفاظت شده باشد در این صورت اگر این حالت ۳ بار در ۳۰ دقیقه قبل رخ دهد این رخداد سبب بروز مسدود سازی جهت حفاظت شده و پیغام خطایی به نام GDU feedback failure of IGBT 7-8 در گروه اول اطلاعات محیطی صادر خواهد شد.

۲۴. تشخیص خطای اتصال به زمین، پیام Earth fault

جریان های فاز +DC و -DCبررسی می‌شود تا خطای اتصال به زمین را در موتور، کابل های موتور، کابل های مقاومت ترمزی، یا مقاومت ترمزی و MCM را سیستم کشف کند هرگونه نابربری این دو جریان به معنای اتصال

به زمین یا خرابی سنسور DCتفاضلی یا خرابی DCU/M است، این رخداد سبب بروز خاموش سازی جهت حفاظت می‌شود، اگر این حالت ۳ بار در ۳۰ دقیقه قبل رخ دهد این رخداد سبب بروز مسدود سازی جهت حفاظت شده و پیغام خطایی به نامEarth fault در گروه اول اطلاعات محیطی صادر خواهد شد.