STM32 MC SDK5.0 培训

2018/06

MCU Application

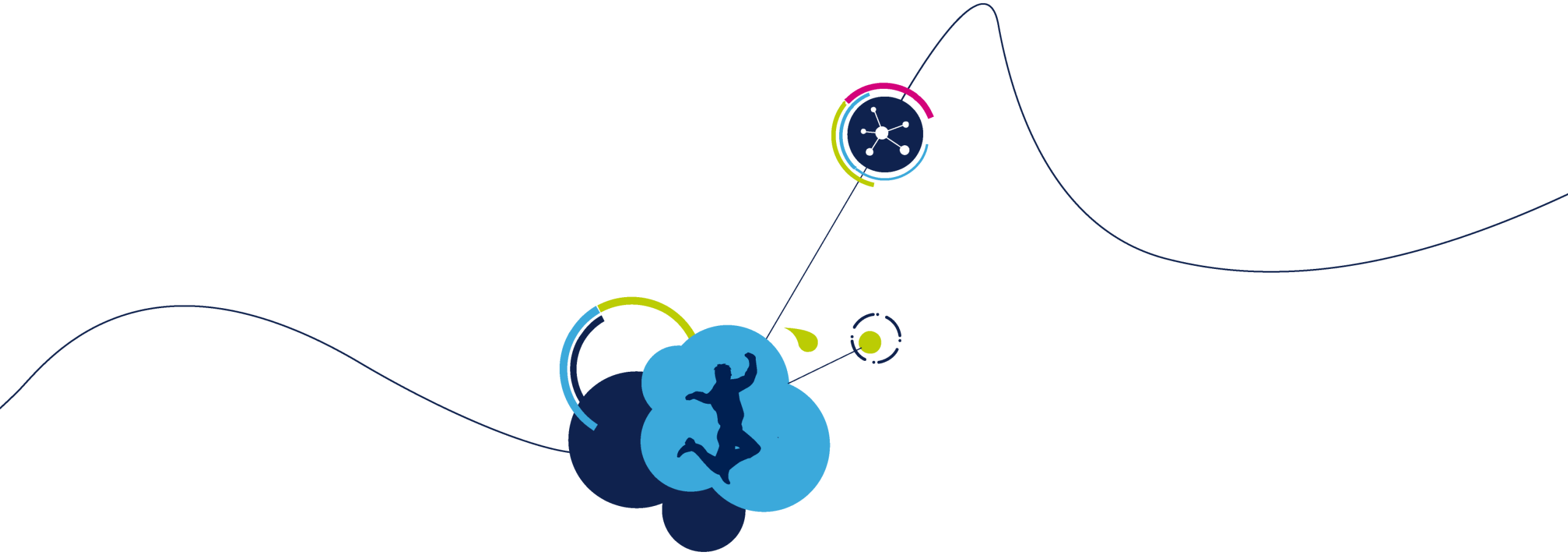
Motor Control LAB

life.augmented

内容

- 上午
  1.MC SDK5.0算法的理论基础 (1小时)
  2.MC SDK5.0详解 (1小时)
  3.实验1：MC SDK5.0 电动机参数识别 (45分钟)
- 下午
  4.MC SDK5.0 工具链及图形用户界面 (1小时)
  5.MC SDK5.0 评估硬件 (15分钟)
  6.实验2：基于MC SDK5.0 API，速度控制与电机启动停止 (20分钟)
  7.实验3：基于MC SDK5.0 PI 组件接口函数做在线参数修改 (20分钟)
  8.实验4：基于MC SDK5.0 状态的切换 (20分钟)
  9.实验5：开放性试验 (20分钟)
  10.实验总结及问答 (45分钟)

life.augmented



1.MC SDK5.0 算法的理论基础

life.augmented

MC SDK5.0算法概述

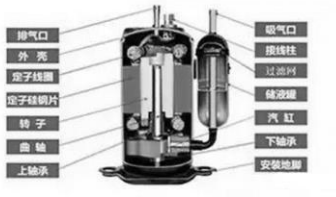
- 目标电机：三相永磁同步电动机（直流无刷电动机）
- 控制方法：矢量控制
- 三相PWM输出方法： SVPWM
- 相电流检测方法：
  - 单电阻电流检测和重构方式
  - 三电阻电流检测和重构方式
  - 隔离型电流传感器检测电流（DCCT或者ACCT）
- 转子位置检测方法
  - 霍尔效应位置传感器
  - 光电增量编码器
  - 无位置传感器的转子速度和位置估计算法
    - 基于估计感应电压的转子位置和速度估计算法
    - 基于高频载波注入的转子位置和估计算法

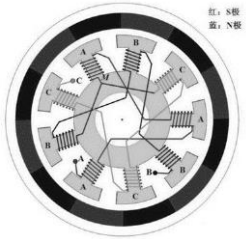
life.augmented

三相永磁同步电动机(直流无刷电动机)

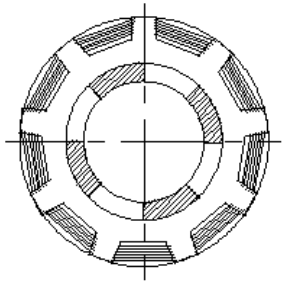
永磁同步电动机
(直流无刷电动机）

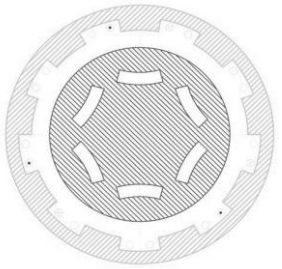
- 转子
  - 内转子/外转子



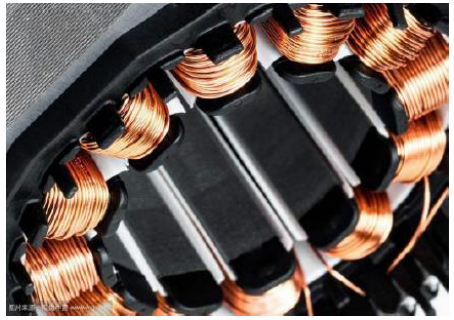


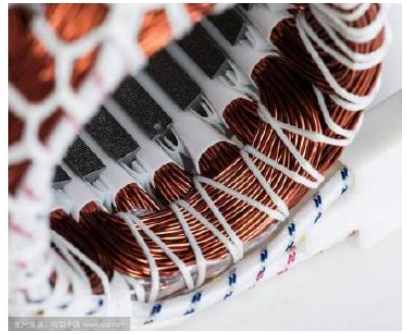
  - 表面贴装磁石或内嵌式磁石





- 绕组(定子) —— 集中绕组/分布绕组





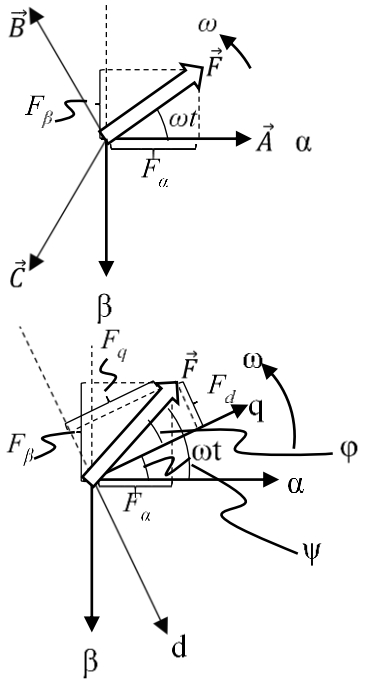
MCSDK5.0应用的永磁同步电动机的数学模型

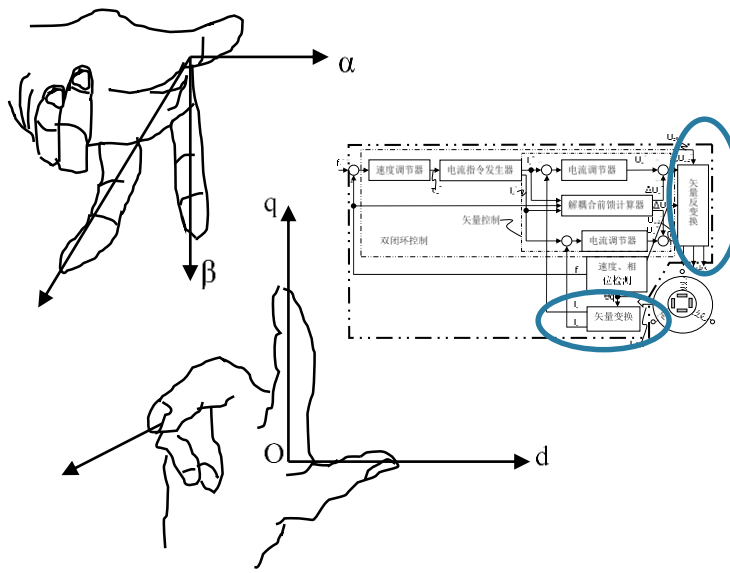
名称	方程式
电压方程	$\begin{bmatrix} U_d \\ U_q \end{bmatrix} = \begin{bmatrix} r + \frac{d}{dt} L_d & -L_q \omega_e \\ L_d \omega_e & r + \frac{d}{dt} L_q \end{bmatrix} \begin{bmatrix} I_d \\ I_q \end{bmatrix} + \begin{bmatrix} 0 \\ k_E \omega_e \end{bmatrix}$
转矩方程	$\tau_e = \frac{3}{2} p \left[k_E + \left(L_d - L_q \right) I_d \right] I_q$
弱磁控制条件	$\left(L_q I_q \right)^2 + \left(k_E + L_d I_d \right)^2 \leq \frac{{U_{1-limit}}^2}{{\omega_e}^2}$
动力学方程	$\tau_e - \tau_L = J \frac{d\omega_r}{dt} + B \omega_r$

life.augmented

矢量控制 — 矢量变换

变换内容		变换前后幅值不变	变换前后瞬时功率不变
从三相到两相的变换	电压	$\begin{bmatrix}u_\alpha\\u_\beta\end{bmatrix}=\begin{bmatrix}1 & 0\\ \frac{1}{\sqrt{3}} & \frac{2}{\sqrt{3}}\end{bmatrix}\begin{bmatrix}u_A\\u_B\end{bmatrix}$	$\begin{bmatrix}u_\alpha\\u_\beta\end{bmatrix}=\begin{bmatrix}\sqrt{\frac{3}{2}} & 0\\ \frac{1}{\sqrt{2}} & \frac{2}{\sqrt{2}}\end{bmatrix}\begin{bmatrix}u_A\\u_B\end{bmatrix}$
	电流	$\begin{bmatrix}i_\alpha\\i_\beta\end{bmatrix}=\begin{bmatrix}1 & 0\\ \frac{1}{\sqrt{3}} & \frac{2}{\sqrt{3}}\end{bmatrix}\begin{bmatrix}i_A\\i_B\end{bmatrix}$	$\begin{bmatrix}i_\alpha\\i_\beta\end{bmatrix}=\begin{bmatrix}\sqrt{\frac{3}{2}} & 0\\ \frac{1}{\sqrt{2}} & \frac{2}{\sqrt{2}}\end{bmatrix}\begin{bmatrix}i_A\\i_B\end{bmatrix}$
从两相到三相的变换	电压	$\begin{bmatrix}u_A\\u_B\\u_C\end{bmatrix}=\begin{bmatrix}1 & 0\\ -\frac{1}{2} & -\frac{\sqrt{3}}{2}\\ -\frac{1}{2} & \frac{\sqrt{3}}{2}\end{bmatrix}\begin{bmatrix}u_\alpha\\u_\beta\end{bmatrix}$	$\begin{bmatrix}u_A\\u_B\\u_C\end{bmatrix}=\sqrt{\frac{2}{3}}\begin{bmatrix}1 & 0\\ -\frac{1}{2} & -\frac{\sqrt{3}}{2}\\ -\frac{1}{2} & \frac{\sqrt{3}}{2}\end{bmatrix}\begin{bmatrix}u_\alpha\\u_\beta\end{bmatrix}$
	电流	$\begin{bmatrix}i_A\\i_B\\i_C\end{bmatrix}=\begin{bmatrix}1 & 0\\ -\frac{1}{2} & -\frac{\sqrt{3}}{2}\\ -\frac{1}{2} & \frac{\sqrt{3}}{2}\end{bmatrix}\begin{bmatrix}i_\alpha\\i_\beta\end{bmatrix}$	$\begin{bmatrix}i_A\\i_B\\i_C\end{bmatrix}=\sqrt{\frac{2}{3}}\begin{bmatrix}1 & 0\\ -\frac{1}{2} & -\frac{\sqrt{3}}{2}\\ -\frac{1}{2} & \frac{\sqrt{3}}{2}\end{bmatrix}\begin{bmatrix}i_\alpha\\i_\beta\end{bmatrix}$
瞬时功率		$p_3=\frac{3}{2}p_2$	$p_3=p_2$
电磁力矩		$\tau_e=\frac{3}{2}p\left[k_E+(L_d-L_q)I_d\right]I_q$	$\tau_e=p\left[k_E+(L_d-L_q)I_d\right]I_q$



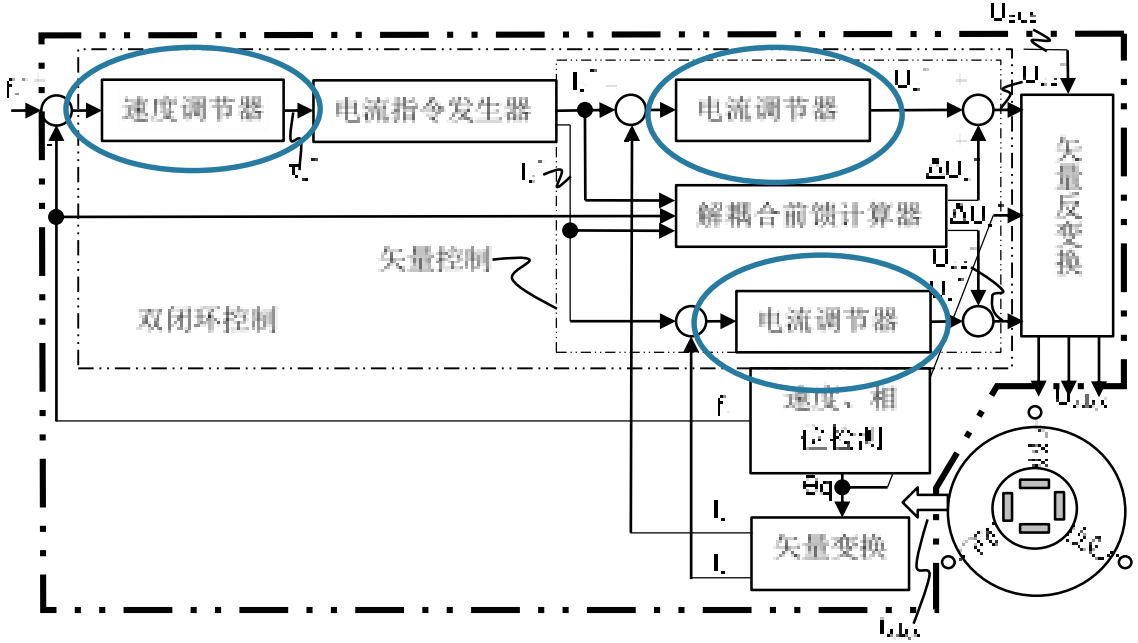


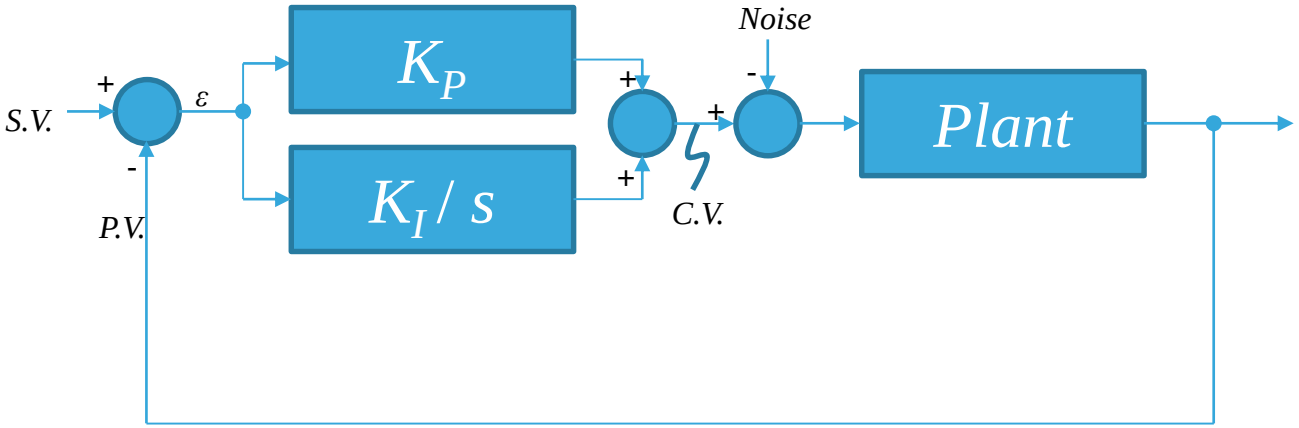
项目	电压	电流
αβ - dq	$\begin{bmatrix}U_d\\U_q\end{bmatrix}=\begin{bmatrix}\cos(\omega t) & \sin(\omega t)\\ -\sin(\omega t) & \cos(\omega t)\end{bmatrix}\begin{bmatrix}u_\beta\\u_\alpha\end{bmatrix}$	$\begin{bmatrix}i_\alpha\\i_\beta\end{bmatrix}=\begin{bmatrix}\cos(\omega t) & \sin(\omega t)\\ -\sin(\omega t) & \cos(\omega t)\end{bmatrix}\begin{bmatrix}I_q\\I_d\end{bmatrix}$
dq - αβ	$\begin{bmatrix}u_\alpha\\u_\beta\end{bmatrix}=\begin{bmatrix}\cos(\omega t) & \sin(\omega t)\\ -\sin(\omega t) & \cos(\omega t)\end{bmatrix}\begin{bmatrix}U_q\\U_d\end{bmatrix}$	$\begin{bmatrix}I_d\\I_q\end{bmatrix}=\begin{bmatrix}\cos(\omega t) & \sin(\omega t)\\ -\sin(\omega t) & \cos(\omega t)\end{bmatrix}\begin{bmatrix}i_\beta\\i_\alpha\end{bmatrix}$

- 变换前后功率和幅值都不改变。
- 变换矩阵可以使用同一个矩阵，但是输入量需要逆序输入，输出量需要顺序输出

life.augmented

矢量控制 — 双闭环控制器

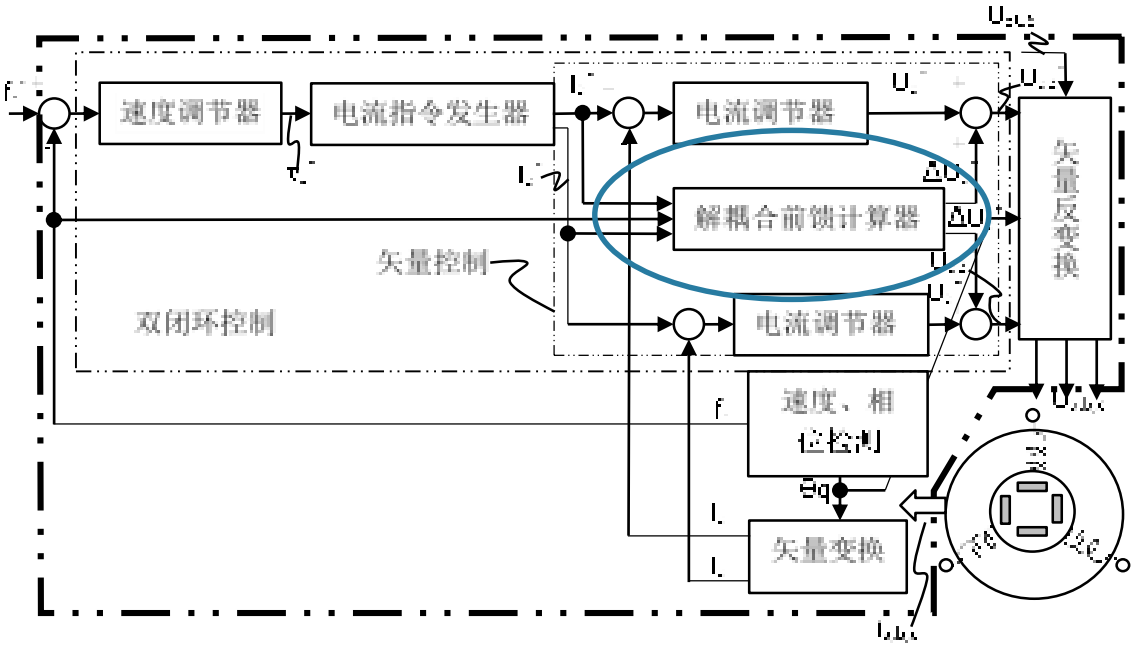




控制器	设定值S.V.	实际值 P.V.	控制值C.V.	对象传递函数
速度控制器	$f_{r\text{-}ref}(f_r^*)$	$f_{r\text{-}fb}(f_r)$	$\tau_{ref}(\tau^*)$ or $I_{q\text{-}ref}(I_q^*)$	$\frac{1}{Js+B}$
电流控制器（忽略dq之间的耦合）	$I_{d/q\text{-}ref}(I_{d/q}^*)$	$I_{d/q\text{-}fb}(I_{d/q})$	$U_{d/q\text{-}ref}(U_{d/q}^*)$	$\frac{1}{L_{d/q}s+r}$

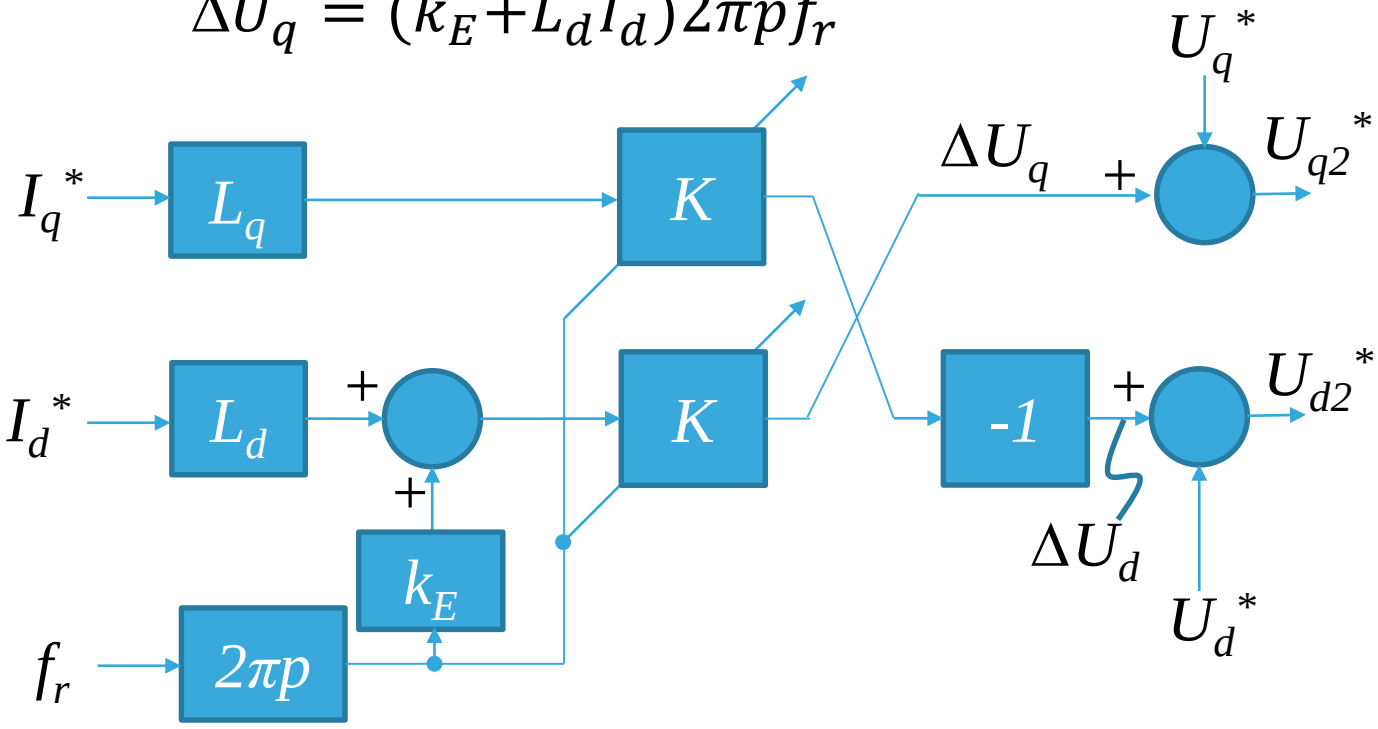
life.augmented

矢量控制 — 电流前馈



$$\Delta U_d = -L_q I_q 2\pi p f_r$$

$$\Delta U_q = (k_E + L_d I_d) 2\pi p f_r$$

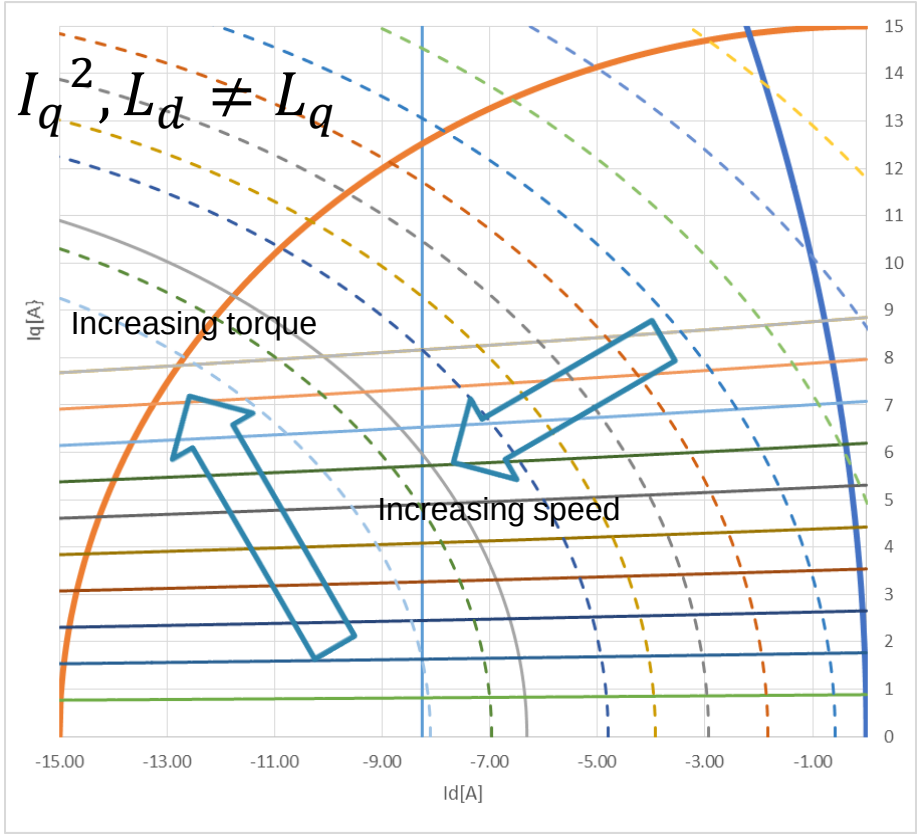


矢量控制 — MTPA与弱磁控制

$$\text{MTPA: } I_d = \begin{cases} -\frac{k_E}{2(L_d - L_q)} + sign(L_d - L_q)\sqrt{\left(\frac{k_E}{2(L_d - L_q)}\right)^2 + I_q^2}, L_d \neq L_q \\ 0, L_d = L_q \end{cases}$$

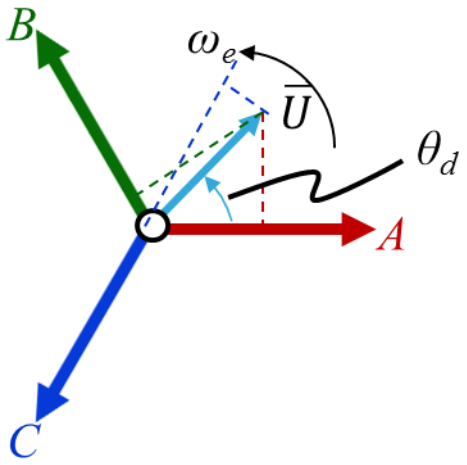
$$sign(x) = \begin{cases} 1, x > 0 \\ 0, x = 0 \\ -1, x < 0 \end{cases}$$

$$\text{弱磁: } I_d = \begin{cases} -\frac{k_E}{L_d} + \frac{\sqrt{\left(\frac{U_{1-limit}}{\omega_e}\right)^2 - (L_q I_q)^2}}{L_d}, \frac{U_{1-limit}}{\omega_e} \geq L_q I_d \\ n/a, \frac{U_{1-limit}}{\omega_e} < L_q I_d \end{cases}$$



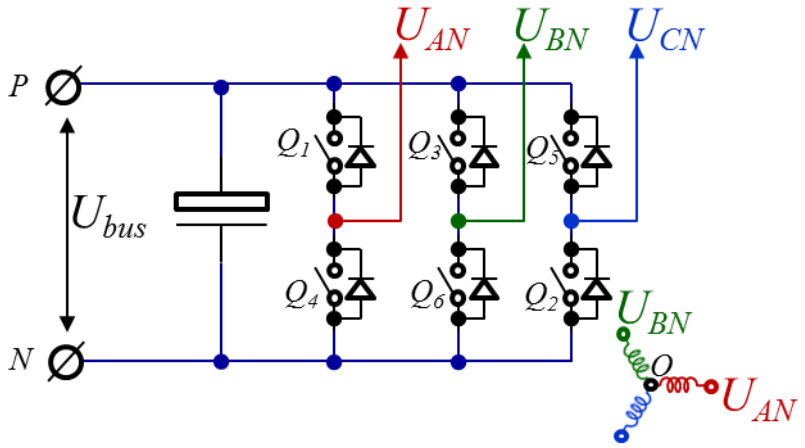
life.augmented

矢量控制 — SVPWM(1/3)



$$U_{AO} = U_m \cos(\theta_d) = U_m \cos(\omega_e t)$$
$$U_{BO} = U_m \cos(\theta_d - 120^\circ) = U_m \cos(\omega_e t - 120^\circ)$$
$$U_{CO} = U_m \cos(\theta_d + 120^\circ) = U_m \cos(\omega_e t + 120^\circ)$$

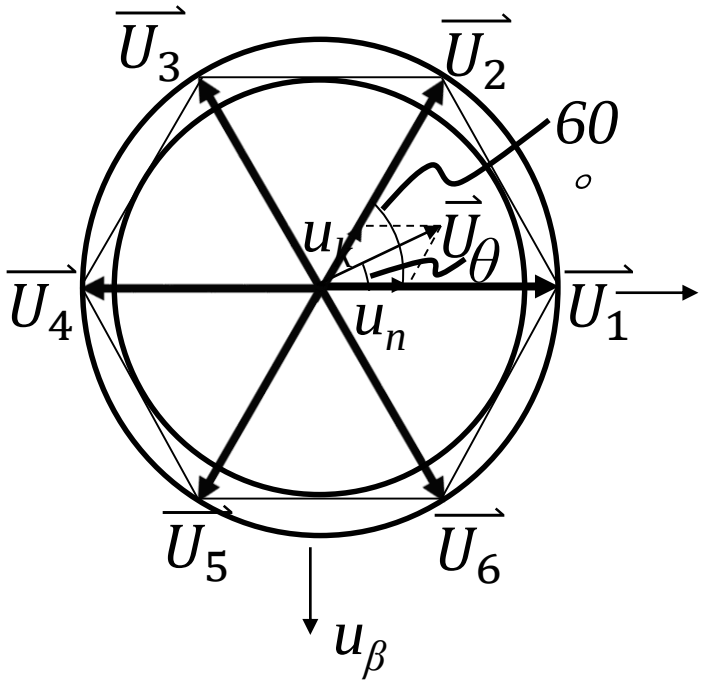
$$|\vec{U}| = U_m$$



uector	$Q_1(Q_4)$	$Q_3(Q_6)$	$Q_5(Q_2)$	U_{AN} [u]	U_{BN}[u]	U_{CN}[u]	U_{AO} [u]	U_{BO}[u]	U_{CO}[u]
$\overrightarrow{U_0}$	OFF(ON)	OFF(ON)	OFF(ON)	0	0	0	0	0	0
$\overrightarrow{U_1}$	ON(OFF)	OFF(ON)	OFF(ON)	U_{bus}	0	0	$2U_{bus}/3$	$-U_{bus}/3$	$-U_{bus}/3$
$\overrightarrow{U_2}$	ON(OFF)	ON(OFF)	OFF(ON)	U_{bus}	U_{bus}	0	$U_{bus}/3$	$U_{bus}/3$	$-2U_{bus}/3$
$\overrightarrow{U_3}$	OFF(ON)	ON(OFF)	OFF(ON)	0	U_{bus}	0	$-U_{bus}/3$	$2U_{bus}/3$	$-U_{bus}/3$
$\overrightarrow{U_4}$	OFF(ON)	ON(OFF)	ON(OFF)	0	U_{bus}	U_{bus}	$-2U_{bus}/3$	$U_{bus}/3$	$U_{bus}/3$
$\overrightarrow{U_5}$	OFF(ON)	OFF(ON)	ON(OFF)	0	0	U_{bus}	$-U_{bus}/3$	$-U_{bus}/3$	$2U_{bus}/3$
$\overrightarrow{U_6}$	ON(OFF)	OFF(ON)	ON(OFF)	U_{bus}	0	U_{bus}	$U_{bus}/3$	$-2U_{bus}/3$	$-U_{bus}/3$
$\overrightarrow{U_7}$	ON(OFF)	ON(OFF)	ON(OFF)	U_{bus}	U_{bus}	U_{bus}	0	0	0

life.augmented

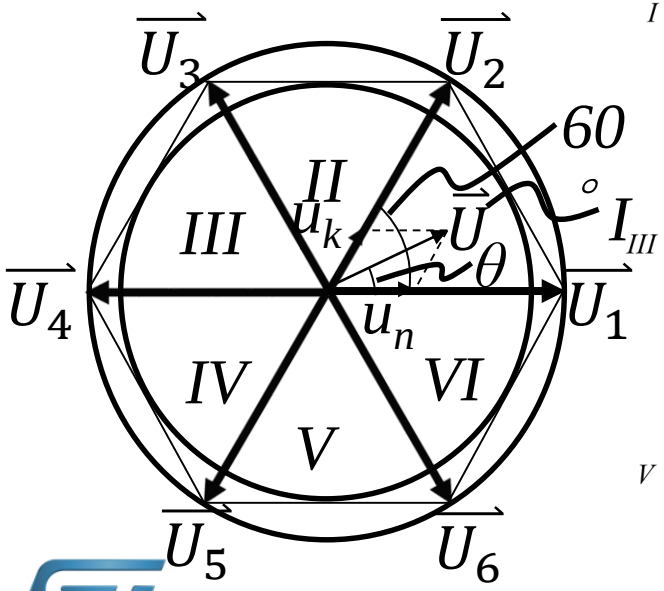
矢量控制 — SVPWM(2/3)

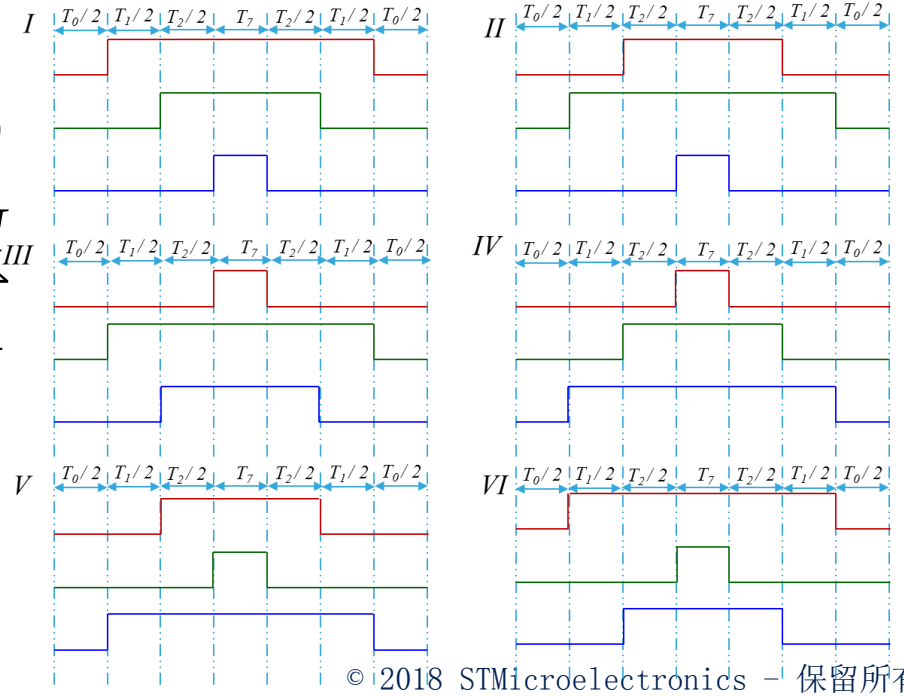


$$\begin{cases} \vec{U}T = T_1\overrightarrow{U_n} + T_2\overrightarrow{U_k} \\ T \geq T_1 + T_2 \end{cases}$$

Sector	I	II	III	IV	V	VI
n	1	3	3	5	5	1
k	2	2	4	4	6	6

$$\frac{|\vec{U}|}{\sin(180° - 60°)} = \frac{u_{n\ or\ k}}{\sin(60° - \theta)} = \frac{u_{k\ or\ n}}{\sin\theta}, 0° \leq \theta \leq 60°$$





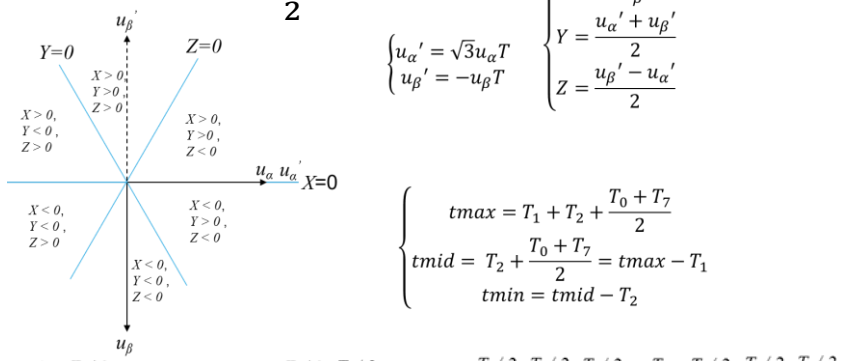
$$T = T_0 + T_1 + T_2 + T_7$$

$$\begin{cases} u_\alpha = |\vec{U}|\cos(60°(sector\ no. - 1) + \theta) \\ u_\beta = -|\vec{U}|\sin(60°(sector\ no. - 1) + \theta) \end{cases}$$

life.augmented

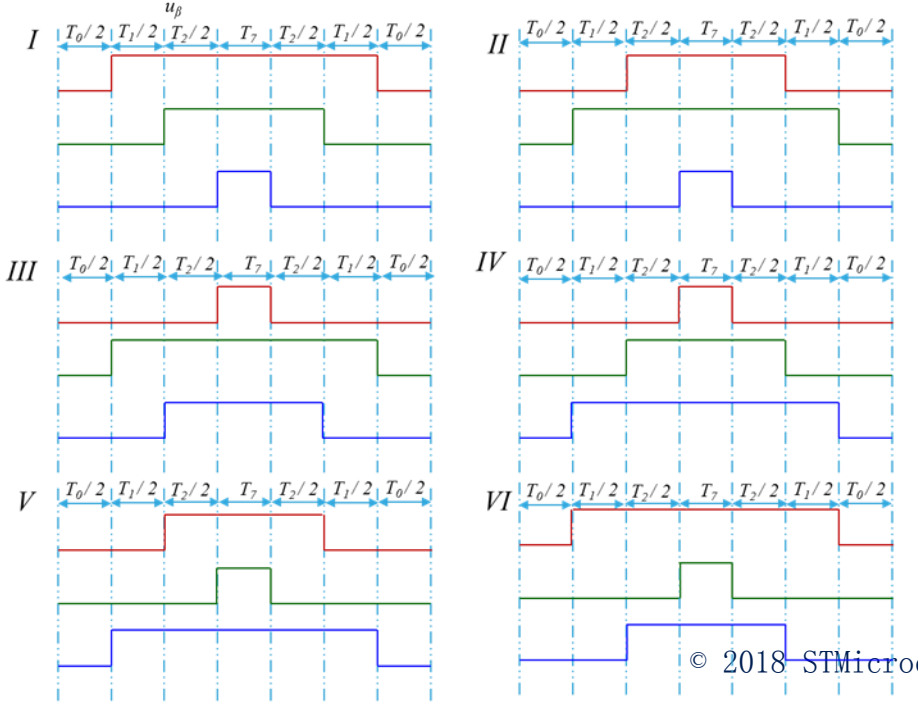
矢量控制 — SVPWM(3/3)

By normalization, if $\overrightarrow{U'} = \frac{\vec{U}}{\frac{\sqrt{3}}{2}}$, because, if $\left|\overrightarrow{U_n}\right| = 1, n = 1,2,3,4,5,6$;then $\left|\vec{U}\right|_{max} = \frac{\sqrt{3}}{2}$,therefore, $\left|\overrightarrow{U'}\right| = 1$.



$$\begin{cases} u_\alpha' = \sqrt{3}u_\alpha T \\ u_\beta' = -u_\beta T \end{cases} \qquad \begin{cases} X = u_\beta' \\ Y = \frac{u_\alpha' + u_\beta'}{2} \\ Z = \frac{u_\beta' - u_\alpha'}{2} \end{cases}$$

$$\begin{cases} tmax = T_1 + T_2 + \frac{T_0 + T_7}{2} \\ tmid = T_2 + \frac{T_0 + T_7}{2} = tmax - T_1 \\ tmin = tmid - T_2 \end{cases}$$

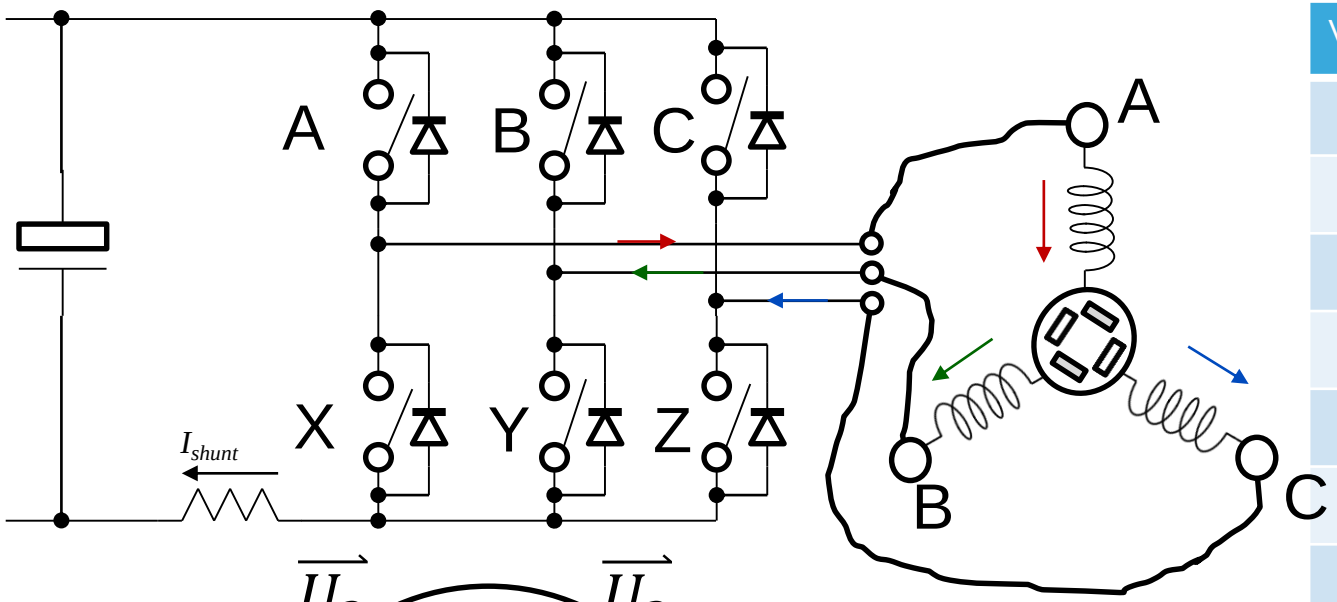


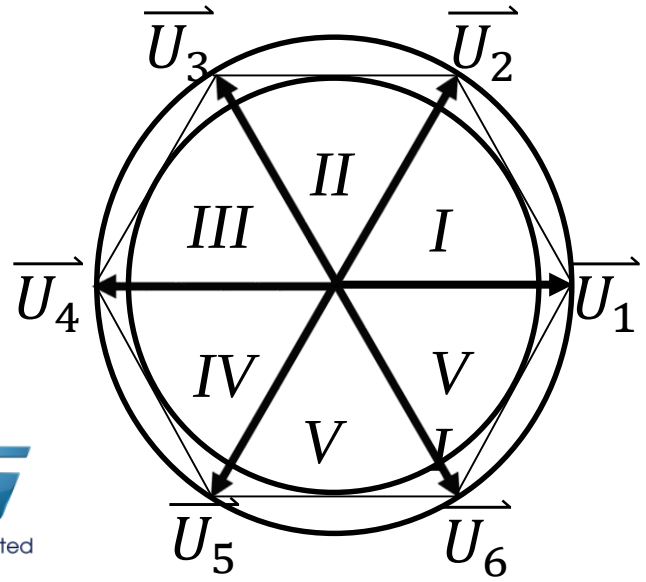
sector	T1	T2	T0+T7
I	*-Z*	*X*	*T + Z – X*
II	*Z*	*Y*	*T – Y – Z*
III	*X*	*-Y*	*T – X + Y*
IV	*-X*	*Z*	*T + X – Z*
V	*-Y*	*-Z*	*T + Y + Z*
VI	*Y*	*-X*	*T + X – Y*

sector	tA	tB	tC
I	*T/2 + (X – Z)/2*	*tA + Z*	*tB – X*
II	*T/2 + (Y – Z)/2*	*tA + Z*	*tA – Y*
III	*T/2 + (Y – X)/2*	*tC + X*	*tA – Y*
IV	*T/2 + (X – Z)/2*	*tA + Z*	*tB – X*
V	*T/2 + (Y – Z)/2*	*tA + Z*	*tA – Y*
VI	*T/2 + (Y – X)/2*	*tC + X*	*tA – Y*

life.augmented

电流采样 — 单电阻(2/8)

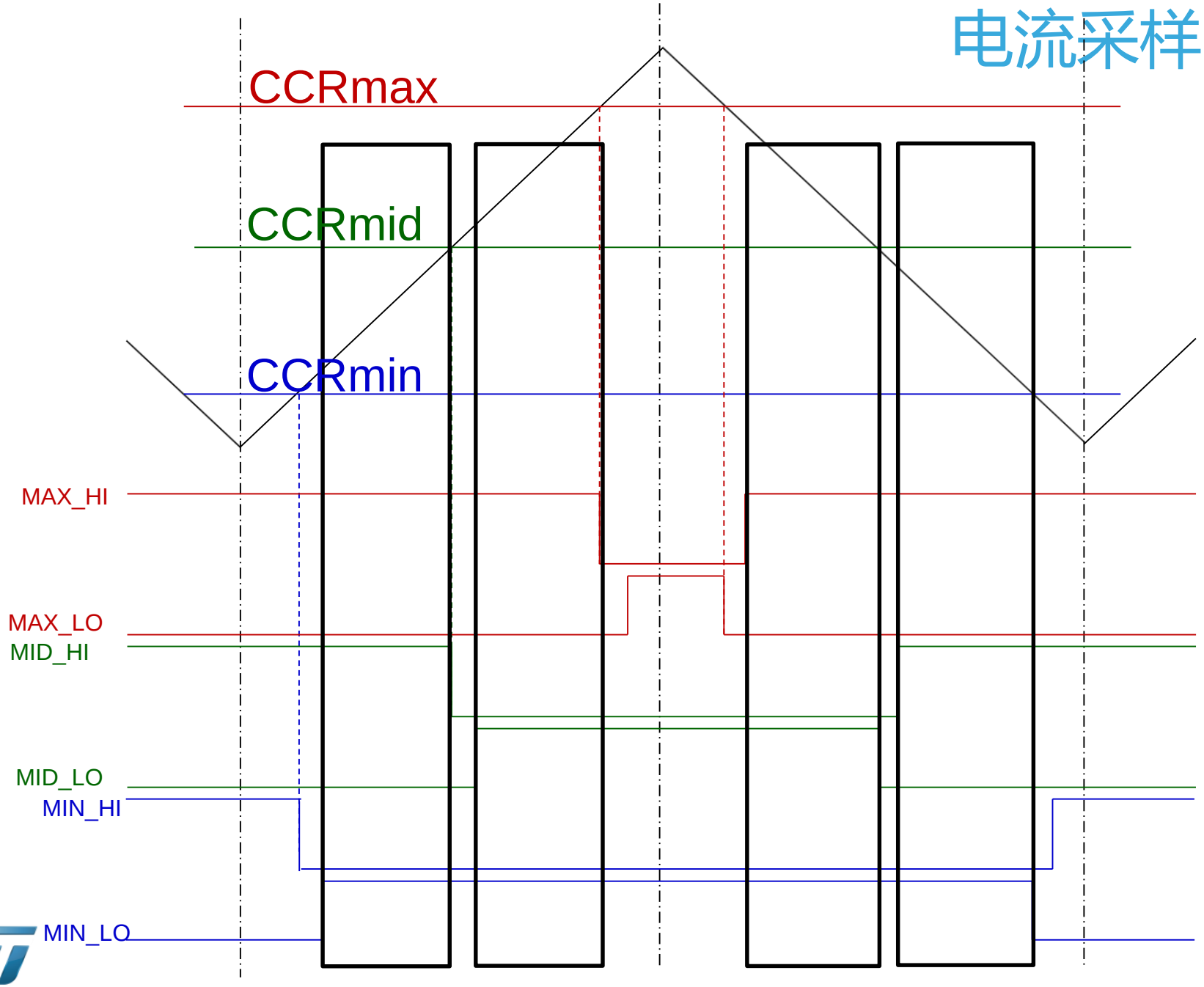




Vector	*A(X)*	*B(Y)*	*C(Z)*	*Ishunt*
$\overrightarrow{U_0}$	OFF(ON)	OFF(ON)	OFF(ON)	*0*
$\overrightarrow{U_1}$	ON(OFF)	OFF(ON)	OFF(ON)	i_A
$\overrightarrow{U_2}$	ON(OFF)	ON(OFF)	OFF(ON)	$-i_C$
$\overrightarrow{U_3}$	OFF(ON)	ON(OFF)	OFF(ON)	i_B
$\overrightarrow{U_4}$	OFF(ON)	ON(OFF)	ON(OFF)	$-i_A$
$\overrightarrow{U_5}$	OFF(ON)	OFF(ON)	ON(OFF)	i_C
$\overrightarrow{U_6}$	ON(OFF)	OFF(ON)	ON(OFF)	$-i_B$
$\overrightarrow{U_7}$	ON(OFF)	ON(OFF)	ON(OFF)	*0*

life.augmented

电流采样 — 单电阻(3/8)



if $T_{ADCtrigger\ delay} > T_{ring}$,needless to add Tring available current signal detection pulse width is:

$Min\_pulsewidth = T_{dead} + T_{on} + T_{ADCtrigger\ delay} + T_{ADC\ s/h}$ else, available current signal detection pulse width is:

$Min\_pulsewidth = T_{dead} + T_{on} + T_{ring} + T_{ADC\ s/h}$

ADC trigger point:

Point 1)CCRmid - $T_{ADCtrigger\ delay}$ -$T_{ADC\ s/h}$ -

Point 2)

if $T_{ADCtrigger\ delay} > T_{ring}$

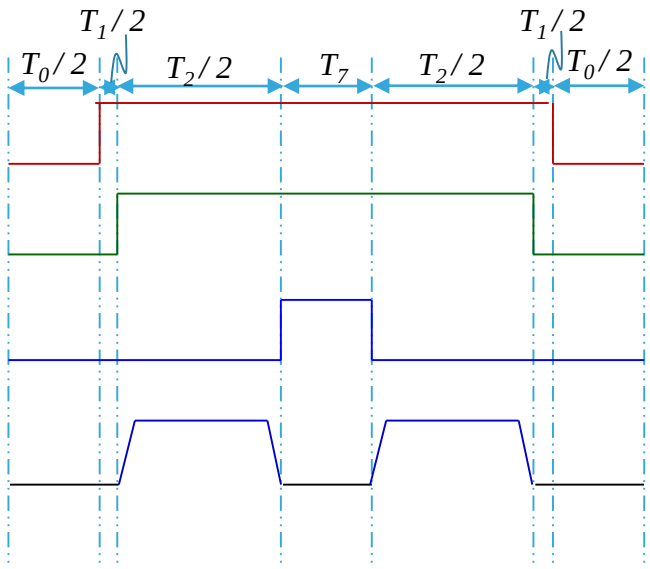
CCRmid + $T_{dead} + T_{on} + T_{ring}$ - $T_{ADCtrigger\ delay}$

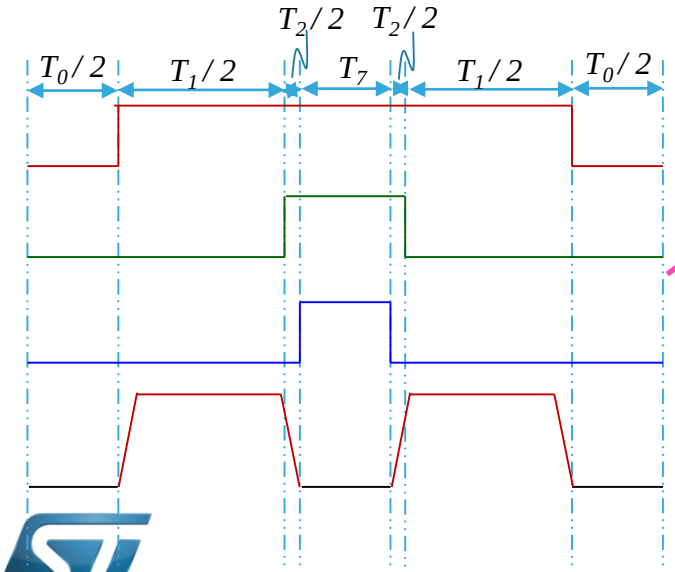
Else

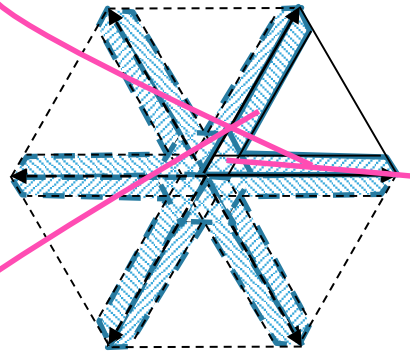
CCRmid + $T_{dead} + T_{on} + T_{ADCtrigger\ delay}$

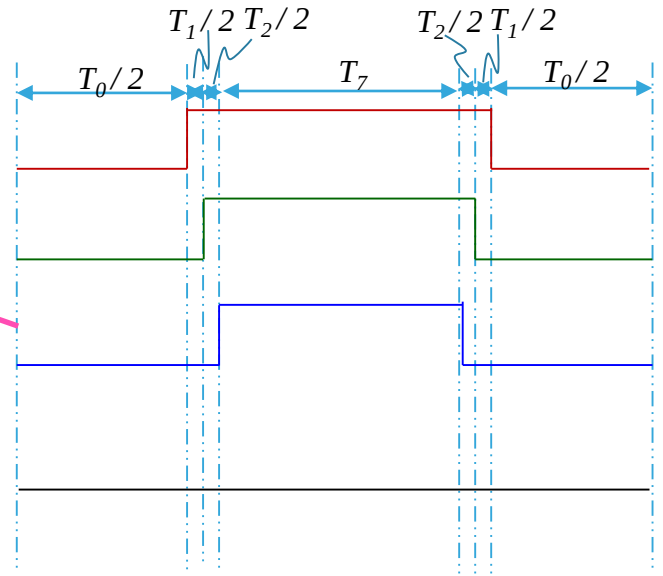
life.augmented

电流采样 — 单电阻(4/8)







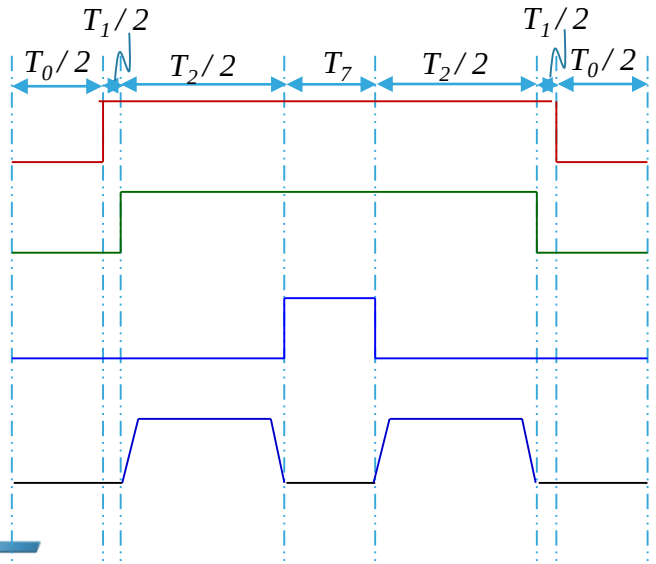


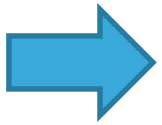
life.augmented

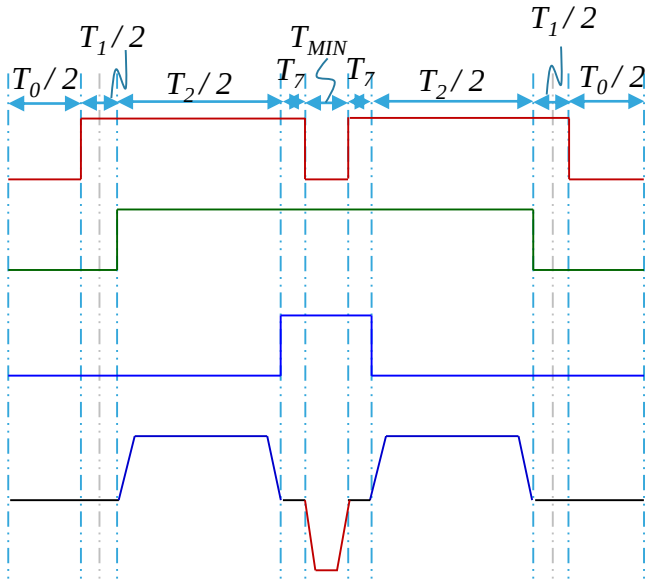
电流采样 — 单电阻(5/8)

ST 专利
US20090284194 A1

ST 专利 (Pat. Pub. No.: US20090284194 A1) 解决单电阻无法采样问题





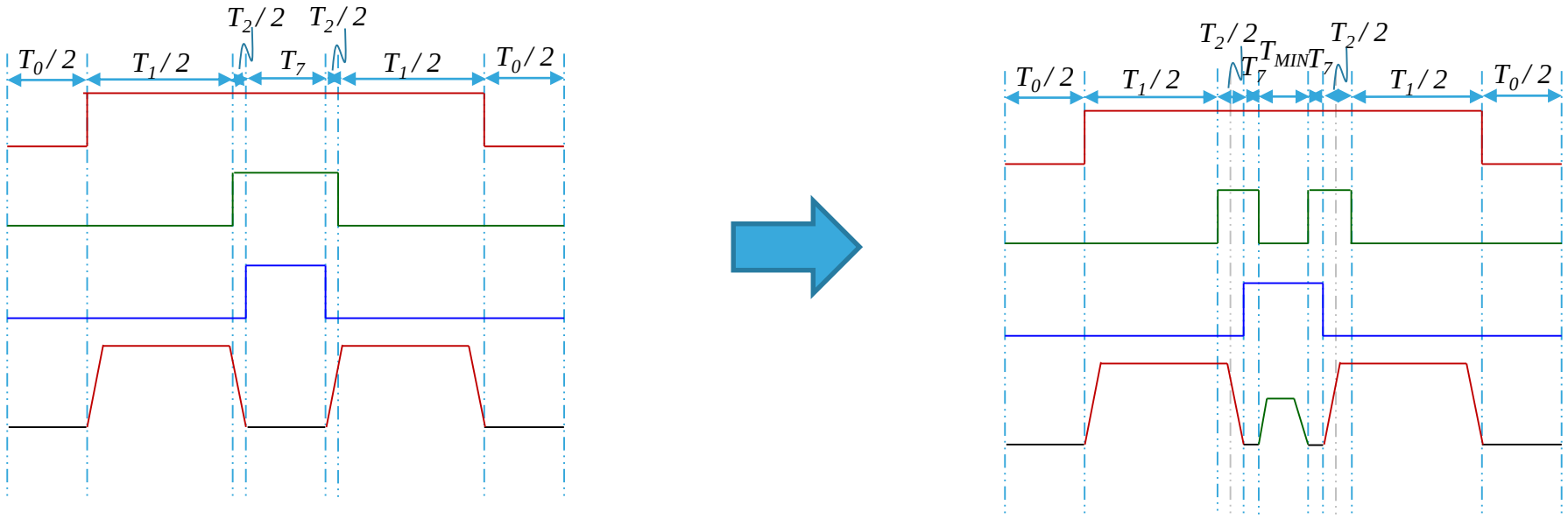


life.augmented

电流采样 — 单电阻(6/8)

ST 专利
US20090284194 A1

ST 专利 (Pat. Pub. No.: US20090284194 A1) 解决单电阻无法采样问题

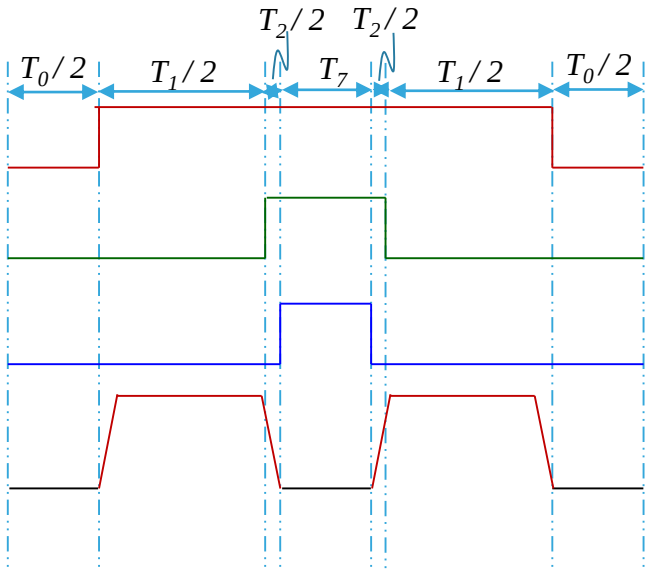


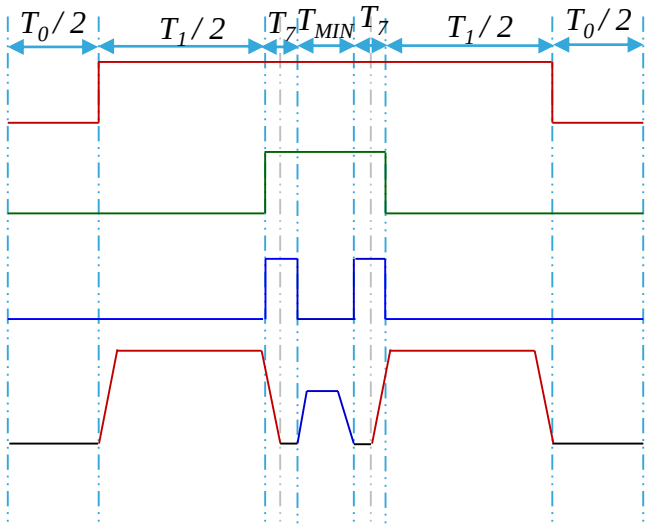
life.augmented

电流采样 — 单电阻(7/8)

ST 专利
US20090284194 A1

ST 专利 (Pat. Pub. No.: US20090284194 A1) 解决单电阻无法采样问题



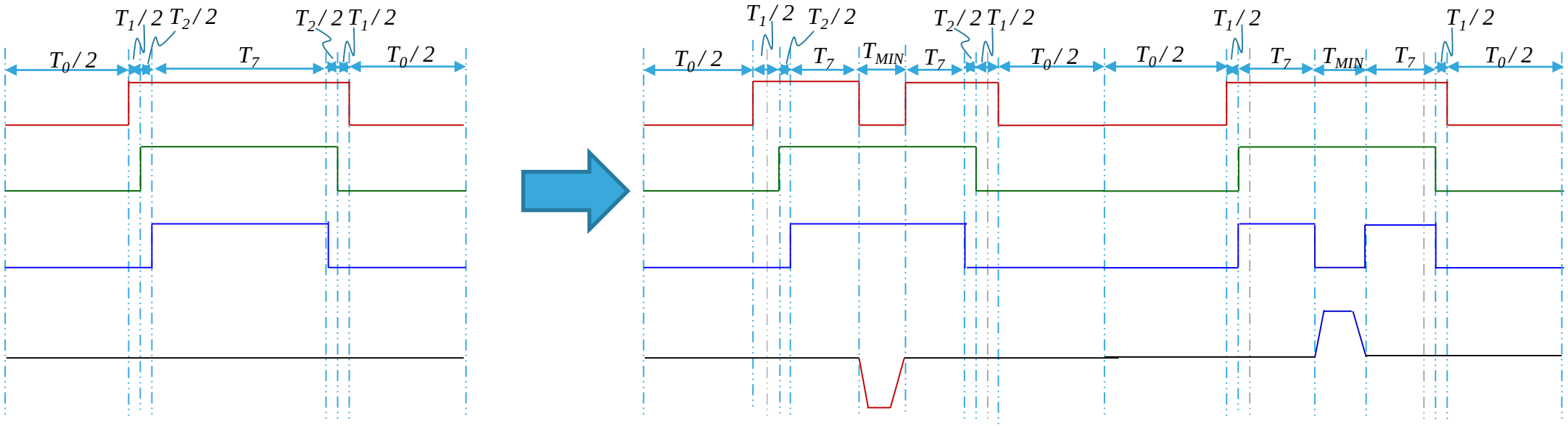


life.augmented

电流采样 — 单电阻(8/8)

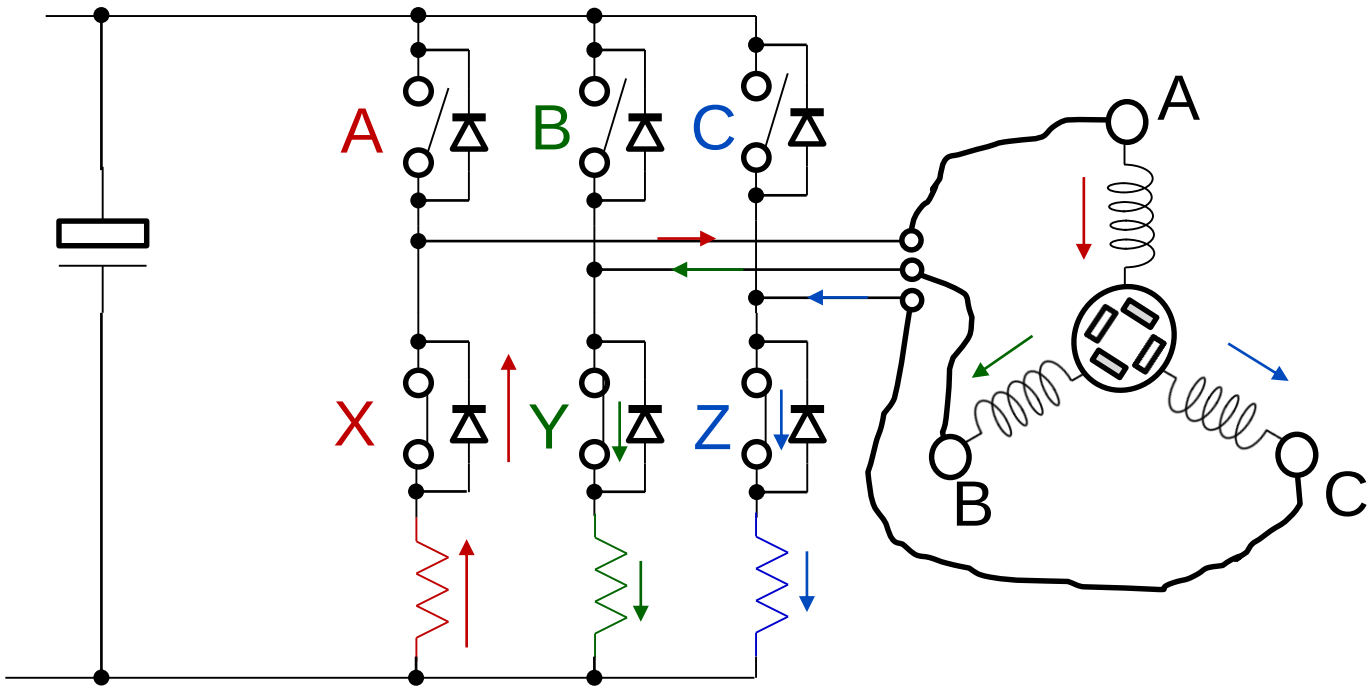
ST 专利
US20090284194 A1

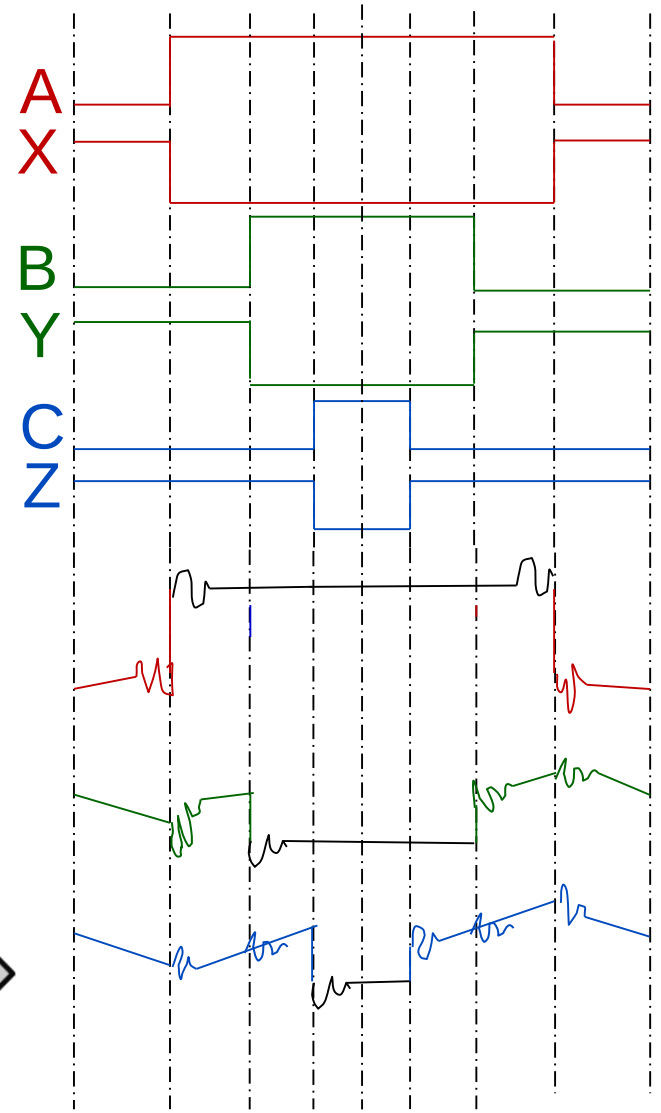
ST 专利 (Pat. Pub. No.: US20090284194 A1) 解决单电阻无法采样问题



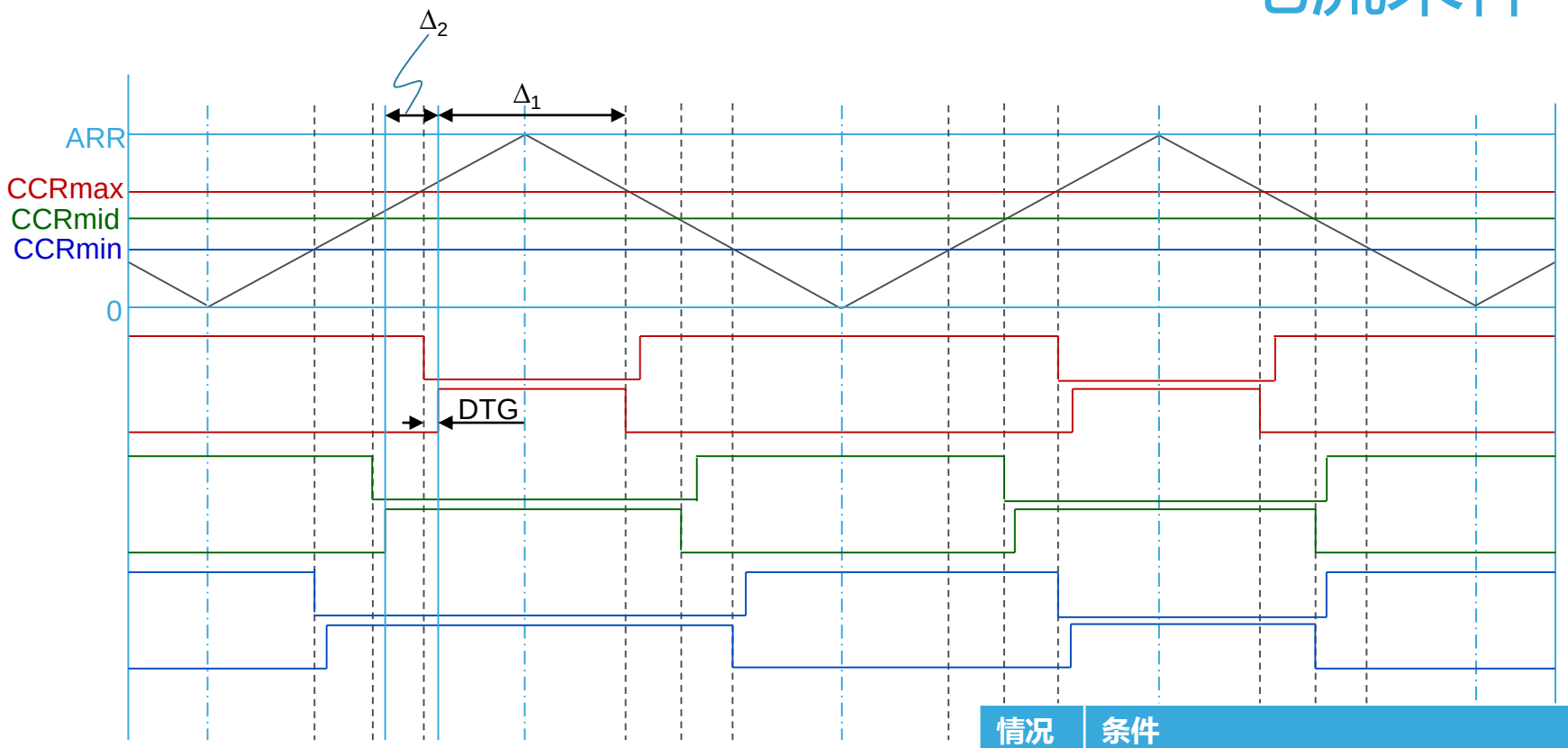
life.augmented

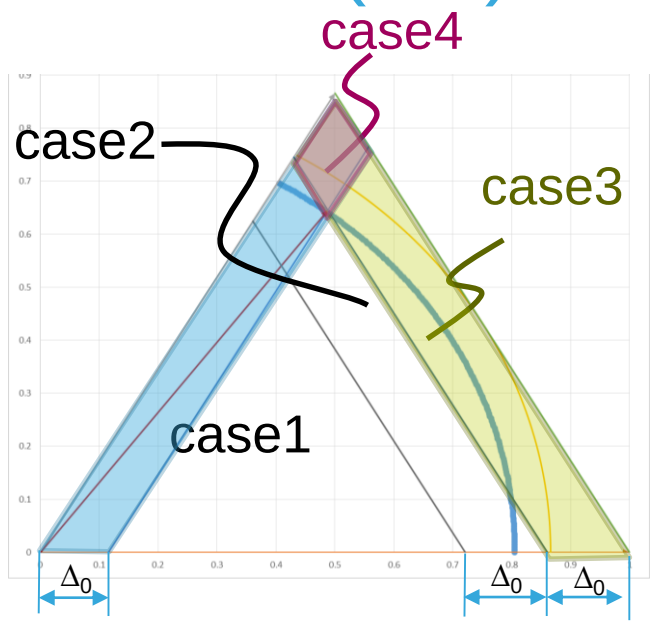
电流采样 — 三电阻(1/2)





电流采样 — 三电阻(2/2)





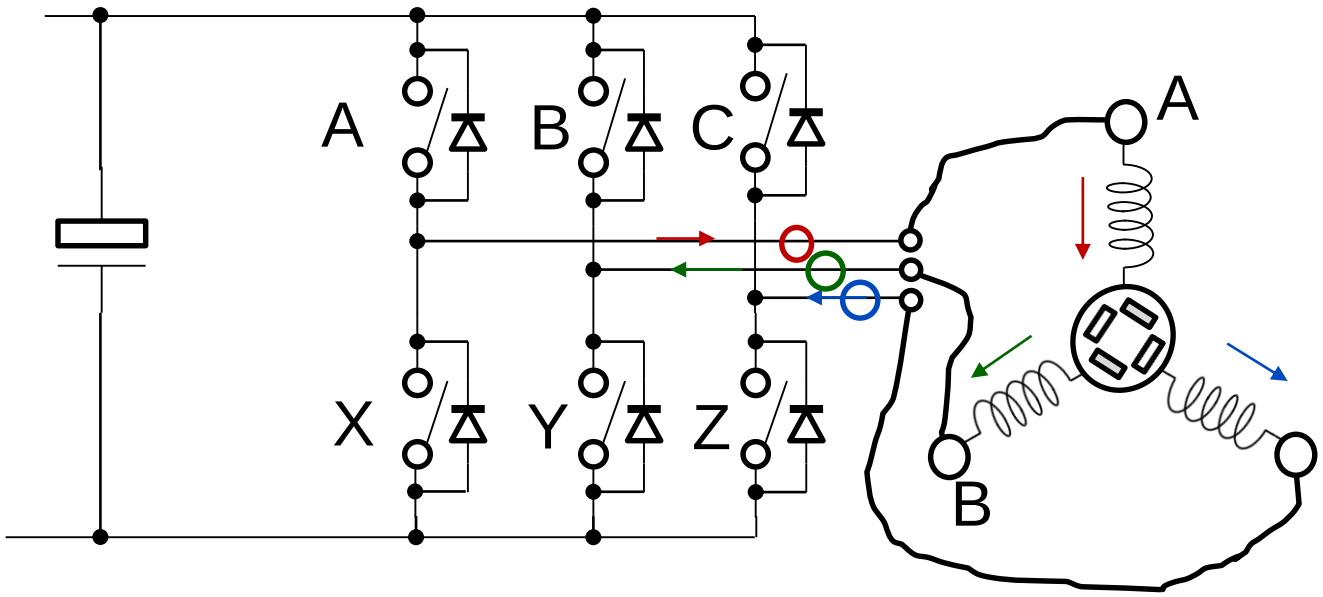
Δ_1 = 2*(ARR – CCRmax – DTG)
Δ_2 = CCRmax – CCRmid

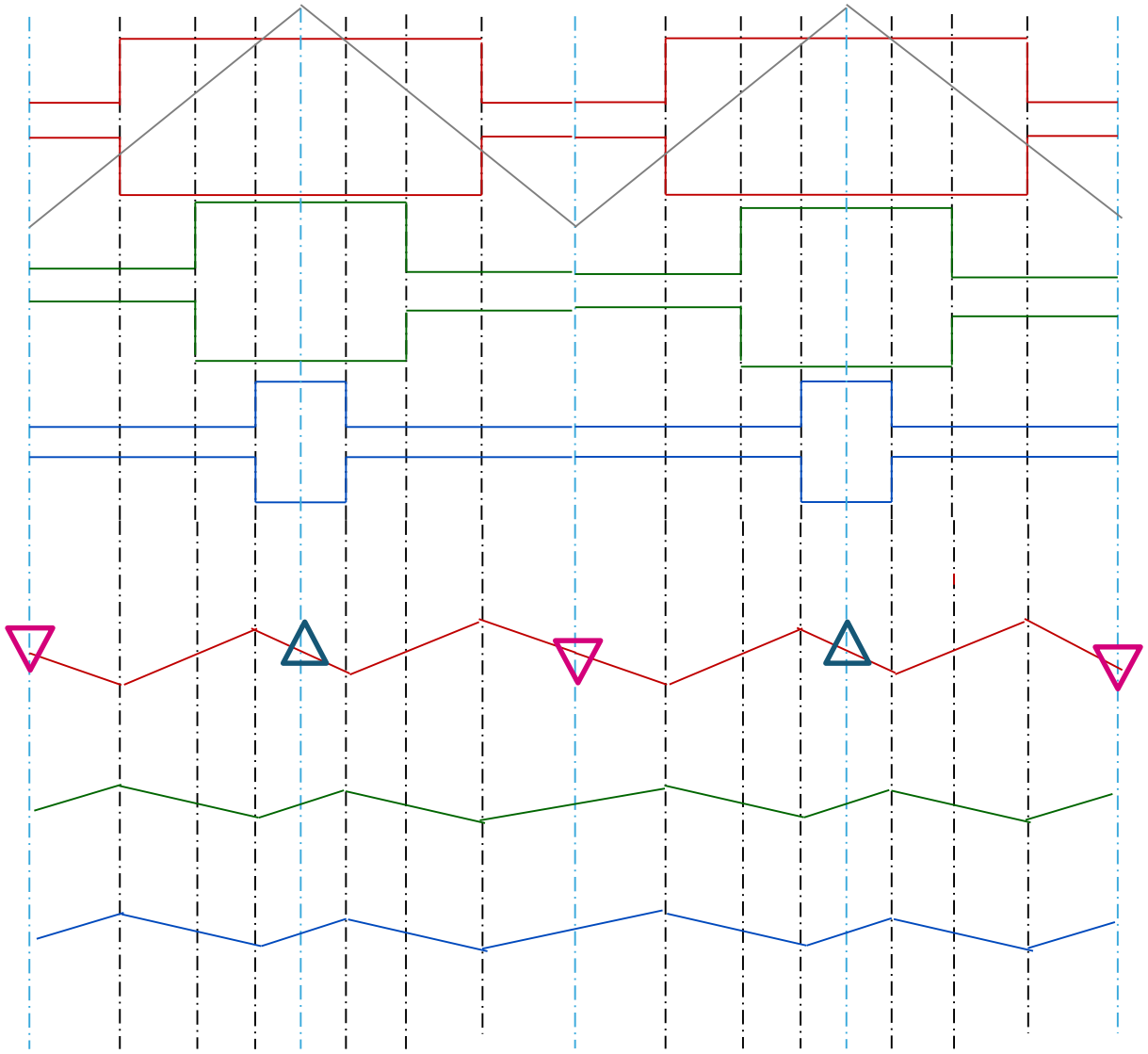
Δ_0 = CNT_Ton+CNT_Trise+CNT_Ring+CNT_TADCSH(COV) , Tring > TADCsta
Δ_0 = CNT_Ton+CNT_Trise +CNT_TADCsta+CNT_TADCSH(COV), TADCsta>=Tring

情况	条件	采样点
1	Δ_1 >max(2*(CNT_Ton+CNT_Trise+CNT_Ring+tdead/2) , CNT_TADCsta+CNT_TADCSH(COV)-tdead/2)	Middle of PWM
2	$\Delta_1 > \Delta_0$	CCRmax+ tdead+ton+tring+ε
3	$\Delta_2 > \Delta_0 > \Delta_1$	CCRmid + tdead+ton+tring+ε
4	$\Delta_1 < \Delta_0$ and $\Delta_2 < \Delta_0$	Not available

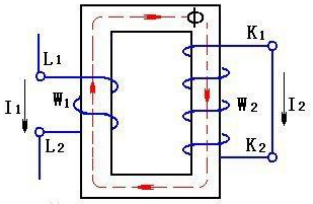
life.augmented

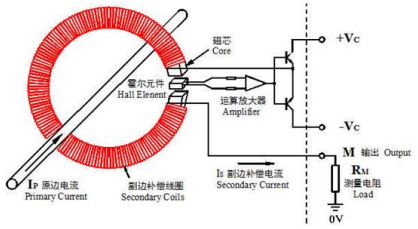
电流采样 — ICS



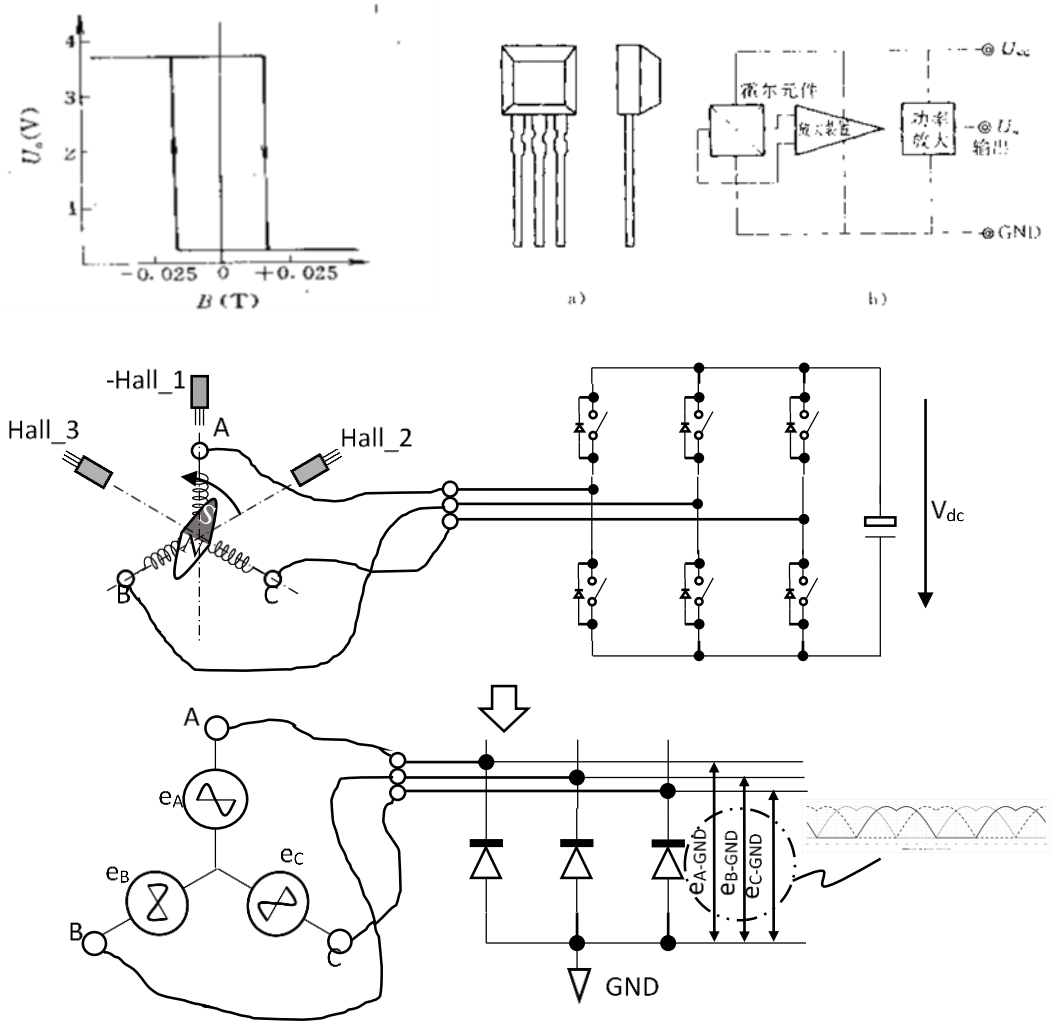


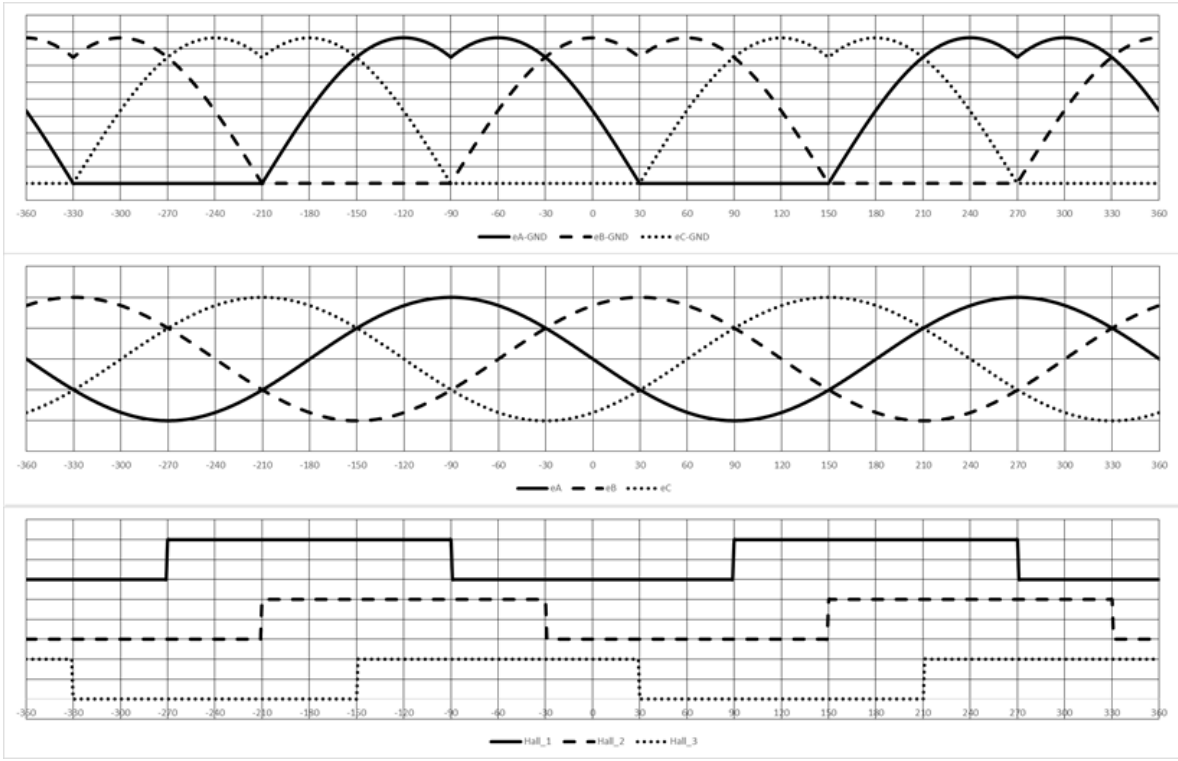
	ACCT	DCCT
频率范围	>0 Hz ~ tens kHz	DC ~ 100kHz
退磁	Need	Need not
成本	low	High





位置速度检测 — Hall传感器

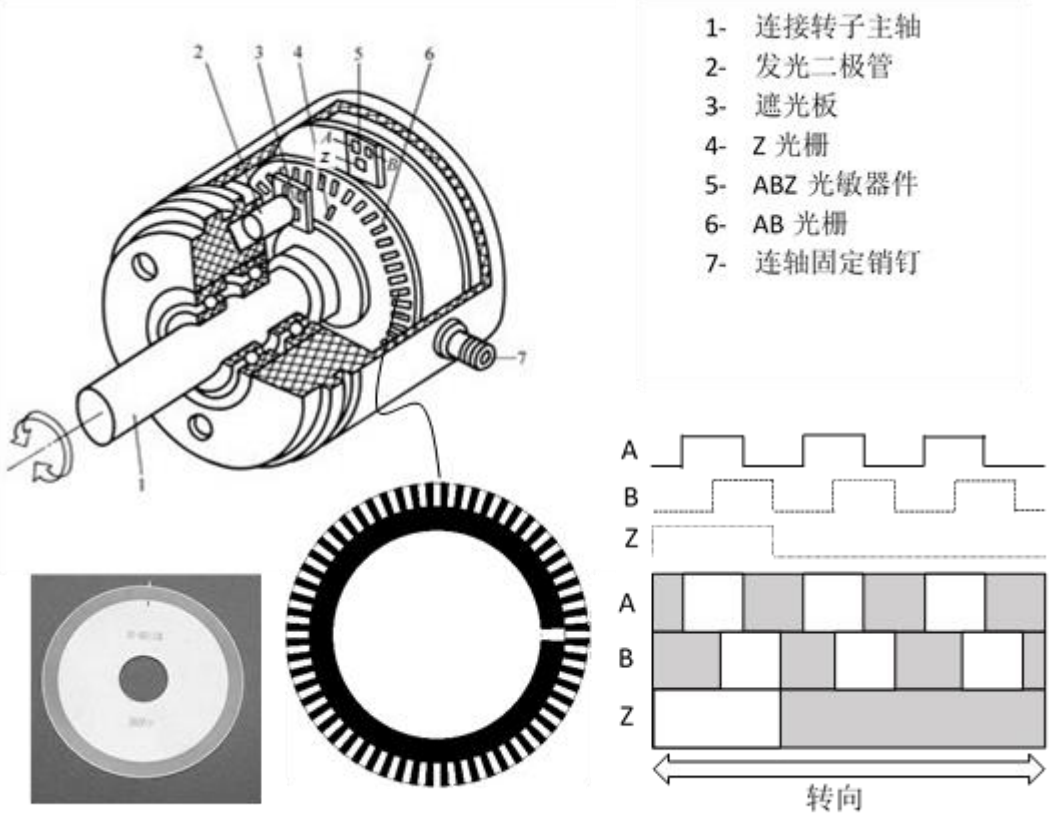


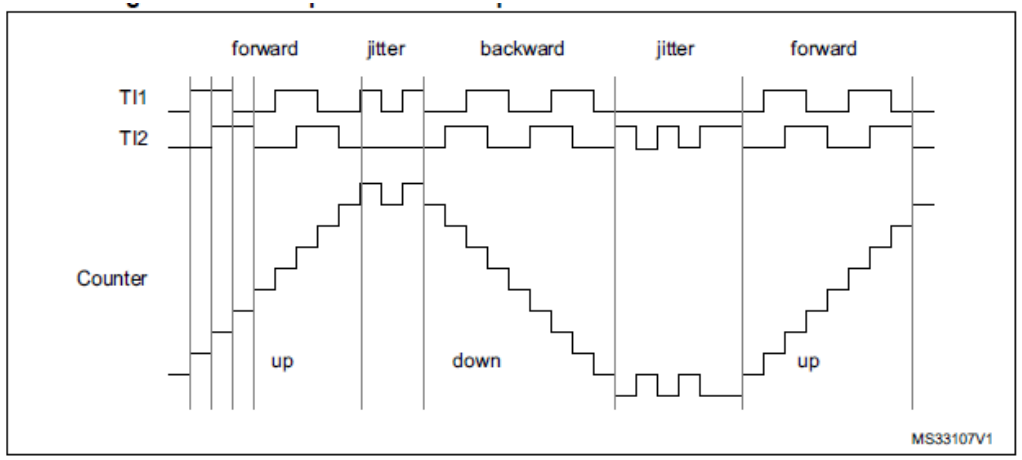


对于60度的Hall信号，可以任意调换三个信号中的任意一个即可得到和120度的处理相似，我们可以很方便使用软件处理。

life.augmented

位置速度检测 — Encoder



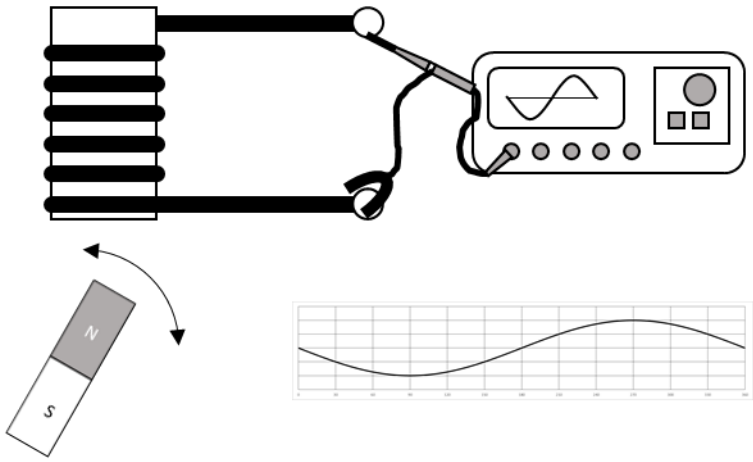


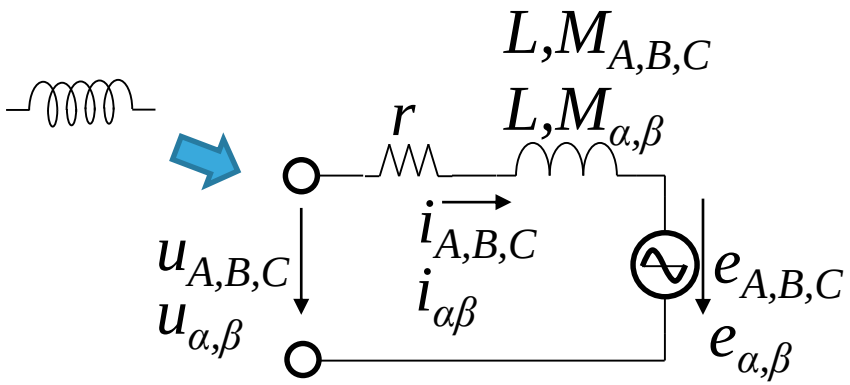
使用增量编码器时，在第一次电机启动，任意保护停止或者MCU复位后都要进行预定位操作。

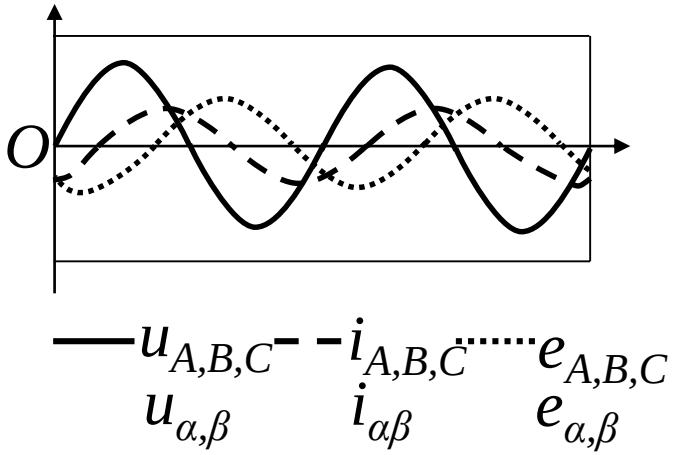
Z信号（一圈一个）可以使用外部中断或者外部Timer捕捉模式，代表编码器的0度位置，可以用于校准角度位置，可以使用DMA模式对编码器模块赋值；

life.augmented

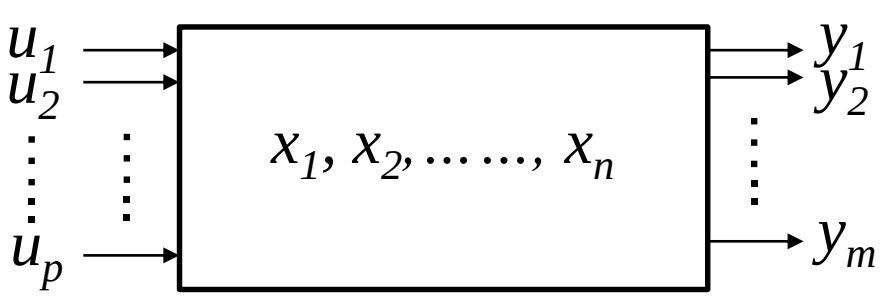
位置速度检测 — 观测器(1/11)

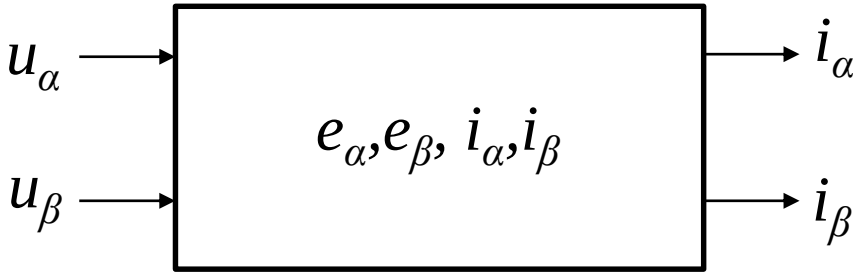






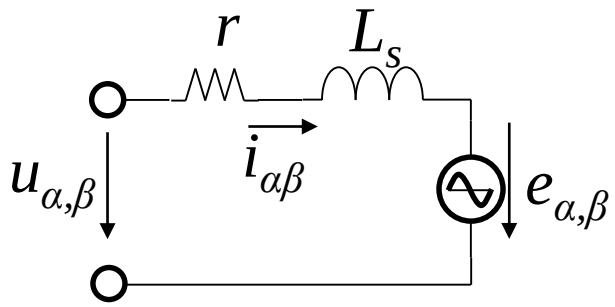
$$u \text{ -> } i\,(\tau_e) \text{ -> } \omega_r, \omega_e = p \cdot \omega_r \text{ - > } e$$

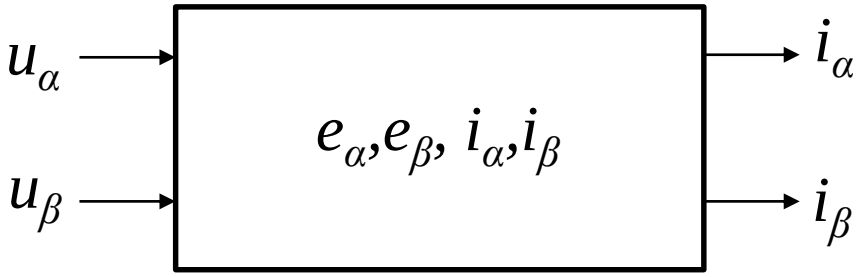


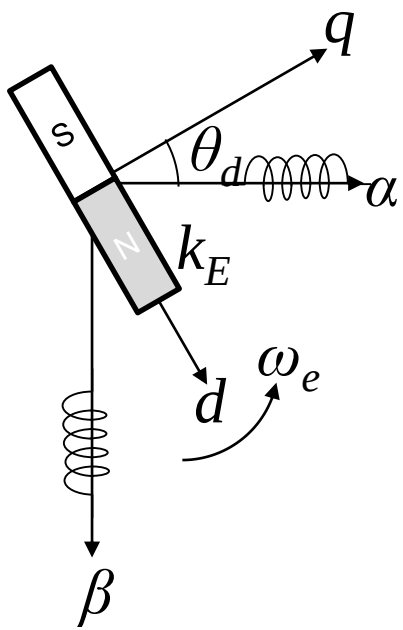


life.augmented

位置速度检测 — 观测器(2/11)







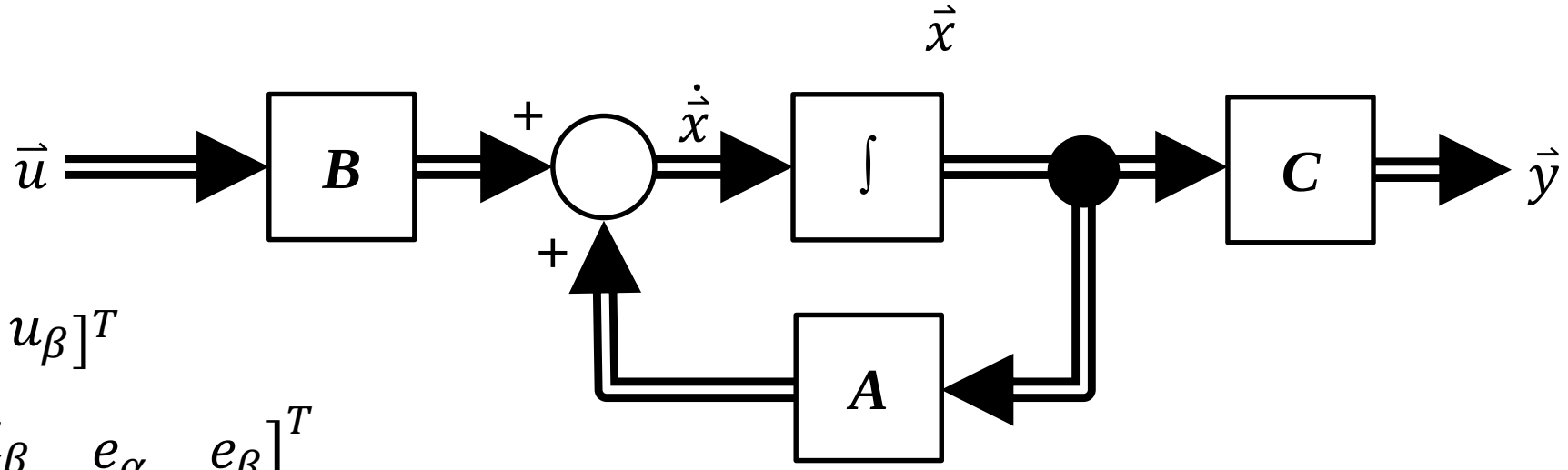
$$\begin{cases} e_\alpha = \dfrac{dk_E sin\theta_d}{dt} = \dfrac{dk_E sin(\theta_{d0}+\omega_e t)}{dt} \\ e_\beta = \dfrac{dk_E cos\theta_d}{dt} = \dfrac{dk_E cos(\theta_{d0}+\omega_e t)}{dt} \end{cases}$$

A linear model

$$\begin{cases} \dfrac{di_\alpha}{dt} = -\dfrac{r}{L_s} i_\alpha - \dfrac{e_\alpha}{L_s} + \dfrac{u_\alpha}{L_s} \\ \dfrac{di_\beta}{dt} = -\dfrac{r}{L_s} i_\beta - \dfrac{e_\beta}{L_s} + \dfrac{u_\beta}{L_s} \\ \dfrac{de_\alpha}{dt} = \omega_e e_\beta \\ \dfrac{de_\beta}{dt} = -\omega_e e_\alpha \end{cases}$$

life.augmented

位置速度检测 — 观测器(3/11)



$$\vec{u} = [u_\alpha \quad u_\beta]^T$$

$$\vec{x} = [i_\alpha \quad i_\beta \quad e_\alpha \quad e_\beta]^T$$

$$\dot{\vec{x}} = \left[\frac{di_\alpha}{dt} \quad \frac{di_\beta}{dt} \quad \frac{de_\alpha}{dt} \quad \frac{de_\beta}{dt}\right]^T$$

$$\vec{y} = [i_\alpha \quad i_\beta]^T$$

$$\begin{cases} \dot{\vec{x}} = A\vec{x} + B\vec{u} \\ \vec{y} = C\vec{x} \end{cases}$$

$$A = \begin{bmatrix} -\frac{r}{L_s} & 0 & -\frac{1}{L_s} & 0 \\ 0 & -\frac{r}{L_s} & 0 & -\frac{1}{L_s} \\ 0 & 0 & 0 & \omega_e \\ 0 & 0 & -\omega_e & 0 \end{bmatrix} \quad B = \begin{bmatrix} -\frac{1}{L_s} & 0 \\ 0 & -\frac{1}{L_s} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

life.augmented

位置速度检测 — 观测器(4/11)

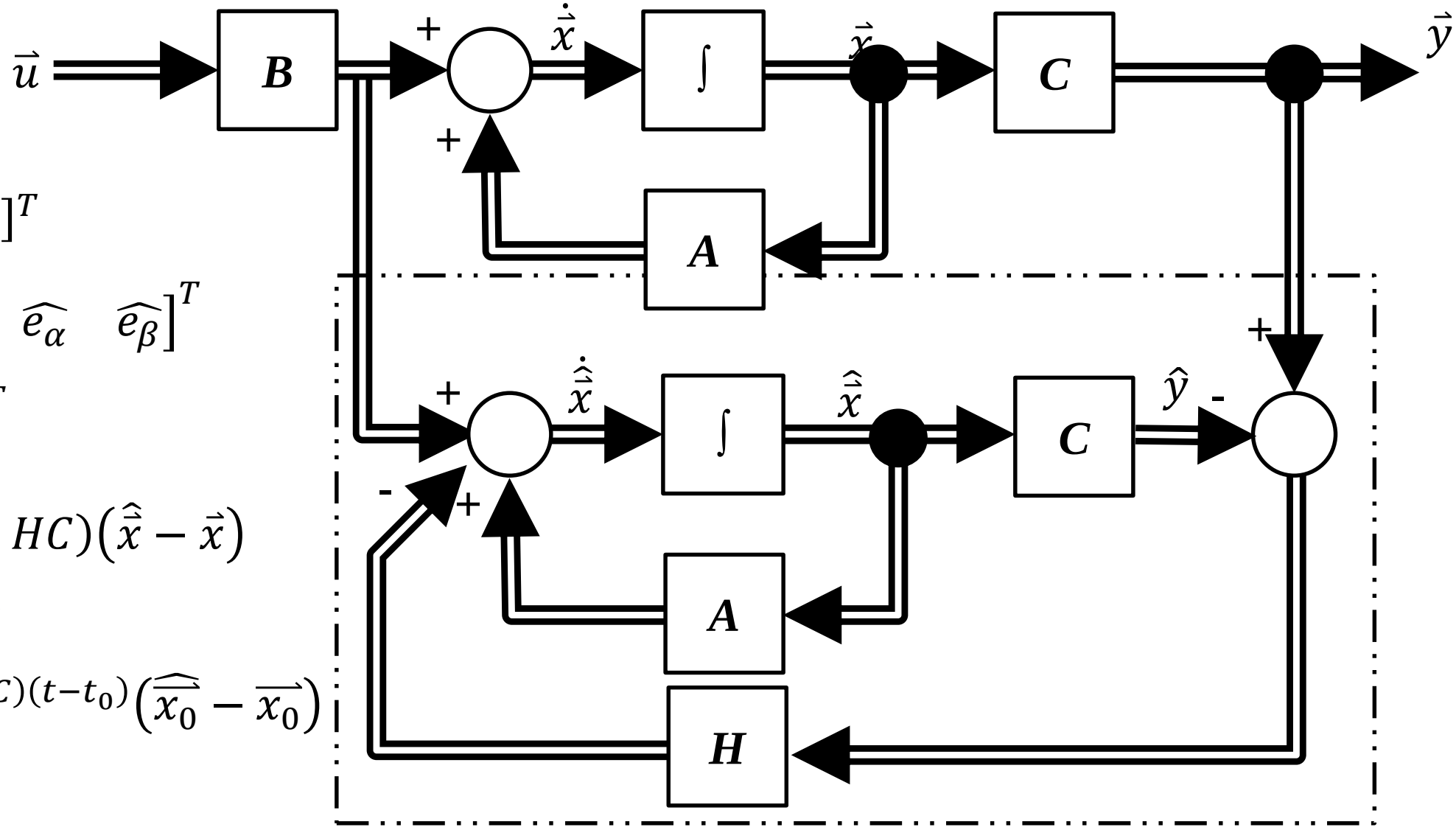
$$\vec{u} = [u_\alpha \quad u_\beta]^T$$

$$\hat{\vec{x}} = [\hat{i_\alpha} \quad \hat{i_\beta} \quad \hat{e_\alpha} \quad \hat{e_\beta}]^T$$

$$\hat{y} = [\hat{i_\alpha} \quad \hat{i_\beta}]^T$$

$$\dot{\hat{\vec{x}}} - \dot{\vec{x}} = (A + HC)(\hat{\vec{x}} - \vec{x})$$

$$\hat{\vec{x}} - \vec{x} = e^{(A+HC)(t-t_0)}(\widehat{\vec{x_0}} - \vec{x_0})$$



位置速度检测 — 观测器(5/11)

$$\begin{cases} \dot{\vec{x}} = A\vec{x} + B\vec{u} \\ \vec{y} = C\vec{x} \end{cases}$$

离散化

$$\dot{\vec{x}} = \frac{d\vec{x}}{dt} \approx \frac{\vec{x}[k] - \vec{x}[k-1]}{T_s}$$

$$\begin{cases} \dfrac{\vec{x}[k] - \vec{x}[k-1]}{T_s} = A\vec{x}[k-1] + B\vec{u}[k-1] \\ \vec{y}[k] = C\vec{x}[k] \end{cases}$$

$$\begin{cases} i_\alpha[k] = \left(1 - \dfrac{rT_s}{L_s}\right) i_\alpha[k-1] - \dfrac{T_s}{L_s} e_\alpha[k-1] + \dfrac{T_s}{L_s} u_\alpha \\ i_\beta[k] = \left(1 - \dfrac{rT_s}{L_s}\right) i_\beta[k-1] - \dfrac{T_s}{L_s} e_\beta[k-1] + \dfrac{T_s}{L_s} u_\beta \\ e_\alpha[k] = e_\alpha[k-1] + \omega_e e_\beta[k-1] \\ e_\beta[k] = e_\beta[k-1] - \omega_e e_\alpha[k-1] \end{cases}$$

life.augmented

位置速度检测 — 观测器(6/11)

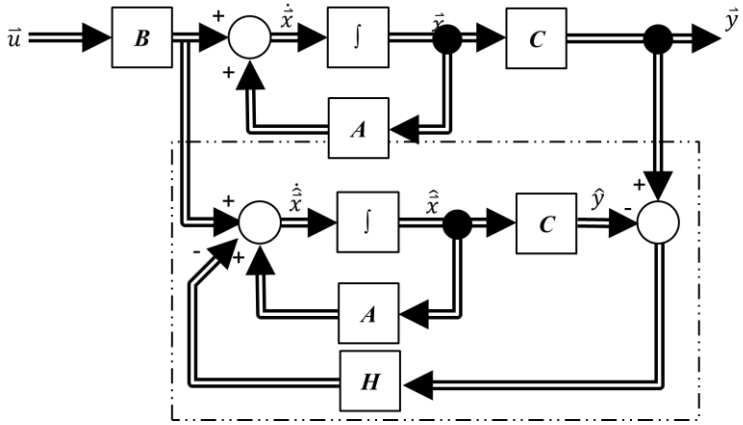
去耦(认为ω_e=0) 将简化马达模型

$$\begin{cases} i_\alpha[k] = \left(1 - \dfrac{rT_s}{L_s}\right) i_\alpha[k-1] - \dfrac{T_s}{L_s} e_\alpha[k-1] + \dfrac{T_s}{L_s} u_\alpha \\ e_\alpha[k] = e_\alpha[k-1] \end{cases}$$

$$A_{reduced} = \begin{bmatrix} 1 - \dfrac{rT_s}{L_s} & -\dfrac{T_s}{L_s} \\ 0 & 1 \end{bmatrix} \xrightarrow[|\lambda I - A| = 0]{\text{特征值}} \begin{cases} \lambda_1 = 1 - \dfrac{rT_s}{L_s} \\ \lambda_2 = 1 \end{cases}$$

life.augmented

位置速度检测 — 观测器(7/11)



$$\dot{\hat{\vec{x}}} = A\hat{\vec{x}} + B\vec{u} + H(\hat{\vec{y}} - \vec{y}) \qquad H = \begin{bmatrix} h_1 & 0 \\ 0 & h_1 \\ h_2 & 0 \\ 0 & h_2 \end{bmatrix}$$

$$\begin{cases} \widehat{i_\alpha}[k] = \left(1 - \dfrac{rT_s}{L_s} + h_1 T_s\right)\widehat{i_\alpha}[k-1] - \dfrac{T_s}{L_s}\widehat{e_\alpha}[k-1] + \dfrac{T_s}{L_s}u_\alpha[k-1] + h_1 T_s(-i_\alpha[k-1]) \\ \widehat{i_\beta}[k] = \left(1 - \dfrac{rT_s}{L_s} + h_1 T_s\right)\widehat{i_\beta}[k-1] - \dfrac{T_s}{L_s}\widehat{e_\beta}[k-1] + \dfrac{T_s}{L_s}u_\beta[k-1] + h_1 T_s(-i_\beta[k-1]) \\ \widehat{e_\alpha}[k] = \widehat{e_\alpha}[k-1] + \omega_e\widehat{e_\beta}[k-1] + h_2 T_s\left(\widehat{i_\alpha}[k-1] - i_\alpha[k-1]\right) \\ \widehat{e_\beta}[k] = \widehat{e_\beta}[k-1] - \omega_e\widehat{e_\alpha}[k-1] + h_2 T_s\left(\widehat{i_\beta}[k-1] - i_\beta[k-1]\right) \end{cases}$$

life.augmented

位置速度检测 — 观测器(8/11)

去耦(设定ω_e=0) 简化观测器模型

$$\begin{cases} \widehat{i_\alpha}[k] = \left(1 - \dfrac{rT_s}{L_s} + h_1 T_s\right)\widehat{i_\alpha}[k-1] - \dfrac{T_s}{L_s}\widehat{e_\alpha}[k-1] + \dfrac{T_s}{L_s}u_\alpha[k-1] + h_1 T_s(-i_\alpha[k-1]) \\ \widehat{e_\alpha}[k] = \widehat{e_\alpha}[k-1] + h_2 T_s\left(\widehat{i_\alpha}[k-1] - i_\alpha[k-1]\right) \end{cases}$$

$$A_{reduced-obs} = \begin{bmatrix} 1 - \dfrac{rT_s}{L_s} + h_1 T_s & -\dfrac{T_s}{L_s} \\ h_2 T_s & 1 \end{bmatrix}$$

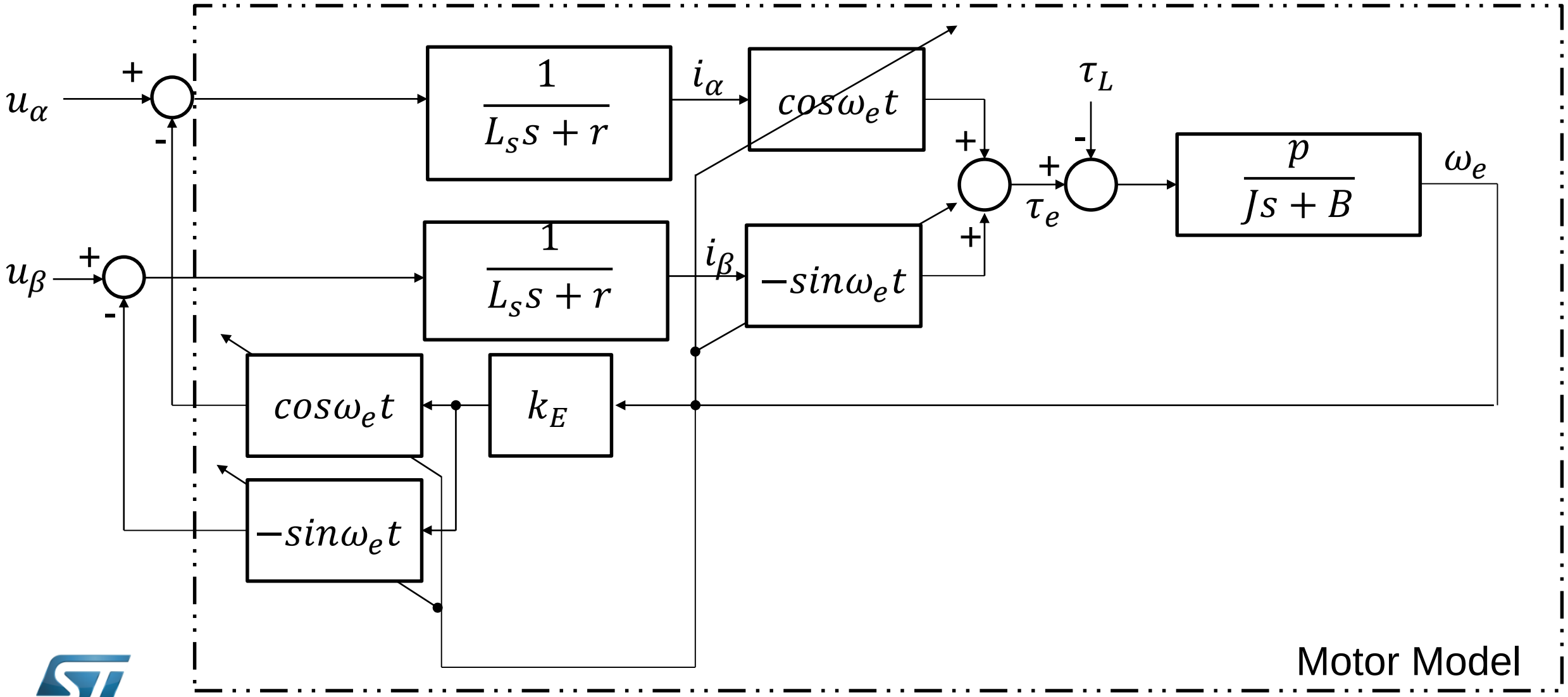
$$\begin{cases} \lambda_{1-obs} = \dfrac{\lambda_1}{k} \\ \lambda_{2-obs} = \dfrac{\lambda_2}{k} \end{cases} \quad k > 1$$

$$|\lambda I - A_{reduced-obs}| = (\lambda - \lambda_{1-obs})(\lambda - \lambda_{2-obs})$$

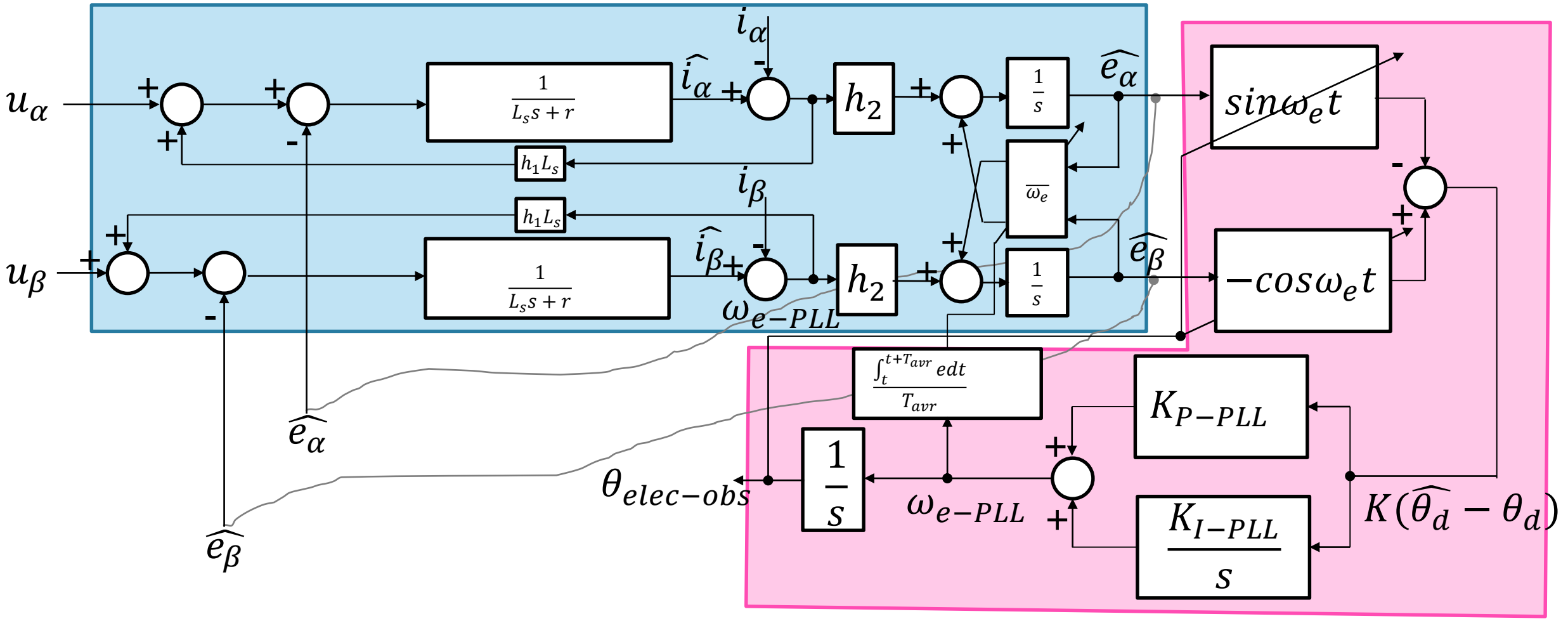
$$\begin{cases} h_1 = \dfrac{\lambda_{1-obs} + \lambda_{2-obs} - 2}{T_s} + \dfrac{r}{L_s} \\ h_2 = \dfrac{L_s(1 - \lambda_{1-obs} - \lambda_{2-obs} + \lambda_{1-obs}\lambda_{2-obs})}{{T_s}^2} \end{cases}$$

life.augmented

位置速度检测 — 观测器(9/11)



位置速度检测 — 观测器(10/11)



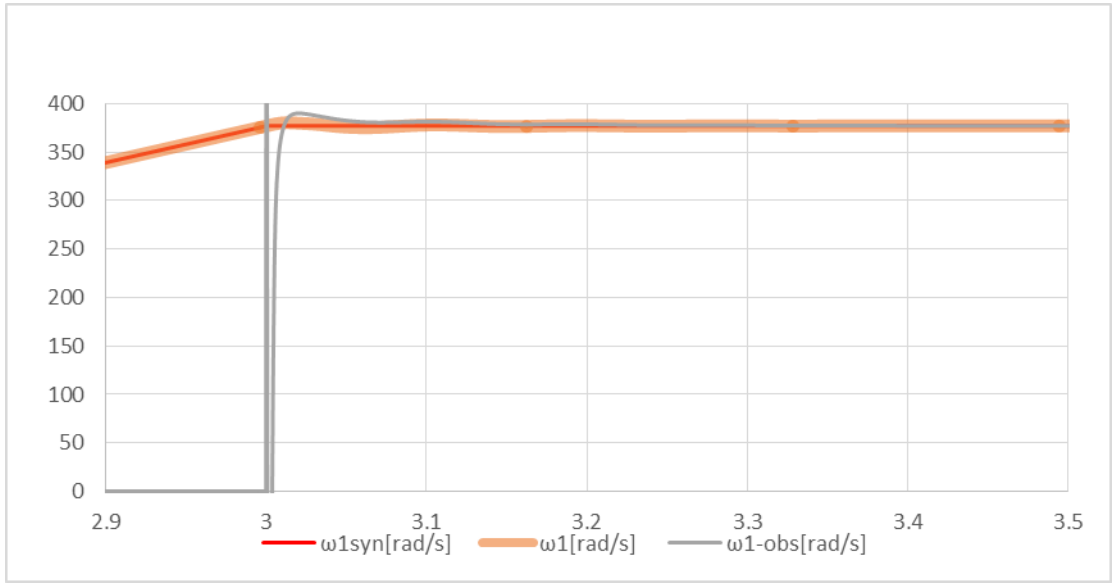
OBSERVER

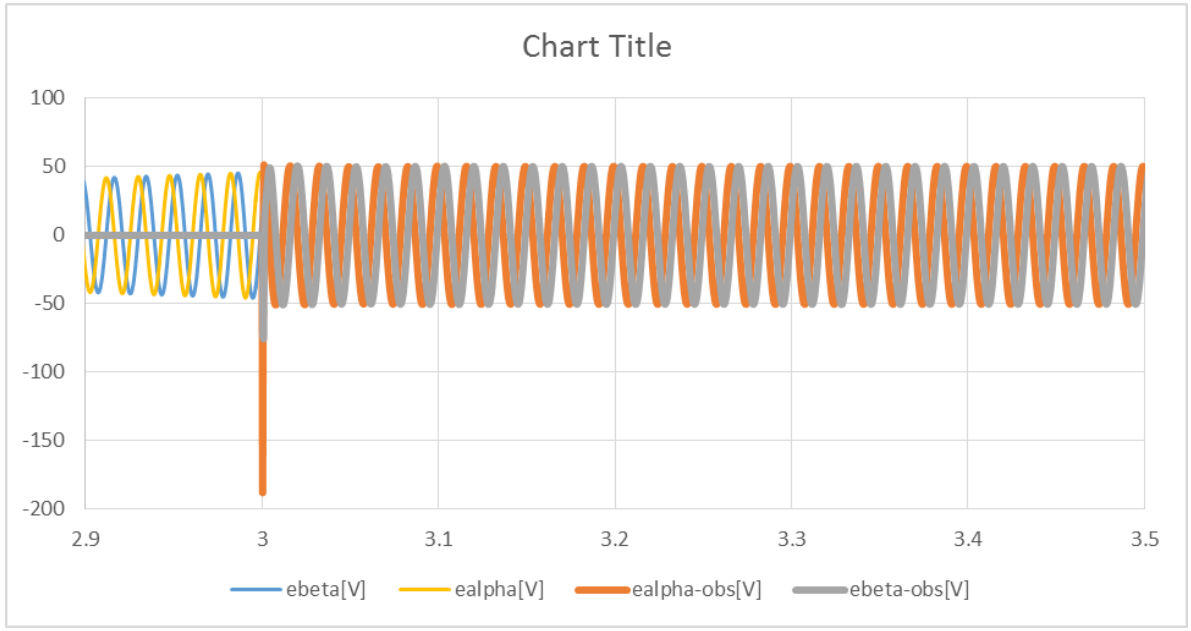
PLL

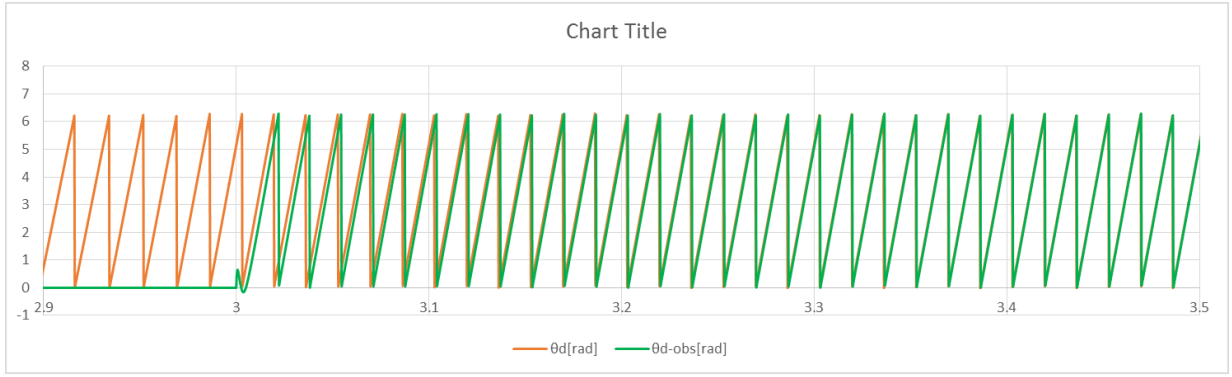
life.augmented

位置速度检测 — 观测器(11/11)

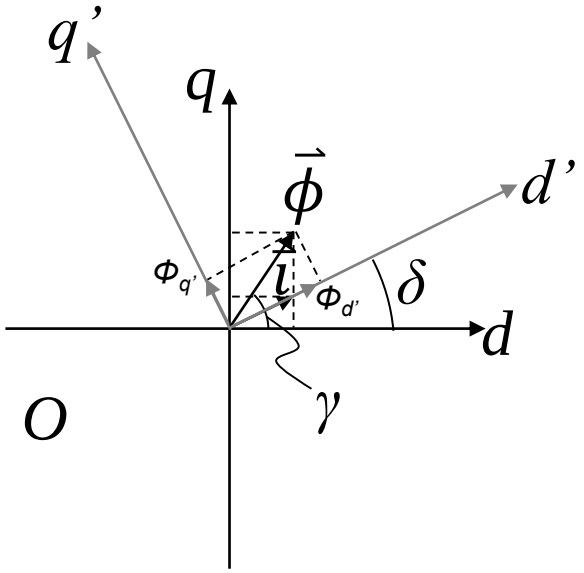
$$
\begin{cases}
\widehat{i_\alpha}[k] = \left(1 - \dfrac{rT_s}{L_s} + h_1 T_s\right)\widehat{i_\alpha}[k-1] - \dfrac{T_s}{L_s}\widehat{e_\alpha}[k-1] + \dfrac{T_s}{L_s}u_\alpha[k-1] + h_1 T_s i_\alpha[k-1] \\
\widehat{i_\beta}[k] = \left(1 - \dfrac{rT_s}{L_s} + h_1 T_s\right)\widehat{i_\beta}[k-1] - \dfrac{T_s}{L_s}\widehat{e_\beta}[k-1] + \dfrac{T_s}{L_s}u_\beta[k-1] + h_1 T_s i_\beta[k-1] \\
\widehat{e_\alpha}[k] = \widehat{e_\alpha}[k-1] + T_s\overline{\omega_1}\widehat{e_\beta}[k-1] + h_2 T_s\left(\widehat{i_\alpha}[k-1] - i_\alpha[k-1]\right) \\
\widehat{e_\beta}[k] = \hat{\beta}[k-1] - T_s\overline{\omega_1}\widehat{e_\alpha}[k-1] + h_2 T_s\left(\widehat{i_\beta}[k-1] - i_\beta[k-1]\right)
\end{cases}
$$







位置速度检测 — HFI(1/4)



$$\tan\delta = \frac{I_q}{I_d}, \tan\gamma = \frac{L_q I_q}{L_d I_d} = \frac{L_q}{L_d}\tan\delta$$

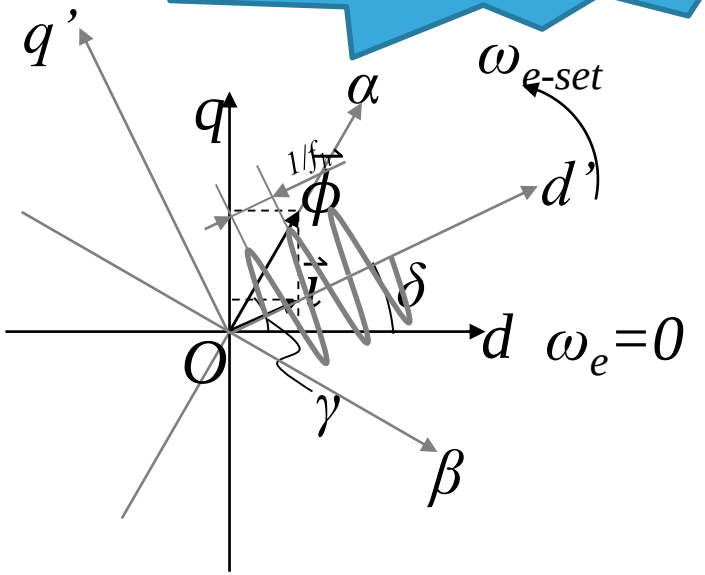
$$\tan(\gamma-\delta) = \frac{\tan\gamma - \tan\delta}{1+\tan\gamma\tan\delta} = \frac{\left(\frac{L_q}{L_d}-1\right)\tan\delta}{1+\frac{L_q}{L_d}tan^2\delta} = \frac{(L_q-L_d)\sin 2\delta}{2\left[L_d+(L_q-L_d)sin^2\delta\right]}$$

$$|\vec{\phi}| = \sqrt{(L_d I_d)^2 + (L_q I_q)^2} = |\vec{\imath}|\sqrt{L_d{}^2 + (L_q{}^2 - L_d{}^2)sin^2\delta}$$

life.augmented

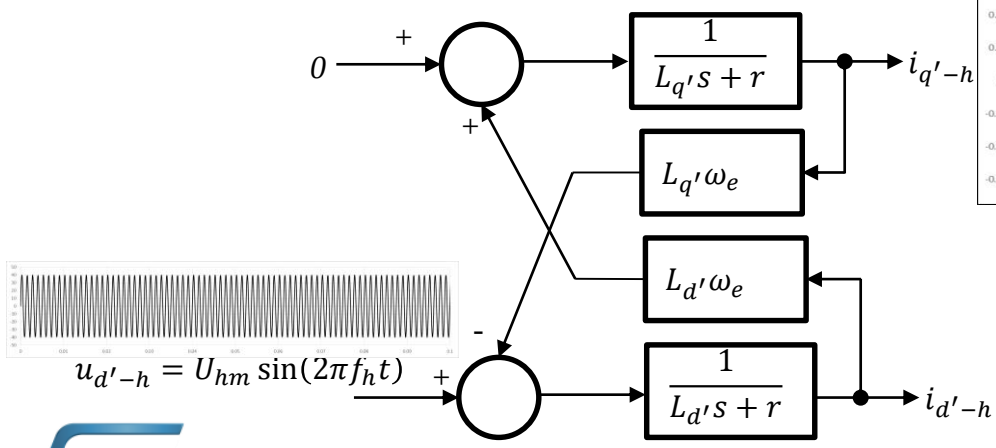
位置速度检测 — HFI(2/4)

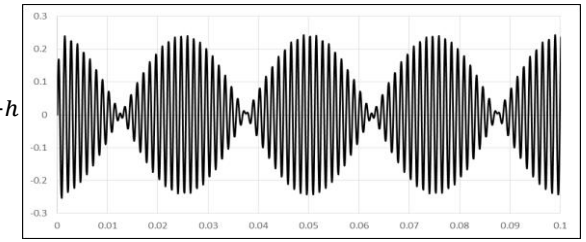
ST 专利
US932563 B1

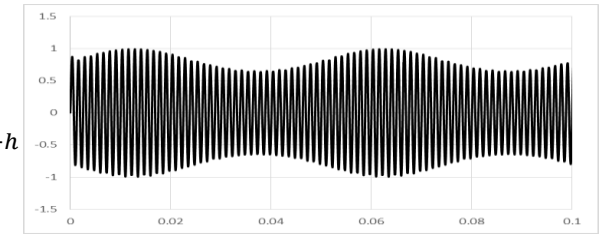


$$|\vec{\phi}| = \sqrt{(L_d I_d)^2 + (L_q I_q)^2} = |\vec{\imath}|\sqrt{L_d{}^2 + (L_q{}^2 - L_d{}^2)sin^2\delta}$$

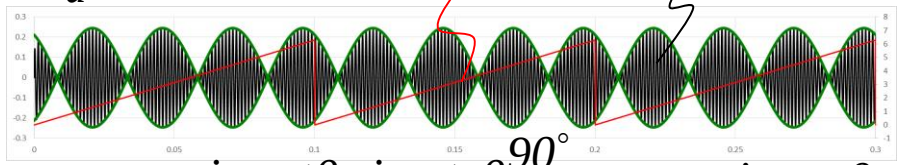
$$|\vec{\phi}| = I_m \sin(2\pi f_h t)\sqrt{L_d{}^2 + (L_q{}^2 - L_d{}^2)sin^2(\omega_e t + \delta_0)}$$





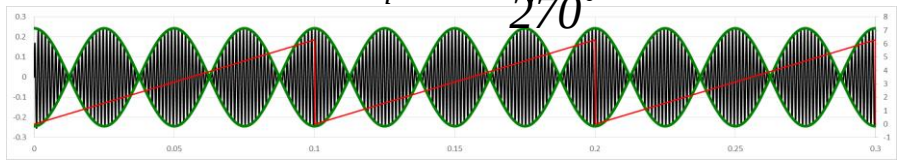


θ_d=30° $\delta=\int\omega_{e\text{-}set}dt$ $i_{q'\text{-}h}$



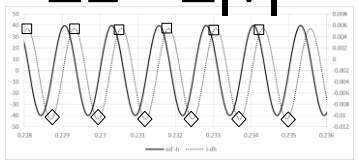
$i_{d'\text{-}h}<0,\ i_{q'\text{-}h}>0$ 90° $i_{d'\text{-}h}>0,\ i_{q'\text{-}h}>0$

θ_d=45° 180° 0° $i_{d'\text{-}h}<0, i_{q'\text{-}h}<0$ $i_{d'\text{-}h}>0, i_{q'\text{-}h}<0,$ 270°

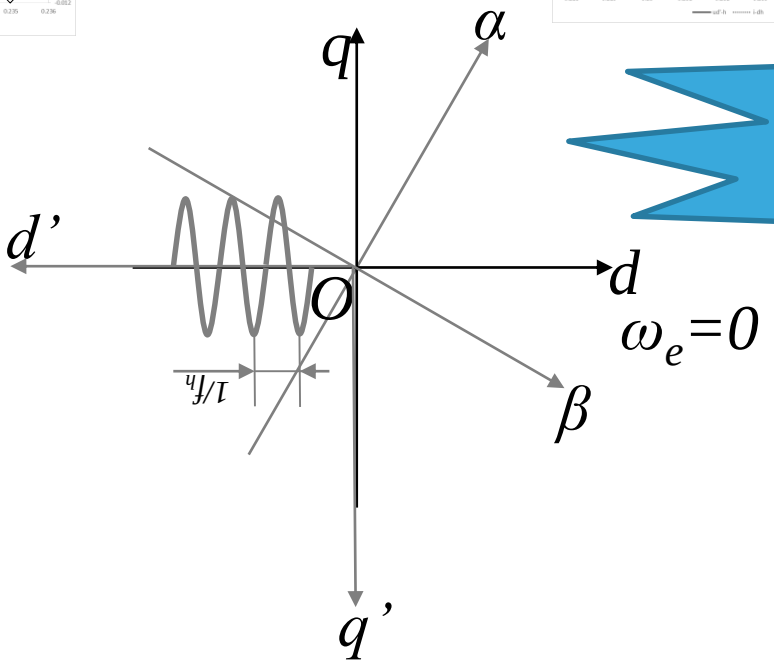


位置速度检测 — HFI(3/4)

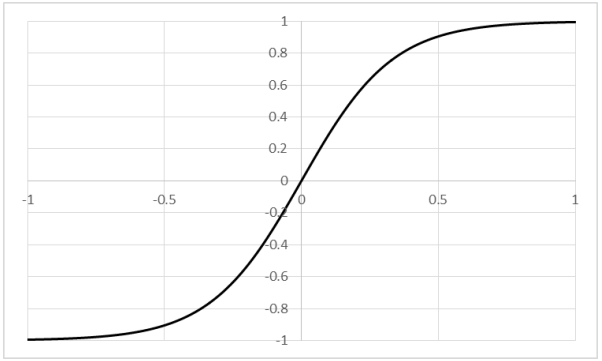
$\Sigma\square < \Sigma|\Diamond|$



$\Sigma\square > \Sigma|\Diamond|$



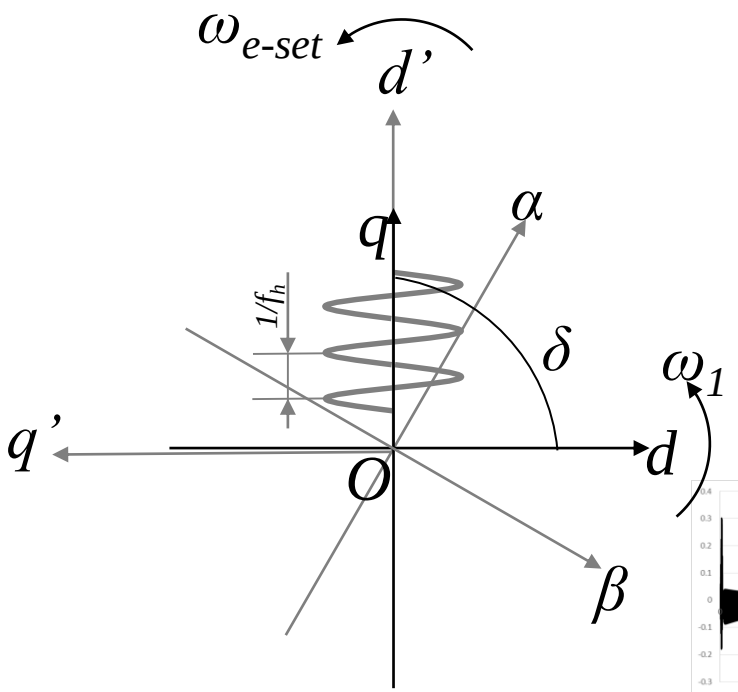
ST Patent
US932563 B1



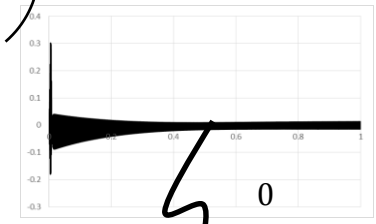
Φ-i 曲线

$u_{d'-h} = U_{hm}\sin(2\pi f_h t)$ → $\frac{1}{L_{d'}s + r}$ → $i_{d'-h}$

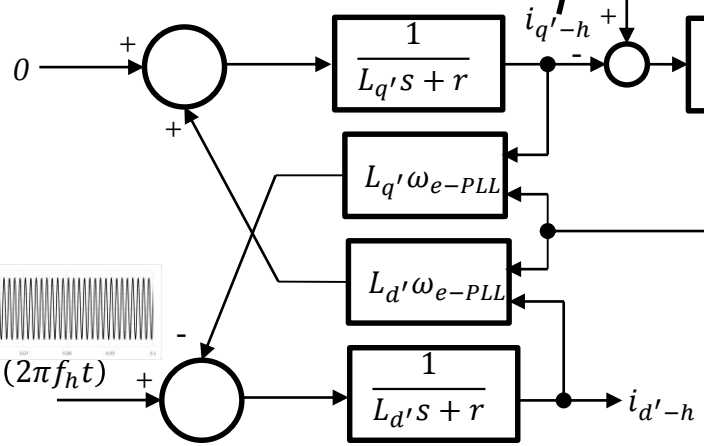
位置速度检测 — HFI(4/4)



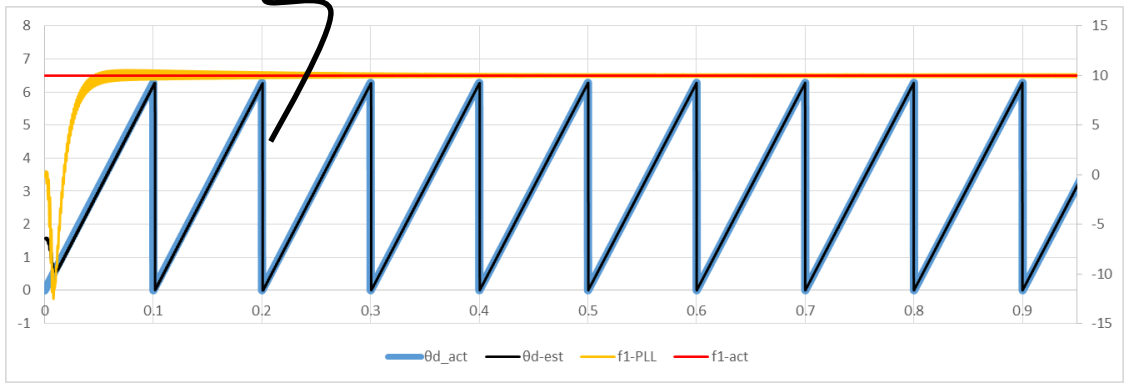
当电机开始转动后，保持δ=0 ，d’轴与q轴重合，那么$i_{q'\text{-}h}$ =0，
也就是如果保持$i_{q'\text{-}h}$ =0， d’-q’ 旋转坐标系的速度等于转子速度$\omega_e = \omega_{e\text{-}set}$.



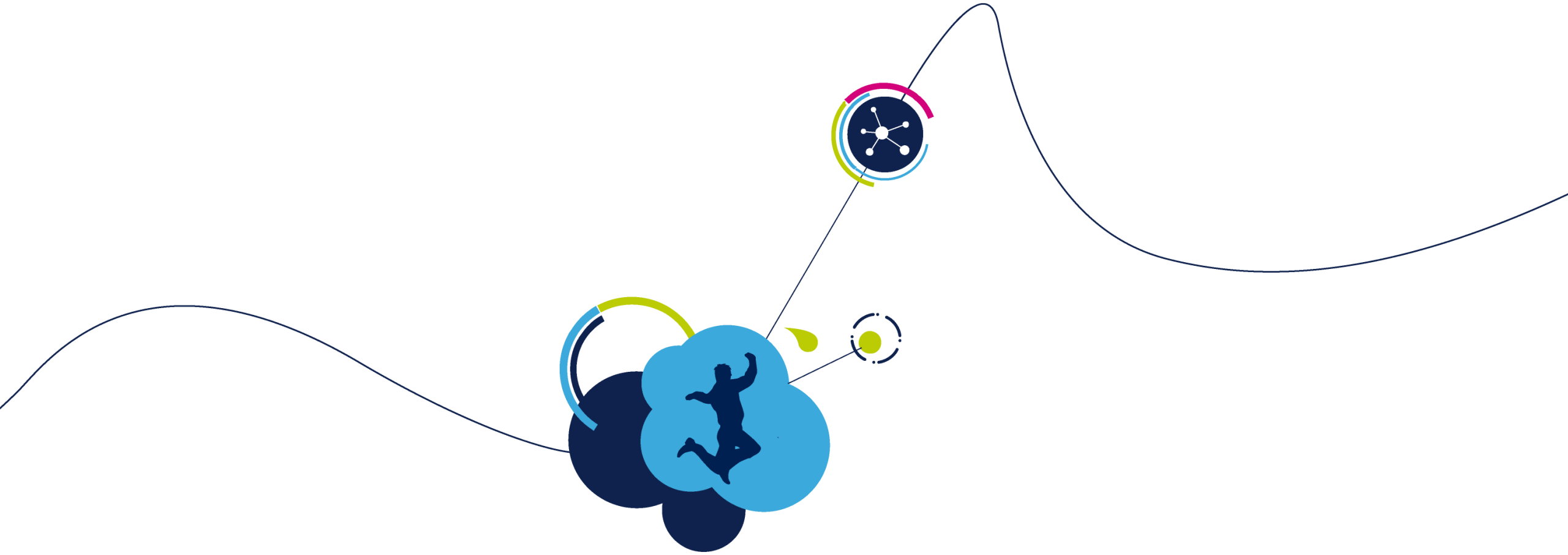
⇦ 使用PLL,使得 $i_{q'\text{-}h}$ =0



$u_{d'-h} = U_{hm}\sin(2\pi f_h t)$



life.augmented



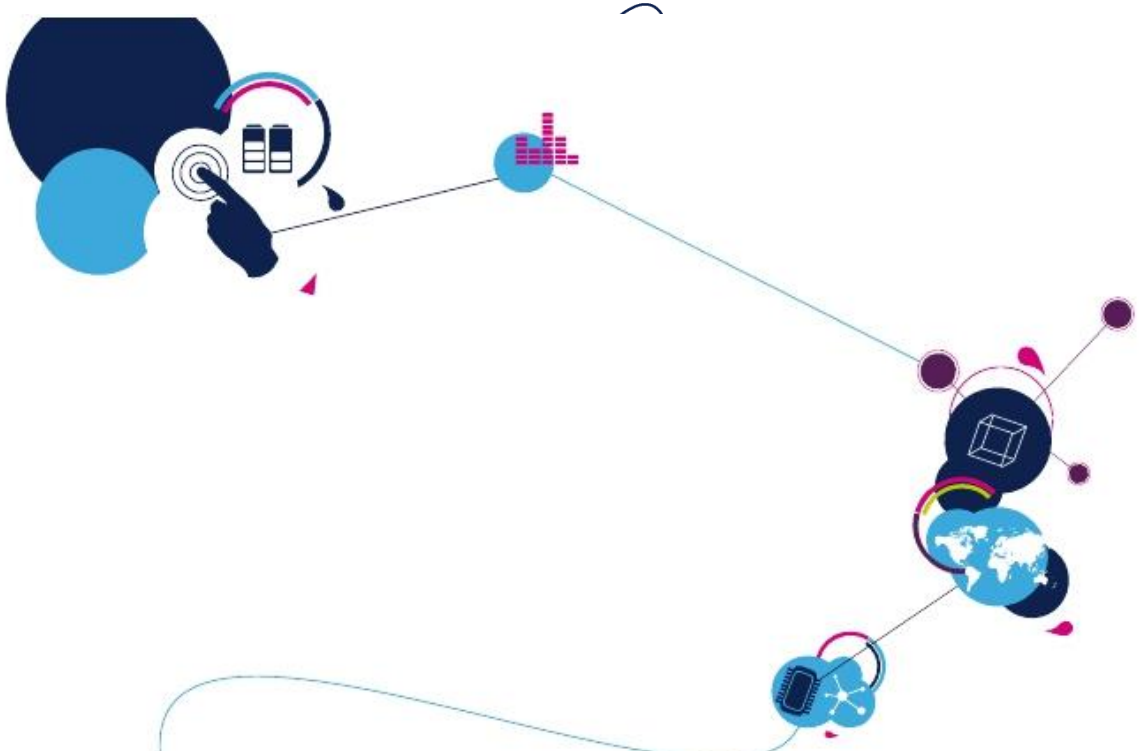
2.MC SDK5.0详解

life.augmented

内容

- STM32 电机控制开发套件 5.0 概览
- SDK 5.0 软件开发

life.augmented



STM32 电机控制开发套件 5.0 概览

life.augmented

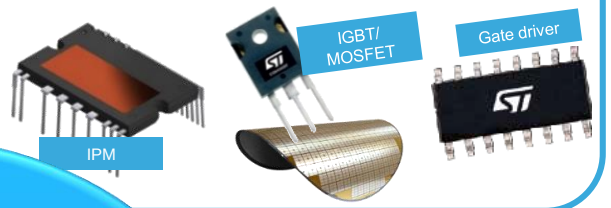
life.augmented

STM32 电机控制开发套件 5.0 概览-历史

MCUs for Motor Control (8-32 bit)

STM8 STM32

Inverters IPM / discretes



Motor Control Ecosystem

ST MC Workbench

PC SW GUI
Full customization and real time communication

HW Boards

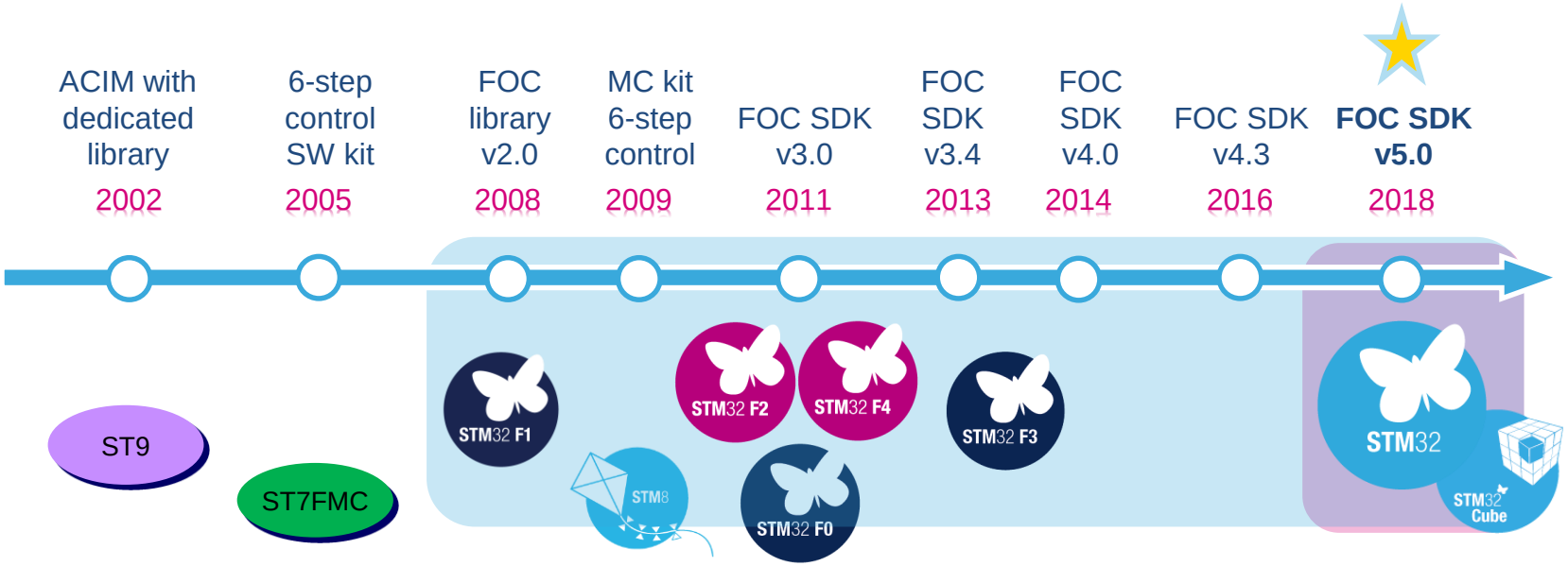
FW library

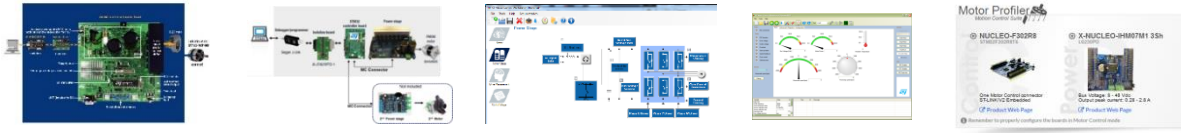
wide range of features & algorithms (FOC – 6step)

Software Development Kit (SDK)

life.augmented

STM32 电机控制开发套件 5.0 概览-历史

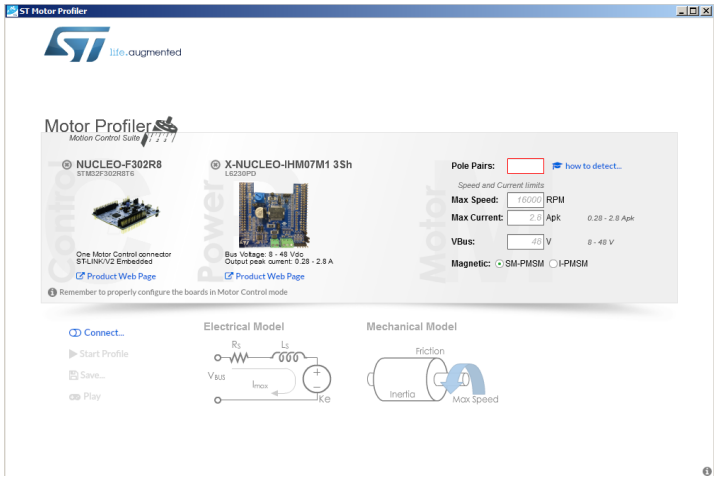




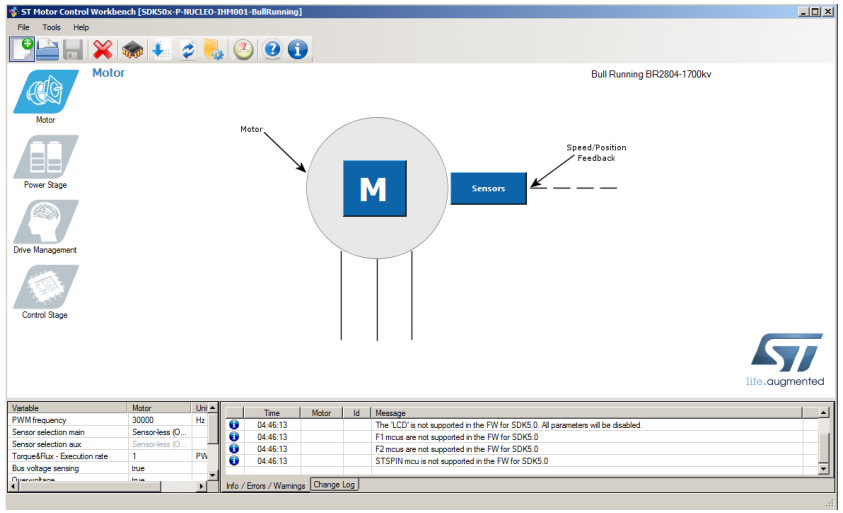
life.augmented

STM32 电机控制开发套件 5.0 概览-SDK 5.0 内容

软件工具



ST Motor Profiler tool

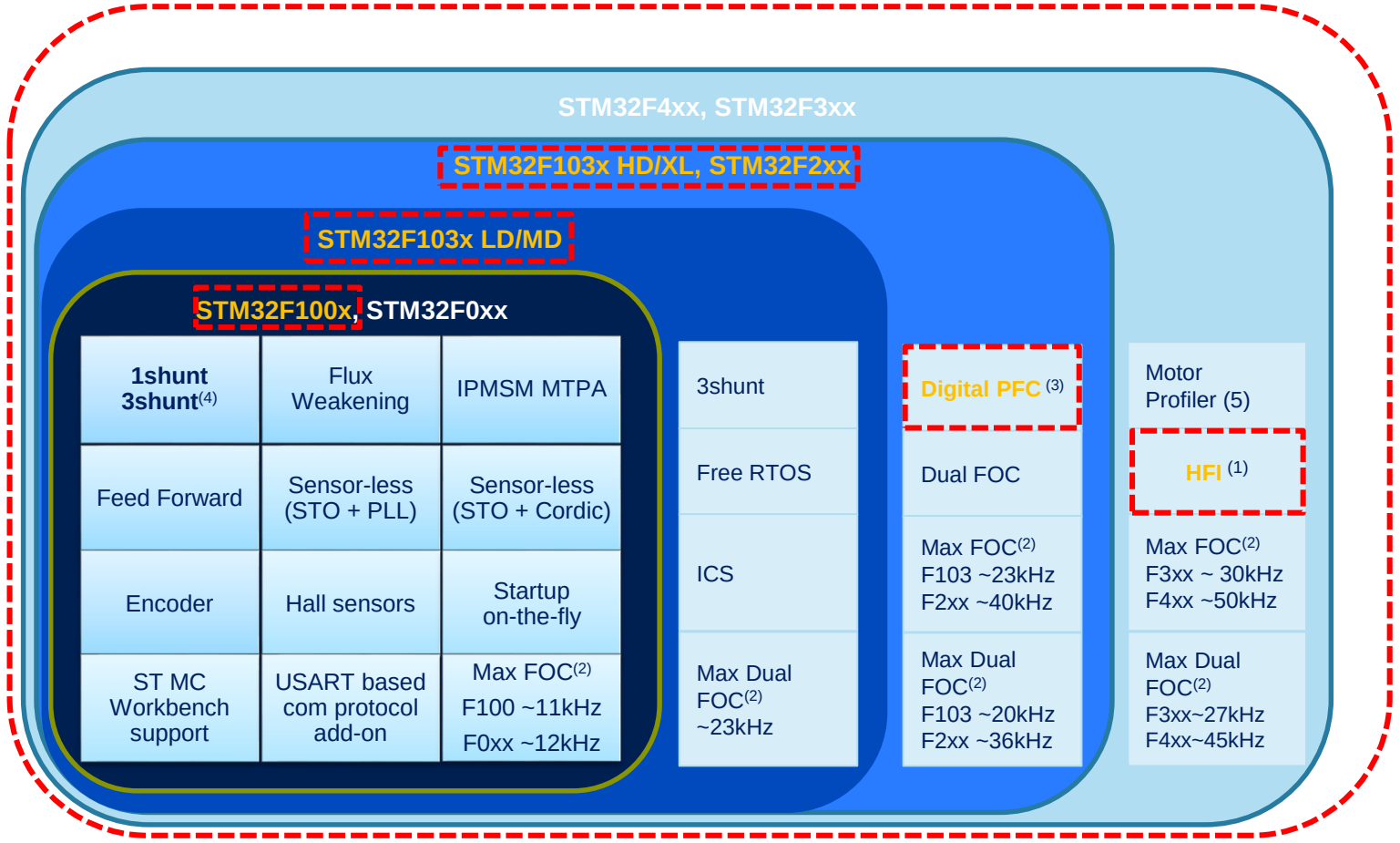


MotorControl Workbench

FOC 软件库

life.augmented

STM32 电机控制开发套件 5.0 概览-软件库功能



XXX **Features and STM32 series to be added in next MC SDK release**

(1)High Frequency Injection

(2)Max FOC estimated in sensorless mode

(3)STM32F103xC/D/E/F/G and STM32F303xB/C

(4)Not for STM32F100

(5)For STM32F30x

life.augmented

功能模块

Sensors

- Speed sensor:
  - HALL sensor
  - Encoder
  - Sensorless
- Current sensor:
  - 1 shunt
  - 3 shunt
  - ICS
- Bus voltage sensor
- Temperature sensor

Actuators

- PWM

Features

- FOC
- Flux weakening
- Feed forward
- Motor profiler
- Single/Dual Motor
- MTPA
- Ramp

PI regulators

- Speed PI
- Iq PI
- Id PI

Microcontrollers

- STM32F0
- STM32F3
- …

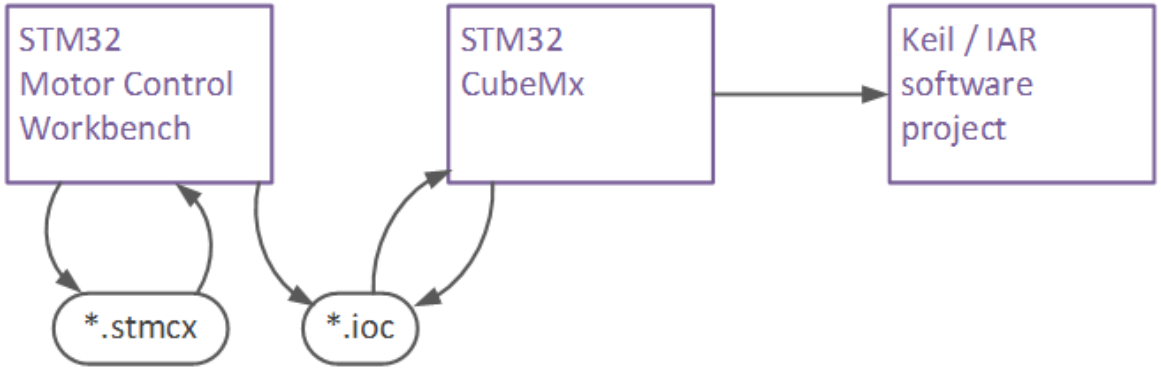
life.augmented

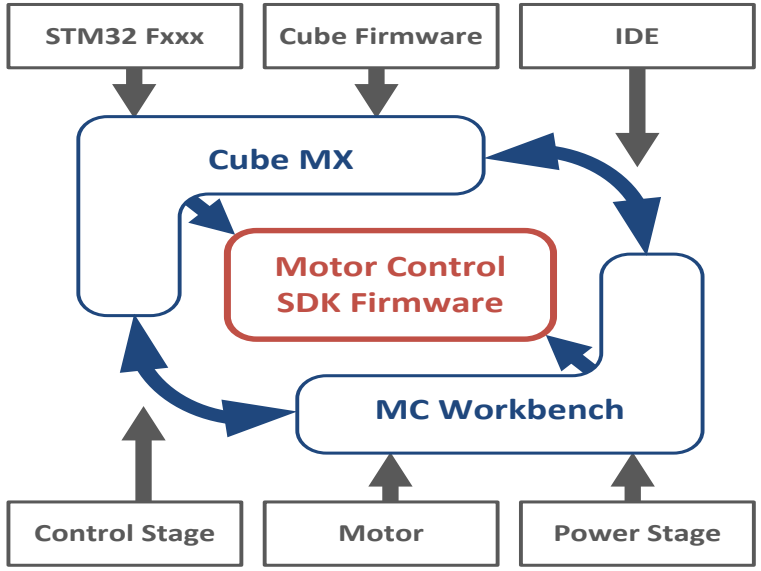
测试条件: PWM 频率20KHz / FOC 电流环控制频率

		MCSDK5.0					
MCU	Config	Workload (%).	Total Code size (kB).	ro data (B).	RAM (B).	FOC Lib (kB).	HAL (kB).
F072RB	1 Shunt	44.3	18.8	609	3126	12.8	5.1
F072RB	3 Shunt	39.4	19.5	653	2910	12.9	4.5
F303RE	1 Shunt	20.4	22.3	4427	2940	14.4	7.8
F303RE	3 Shunt	18.1	23.6	4179	2884	16.1	7.5
F446RE	1 Shunt	10.2	19.7	625	3122	14.4	5.3
F446RE	3 Shunt	8.2	17.8	603	2840	13.1	4.7
F303VE	DUAL/3S	38.2	20.8	4449	4724	13.1	7.7
F415ZG	DUAL/3S	18.3	19.3	761	4484	14.7	4.6

life.augmented

代码生成流程

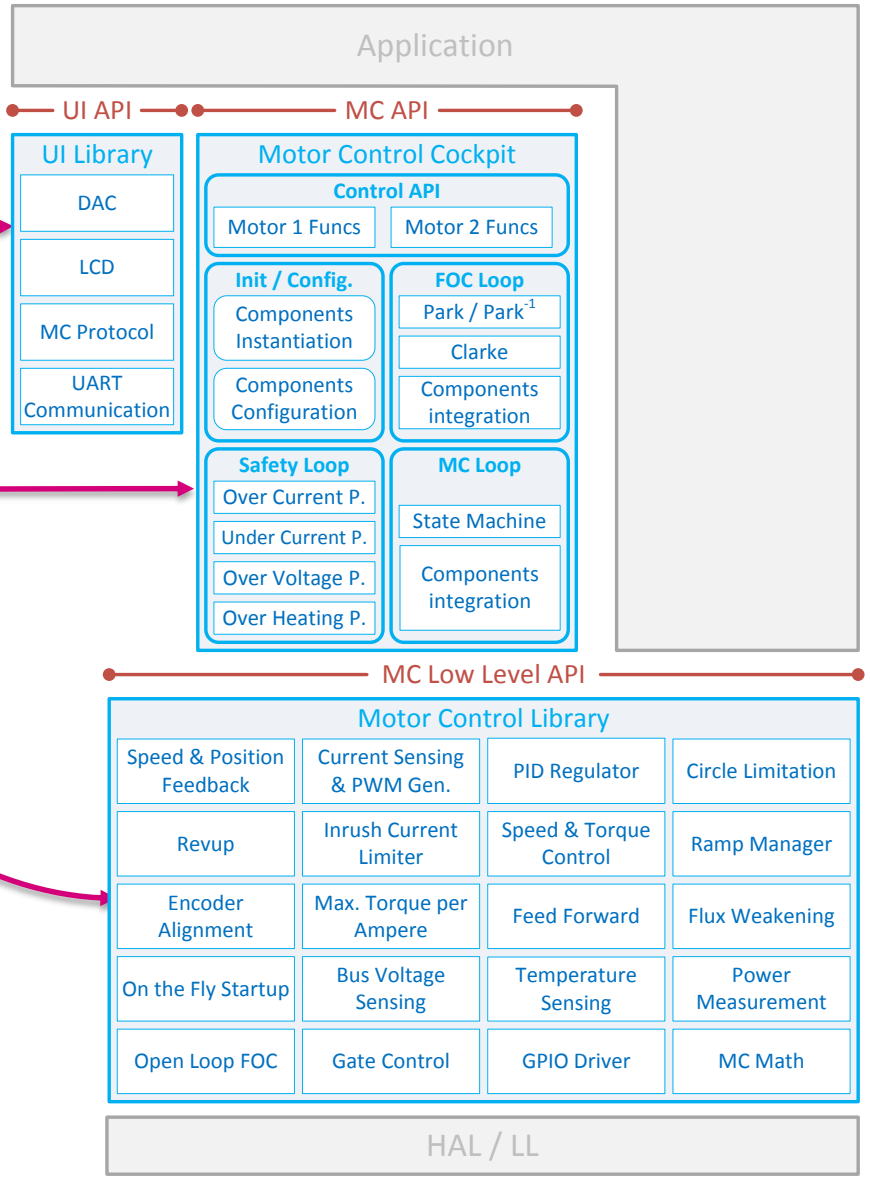




STM32 电机控制开发套件 5.0 概览-软件库架构

整个软件库由三部分组成

- 用户界面库
- 电机驾驶舱
- 电机控制库



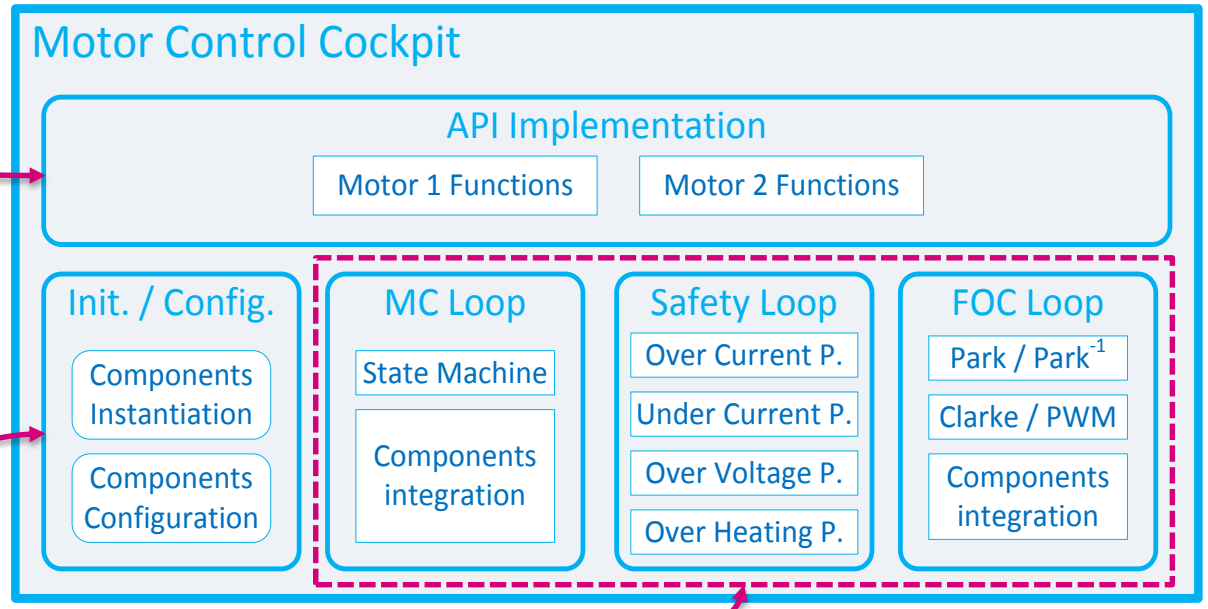
life.augmented

STM32 电机控制开发套件 5.0 概览-软件库架构

电机驾驶舱

电机控制接口
通过 MC API来实现

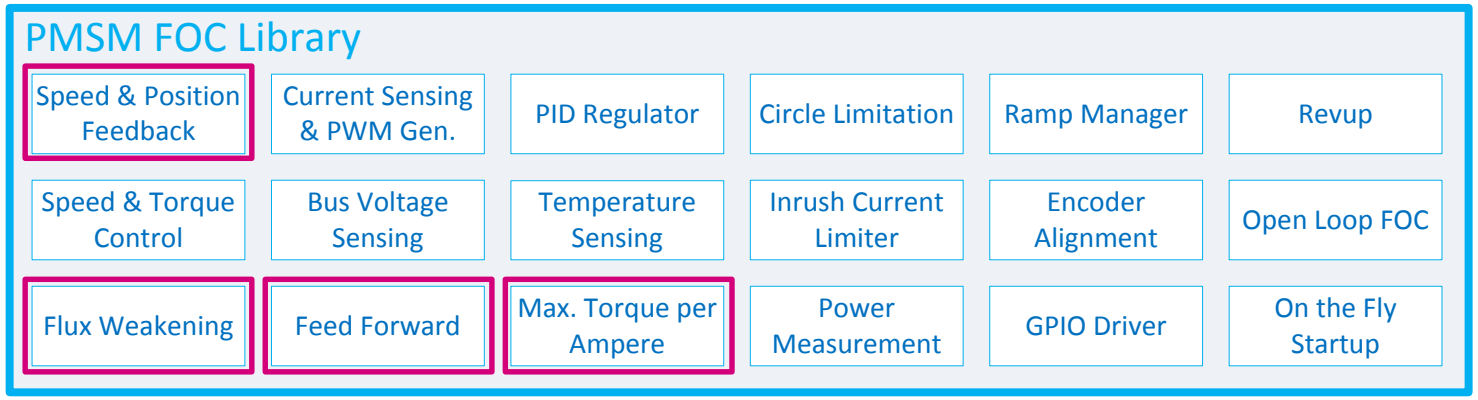
电机控制配置
实例化并配置所有需要的组件。



电机控制动态
实现对电机动的动态控制：
- FOC控制环路(高频任务)
- 电机控制环路(中频任务)
- 安全控制环路 (安全任务)

life.augmented

电机控制库



软件控制库

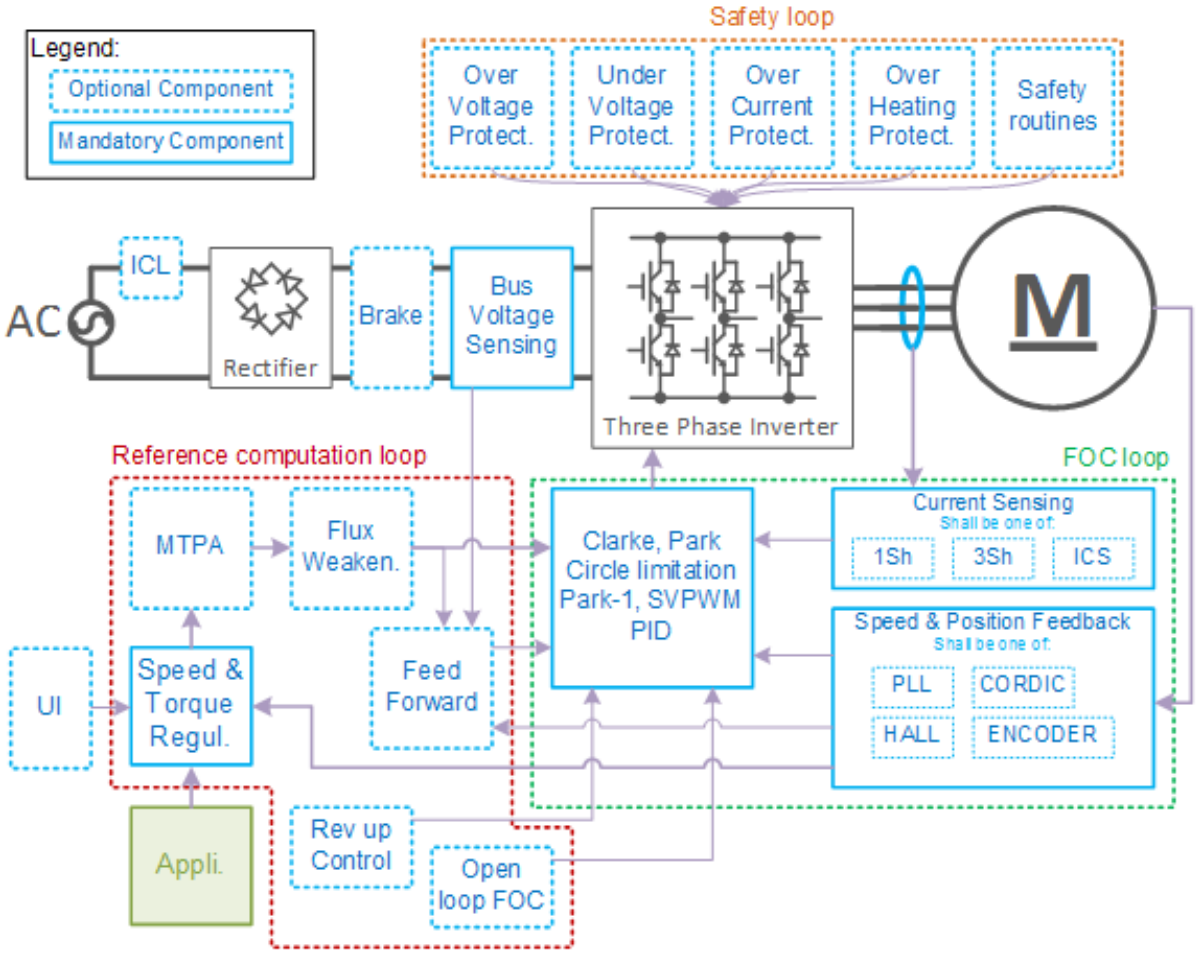
- 是诸多组件的集合。每一个组件实现电机控制的一个功能例如，速度和位置检测, 电流检测, PID算法等等…
- X-CUBE-MCSDK [ ] 中的组件不提供源代码，以库的形式提供。

life.augmented

用户界面库

用户界面库 包含负责通讯的组件。电机控制代码通过这些组件控制串口和DAC与外界通讯。通过这个库我们可以连接MCU和Workbench。在Workbench中实现对电机运行状态的监控。

STM32 电机控制开发套件 5.0 概览-软件库架构



程序由 Workbench 自动生成

基于用户在Workbench工程中的配置。

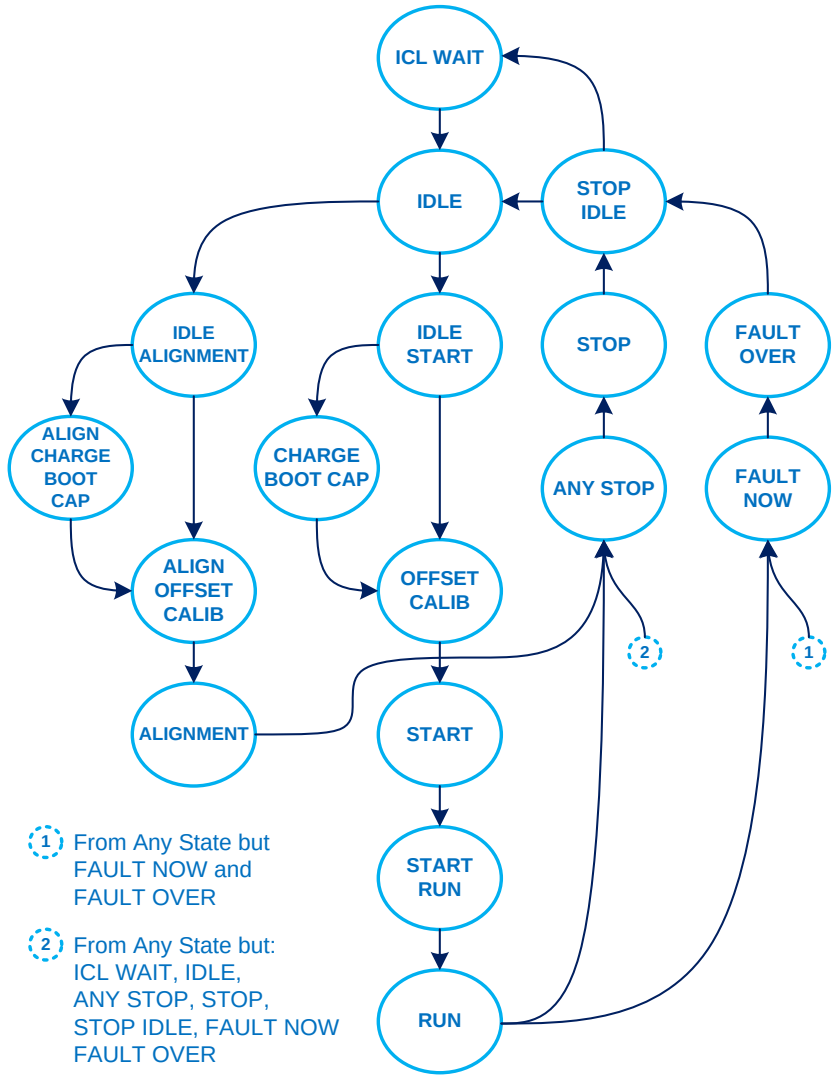
自动生成程序的优点

简短: 只生成真正需要的组件。

简单: 程序中没有(#ifdef / #else / #endif)等条件编译。

高效: 没有虚函数. 直接调用函数实现。

life.augmented



电机控制状态机

- 程序中每一个电机都有一个独立运行的状态机来控制电机状态。
- 用户不可以直接操作状态机，而应该通过 SDK 提供的 API 来改变电机状态。

life.augmented

STM32 电机控制开发套件 5.0 概览-从 SDK 4.3 到 SDK5.0 的改进

改进点 (相对与4.3)	改进的理念
简化程序结构(不再面向对象) (固件) • 不再面向对象编程 • 不再数据隐藏 • 删除虚拟函数	• 易于调试 • 易于扩展开发
基于STM32Cube (固件) • 用HAL/LL代替原来的SPL	• 与STM32 固件开发策略保持一致 • 与STM32新产品无缝对接。现在支持的系列包括F0,F2,F4,F3等，还会加入L4,F7,H7等新产品，及后续上市的更新的产品。STM32产品线全线覆盖
建立了CUBEMX 与 WORKBENTCH 之间的联系(工具) • Cube MX => 应用CUBEMX初始化全部所需要的MCU外围模块。	• 与STM32开发工具策略保持一致 • 应用CUBE MX对MCU的外围设备做初始化 • 简化用户开发流程

life.augmented

简化程序结构

SDK 4.3

PWMnCurrFdbkClass.c

```
110   *         of phase A and B in Curr_Components format.
111   * @retval none.
112   */
113  void PWMC_GetPhaseCurrents(CPWMC this,Curr_Components* pStator_Currents)
114  {
115    int16_t hIa, hIb;
116    pVars_t pVars = CLASS_VARS;
117    (CLASS_METHODS.pPWMC_GetPhaseCurrents)(this, pStator_Currents);
118    hIa = pStator_Currents->qI_Component1;
119    hIb = pStator_Currents->qI_Component2;
120    pVars->hIa = hIa;
121    pVars->hIb = hIb;
122    pVars->hIc = -hIa - hIb;
123  }
```

SDK 5.0

pwm_curr_fdbk.c

PWMC_GetPhaseCurrents(PWMC_Handle_t *, Curr_Components *)

```
108    * @retval none.
109   */
110  void PWMC_GetPhaseCurrents(PWMC_Handle_t *pHandle,Curr_Components* pStator_Currents)
111  {
112    pHandle->pFctGetPhaseCurrents(pHandle, pStator_Currents);
113  }
```

不再面向对象编程

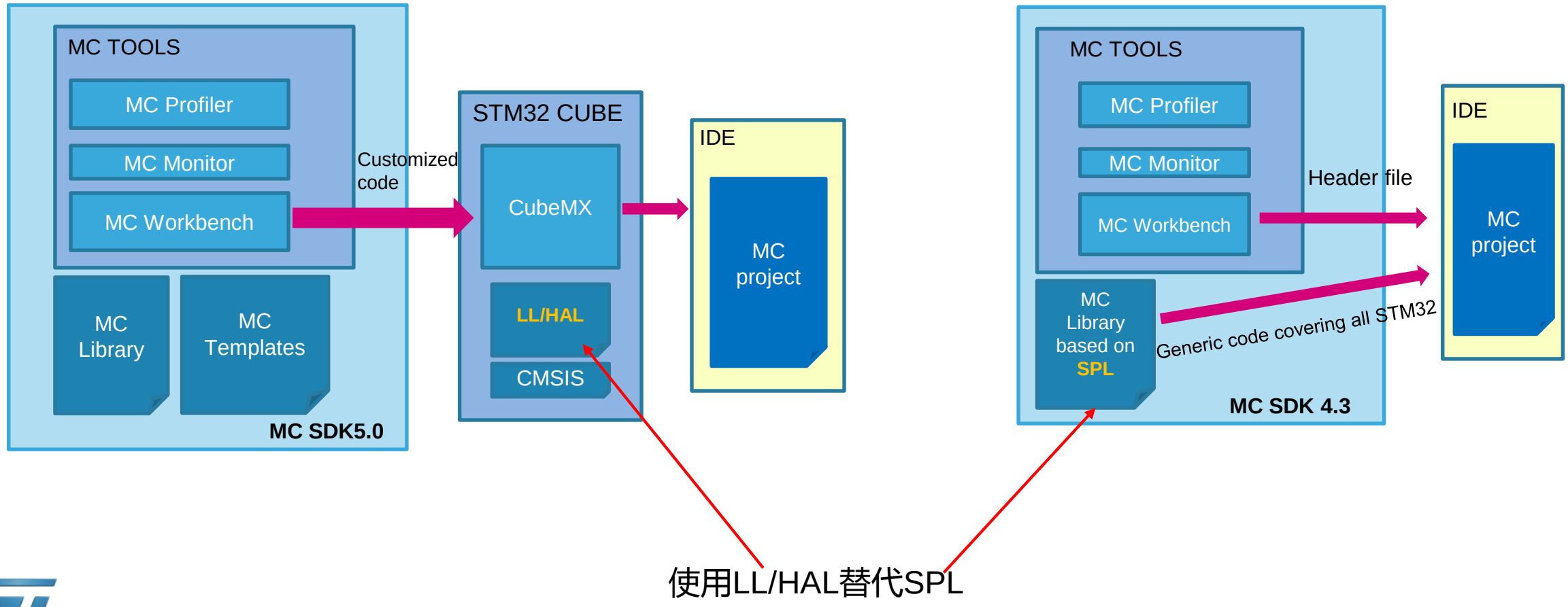
- 用组件替代类
- 用函数替代方法

不再数据隐藏

- 组件所有的全局变量都定义在 PWMC_Handle_t 里，这个结构体可以在文件 pwm_curr_fdbk.h 中找到. 在 SDK 4.3 中class 的部分数据成员隐藏在文件 PWMnCurrFdbkPrivate.h, 中调用时只在class内部才转换成能看到的数据结构.

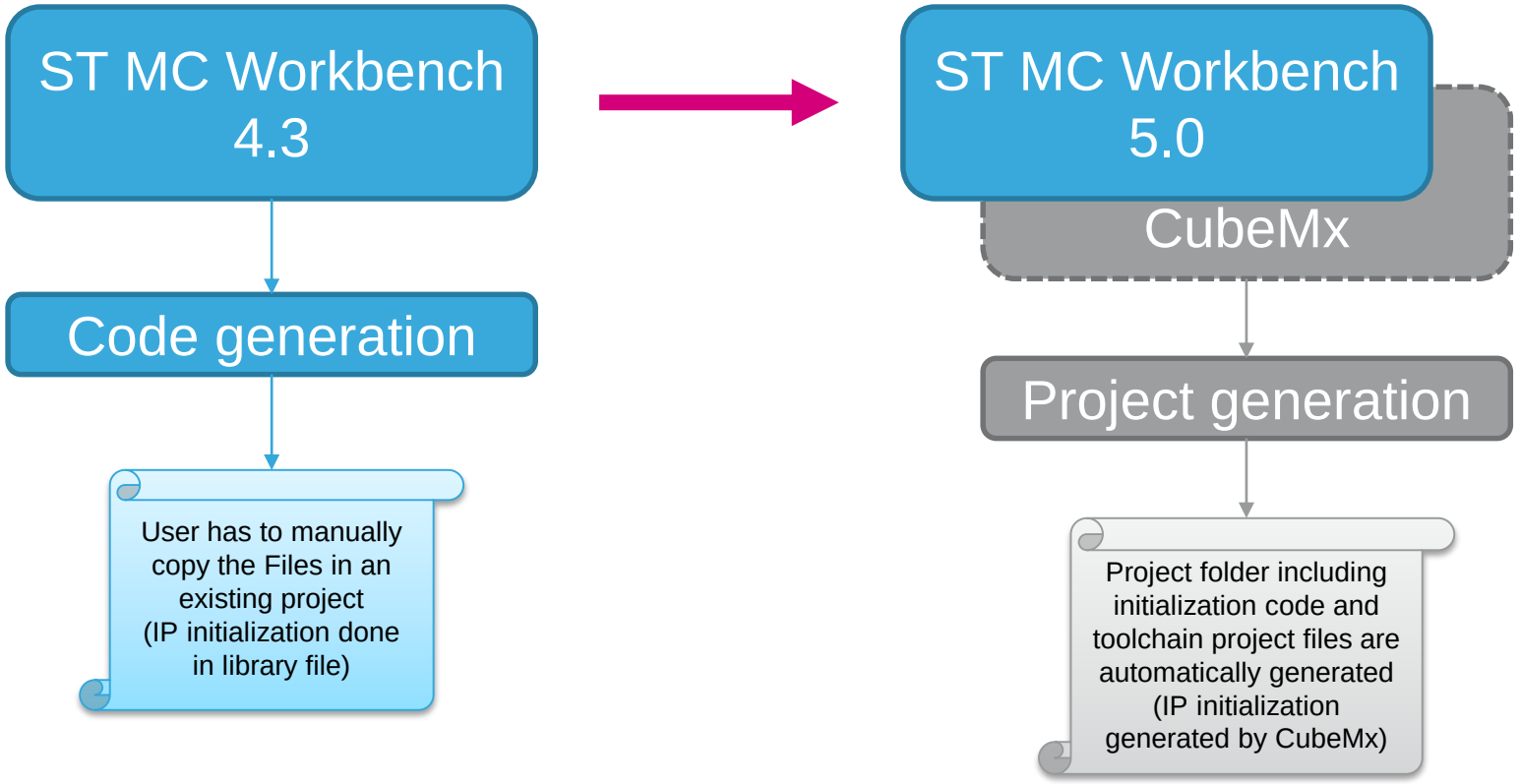
life.augmented

STM32 电机控制开发套件 5.0 概览-从 SDK 4.3 到 SDK5.0 的改进



life.augmented

STM32 电机控制开发套件 5.0 概览-从 SDK 4.3 到 SDK5.0 的改进

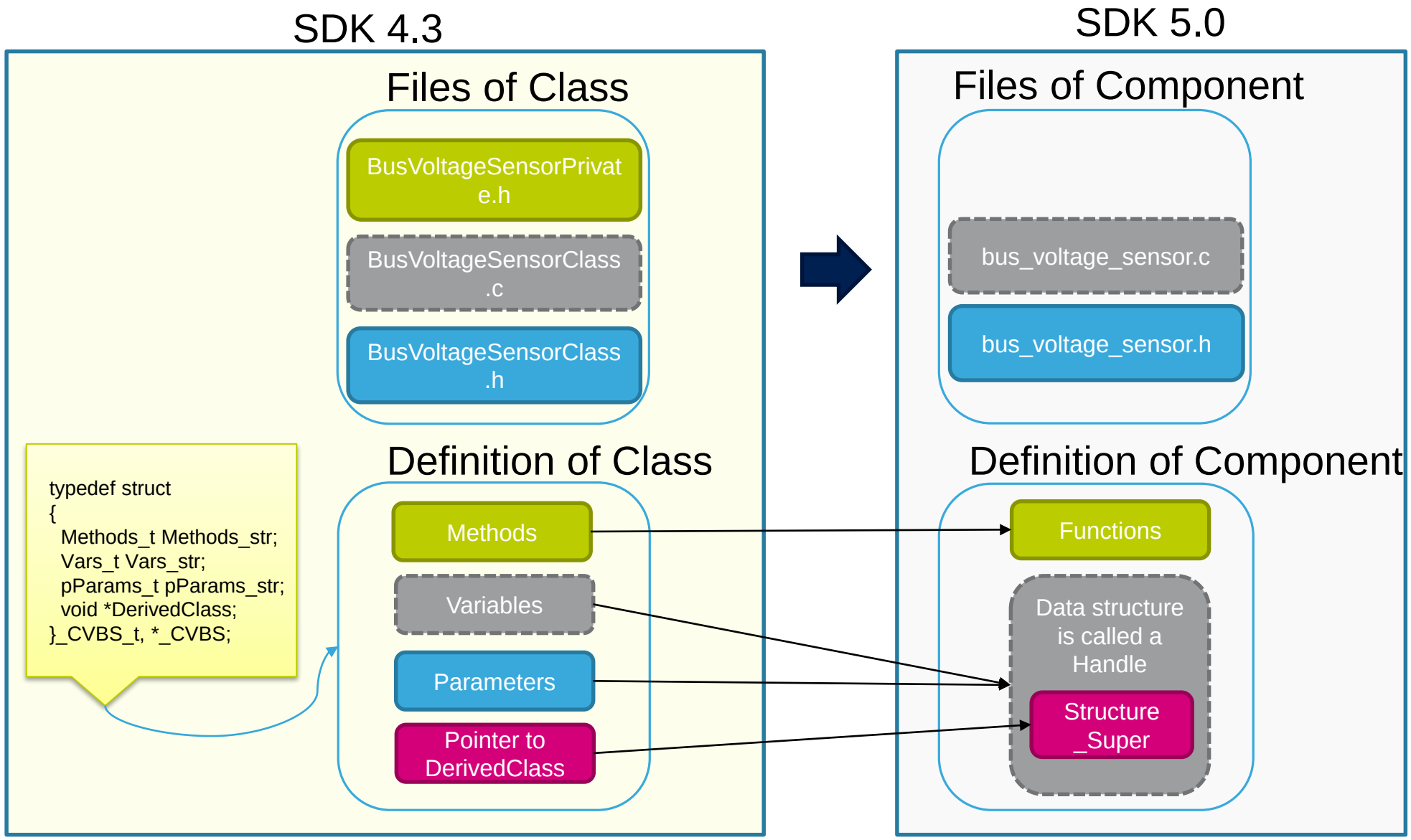


Cube MX 初始化外设

Workbench 调用 CubeMx 生成整个程序，程序中包含了所有必要的源文件和头文件。但 CubeMx 运行在后台，无需客户操作。从使用者角度，表面上所有的生成程序的工作都是在 Workbench 完成的。

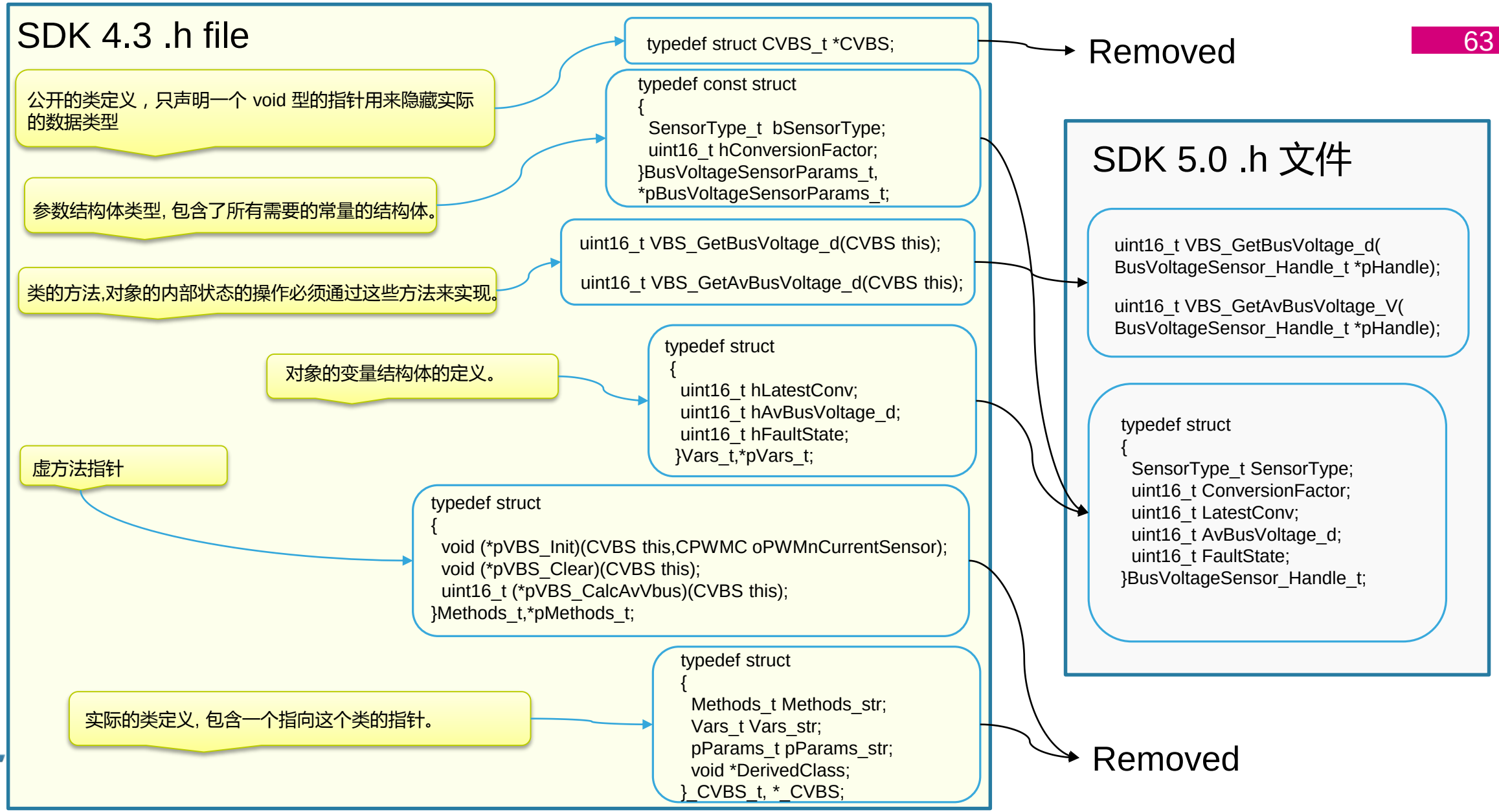
life.augmented

STM32 电机控制开发套件 5.0 概览-组件和类的对比



life.augmented

STM32 电机控制开发套件 5.0 概览-组件和类的对比



life.augmented

STM32 电机控制开发套件 5.0 概览-组件和类的对比

SDK 4.3 .c file

```
#define BCLASS_VARS   ((_CVBS)this)->Vars_str

uint16_t VBS_GetAvBusVoltage_d(CVBS this)
 {
   return(BCLASS_VARS.hAvBusVoltage_d);
 }
```

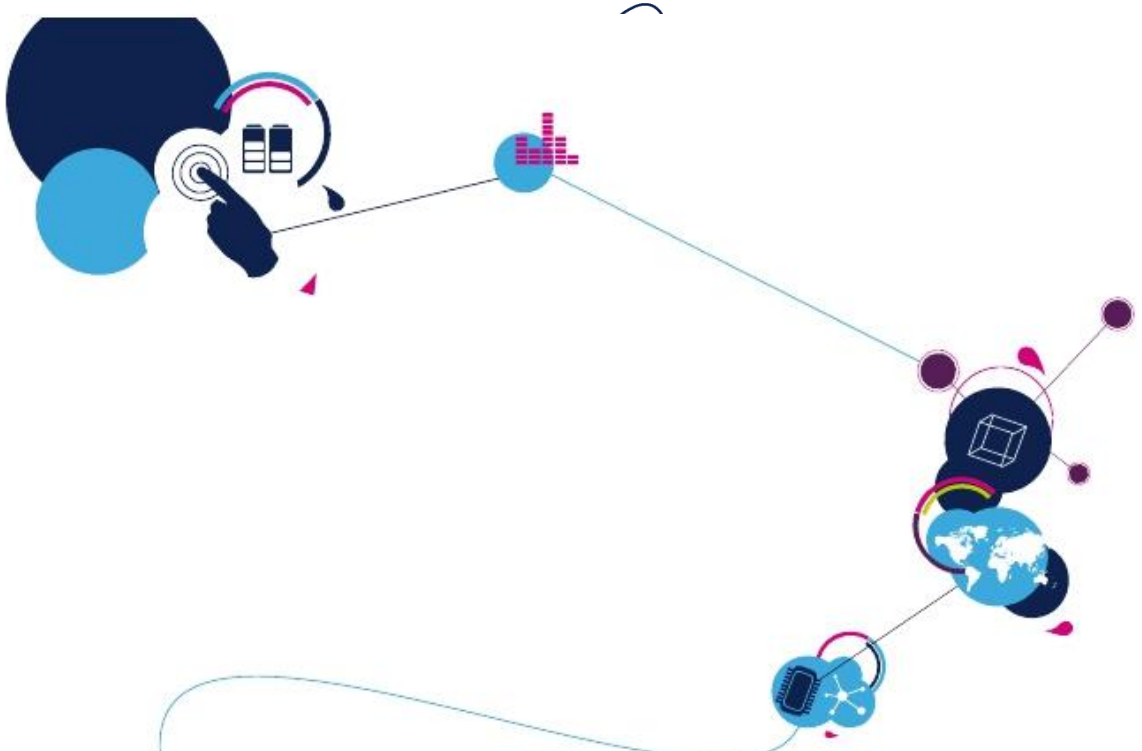
操作变量需要先将void*的指针转换成实际数据类型的指针，效率低.

SDK 5.0 .c file

```
uint16_t VBS_GetAvBusVoltage_d(BusVoltageSensor_Handle_t *pHandle)
 {
   return(pHandle->AvBusVoltage_d);
 }
```

没有数据隐藏直接操作变量所在的结构体。

life.augmented



SDK 5.0 软件开发

life.augmented

life.augmented

- **IDE 是 IAR Embedded Workbench 8**
- **开发板选用 P-NUCLEO-IHM001 或者 P-NUCLEO-IHM002**

低成本电机开发套件

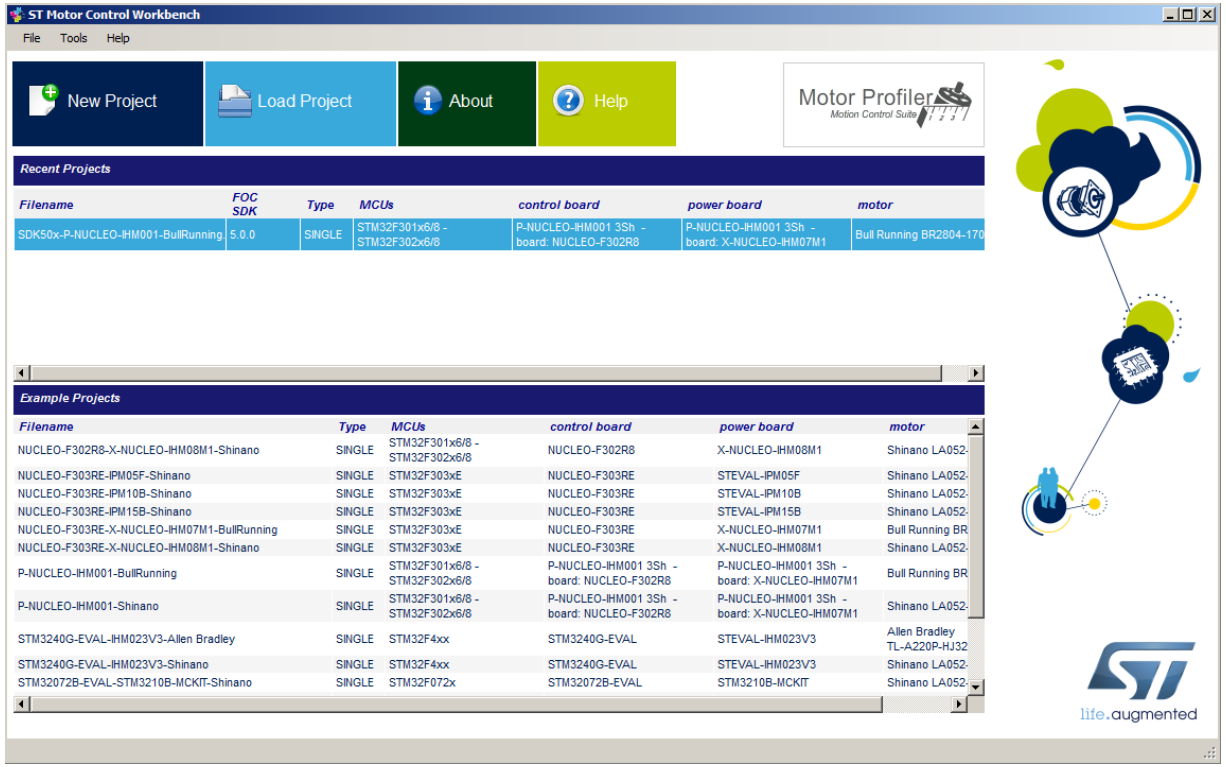
此开发套件被设计用于评估 BLDC 电机控制(最高电压 36V, 最大电流 1.4 A)

整个套件包括

- NUCLEO-F302R8
  基于 STM32F302 的控制板。
- X-NUCLEO-IHM07M1
  基于 L6230 的功率板。
- BLDC 电机
- 12V 电源.（仅P-NUCLEO-IHM002包含）

life.augmented

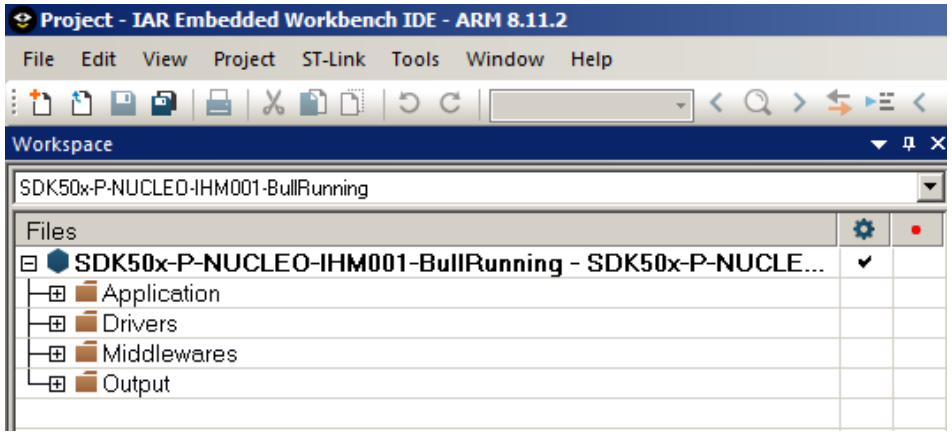
SDK 5.0 软件开发-开发平台介绍



打开 MotorControl Workbench 5.0.3. 在 Example Project 窗口中有很多的例程. 我们来打开 SDK50x-P-NUCLEO-IHM001-BullRunning.stmcx 来生成一个 IAR 的例程代码，然后打开这个工程。

life.augmented

SDK 5.0 软件开发-工程的详细结构



在打开的工程中，有四个文件夹。

- Application.
  和客户的应用紧密相关的程序。通常客户只需要更改这一部分的代码。我们在后面会有更详细的说明。

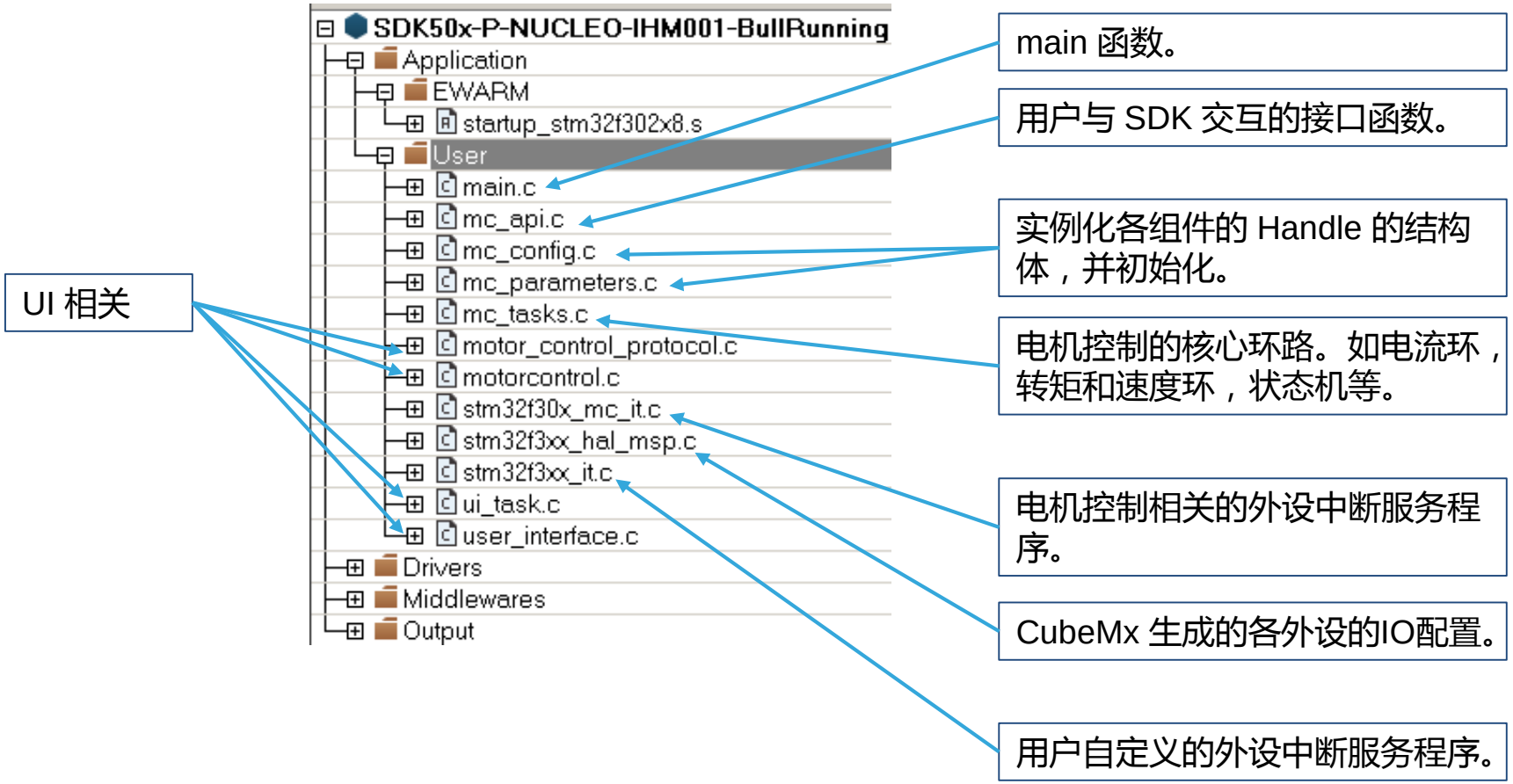
- Drivers
  MCU 的外设驱动代码，由 CubeMx 生成。另外还包含 ARM 的CMSIS 库.

- Middleware
  电机控制库中的各个组件的代码和库文件。

- Output
  链接器输出的文件如.out文件， .map 文件等。

life.augmented

SDK 5.0 软件开发-工程的详细结构



life.augmented

SDK 5.0 软件开发-工程的详细结构


```c
int main(void)
{
  /* USER CODE BEGIN 1 */

  /* USER CODE END 1 */

  /* MCU Configuration--------------------------

  /* Reset of all peripherals, Initializes th
  HAL_Init();

  /* USER CODE BEGIN Init */

  /* USER CODE END Init */

  /* Configure the system clock */
  SystemClock_Config();

  /* USER CODE BEGIN SysInit */

  /* USER CODE END SysInit */

  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_ADC1_Init();
  MX_DAC_Init();
  MX_TIM1_Init();
  MX_USART2_UART_Init();
  MX_MotorControl_Init();

  /* Initialize interrupts */
  MX_NVIC_Init();
  /* USER CODE BEGIN 2 */

  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {

  /* USER CODE END WHILE */

  /* USER CODE BEGIN 3 */

  }
  /* USER CODE END 3 */

}
```

在 main.c 文件中，函数 int main(void) 最重要。它包含以下内容：

- System configure
- Peripherals configure
- Motor Control configure
- Interrupt configure
- Main loop

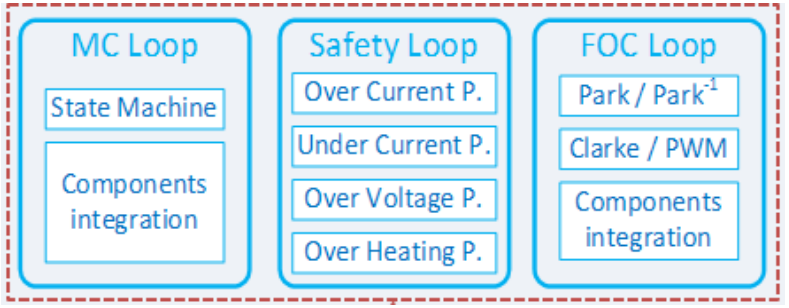
life.augmented

SDK 5.0 软件开发-工程的详细结构

- MCboot(MCI_Handle_t*[], MCT_Handle_t*[]) : void
- MC_Scheduler(void) : void
- TSK_MediumFrequencyTaskM1(void) : void
- FOC_Clear(uint8_t) : void
- FOC_InitAdditionalMethods(uint8_t) : void
- FOC_CalcCurrRef(uint8_t) : void
- TSK_SetChargeBootCapDelayM1(uint16_t) : void
- TSK_ChargeBootCapDelayHasElapsedM1(void)
- TSK_SetStopPermanencyTimeM1(uint16_t) : void
- TSK_StopPermanencyTimeHasElapsedM1(void)
- TSK_HighFrequencyTask(void) : uint8_t
- FOC_CurrController(uint8_t) : uint16_t
- MC_RequestRegularConv(uint8_t, uint8_t) : void
- MC_GetRegularConv(void) : uint16_t
- MC_RegularConvState(void) : URRC_State_t
- TSK_SafetyTask(void) : void
- TSK_SafetyTask_PWMOFF(uint8_t) : void
- GetMCI(uint8_t) : MCI_Handle_t*
- GetMCT(uint8_t) : MCT_Handle_t*
- TSK_HardwareFaultTask(void) : void
- mc_lock_pins(void) : void

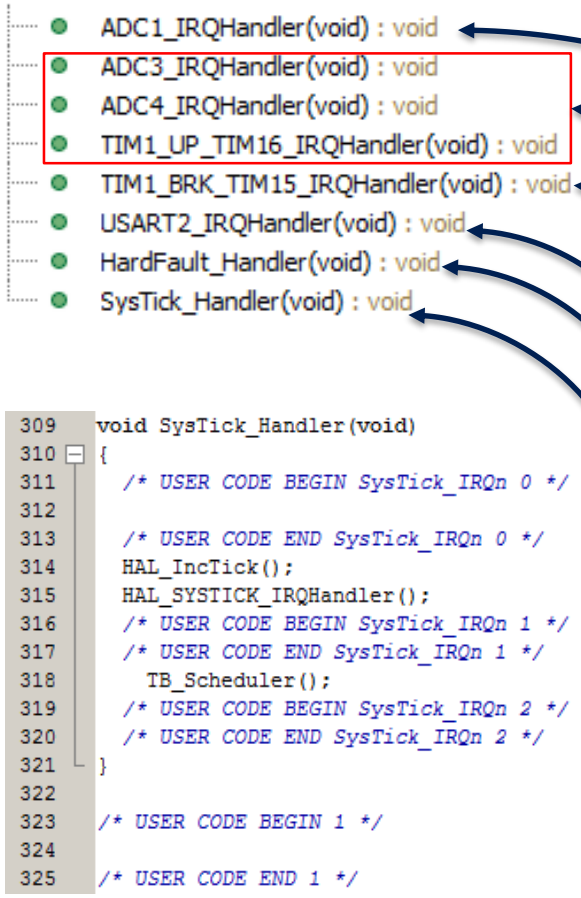
在文件 mc_tasks.c 中核心的是下面的几个环路：

- MC loop
- Torque or speed loop
- Current loop
- Safety loop



life.augmented

SDK 5.0 软件开发-工程的详细结构

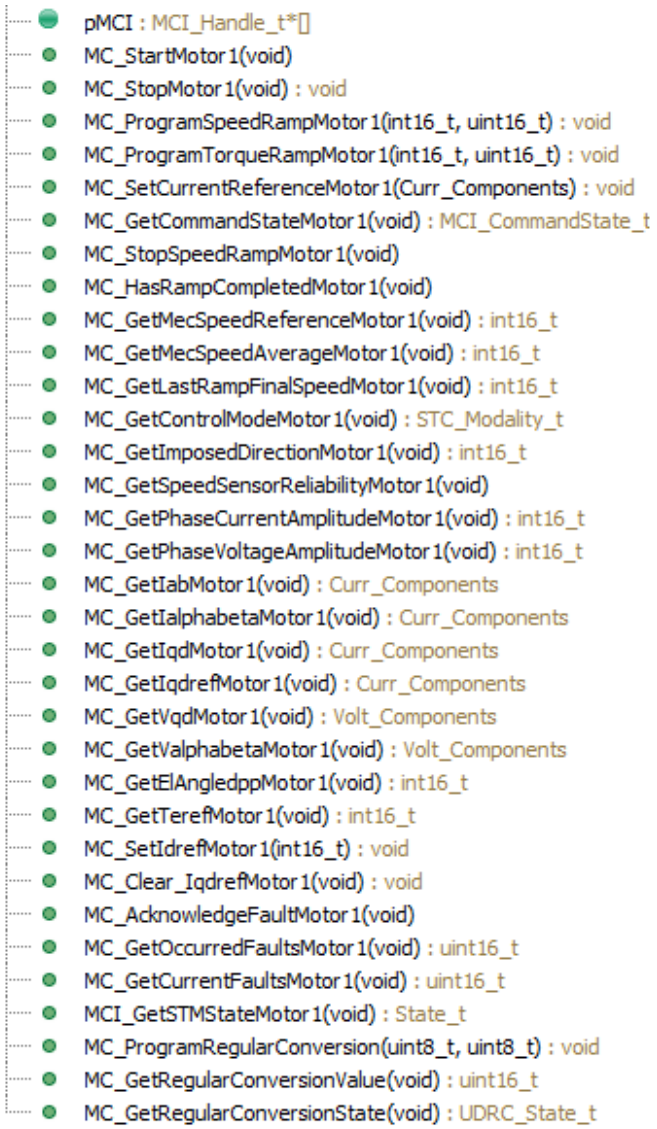


在文件 stm32f30x_mc_it.c 中有以下的中断服务程序：

- ADC 中断：HighFrequencyTask() 在这个函数中被调用。调用的频率可以在 workbench 中设定。最大的频率可以和 PWM 频率相同。
- 此例程中未启用。
- TIM1 Break 中断。
- UART 中断。
- System hard fault 中断。
- System Tick 中断：周期为 500us。

life.augmented

SDK 5.0 软件开发-工程的详细结构



在 mc_api.c 中有一系列的函数来实现与对电机的控制，我们称之为 MC API 。这些函数是用户和 SDK 之间的桥梁。下面列出了一些最常用的函数：

- MC_StartMotor1
- MC_StopMotor1
- MC_ProgramSpeedRampMotor1
- MC_ProgramTorqueRampMotor1
- MC_GetMecSpeedReferenceMotor1
- MC_GetMecSpeedAverageMotor1
- MCI_GetSTMStateMotor1
- MC_GetOccurredFaultsMotor1
- MC_AcknowledgeFaultMotor1
- MC_GetImposedDirectionMotor1

life.augmented

STM32 Motor Control SDK 可以用一片 STM32 MCU 同时驱动两台电机。为了简化程序 MC API 为每一台电机提供了一组单独的函数。这样把函数的参数限制到最简化。我们从函数的名字可以看出它是用来控制哪个电机的. 这是与 SDK 4.3 最主要的不同.

SDK 4.3	SDK 5.0
bool MCI_StartMotor(CMCI this)	bool MC_StartMotor1(void) bool MC_StartMotor2(void)
CMCI oMCI[MC_NUM]; MCboot(oMCI,oMCT) MCI_StartMotor(oMCI[M1])	Directly call function MC_StartMotor1()

life.augmented

如果 Motor 1 的状态机正处于 IDLE 状态, 命令被立即执行启动电机，返回值为 true。否则命令被丢弃，返回值为 false。调用者可以检查返回值来得知命令是被执行还是被丢弃。

```
bool MC_StartMotor1(void)
{
return MCI_StartMotor(
pMCI[M1] );
}
```

如果 Motor 1 的状态机正处于 RUN 或者 START 状态, 命令被立即执行停止电机。否则命令被丢弃。

```
void MC_StopMotor1(void)
{
MCI_StopMotor( pMCI[M1] );
}
```

life.augmented

设置速度指令。. 在调用此函数后速度指令在 hDurationms 设置的时间内由当前速度变化到 hFinalSpeed 设置的目标速度。

```
void MC_ProgramSpeedRampMotor1( int16_t hFinalSpeed, uint16_t hDurationms )
{
    MCI_ExecSpeedRamp( pMCI[M1], hFinalSpeed, hDurationms );
}
```

设置转矩指令。在调用此函数后转矩指令在 hDurationms 设置的时间内由当前转矩变化到 hFinalTorque 设置的目标转矩。

```
void MC_ProgramTorqueRampMotor1( int16_t hFinalTorque, uint16_t hDurationms )
{
    MCI_ExecTorqueRamp( pMCI[M1], hFinalTorque, hDurationms );
}
```

life.augmented

返回 Motor 1 转子当前的指令机械转速, 数字量1代表0.1Hz。

```
int16_t MC_GetMecSpeedReferenceMotor1(void)
{
    return MCI_GetMecSpeedRef01Hz( pMCI[M1] );
}
```

返回 Motor 1 转子当前实测的平均机械转速, 数字量1代表0.1Hz。

```
int16_t MC_GetMecSpeedAverageMotor1(void)
{
    return MCI_GetAvrgMecSpeed01Hz( pMCI[M1] );
}
```

life.augmented

返回 Motor 1 状态机当前的状态。下面表格列举了一些常用的电机状态。

```
State_t  MCI_GetSTMStateMotor1(void)
{
    return MCI_GetSTMState( pMCI[M1] );
}
```

Frequently used state

State	Code	Simple Description
IDLE	0	Idle
START	4	In start up process
START_RUN	5	In start up process
RUN	6	Run normal
ANY_STOP	7	Stop
STOP	8	Stop
STOP_IDLE	9	Stop
FAULT_NOW	10	Fault
FAULT_OVER	11	Fault

life.augmented

返回 Motor 1 发生过的故障的代码。

```
uint16_t MC_GetOccurredFaultsMotor1(void)
{
    return MCI_GetOccurredFaults( pMCI[M1] );
}
```

Fault List

State	Code	Simple Description
MC_NO_ERROR	0x0000	No fault
MC_NO_FAULTS	0x0000	No fault
MC_FOC_DURATION	0x0001	FOC calculation time out
MC_OVER_VOLT	0x0002	Bus over voltage
MC_UNDER_VOLT	0x0004	Bus under voltage
MC_OVER_TEMP	0x0008	Over temperature
MC_START_UP	0x0010	Start up failure
MC_SPEED_FDBK	0x0020	Speed feedback is not reliable
MC_BREAK_IN	0x0040	Hardware break in
MC_SW_ERROR	0x0080	Software fault

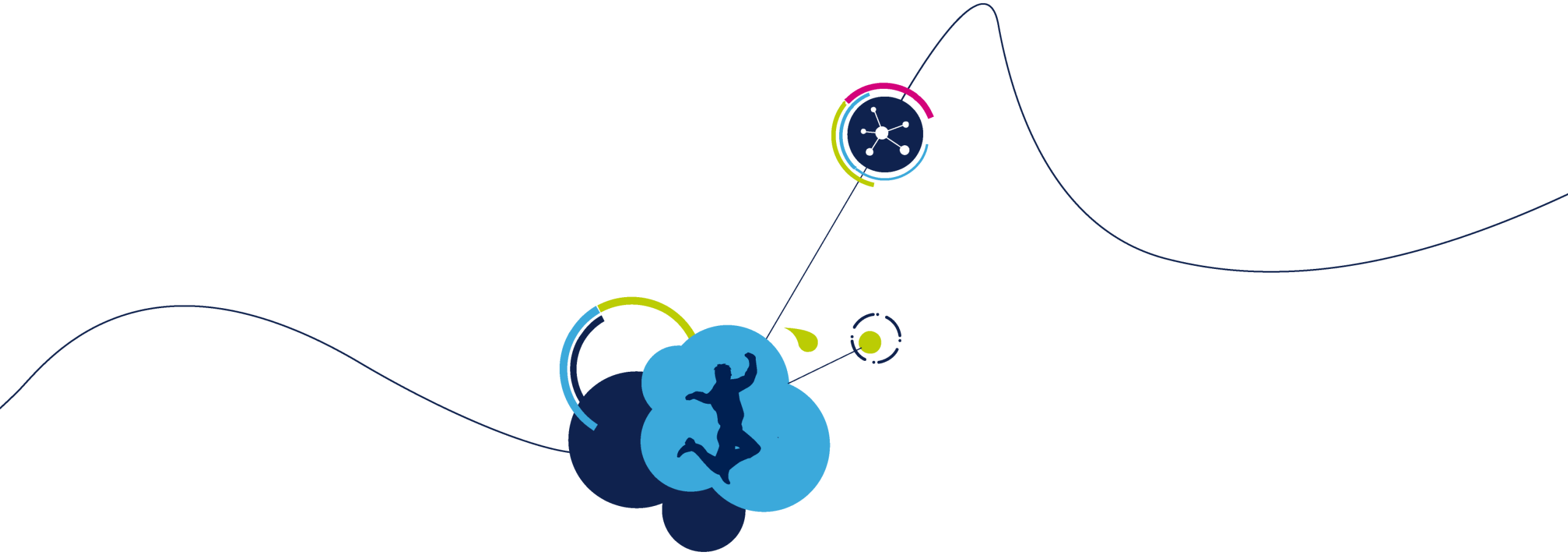
life.augmented

应答 Motor 1 发生过的故障。用户调用这个函数前，如果电机发生了故障。电机将停留在FAULT_OVER状态，并保留故障代码。在调用了这个函数之后，状态机将清除故障代码的记录，并恢复到 IDLE 状态。

```
bool MC_AcknowledgeFaultMotor1( void )
{
    return MCI_FaultAcknowledged( pMCI[M1] );
}
```

返回由最后一个指令设置的Motor 1的方向。如果最终的速度或转矩指令为负函数返回 -1，否则函数返回 1 。

```
int16_t MC_GetImposedDirectionMotor1(void)
{
    return MCI_GetImposedMotorDirection( pMCI[M1] );
}
```



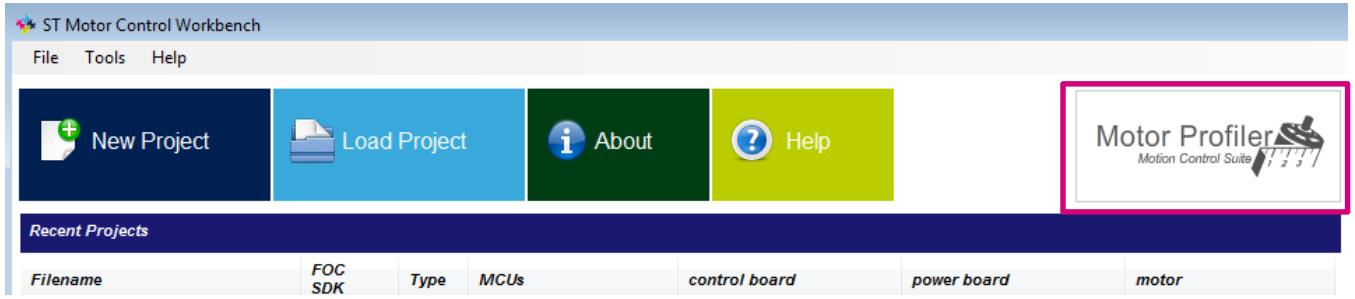
3.实验1：MC SDK5.0 电动机参数识别

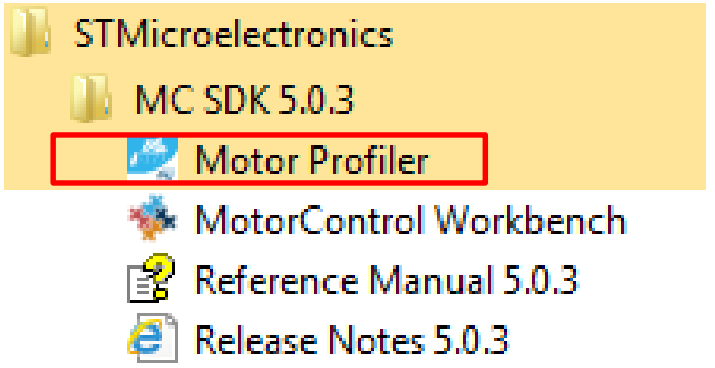
life.augmented

电机参数测量

可通过下列方式打开ST电机参数测量工具：

- 使用ST MC Workbench GUI上的专用按钮
- 直接从安装文件夹运行

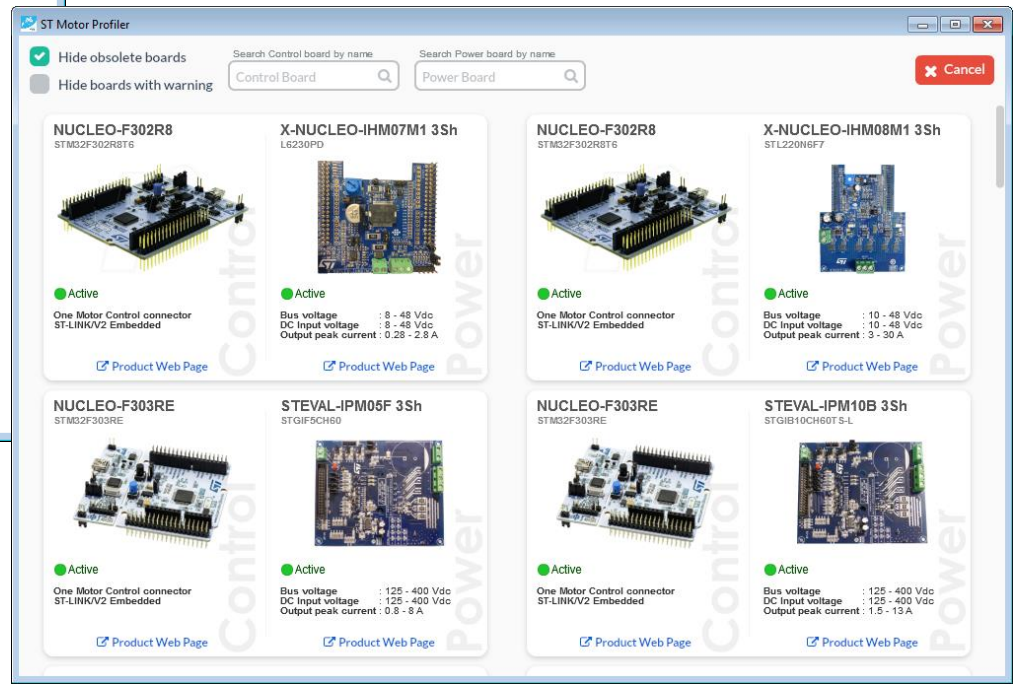




life.augmented

电机参数测量

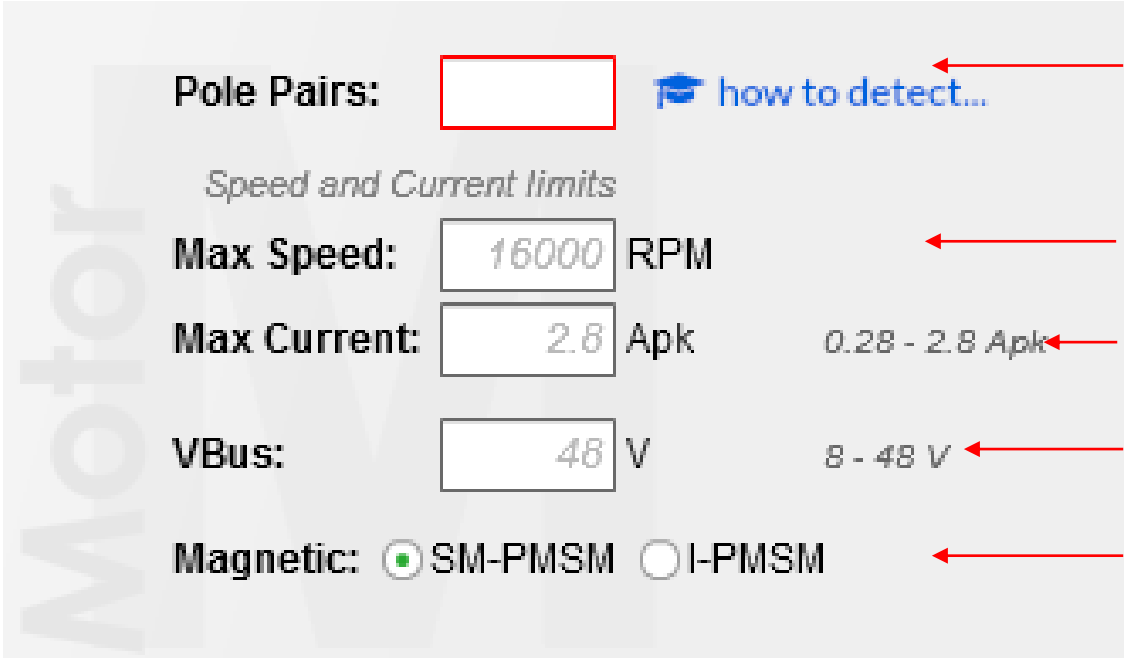
硬件搭配示例列表



life.augmented

电机参数测量

SM-PMSM参数示例



电机极对数 (必填)

电机最大转速 (可选)

电机允许的最大电流 (可选)

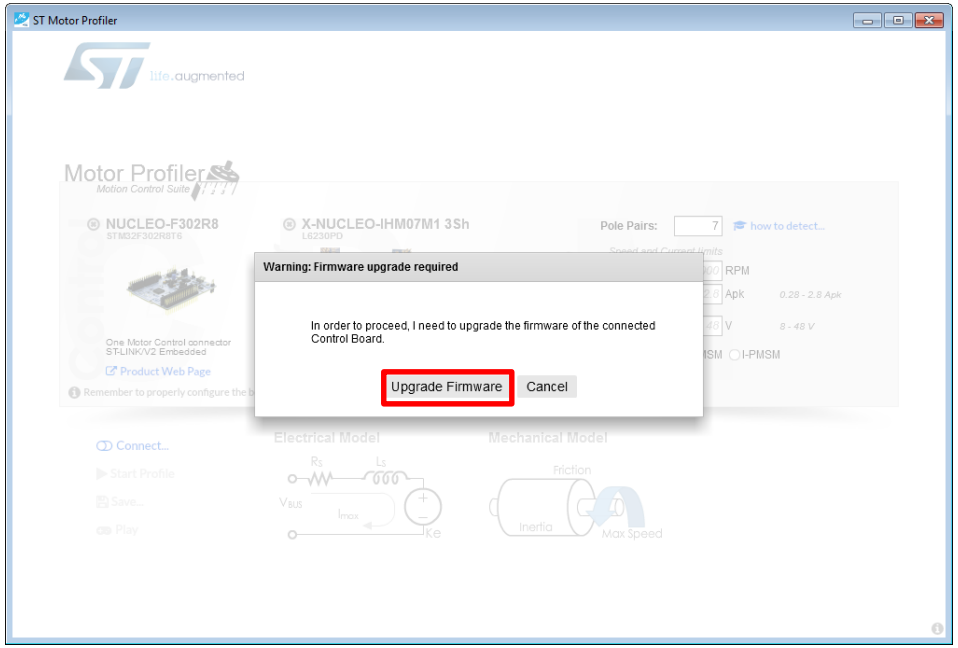
额定总线电压 (可选)

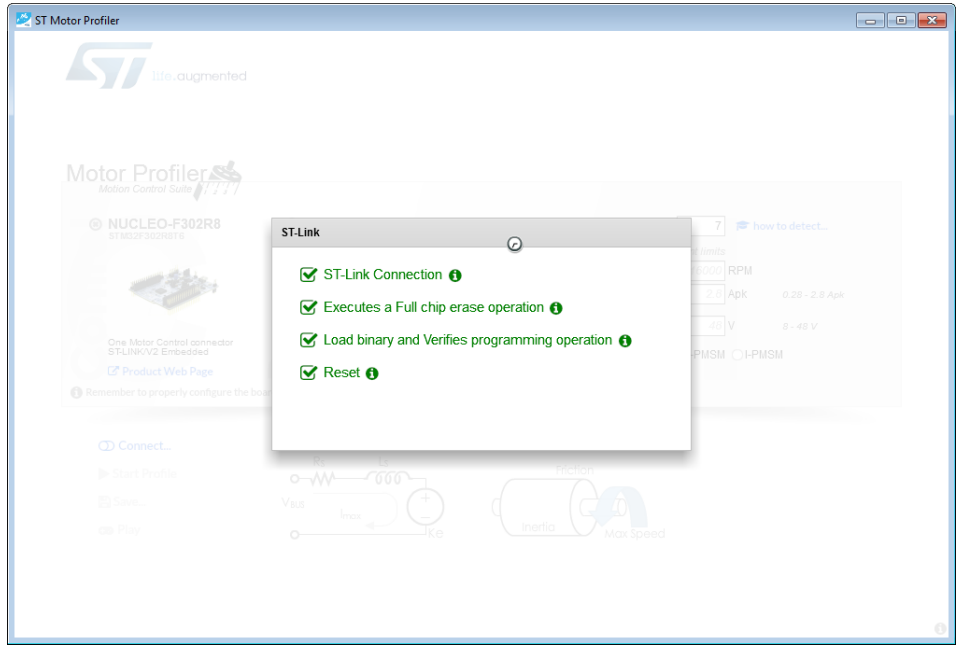
磁体内置类型 (必填)

life.augmented

电机参数测量

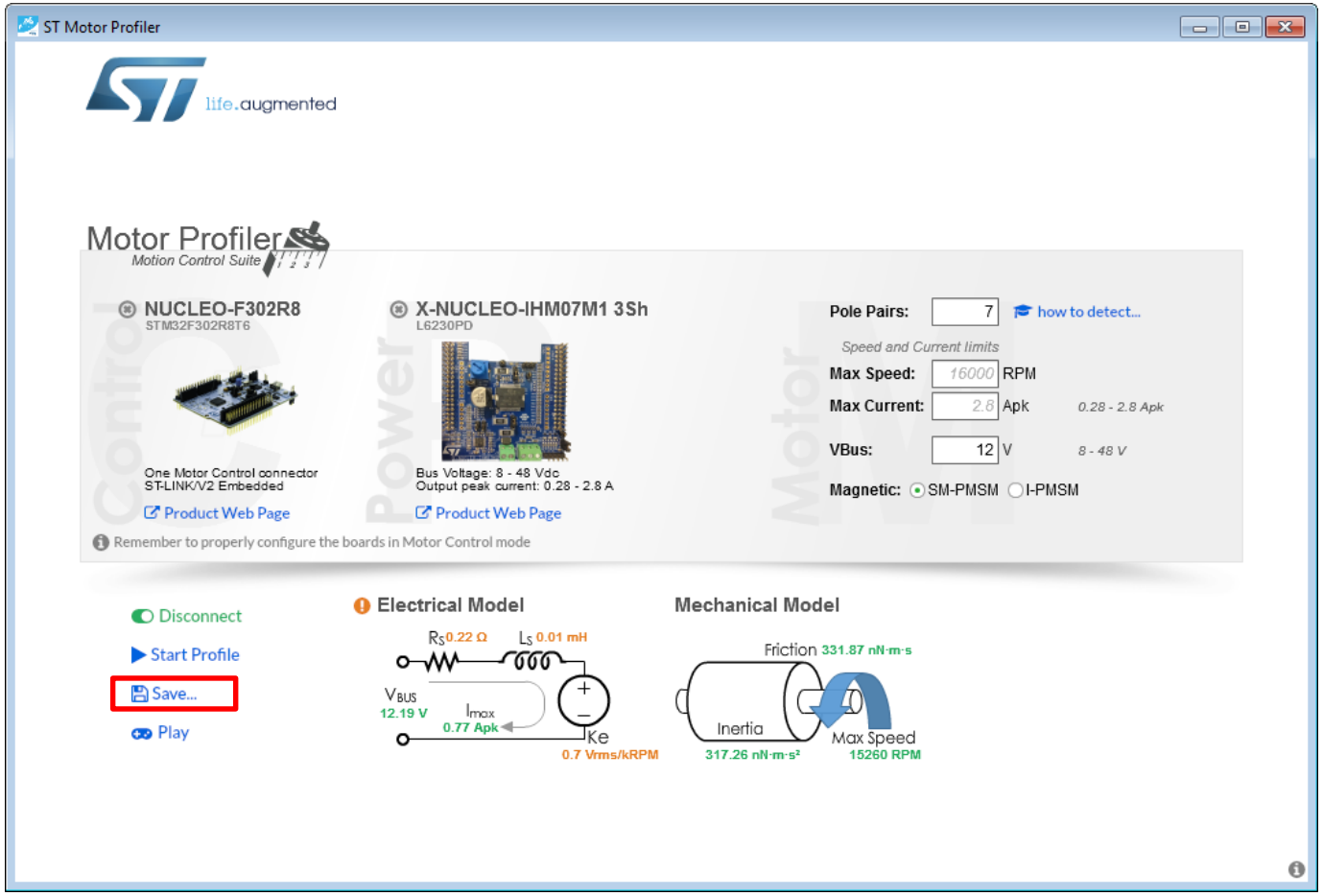
如需更新固件：

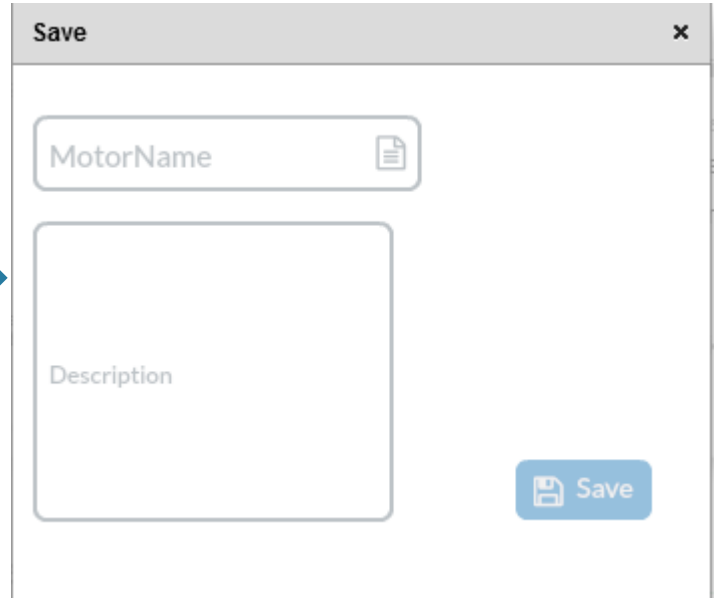




电机参数测量

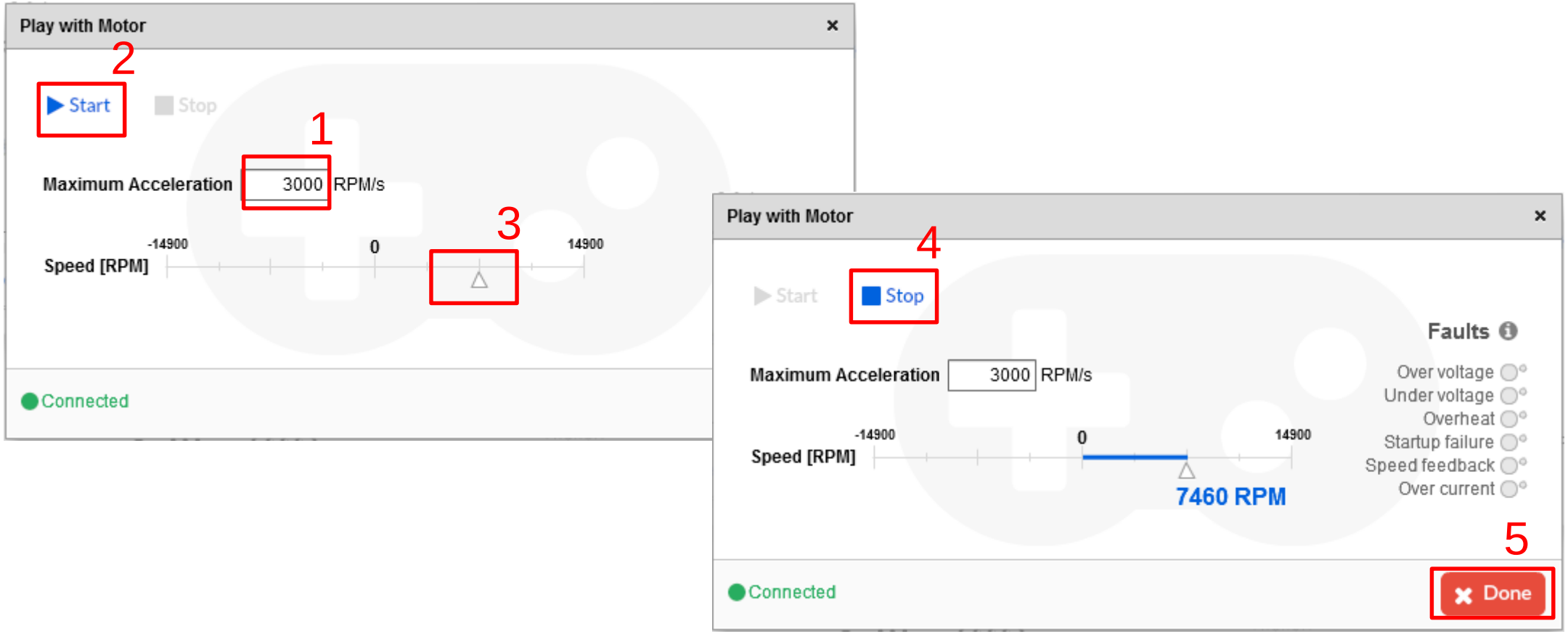
测量结果：

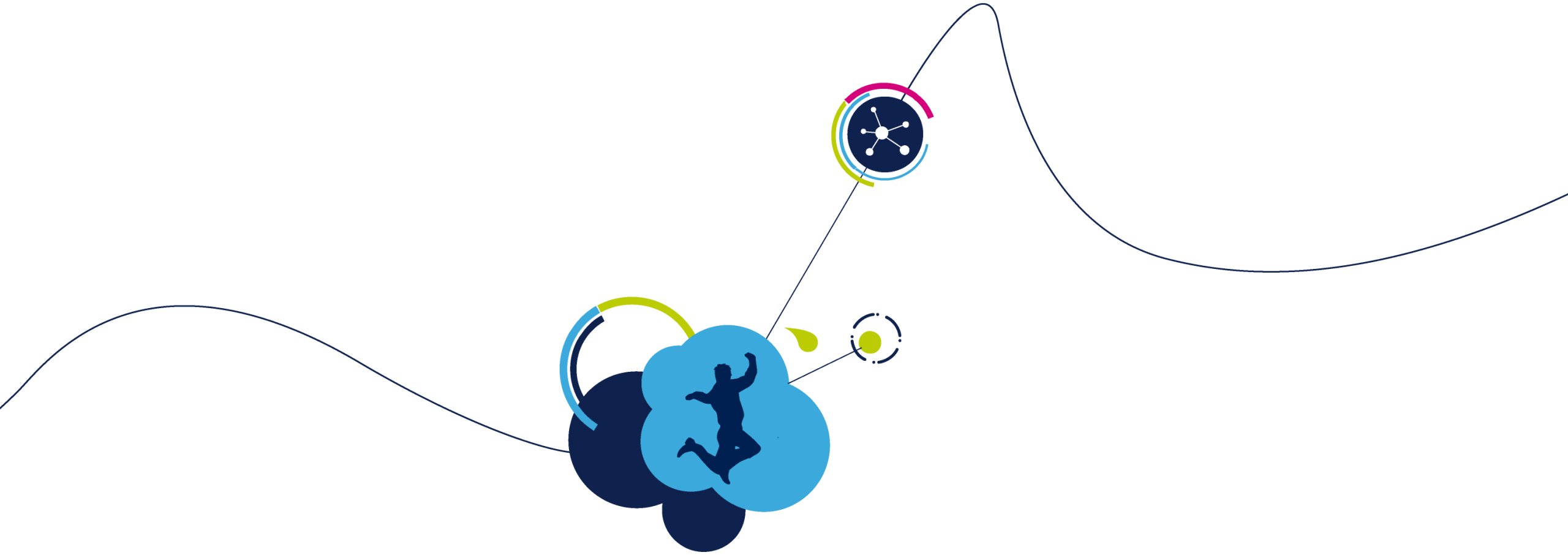




life.augmented

点击*Play*按钮可转动测量过的电机





4 .MC SDK5.0 工具链及图形用户界面

life.augmented

目录

- **MC SDK5.0 培训材料(IDE/GUI)**
  - 软件工具的下载和安装
  - ST MC Workbench
  - MC Project 的生成，编译和下载
  - 电机控制及监控
  - Workbench 关键配置参数
  - 电机参数测量

life.augmented

软件工具的安装

请预先安装下列PC软件工具：

- ST MC Workbench (v5.0.3)
- STM32CubeMx (v4.25)
- ST-LINK/V2 (v4.2.0)
- IDE:
  - IAR Embedded Workbench for Arm (v8.xx)
    http://www.iar.com/

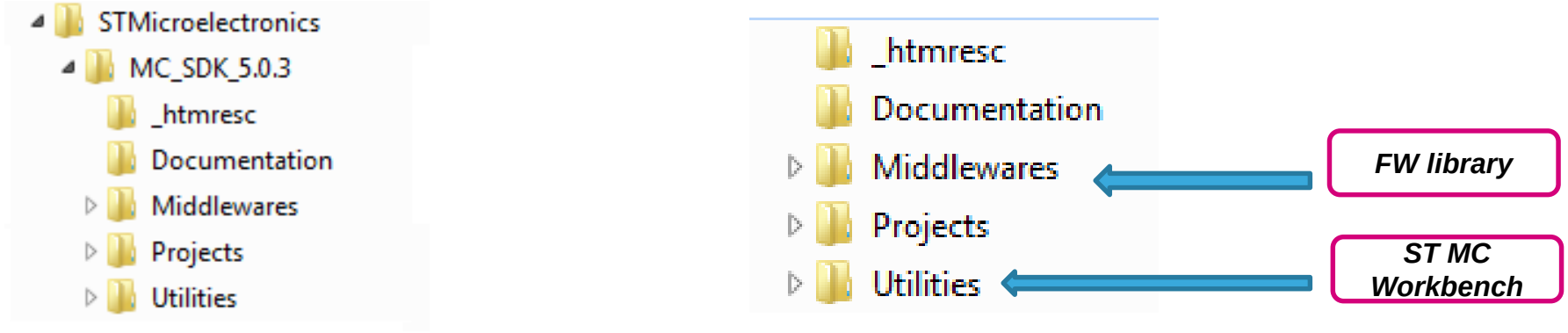
life.augmented

ST MC Workbench 的安装

请从上www.st.com下载STM32 MC SDK，并安装。

STM32 MC SDK 固件 (X-CUBE-MCSDK and X-CUBE-MCSDK-FUL) 包括PMSM 固件库(FOC控制) 和STM32 MC Workbench。

安装后，可以看到下列文件夹：



life.augmented

ST MC Workbench 相关文档

可从www.st.com上获得STMicroelectronics的文档：

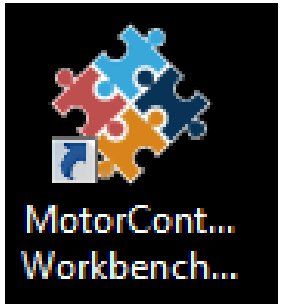
Title	Type	Contents
DB3548: STM32 MC SDK software expansion for STM32Cube	Data Brief	数据手册。MC SDK5.0作为CubeMX的一个扩展软件包。
UM2374: Getting started with STM32 motor control SDK v5.0	User Manual	MC SDK5.0入门手册
UM2380: STM32 motor control SDK v5.0 tools	User Manual	MC SDK5.0 外部工具(对应CubeMX而言)
AN5143: How to migrate motor control application software from SDK v4.3 to SDK v5.0	Application note	介绍如何从SDK4.3转到SDK5.0。
UM2312: Development checklist for STM32Cube Expansion Packages	User Manual	STMCube扩展软件包开发相关标准一览表
UM2285: Development guidelines for STM32Cube Expansion Packages	User Manual	STM32Cube 开发指导
UM1718: STM32CubeMx for STM32 configuration and initialization C code generation user manual	User Manual	STM32Cube 入门手册
UM0892: STM32 ST-LINK utility software description user manual	User Manual	STM32 ST-LINK用户手册

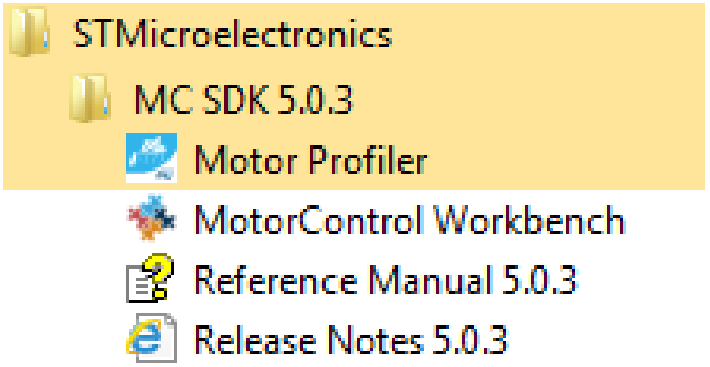
life.augmented

软件启动

下列方式可启动ST MC Workbench软件工具：

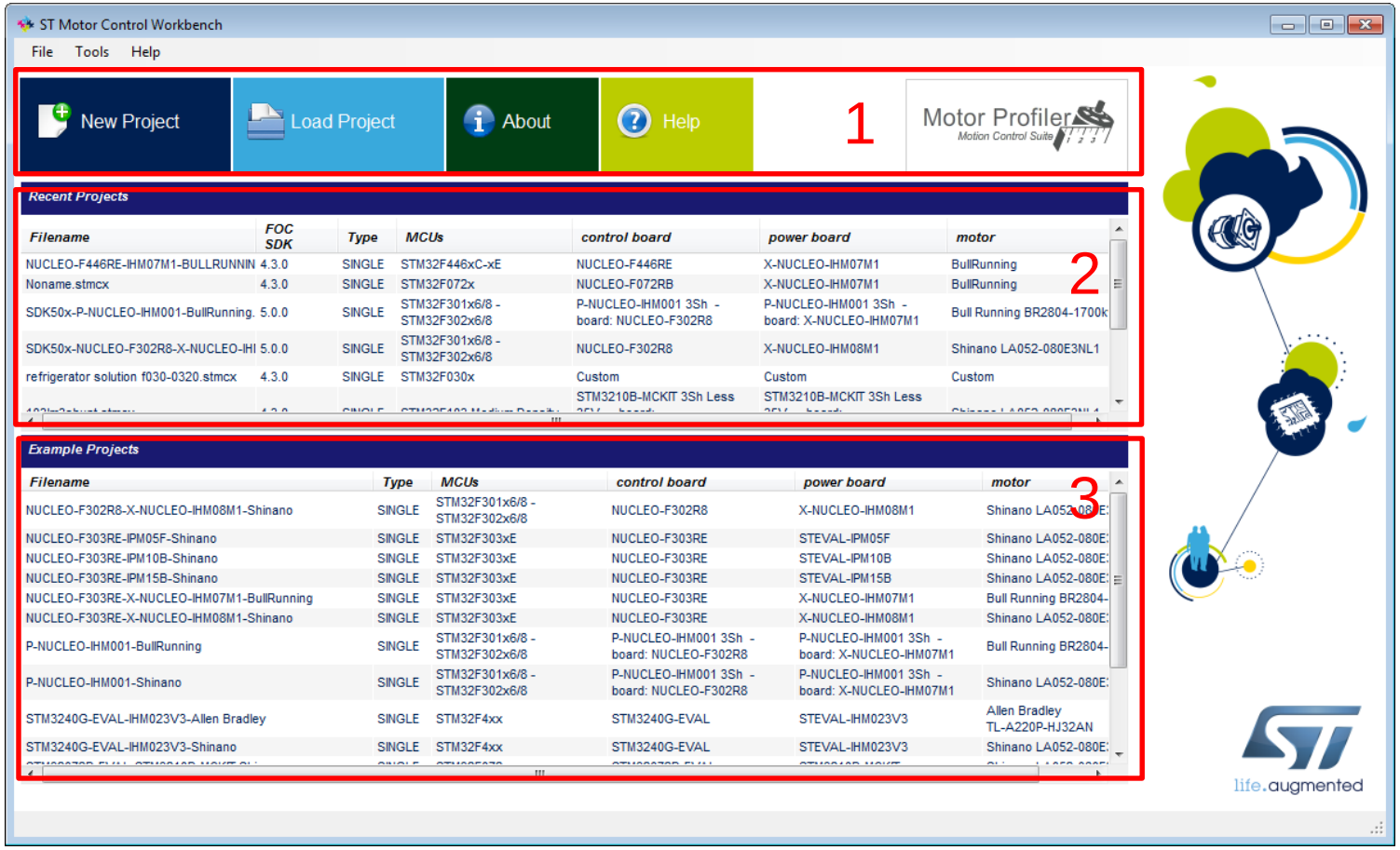
- 单击其图标
- 从安装文件夹路径直接启动





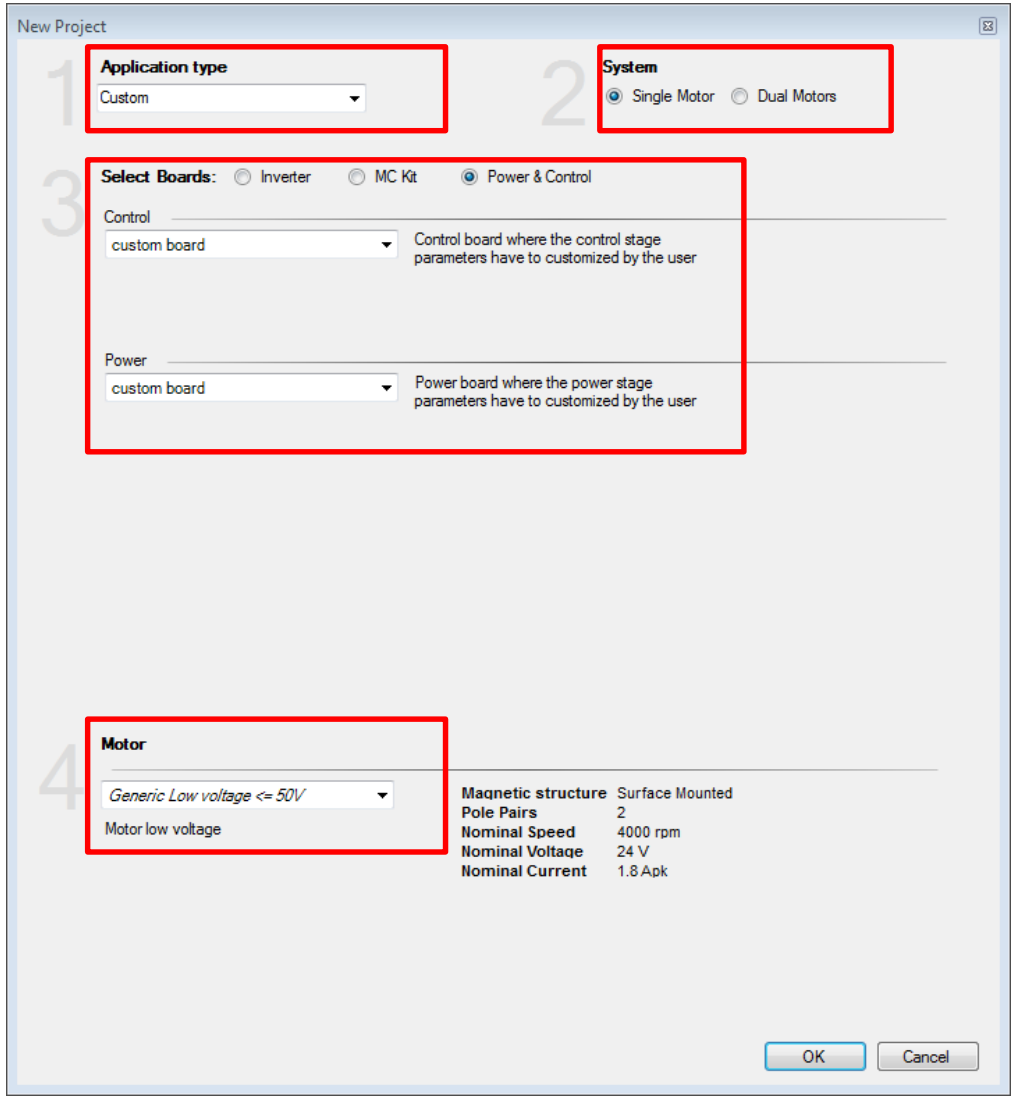
life.augmented

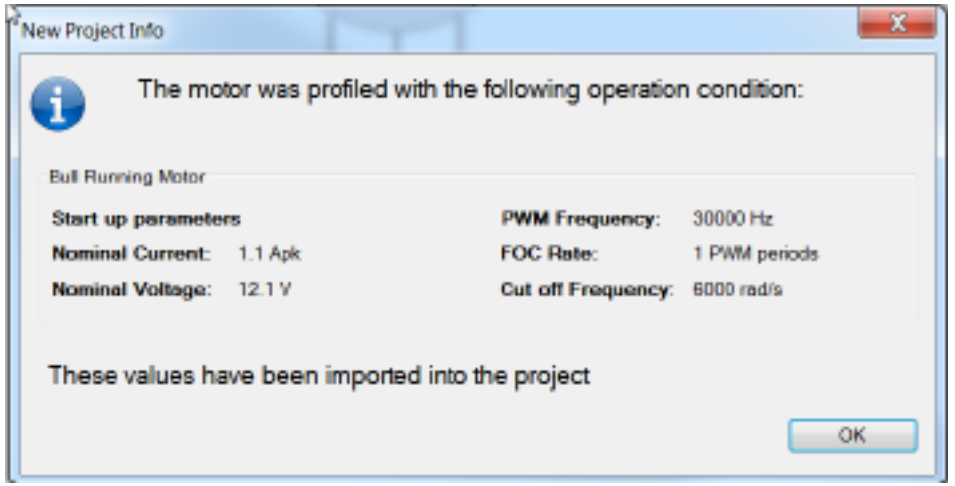
Workbench



1. 用户按钮区用于创建新工程，加载已有工程或启动ST电机参数测量工具。
2. 最近的工程区用于加载近期的工程。
3. 例程区用于加载工程示例。

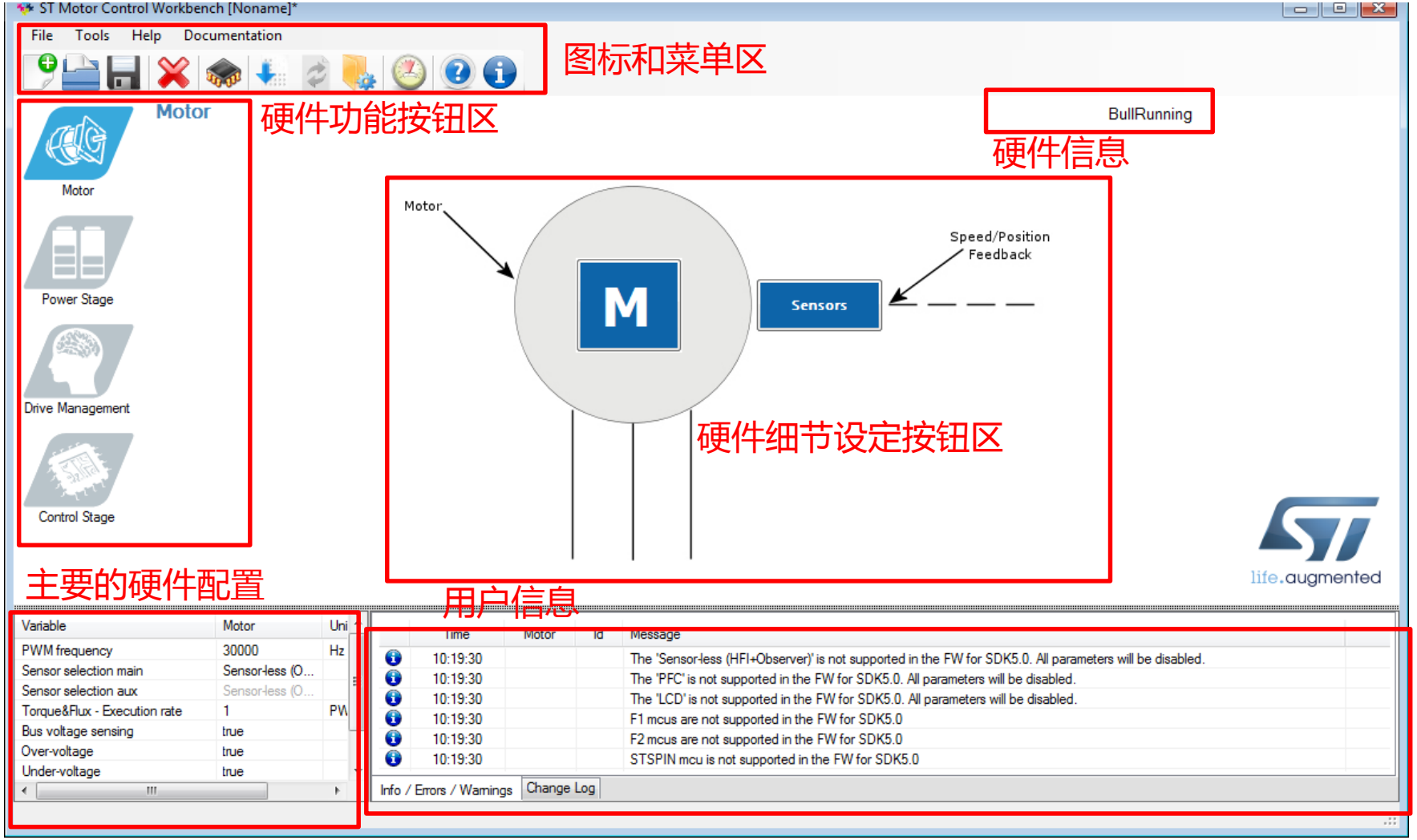
创建新工程





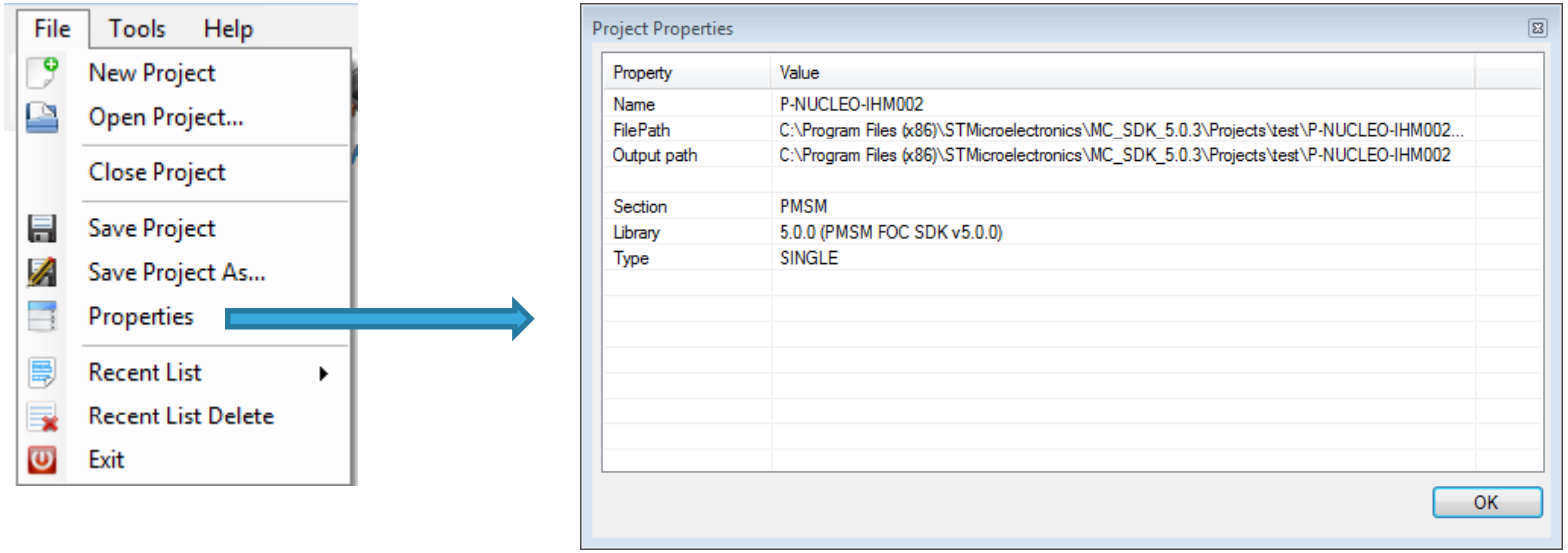
life.augmented

硬件配置窗口



life.augmented

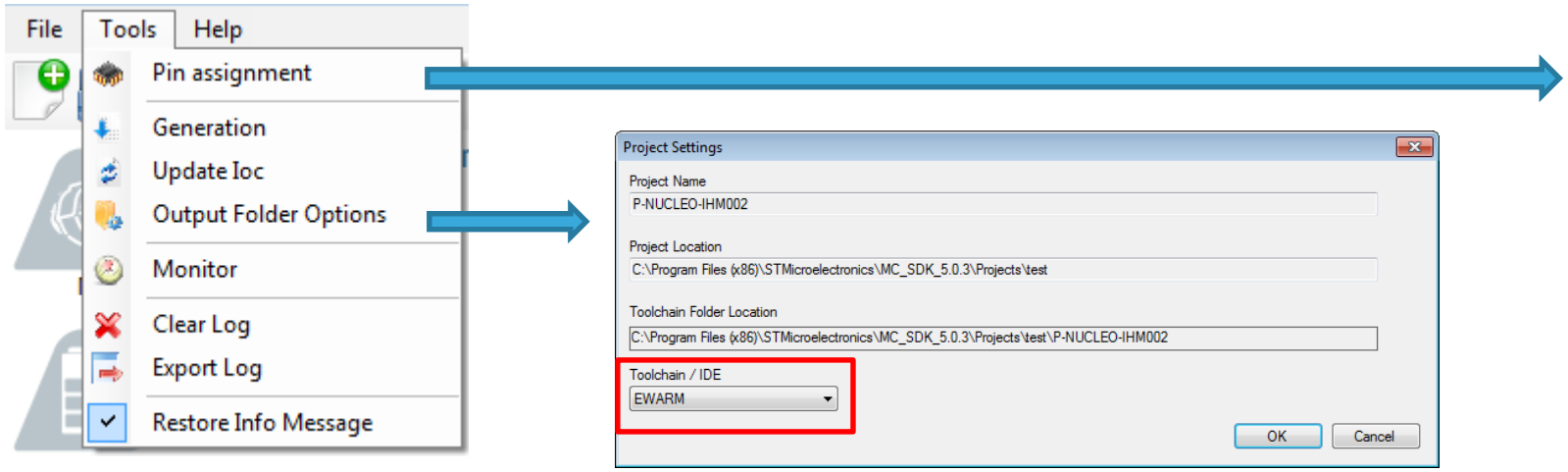
Workbench菜单 (1)

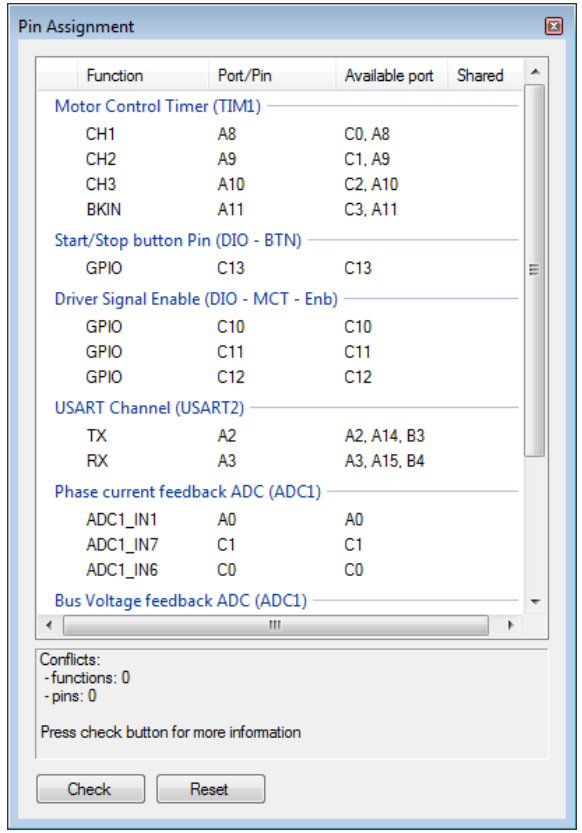


- ✓ New Project: 创建一个新工程
- ✓ Open Project…: 打开一个现有的工程
- ✓ Close Project: 关闭现在的工程
- ✓ Save Project: 使用相同文件名保存打开的工程
- ✓ Save Project As…: 将打开的工程保存为指定的文件名
- ✓ Properties: 查看工程属性
- ✓ Recent List: 从最近打开过的工程列表中加载一个现有工程
- ✓ Recent List Delete: 删除最近的工程列表
- ✓ Exit: 从硬件配置窗口退出

life.augmented

Workbench菜单 (2)





- ✓ Pin assignment:检查MCU的引脚分配和剩余的可用引脚
- ✓ Generation:根据所选的IDE，生成MC应用工程文件
- ✓ Update ioc:更新当前工程的.ioc文件，仅限FW参数，不包含硬件配置
- ✓ Output Folder Option:建立工程环境的路径
- ✓ Monitor:监控并转动电机
- ✓ Clear Log:清除用户信息表
- ✓ Export Log: 将用户信息表以文本格式导出到日志文件
  Restore Info Message: 需要时显示用户信息表

life.augmented

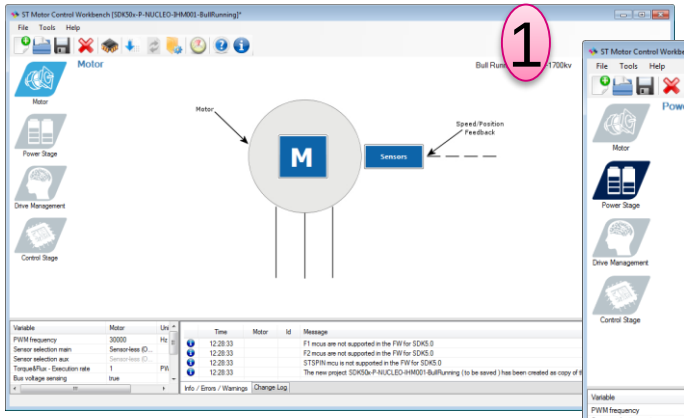
工具栏图标

- ✓ New：创建一个新工程
- ✓ Load：打开一个现有的工程
- ✓ Save：使用相同文件名保存打开的工程
- ✓ Clear Log：清除用户信息表
- ✓ Pin assignment ： 检查MCU的引脚分配和剩余的可用引脚
- ✓ Generation: 根据所选的IDE，生成MC应用工程文件
- ✓ Update Ioc: 更新当前工程的.ioc文件，仅限FW参数，不包含硬件配置
- ✓ Output Folder Option: 建立工程环境的路径.
- ✓ Click to open monitor: 监控并转动电机
- ✓ Help: 提供在线帮助文件的访问接口
- ✓ About: 显示该应用的版本号

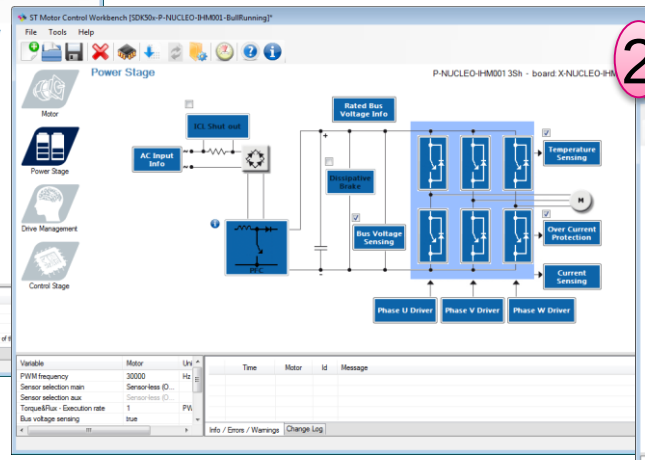
life.augmented

Workbench

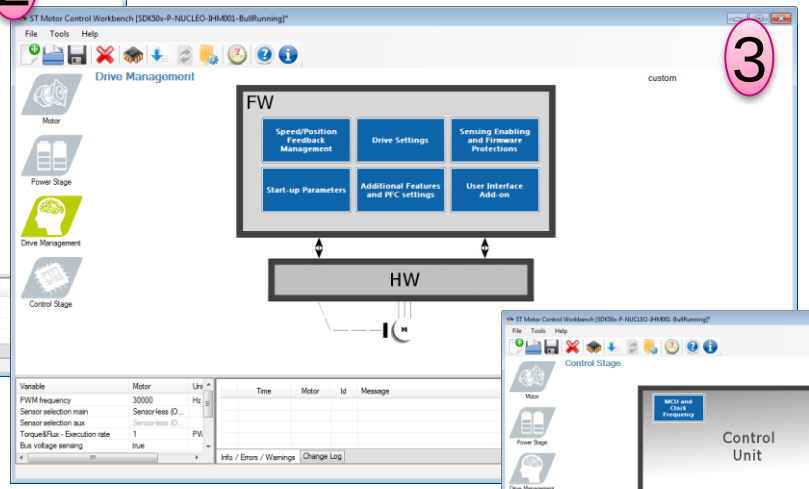
根据客户需求快速设置电机库



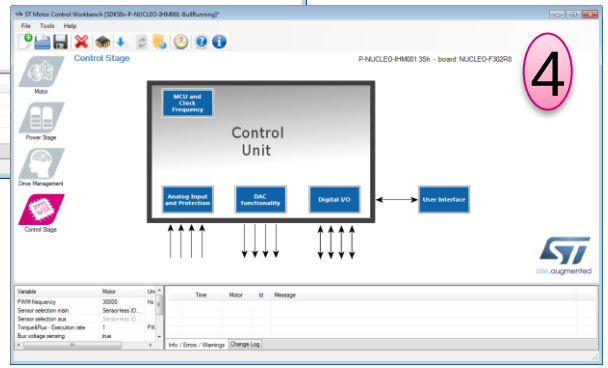
电机参数



硬件驱动

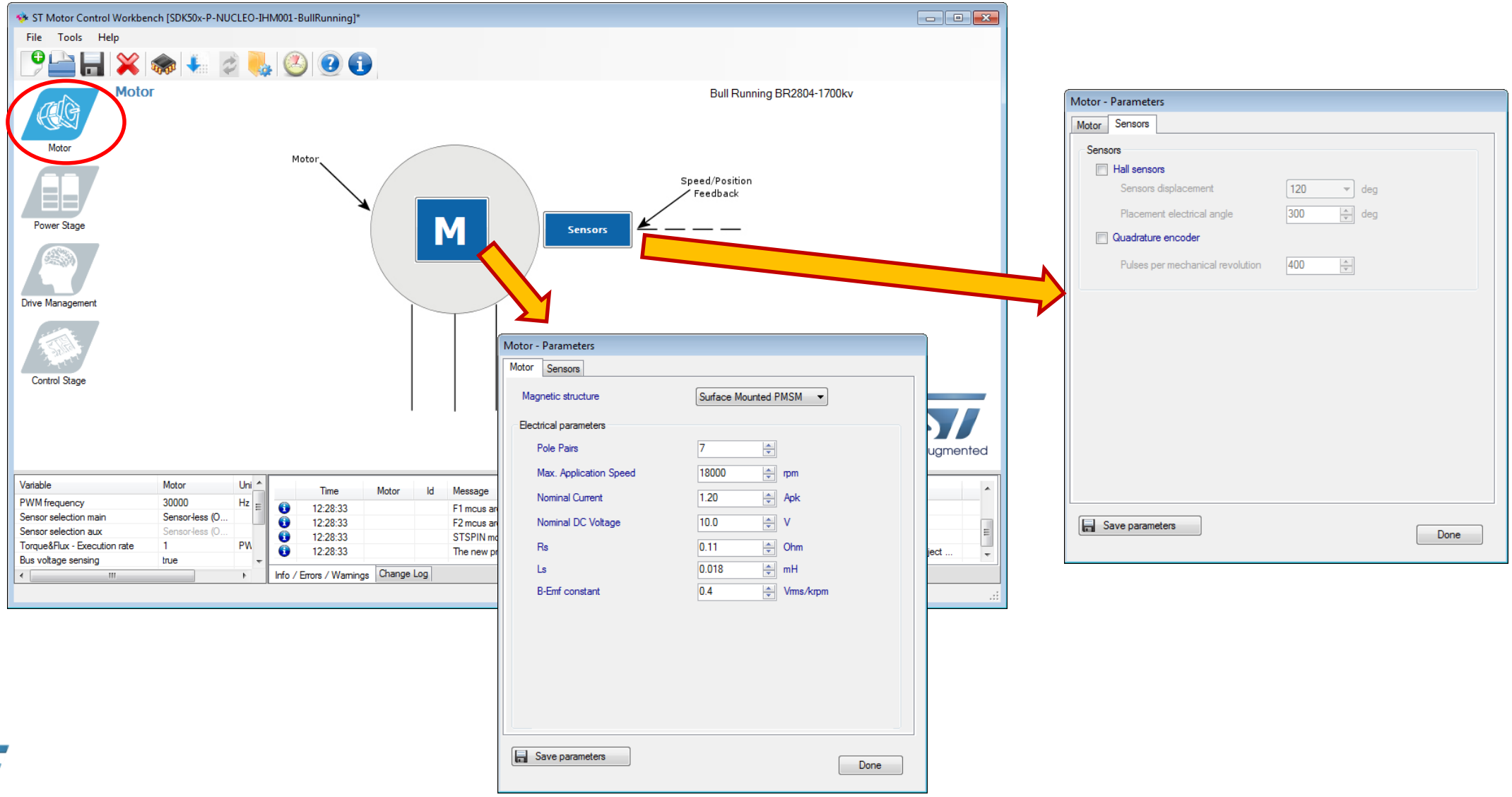


驱动控制管理



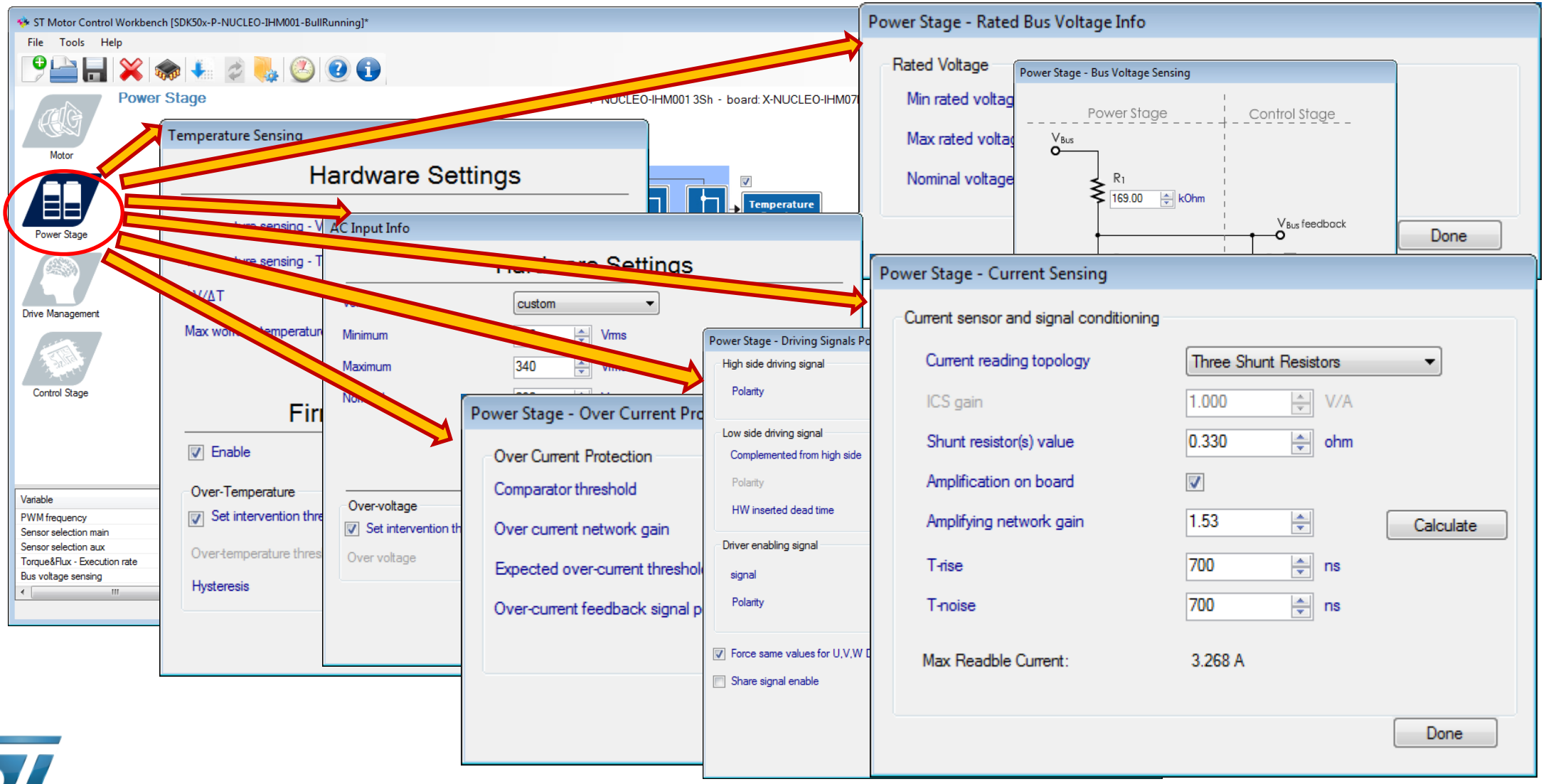
单片机相关

电机参数



life.augmented

硬件驱动

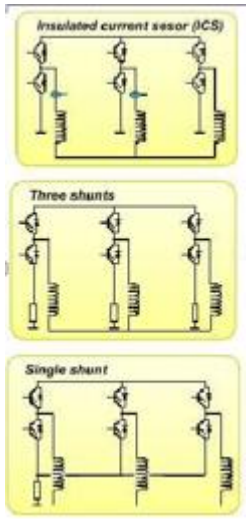


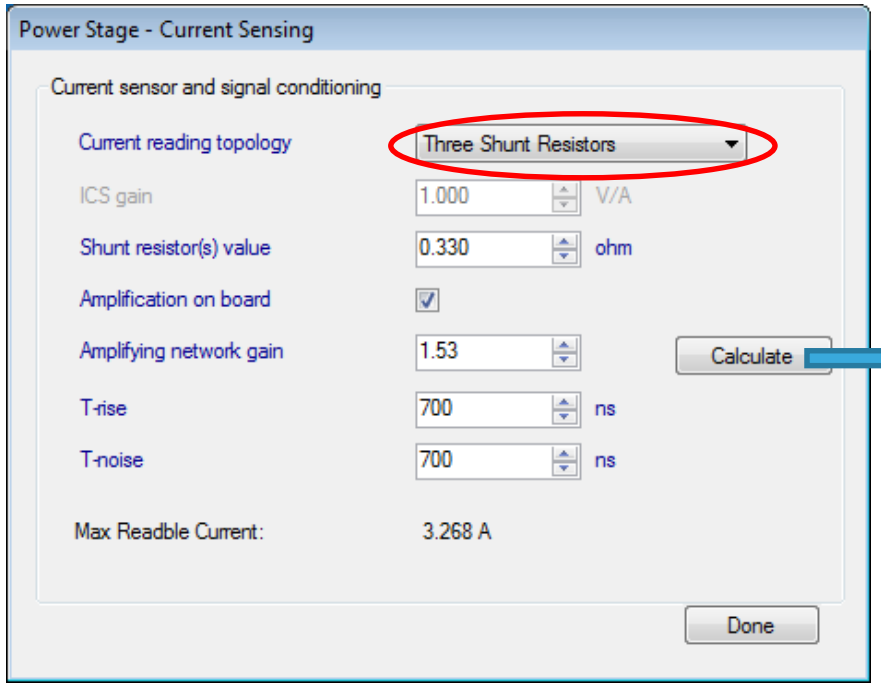
life.augmented

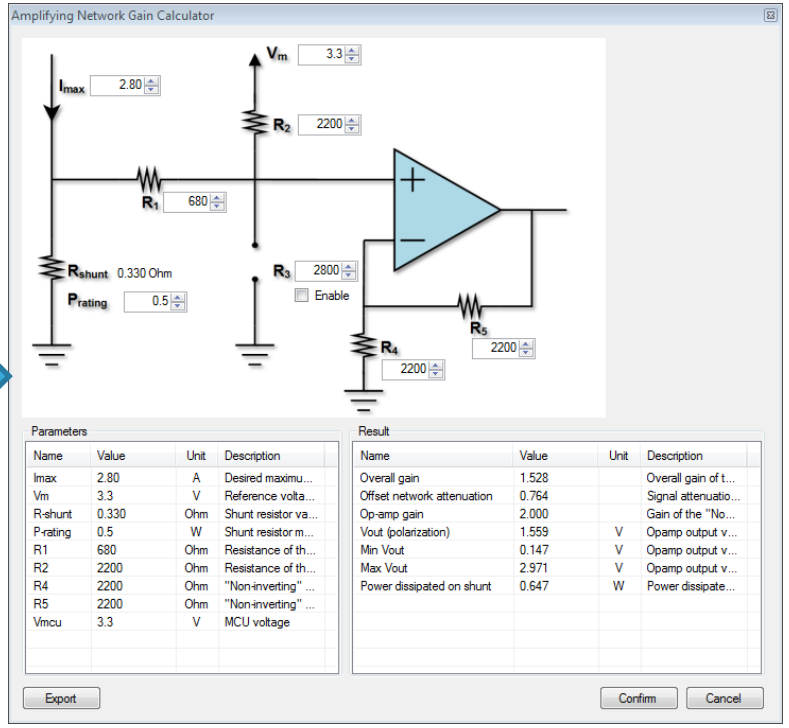
硬件驱动

电流采样拓扑结构选择：

$$(1+R5/R4)\cdot\frac{(R2\,//\,R3)}{((R2\,//\,R3)+R1)}$$

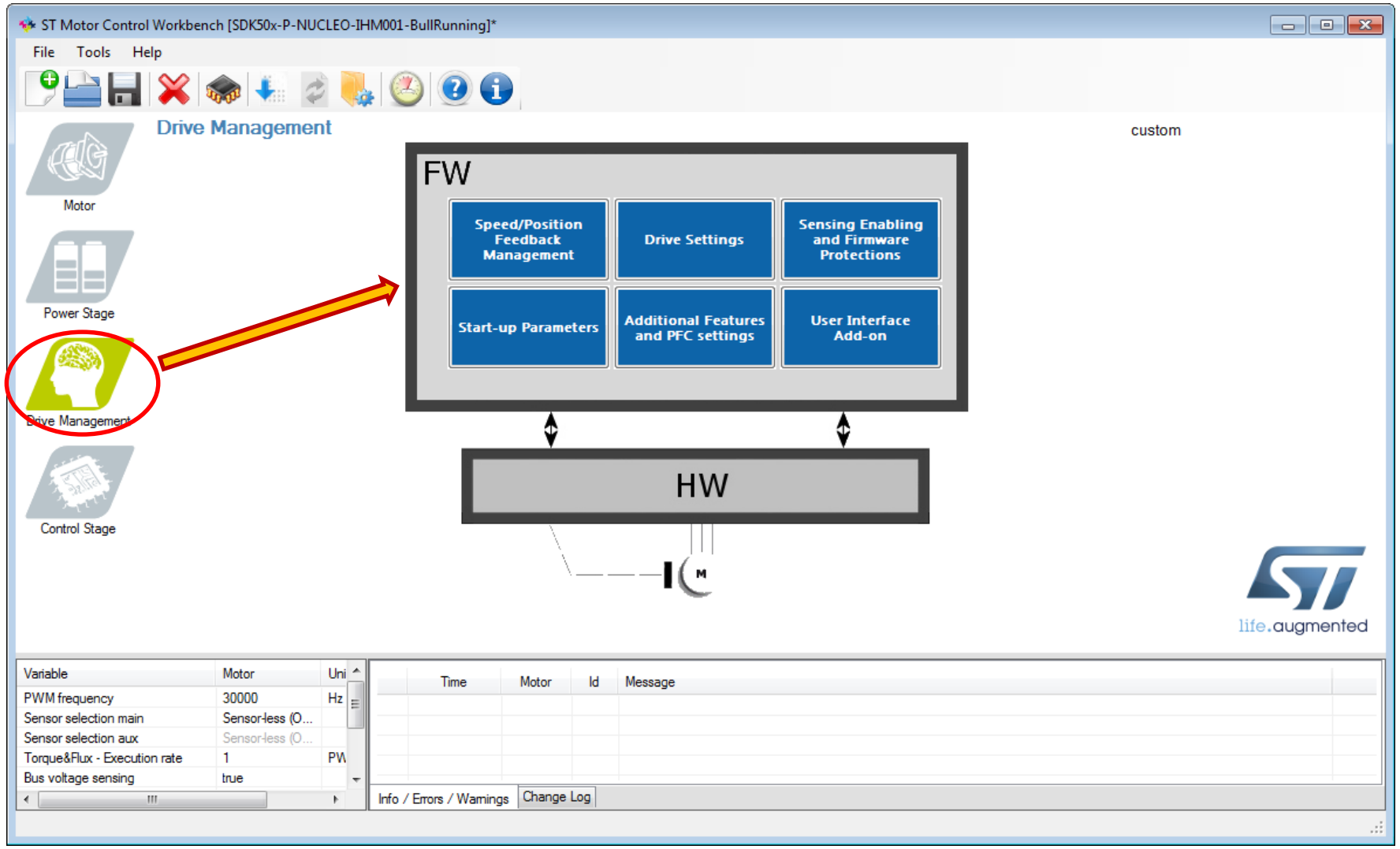




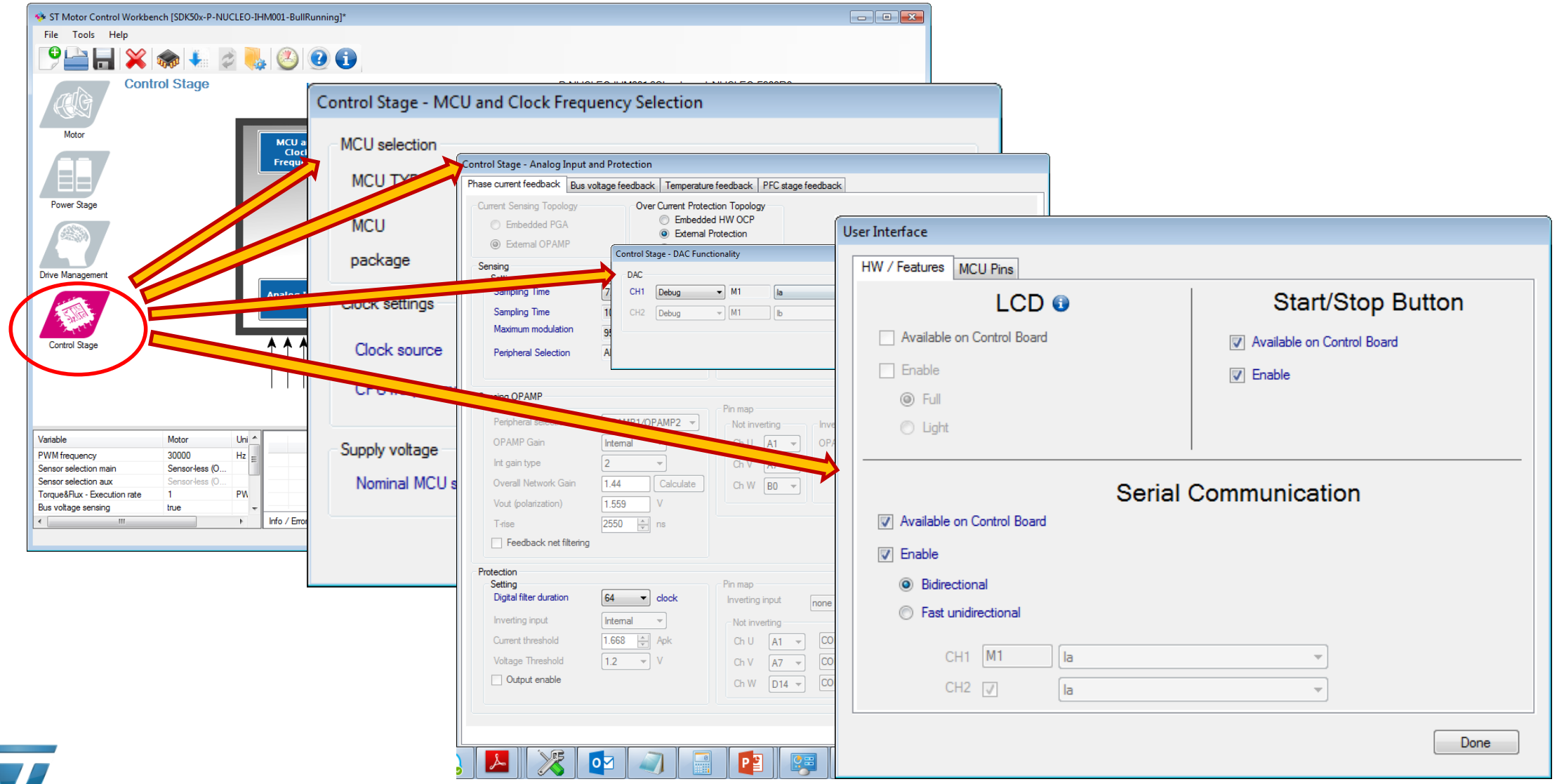


life.augmented

驱动控制管理



单片机相关



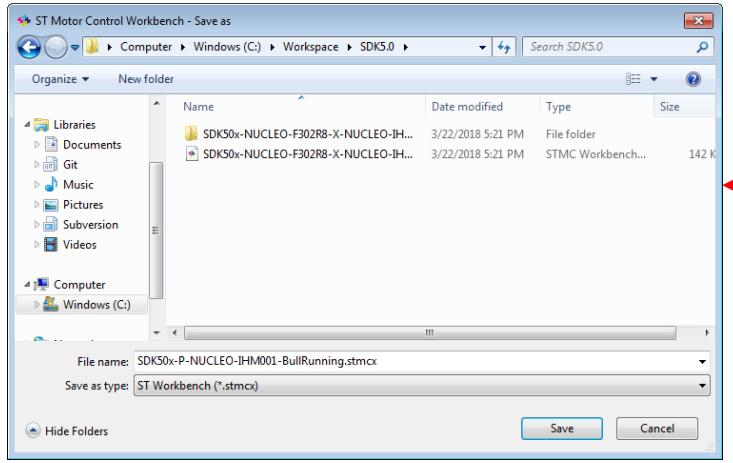
life.augmented

MC工程的生成

所有的参数配置完成后，点击生成图标，可根据所选的IDE生成MC应用工程：

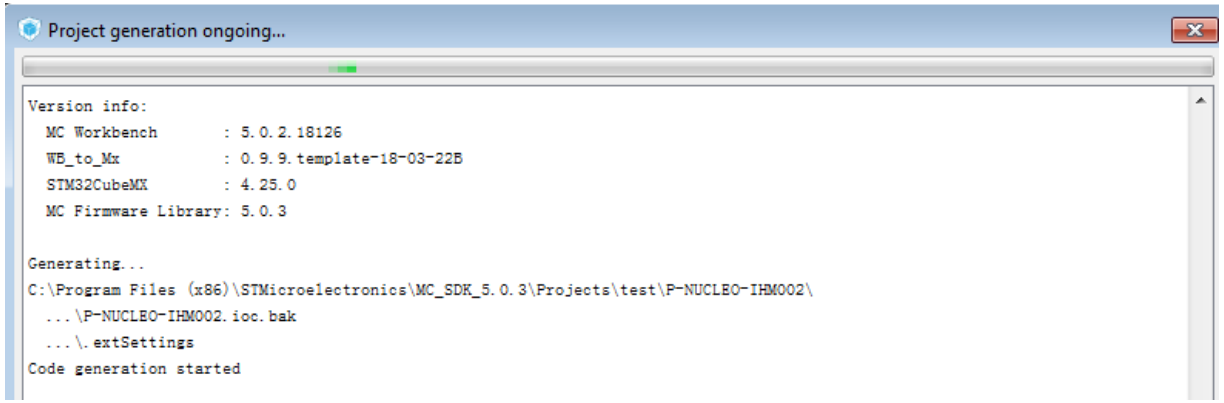
- 用户不再需要将MC Workbench输出文件拷贝到其工程下。
- STM32CubeMx作为接口被MC Workbench后台调用，来生成选择的IDE项目工作框架。

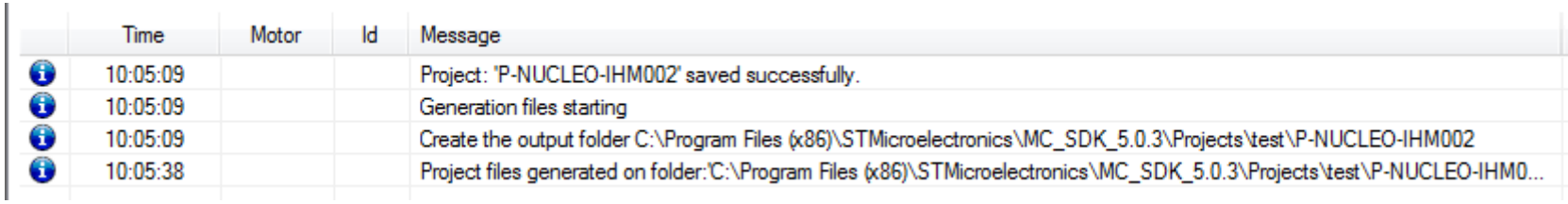
MC工程的生成



如果当前工程尚未保存，会显示一个文件管理器窗口，询问是否将当前工程设置保存为一个新项目。

如果当前工程已保存过，会显示一个进程窗口，告知用户脚本在执，同时更新用户信息表





life.augmented

MC工程的生成

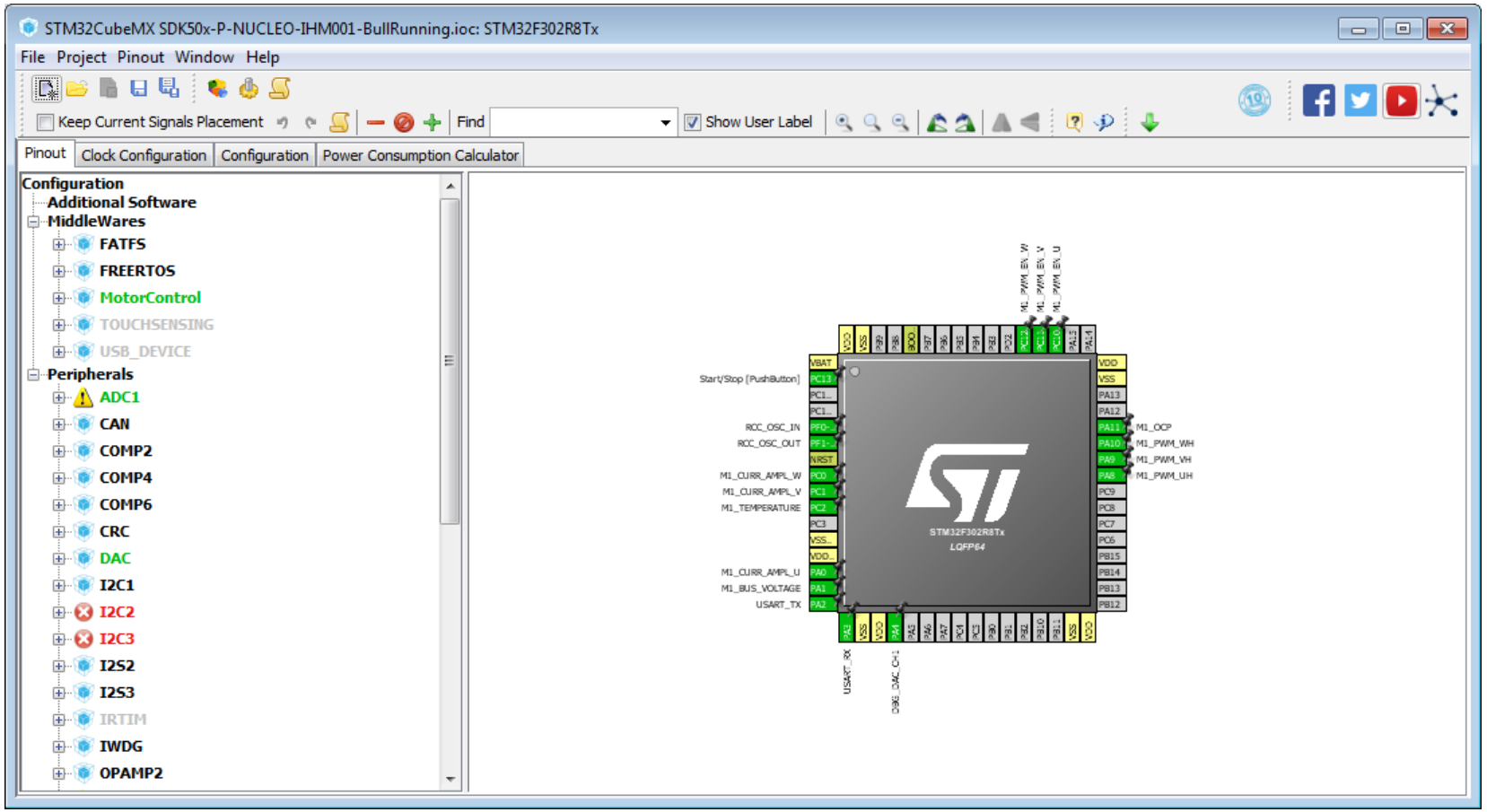
生成的IAR工程文件保存在如下路径：
$ProjectFolder\EWARM

如果不需要配置其他外设，可直接打开.eww文件。

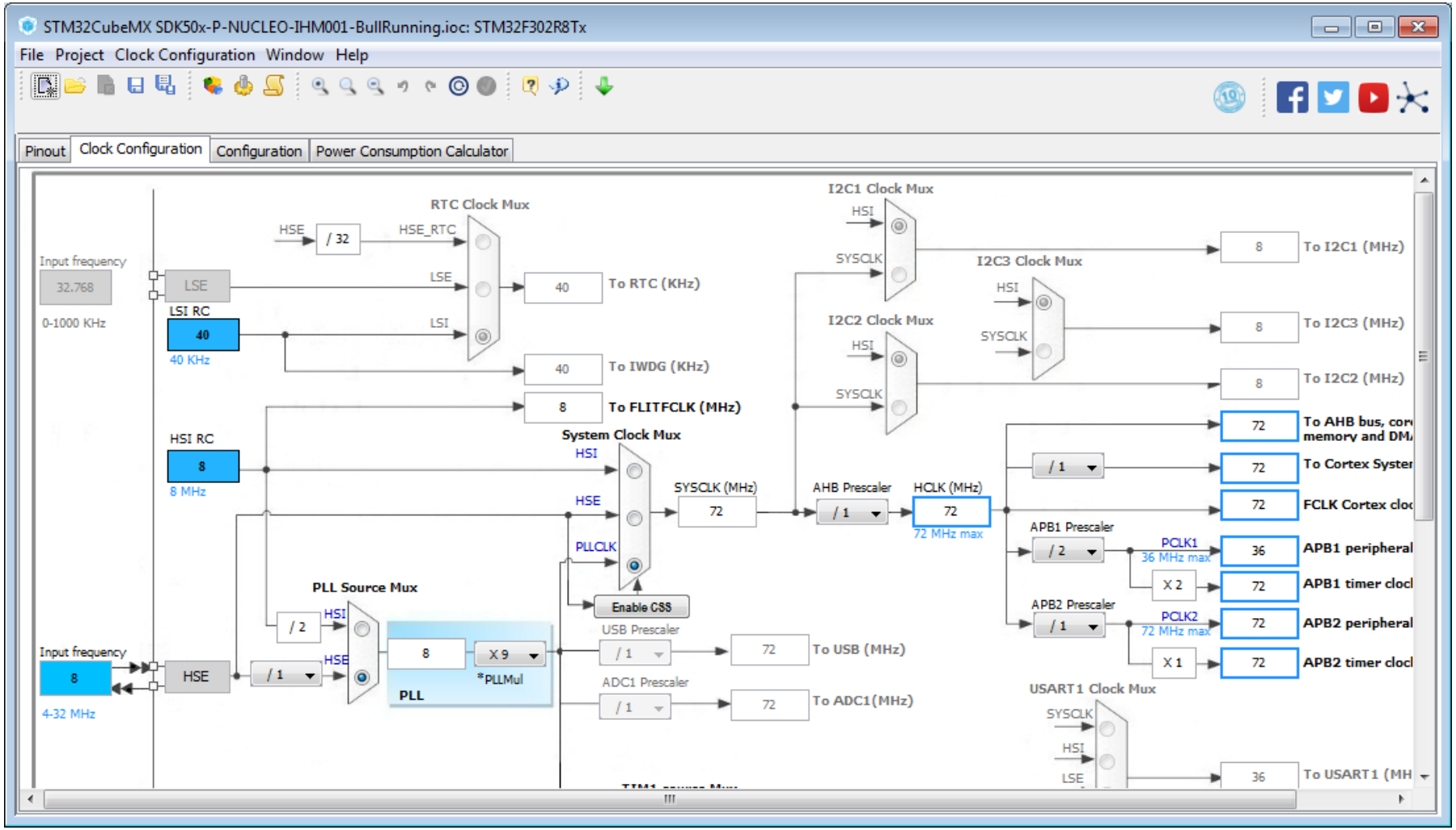
如果需要配置其他外设，请用CubeMX打开.ioc文件进行添加。
此操作最好在所有电机控制相关配置完成后进行，否则请在Workbentch中使用update ioc来更新oc文件和FW参数(不包含硬件配置)

life.augmented

打开.ioc文件，STM32CubeMx将被打开。
图形化的引脚分配配置

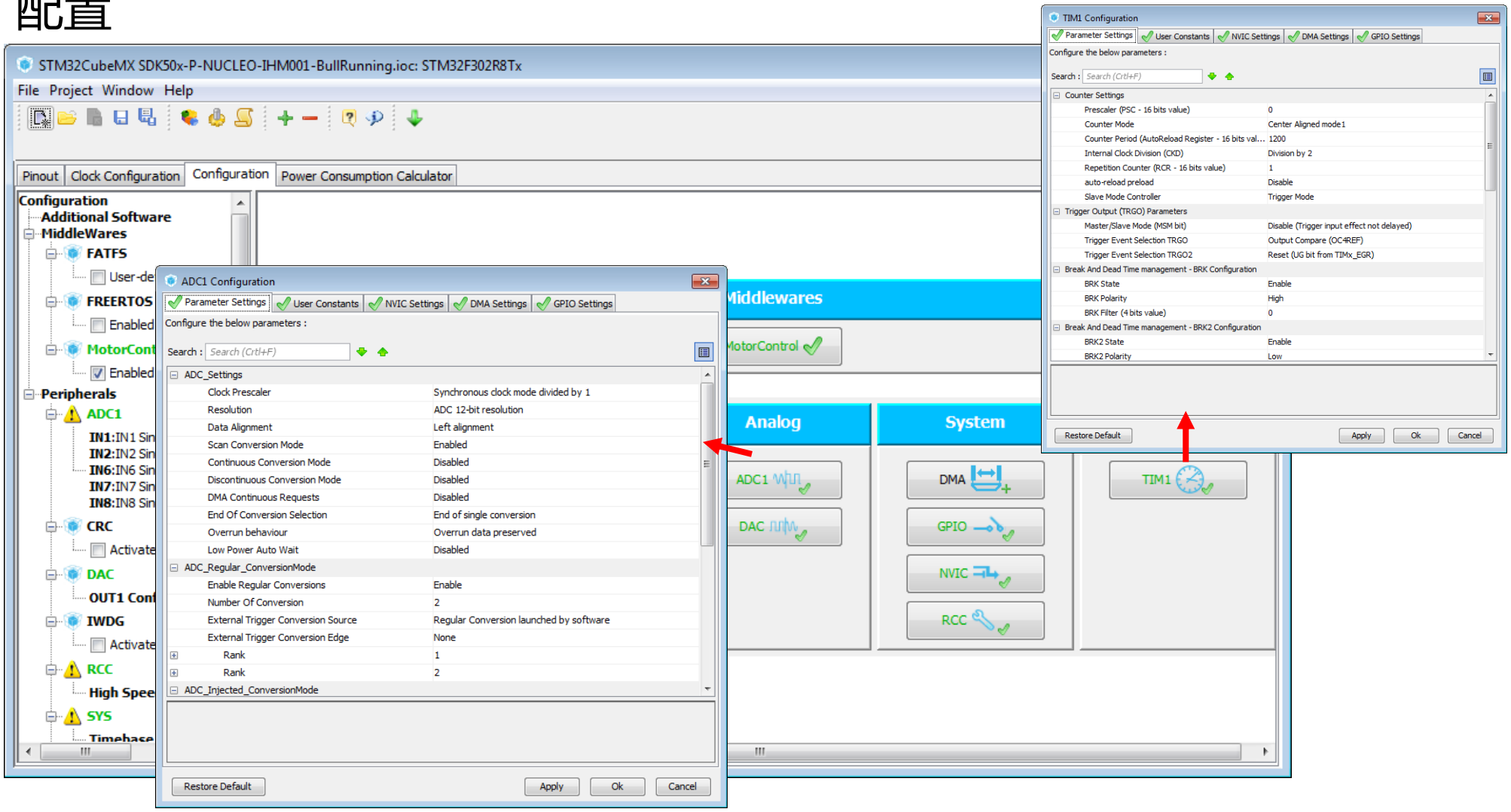


时钟树结构



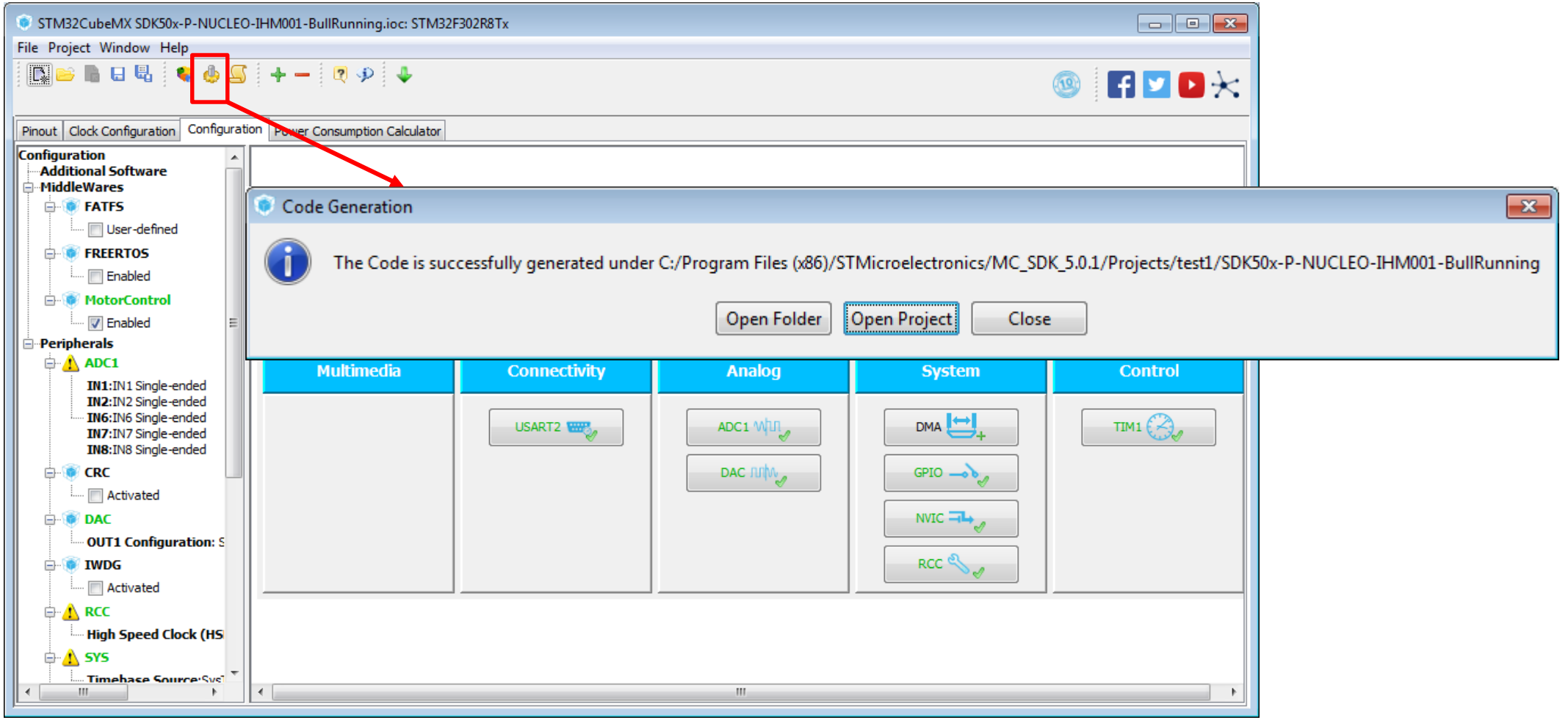
life.augmented

配置



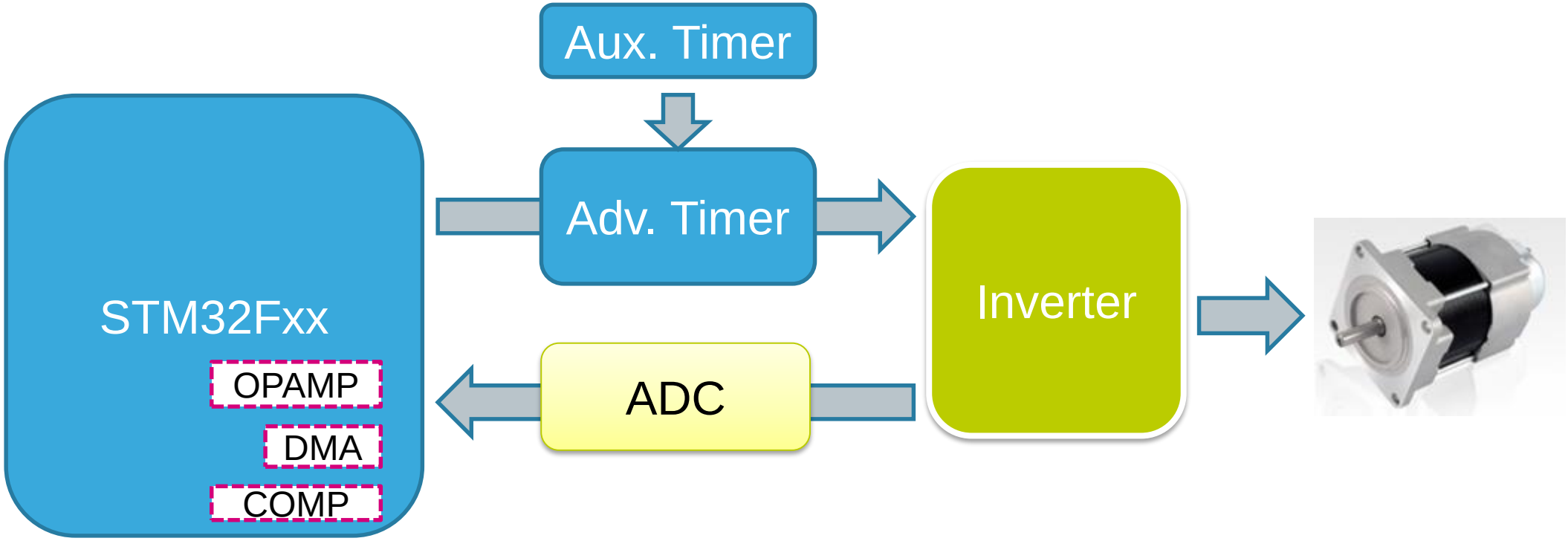
life.augmented

代码生成：



Motor Control 相关外设

在CubeMx中请不要修改电机控制相关外设的设定!



life.augmented

Motor Control 相关外设

三电阻电流采样所用资源

MCU	Single/Dual	Adv.Timer	DMA	ADC	Note
F0	Single	TIM1	DMA1_CH1	ADC1	The dual drive mode and the internal PGA are not available
F3	Single	TIM1	None	ADC1	Using one ADC
F3	Single/Dual	TIM1(TIM8)	None	ADC1 ADC2	Using two ADC
F3	Single/Dual	TIM1(TIM8)	None	ADC1 ADC2 (ADC3 ADC4)	Using four ADC
F4	Single/Dual	TIM1 (TIM8)	-	ADC1 ADC2	Using two ADC, ADC is used in time sharing.

life.augmented

Motor Control 相关外设

单电阻采样所用资源(1)

MCU	Single/Dual	Adv. Timer	Aux. Timer	DMA	ADC	Note
F0	Single	TIM1	TIM3 (CH4)	DMA1_CH2 DMA1_CH3 DMA1_CH4	ADC1	F030x/F031x device configuration
F0	Single	TIM1	TIM15 (CH1)	DMA1_CH2 DMA1_CH5 DMA1_CH4	ADC1	F051x device configuration
F0	Single	TIM1	TIM15 (CH1)	DMA1_CH2 DMA1_CH5 DMA1_CH4	ADC1	F072x device configuration
F3	Single	TIM1	-	DMA1_CH4 DMA1_CH5	ADC1	-
F3	Dual	TIM1 TIM8	-	DMA1_CH4 DMA1_CH5 DMA2_CH1 DMA2_CH2	ADC1 ADC2	-

life.augmented

Motor Control 相关外设

单电阻采样所用资源(2)

MCU	Single/Dual	Adv. Timer	Aux. Timer	DMA	ADC	Note
F4	Single/Dual	TIM1	TIM5 (CH4)	DMA1,stream1,ch6; DMA2,stream4,ch6	ADC1	Option1:used by first motor when it is configured in single shunt, or by second motor when the first one isn't in single shunt. ADC1 used for general purpose conversions
F4	Single/Dual	TIM8	TIM4 (CH2)	DMA1,stream3,ch2; DMA2,stream7,ch7	ADC1	Option1:used by second motor when it is configured in single shunt and when first motor isn't in single shunt. ADC1 used for general purpose conversions
F4	Single/Dual	TIM8	TIM5 (CH4)	DMA1,stream1,ch6; DMA2,stream7,ch7	ADC3	Option2:used by first motor when it is configured in single shunt, or by second motor when the first one isn't in single shunt. ADC1 used for general purpose conversions
F4	Single/Dual	TIM1	TIM4 (CH2)	DMA1,stream3,ch2; DMA2,stream4,ch6	ADC1	Option2:used by second motor when it is configured in single shunt and when first motor is also in single shunt. ADC1 used for general purpose conversions

F4xx用于单电机时，可在option 1和option 2间进行选择。

life.augmented

Motor Control 相关外设

ICS电流采样所用资源

MCU	Single/Dual	Advanced Timer	DMA	ADC	Note
F3	Single	TIM1 or TIM8	None	ADC1 ADC2	Used by the first or second motors depending on the user selection.
F3	Dual	TIM1 TIM8	None	ADC1 ADC2 ADC3 ADC4	-
F4	Single	TIM1 or TIM8	None	ADC1 ADC2	Used by the first or second motors depending on the user selection.
F4	Dual	TIM1 TIM8	None	ADC1 ADC2	-

life.augmented

ADC的配置

描述	下载地址
STM32F0模数转换模块（ADC）介绍	下载地址
STM32F1其余模块（包括ADC和计数器）介绍	下载地址
STM32F2模数转换模块（ADC）介绍	下载地址
STM32F3模数转换模块（ADC）介绍	下载地址
How to get the best ADC accuracy in STM32Fx Series and STM32L1 Series devices	下载地址
STM32™'s ADC modes and their applications	下载地址
Improving STM32F1x and STM32L1x ADC resolution by oversampling	下载地址
使用STM32F2xx 和STM32F4xx 微控制器时如何提高ADC 测量精度	下载地址
STM32F30x ADC modes and application	下载地址

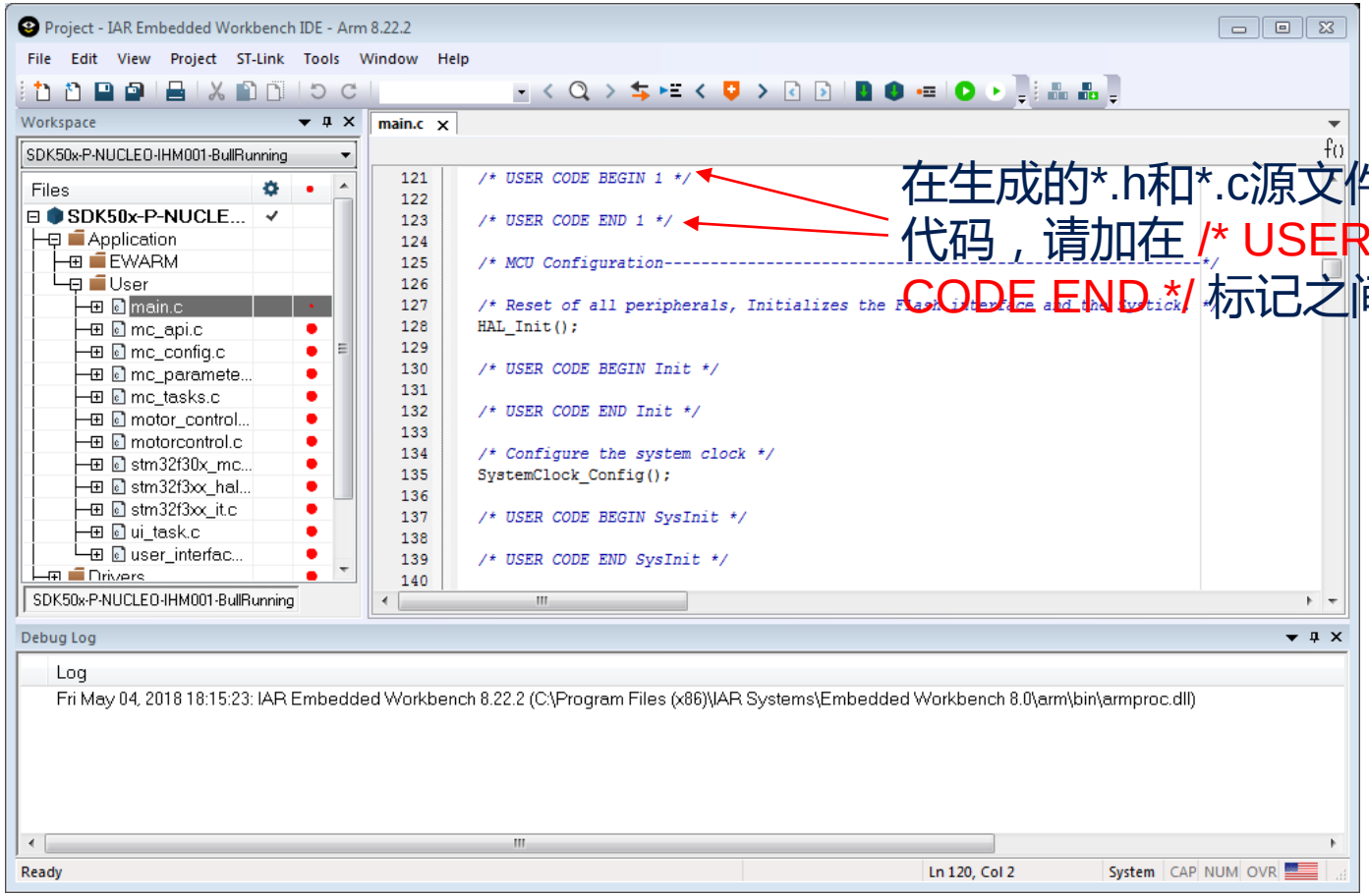
life.augmented

Timer 的配置

描述	链接
STM32F0定时器模块（TIM）介绍	下载地址
STM32F1其余模块（包括ADC和计数器）介绍	下载地址
STM32F2定时器模块（TIM）介绍	下载地址
STM32F3通用定时器模块（TIM）介绍	下载地址
STM32F3定时器模块新增功能介绍	下载地址
General-purpose timer cookbook	下载地址
STM32F1xx、STM32F2xx、STM32F4xx、STM32L1xx、 STM32F30/31/37/38x 定时器概览	下载地址

life.augmented

IAR Embedded Workbench for Arm (v8.xx)

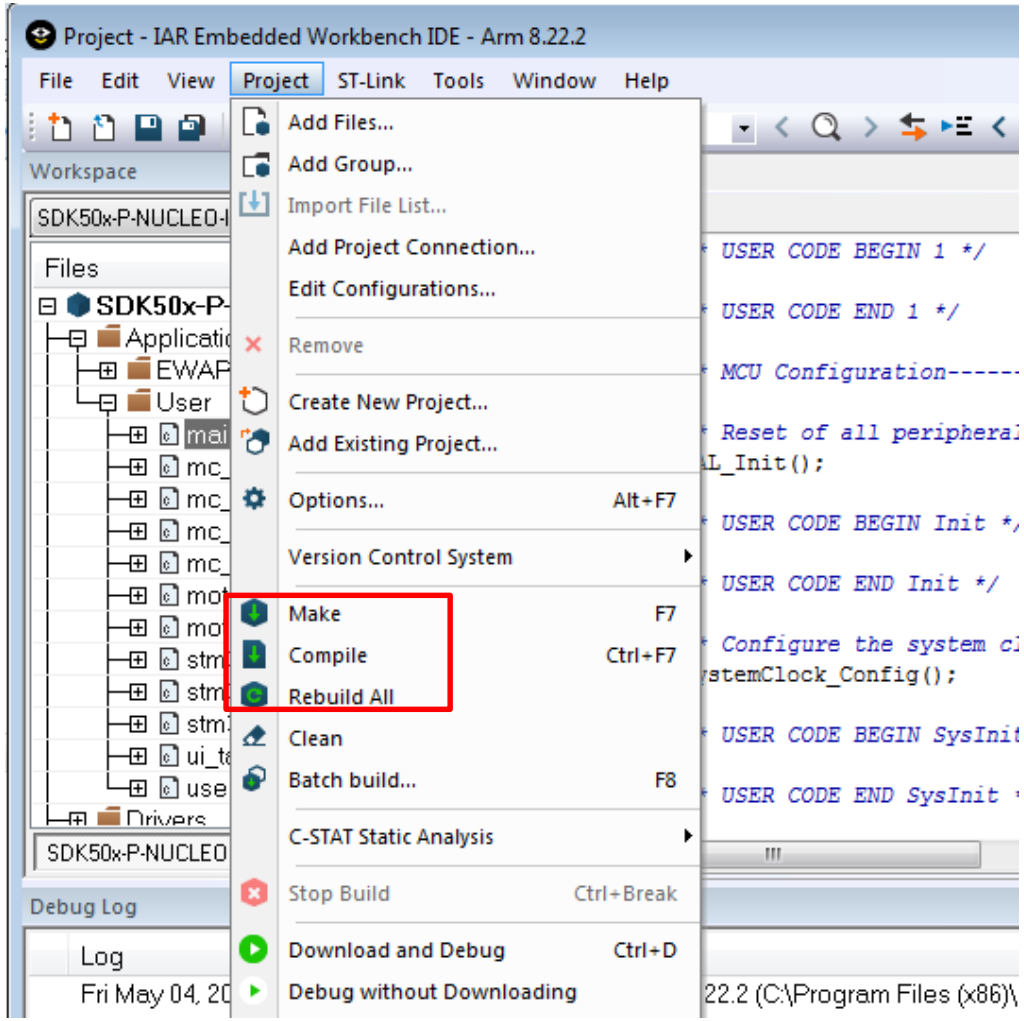


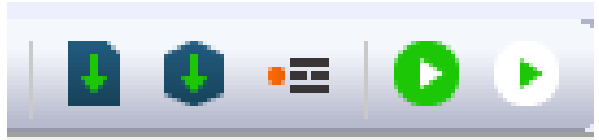
在生成的*.h和*.c源文件中，用户如果添加自己的应用代码，请加在 /* USER CODE BEGIN */和/* USER CODE END */ 标记之间。

在IAR Embedded Workbench for Arm v8.xx中，C编译器的优化选择请设定为 ***Speed.***

life.augmented

MC 工程编译和下载





Compile

Make

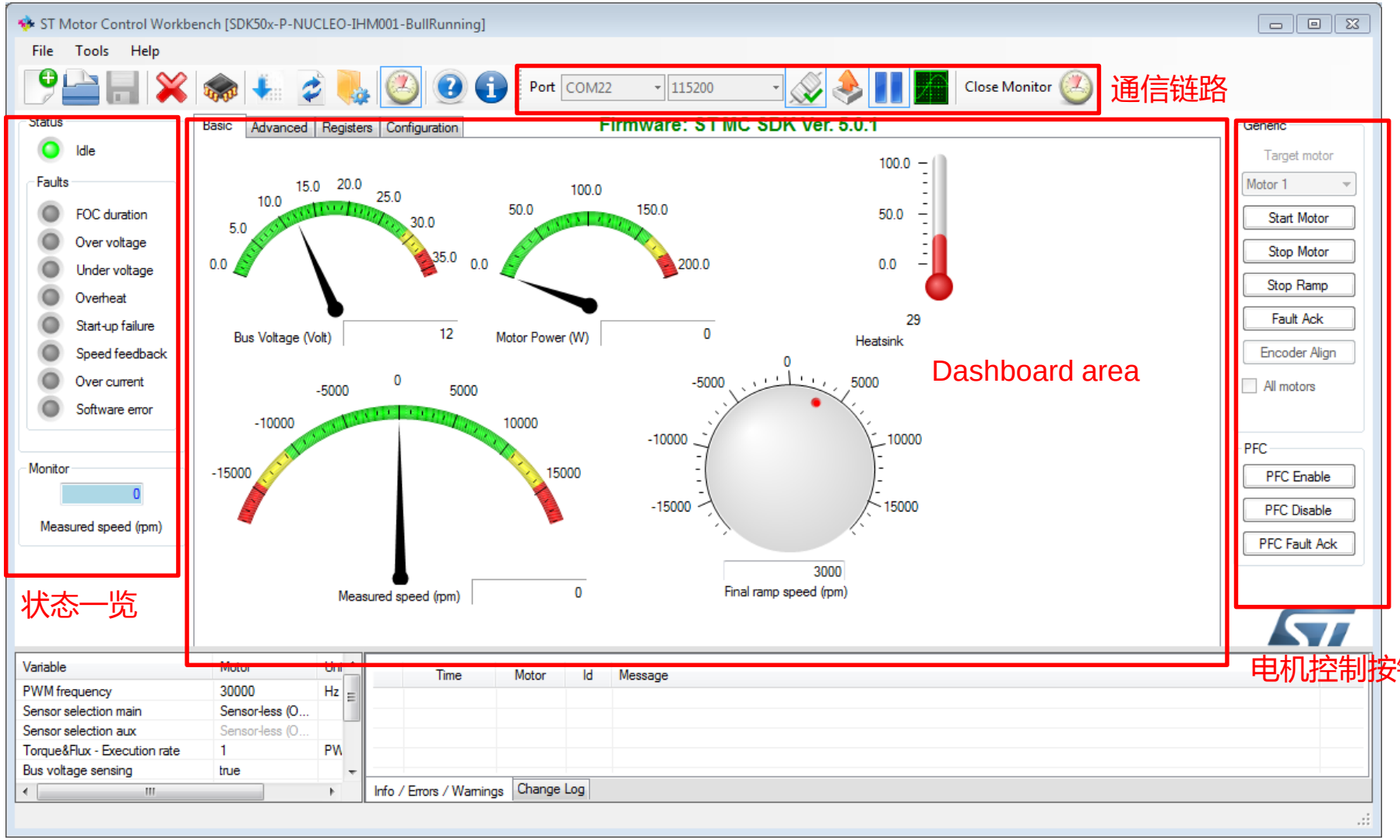
Toggle Breakpoint

Download and Debug

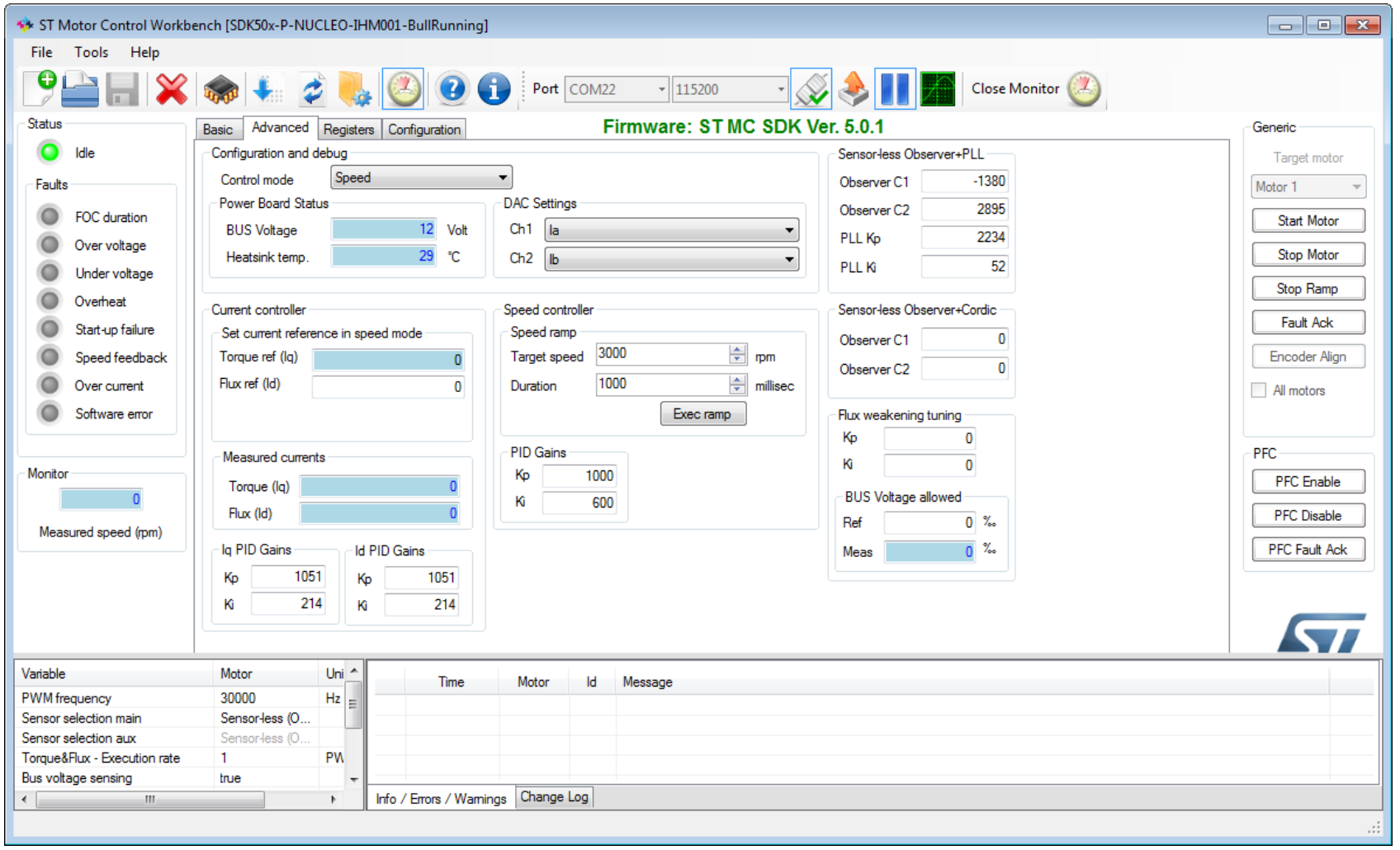
Debug without Downloading

life.augmented

电机控制和监控

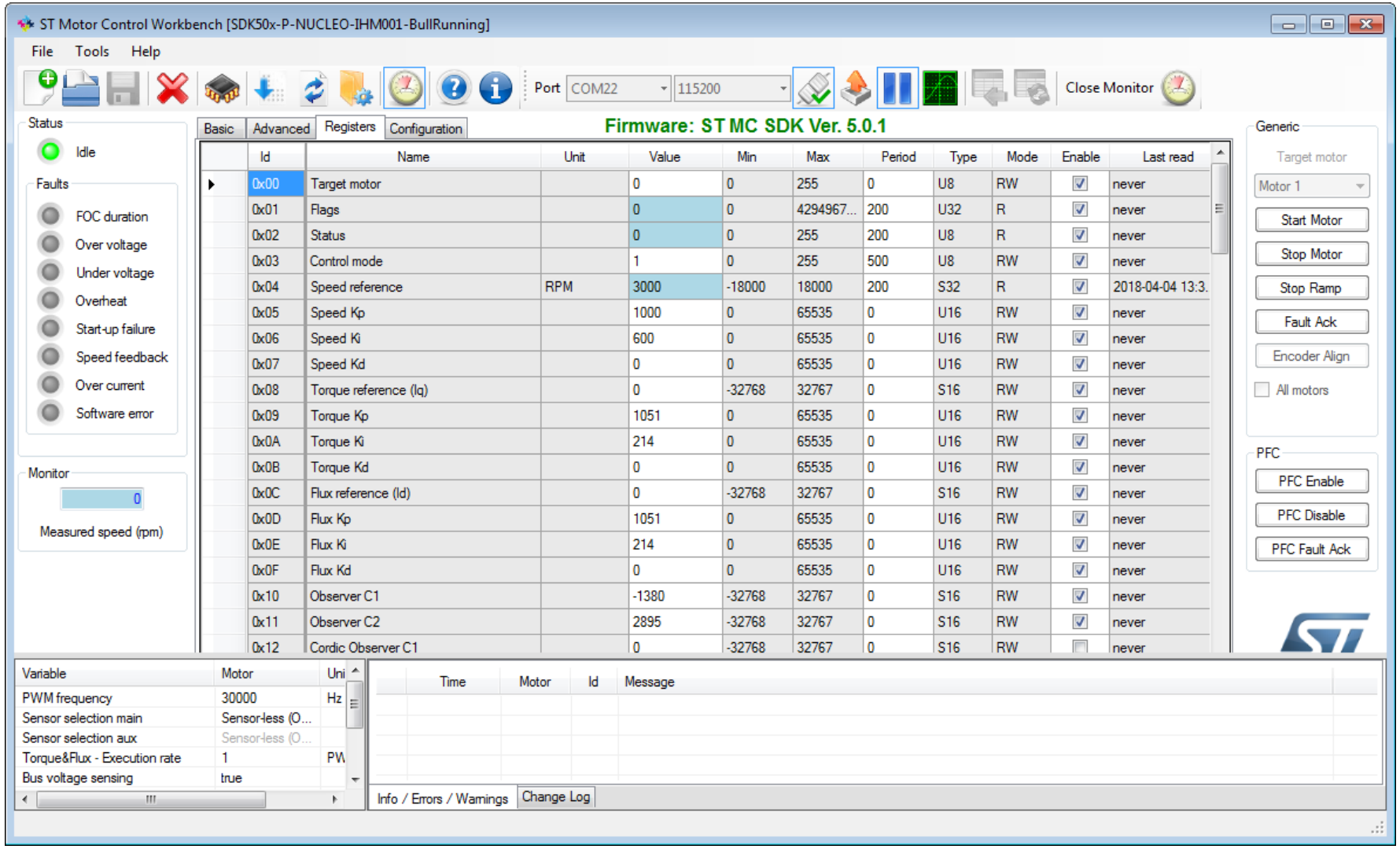


电机控制和监控



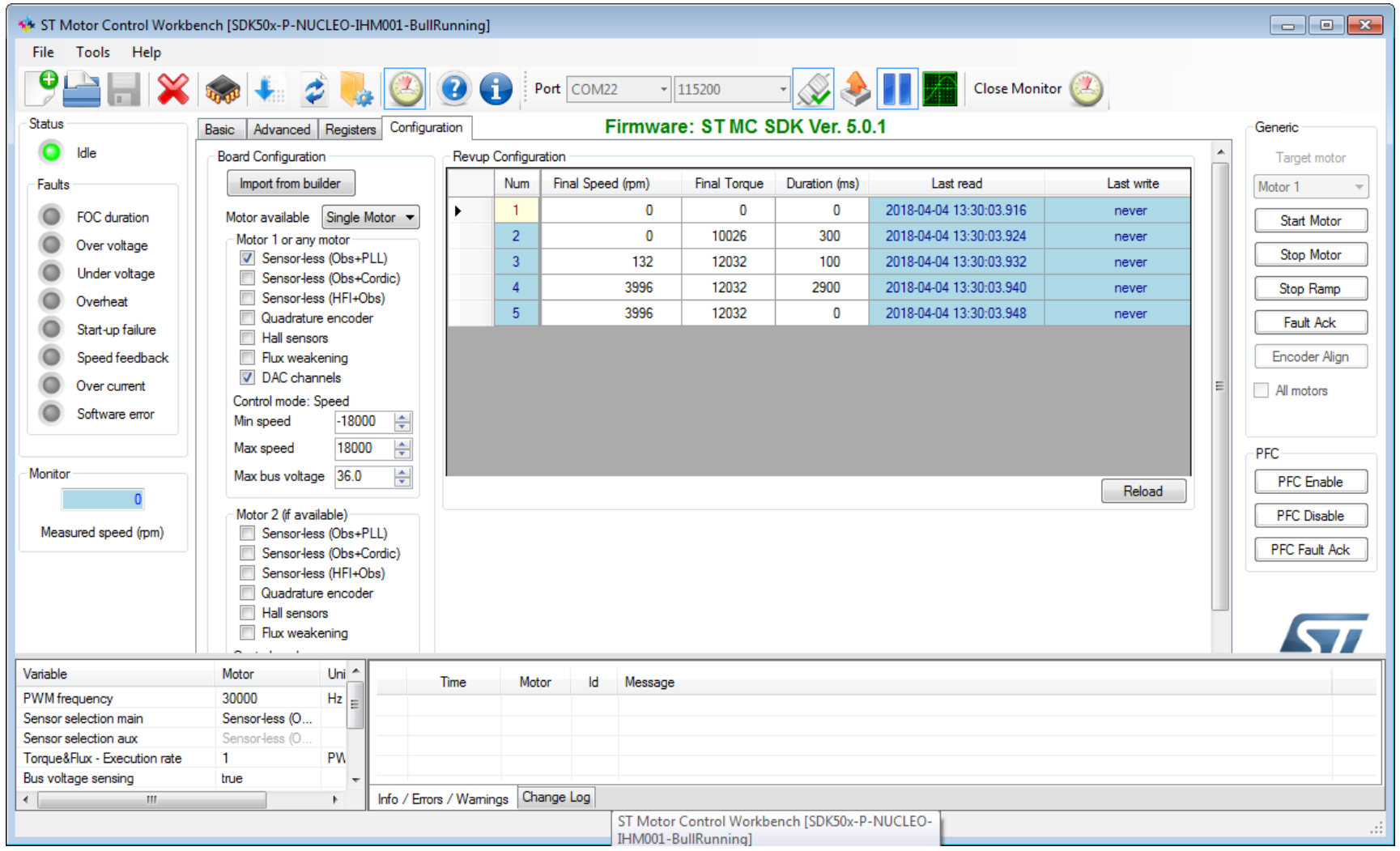
life.augmented

电机控制和监控



life.augmented

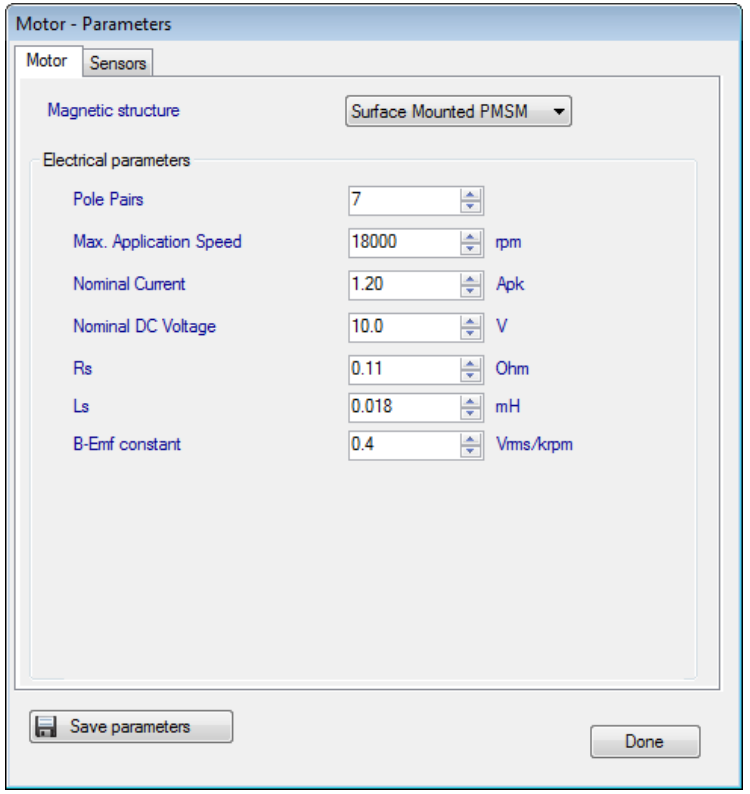
电机控制和监控

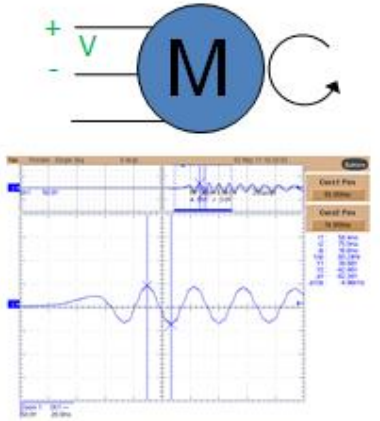


life.augmented

Workbench关键配置参数说明

- 马达参数配置
  - 极对数
  - 最大转速
  - 最大电流
  - 供电电压
  - 电机电阻
  - 电机电感
  - 电机反电势常数

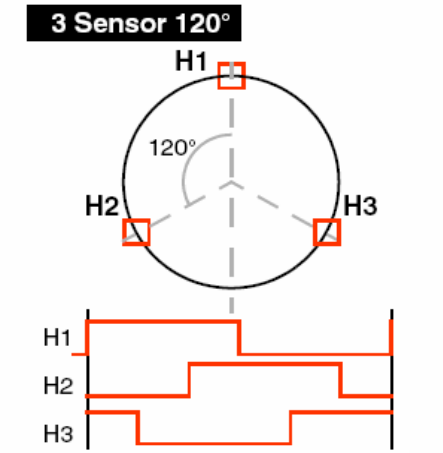


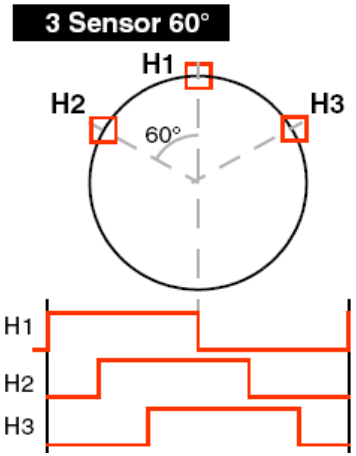


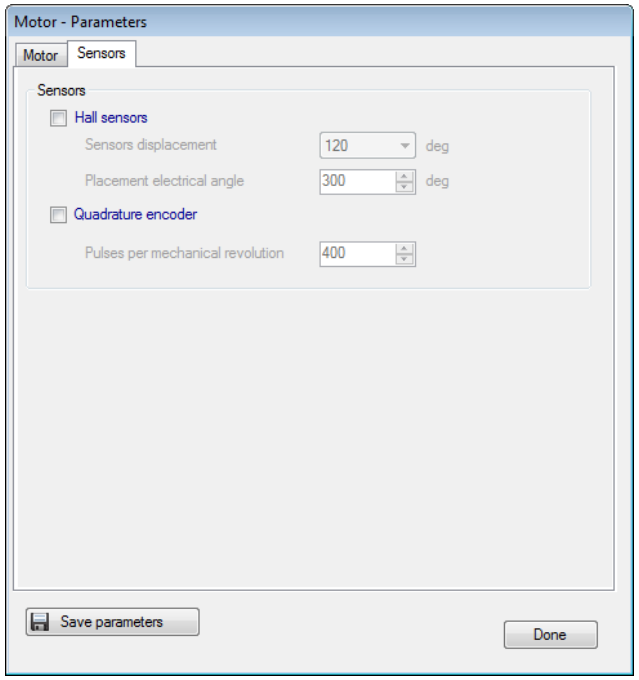
life.augmented

Workbench 关键配置参数说明

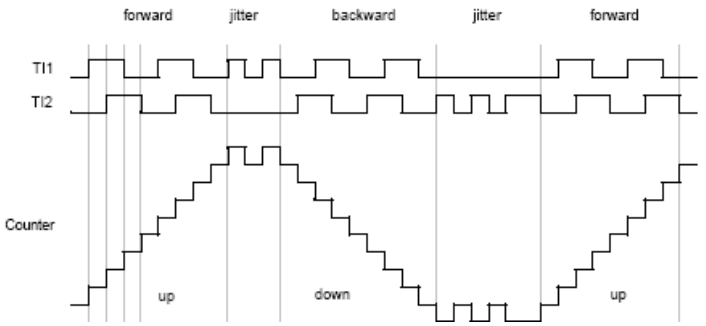
- Hall同步电角度：
  - 电机Hall A的上升沿到电机A相反电动势最高点的延迟角度。
  - 默认电机A相的反电动势最高点作为电角度的0度；



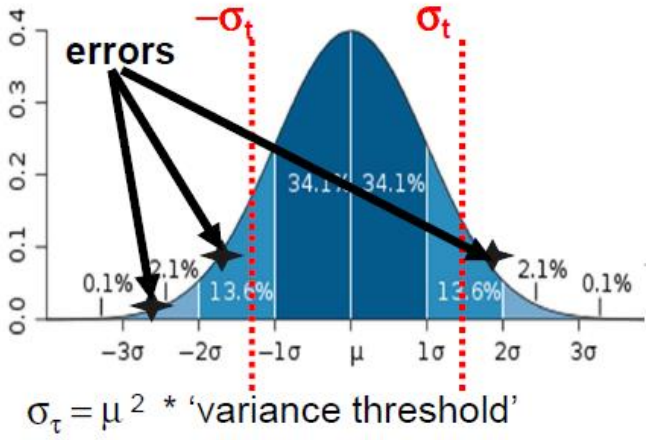




- 编码器参数
  - Resolution更清晰，分辨率，单位：Pulse/Round
  - 旋转一圈的编码器脉冲个数



Workbench 关键配置参数说明

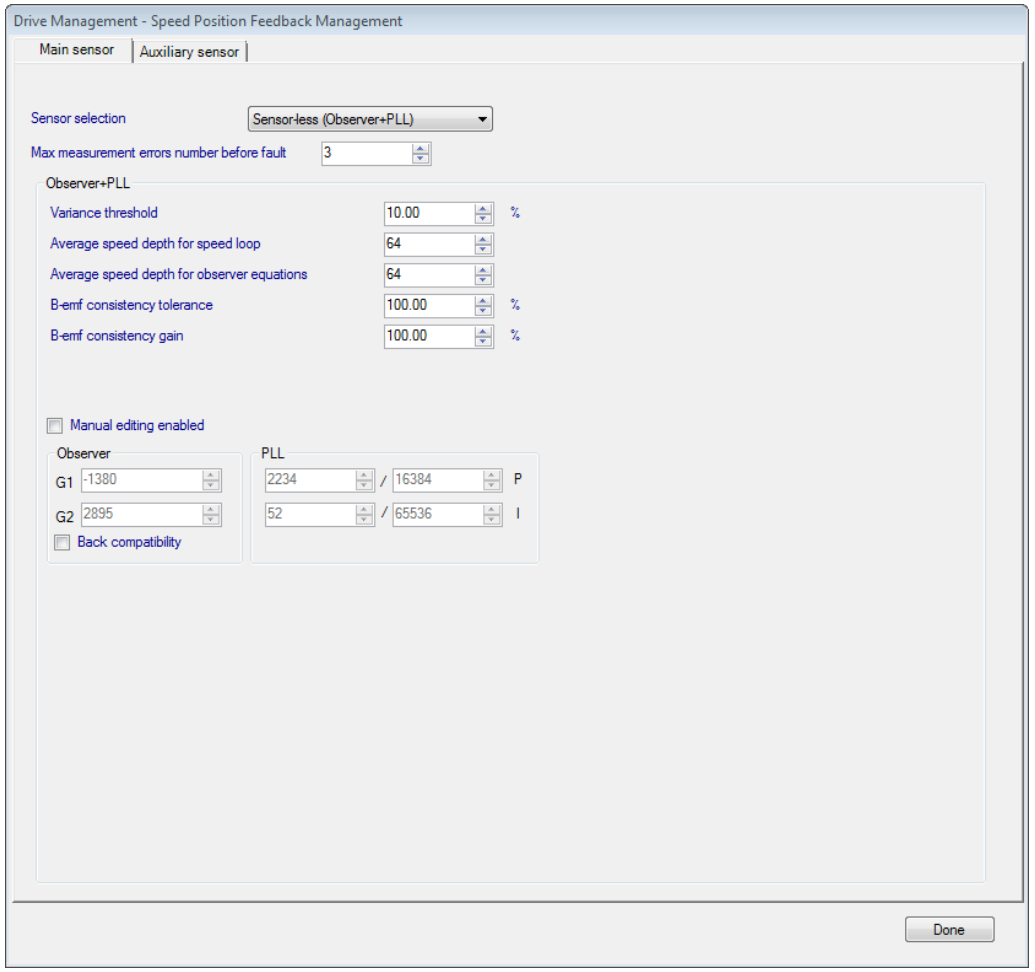


Speed Average value

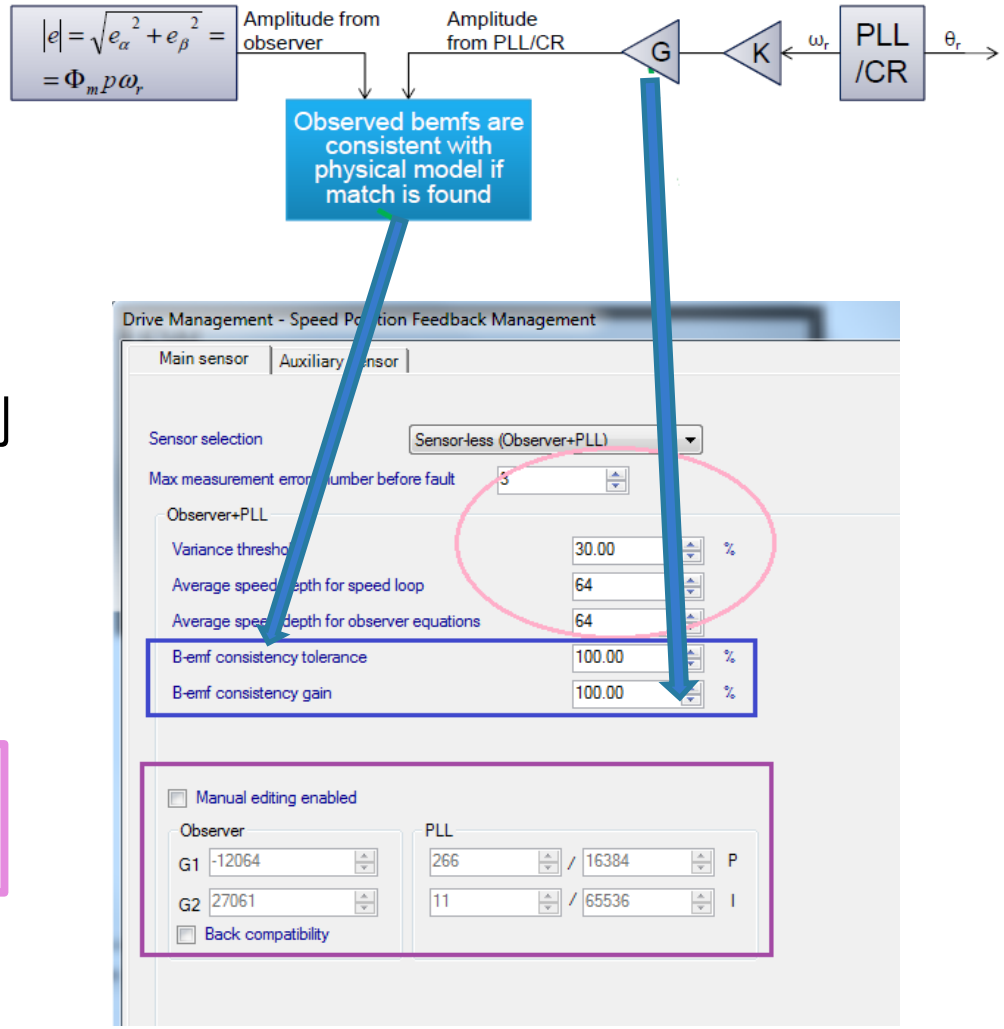
$$\mu = \frac{\sum_{i=1}^{64} \omega_i}{64}$$

Speed Variance

$$\sigma^2 = \frac{\sum_{i=1}^{64} (\omega_i - \mu)^2}{64}$$



Workbench 关键配置参数说明



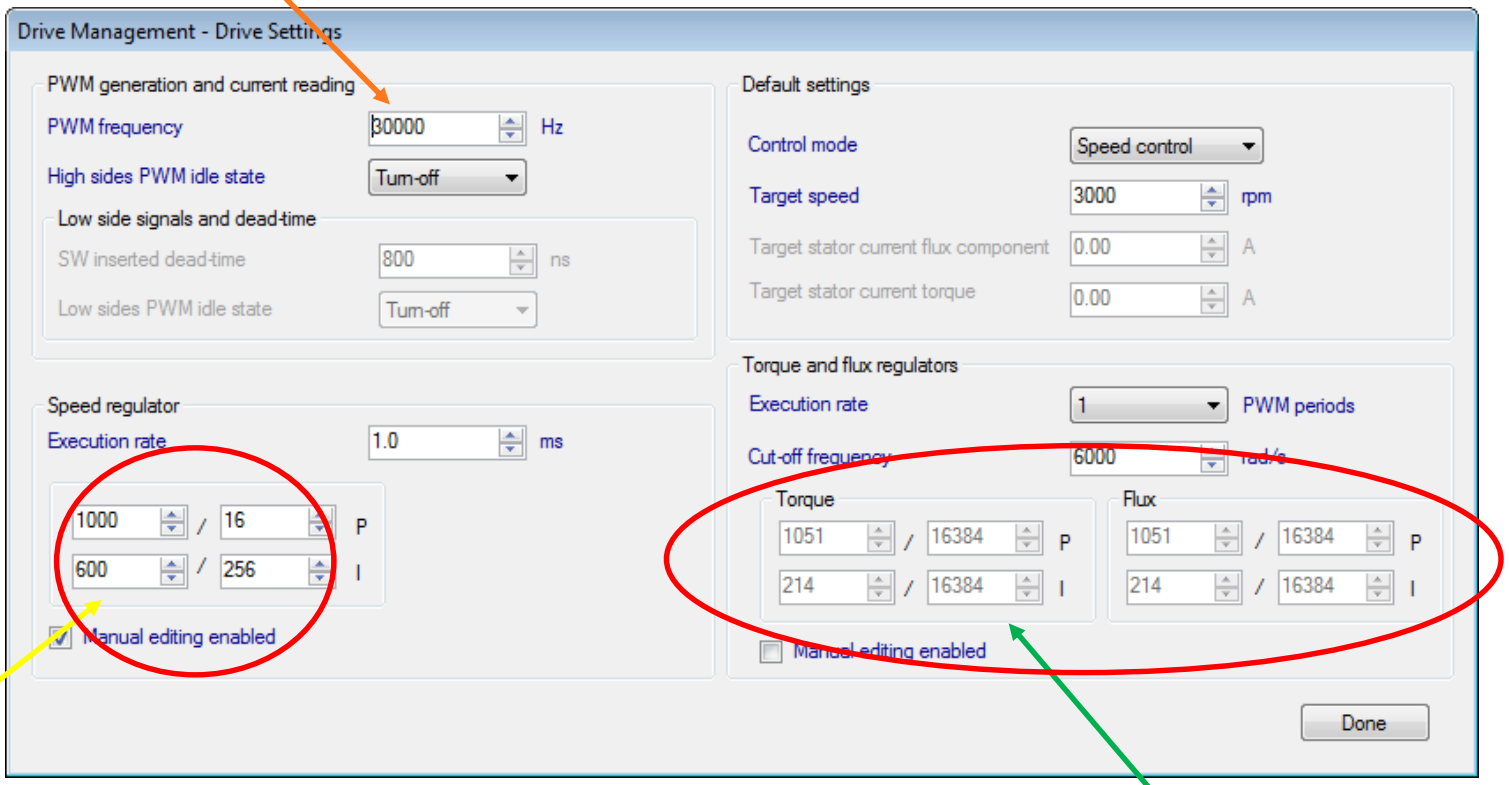
- BEMF收敛：
  - 一般用于堵转判断
  - 设定为100%时，忽略此判断

根据参数计算得到的系数，一般情况下无需修改，保持为默认数据

life.augmented

Workbench 关键配置参数说明

PWM载波频率



速度环Kp & Ki

电流环Kp & Ki

life.augmented

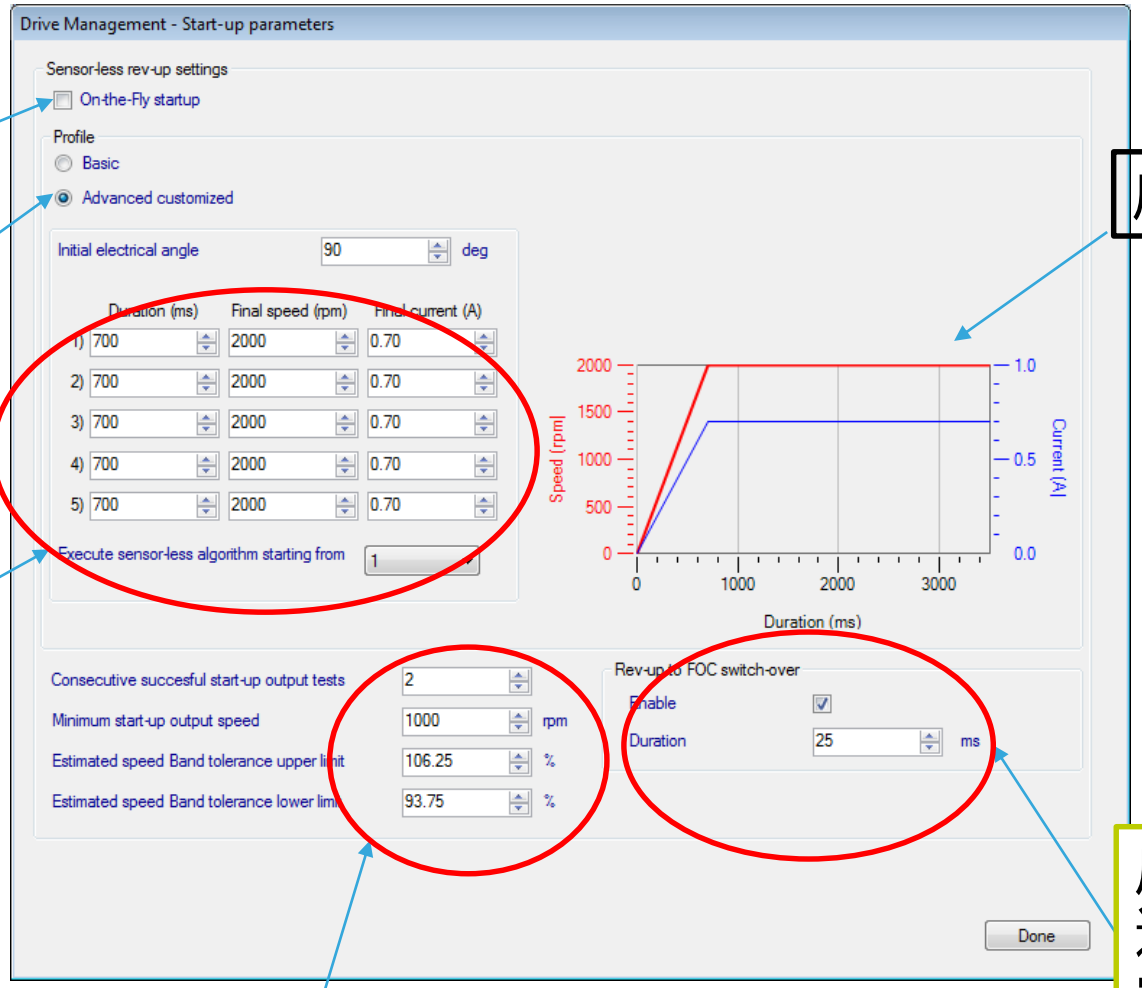
Workbench 关键配置参数说明

- 配置无传感器开环启动参数
  - 有传感器不用配置

顺逆风启动检测使能

折线或曲线启动选择

启动曲线配置，包含速度，电流，时间



启动曲线

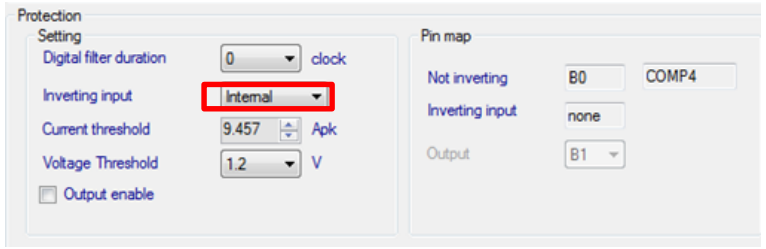
启动-FOC过渡期设定

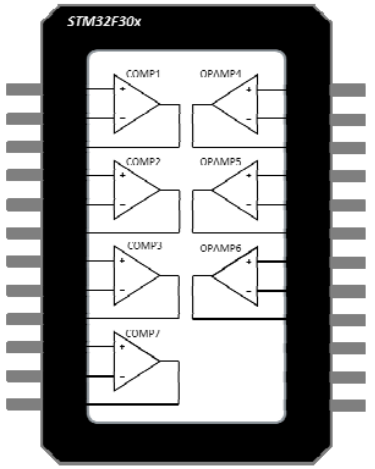
切入闭环条件判定

life.augmented

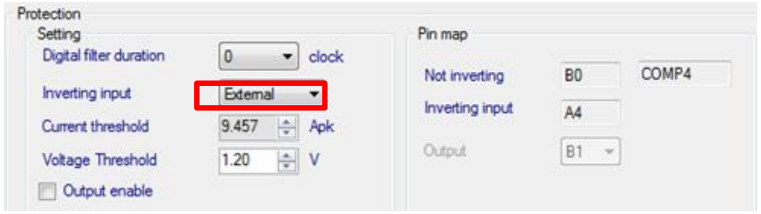
COMP 的配置

- 3 种方式设定OCP的过流阈值
  - 内部参考值，省一个管脚，但是该值较粗（只有有限的几个值可选）

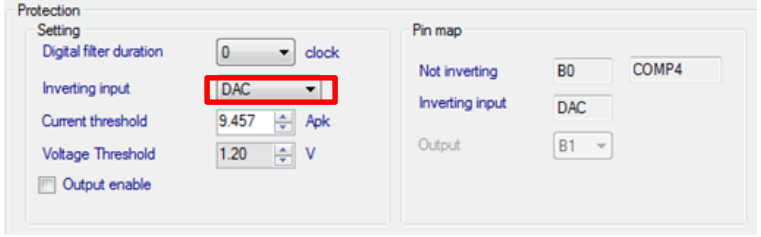




  - 外部参考值，需要一个管脚（PA4），但是该值容易调整

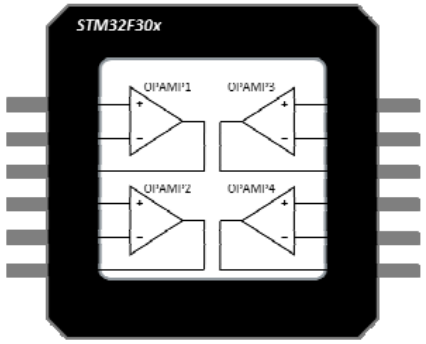


  - DAC通道，需要一个管脚（DAC窗口定义），该值容易调整



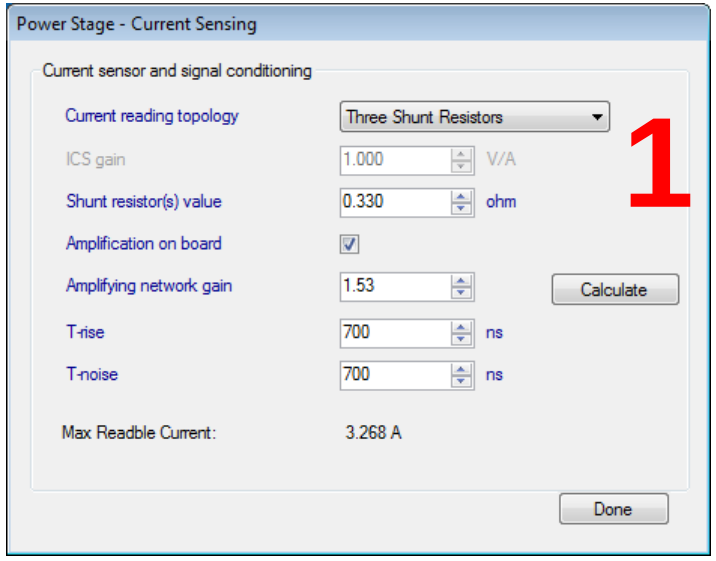
life.augmented

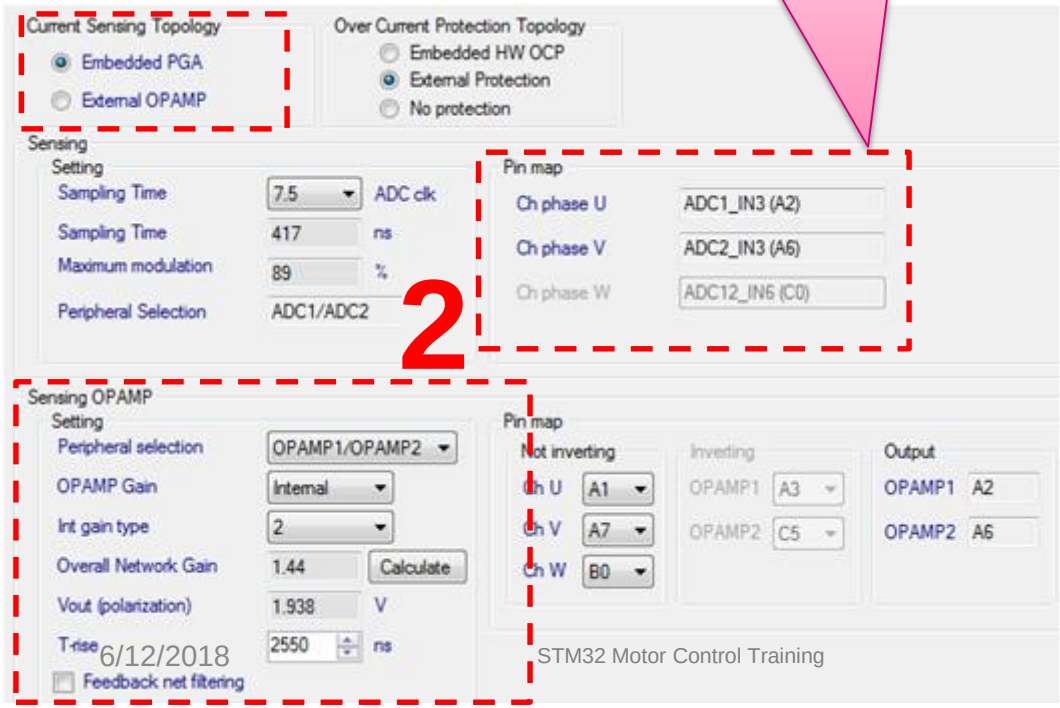
内置高速运放的配置



MCU	Configuration	or
STM32F303	OPAMP1+ADC1 OPAMP2+ADC2	OPAMP3+ADC3 OPAMP4+ADC4
STM32F302	OPAMP1+ADC1 OPAMP2+ADC2	

不能选择ADC的引脚，因为是和OPAMP共用同一个引脚



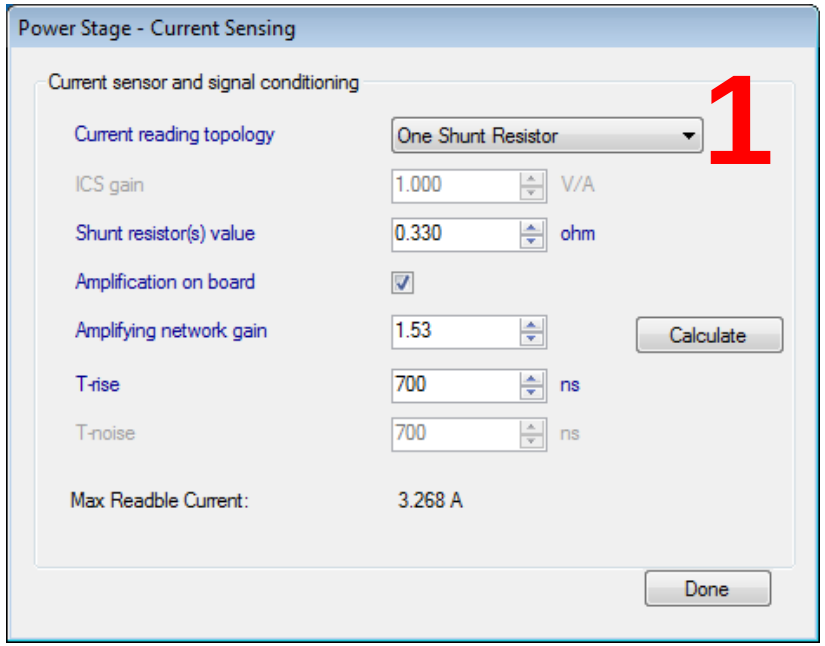


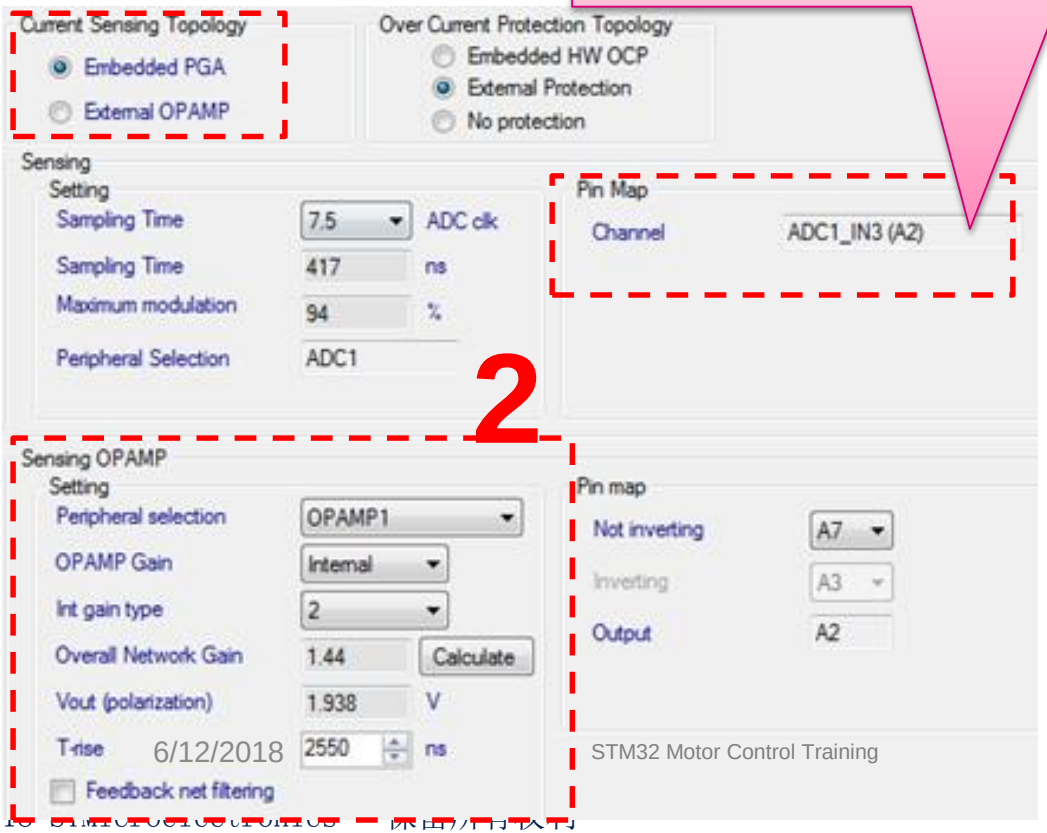
life.augmented

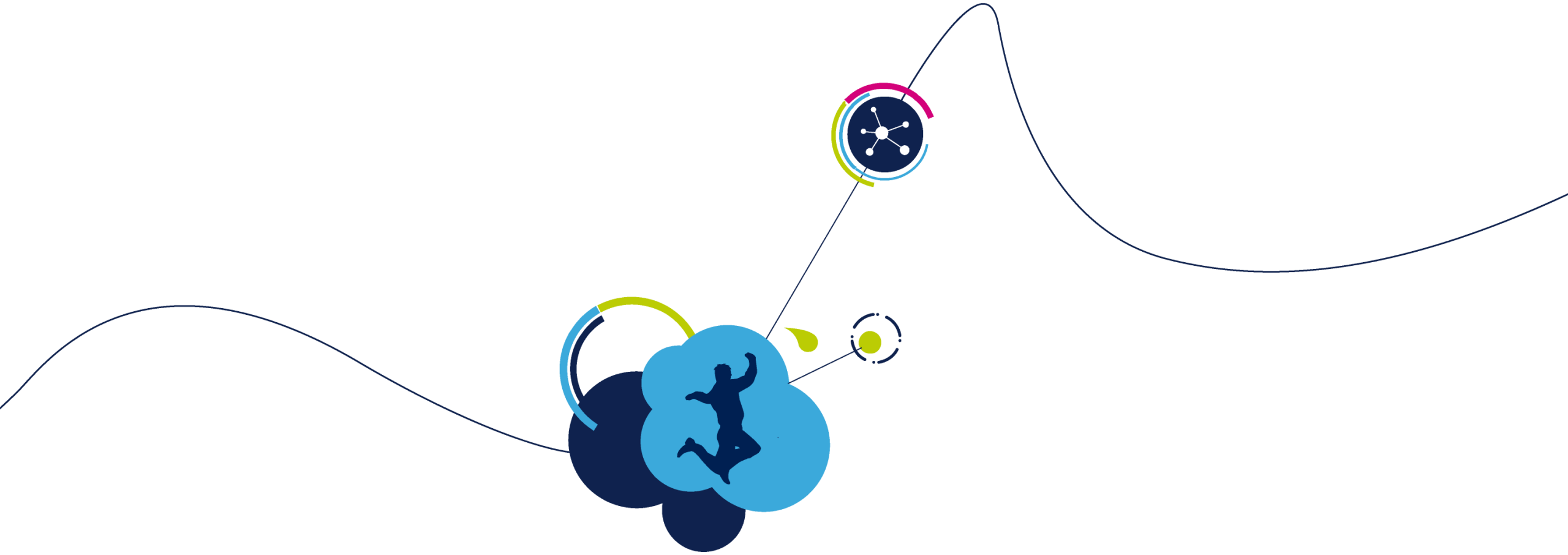
内置高速运放的配置

MCU	Configuration	or	or	or
STM32F303	OPAMP1+ADC1	OPAMP2+ADC2	OPAMP3+ADC3	OPAMP4+ADC4
STM32F302	OPAMP1+ADC1			

不能选择ADC的引脚，因为是和OPAMP共用同一个引脚







5 .MC SDK5.0 评估硬件

life.augmented

支持组成灵活的各类马达控制系统

灵活的各种EVB电机控制平台

EVB控制板

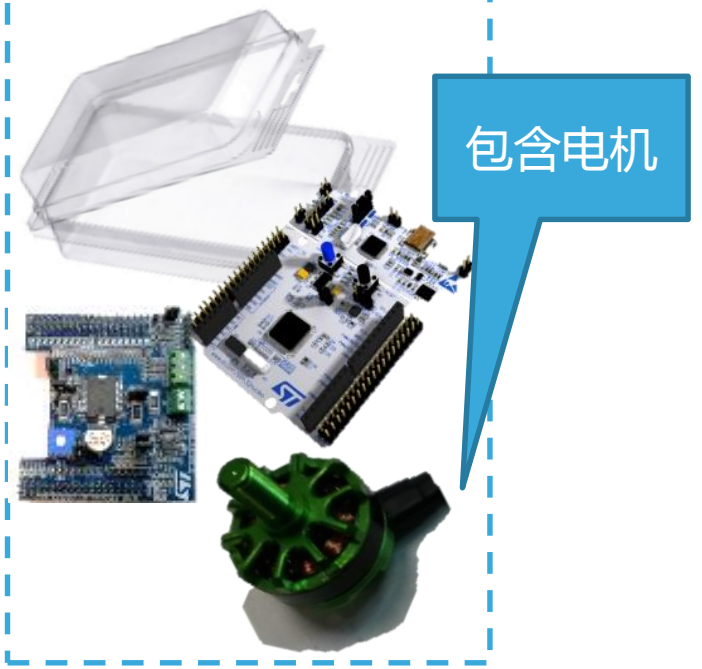
MC Connector (连接电缆)

EVB功率板

完整的电机控制驱动板

Nucleo + X-NUCLEO板

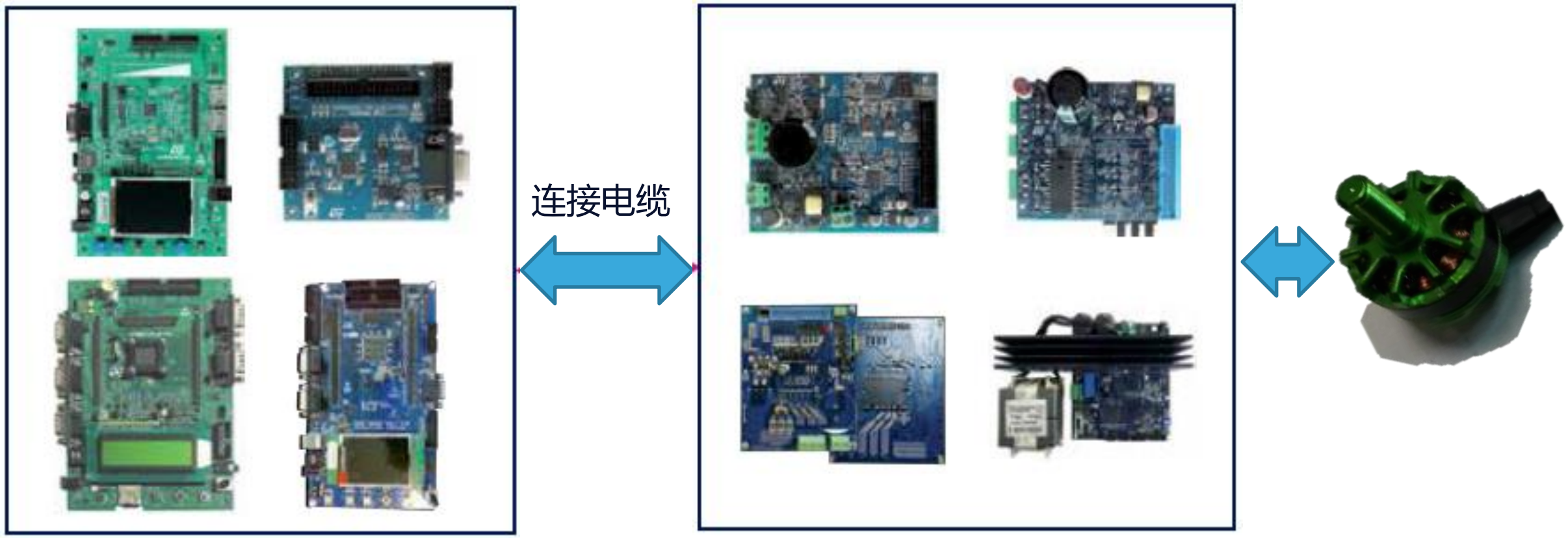
电机控制套件



life.augmented

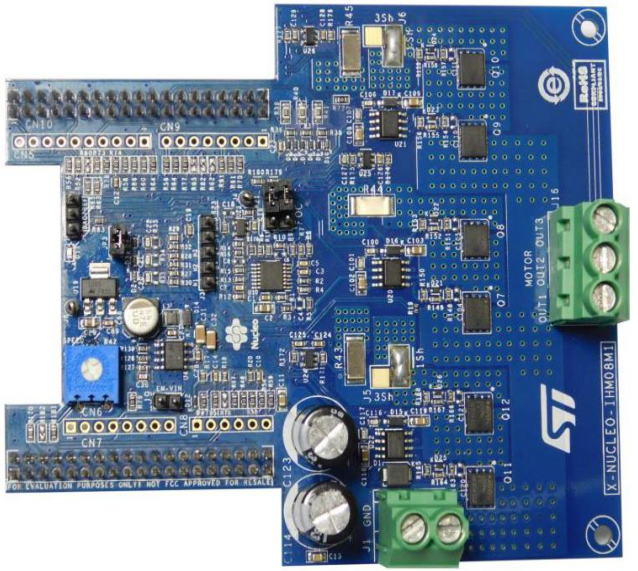
MCU EVB + PB EVB + Motor

- 整合的MCU EVB和功率板EVB满足客户的应用需求



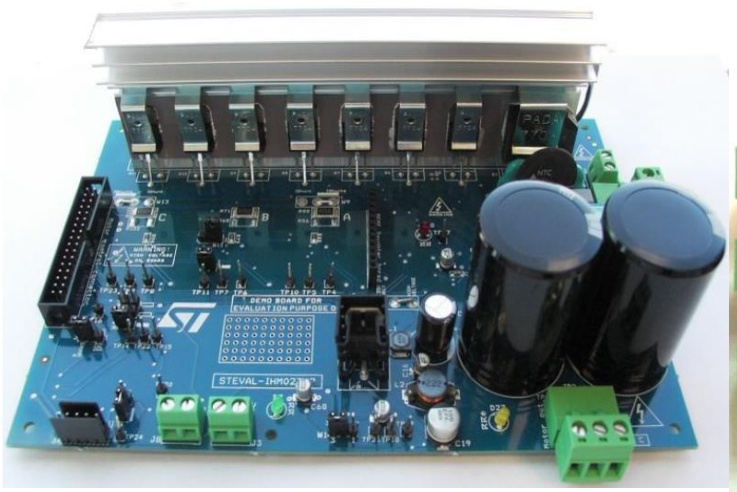
直流供电的NUCLEO功率板简介

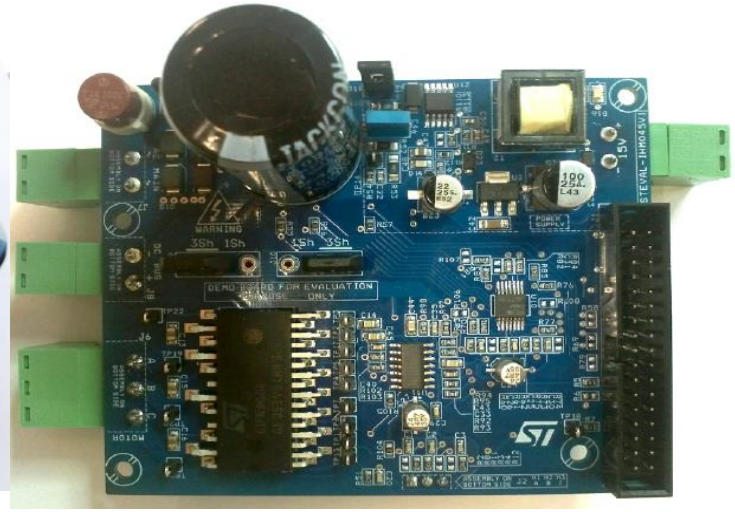
	PB Name	Power(W)/Current(A)	Voltage(V)	Motor Type/Control Type	Application
1	X-NUCLEO-IHM07M1	1.4A(2.8A)	8-48V DC	PMSM/BLDC FOC/6 Step 1/3 shunt	Drones
2	X-NUCLEO-IHM08M1	15A	10-48V DC	PMSM/BLDC FOC/6 Step 1/3 shunt	Drones,e-bikes,drills,pumps
3	X-NUCLEO-IHM11M1	1.3A	1.8-10 VDC	BLDC FOC 1 shunt	Drones

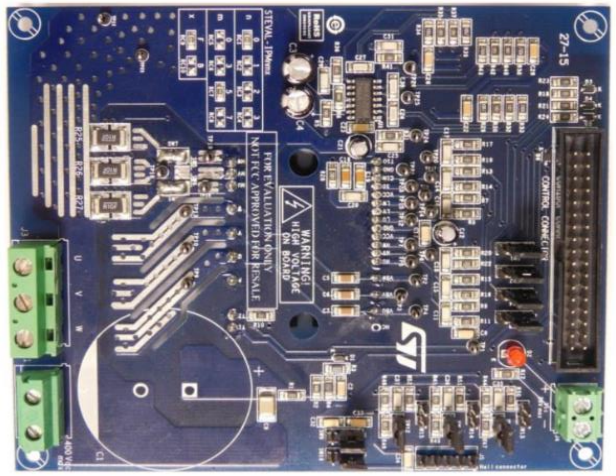


直流供电的EVB功率板简介

No.	PB Name	Power(W)/Current(A)	Voltage(V)	Motor Type/Control Type	Application
4	STM3210B-MCKIT	7A	42V DC	PMSM/BLDC FOC/6 Step 1/3 shunt	Drones,e-bikes,drills,pumps
5	STEVAL-IHM023V3	1KW	125-400V DC	PMSM/BLDC FOC 3 shunt	Pumps,compressors,wash machines
6	STEVAL-IHM028V2	2KW	125-400V DC	PMSM/BLDC FOC 1/3 shunt	Pumps,compressors,wash machines
7	STEVAL-IHM025V1	1KW	125-400V DC	PMSM/BLDC FOC 1/3 shunt	Pumps,compressors,wash machines
8	STEVAL-IHM045V1	100W	40-400V DC	PMSM/BLDC FOC 1/3 shunt(Dual motor)	Pumps,compressors,fans,dish washers
9	STEVAL-IPM05F	500W	125-400V DC	PMSM/BLDC FOC 1/3 shunt	Water pumps,fans,dish washers
10	STEVAL-IPM10B	1.2KW	125-400V DC	PMSM/BLDC FOC 1/3 shunt	Pumps,compressors,air conditioning
11	STEVAL-IPM15B	1.5KW	125-400V DC	PMSM/BLDC FOC 1/3 shunt	Pumps,compressors,fans,dish washers







life.augmented

SDK5.0 Workbench 支持的MCU列表

MCU Family	MCU Type	Target MCU	Package	Board
F3	302	STM32F302VB STM32F302VC STM32F302R8	LQFP100 LQFP100 LQFP64	- - NUCLEO-F302R8
	303	STM32F303VB STM32F303VC STM32F303ZE STM32F303VE STM32F303RE	LQFP100 LQFP100 LQFP144 LQFP100 LQFP64	- - - STM32303E-EVAL NUCLEO-F303RE
F0	030	STM32F030RC STM32F030R8	LQFP64 LQFP64	- NUCLEO-F030R8
	031	STM32F031C6	LQFP48	
	051	STM32F051R8 STM32F051C8	LQFP64 LQFP64	
	072	STM32F072VB STM32F072RB	LQFP100 LQFP64	STM32072B-EVAL NUCLEO-F072RB
F4	4xx	STM32F417IG STM32F415ZG STM32F407IG	UFBGA176 LQFP144 UFBGA176	STM3241G-EVAL STEVAL-IHM039V1 STM3240G-EVAL
	446	STM32F446ZE STM32F446RE	LQFP144 LQFP64	STM32446E-EVAL NUCLEO-F446RE

所有 SDK4.3 支持的MCU SDK5.0 最终会全部支持!

SDK5.0 ST 参考板(1/2)

5.0 支持的控制板

Product	MCU	Serie	Line	DIE	WB	WB Type	CubeMx board
STM32	F030R8	F0	F030	440	NUCLEO-F030R8	Control board	NUCLEO-F030R8_STM32F030R8
STM32	F072RB	F0	F072	448	NUCLEO-F072RB	Control board	NUCLEO-F072RB_STM32F072RB
STM32	F302R8	F3	F302	439	NUCLEO-F302R8	Control board	NUCLEO-F302R8_STM32F302R8
STM32	F303RE	F3	F303	446	NUCLEO-F303RE	Control board	NUCLEO-F303RE_STM32F303RE
STM32	F072VB	F0	F072	448	STM32072B-EVAL	Control board	STM32072B-EVAL_STM32F072VB
STM32	F303VE	F3	F303	446	STM32303E-EVAL	Control board	STM32303E-EVAL_STM32F303VE
STM32	F446RE	F4	F446	421	NUCLEO-F446RE	Control board	NUCLEO-F446RE_STM32F446RE
STM32	F407IG	F4	F407	413	STM3240G-EVAL	Control board	STM3240G-EVAL_STM32F407IG
STM32	F417IG	F4	F417	413	STM3241G-EVAL	Control board	STM3241G-EVAL_STM32F417IG
STM32	F446ZET	F4	F446	421	STM32446E-EVAL	Control board	STM32446E-EVAL_STM32F4446ZET
STM32	F415ZGT8	F4	F415		STEVAL-IHM039V1	Control board	

life.augmented

SDK5.0 ST 参考板(2/2)

5.0 支持的功率板

Part Number	General Description
STEVAL-IHM023V3	1 kW 3-phase motor control evaluation board featuring L6390 drivers and STGP10H60DF IGBT
STEVAL-IHM028V2	2 kW 3-phase motor control evaluation board featuring the STGIPS20C60 IGBT intelligent power module
STEVAL-IHM045V1	3-phase high voltage inverter power board for FOC based on the STGIPN3H60A (SLLIMM™-nano)
STEVAL-IPM05F	Motor control power board based on the SLLIMM™ 2nd series of IGBT IPMs
STEVAL-IPM07F	Motor control power board based on the SLLIMM™ 2nd series of IGBT IPMs
STEVAL-IPM10B	Motor control power board based on the SLLIMM™ 2nd series of IGBT IPMs
STEVAL-IPM10F	Motor control power board based on the SLLIMM™ 2nd series of IGBT IPMs
STEVAL-IPM15B	Motor control power board based on the SLLIMM™ 2nd series of IGBT IPMs
X-NUCLEO-IHM07M1	Three-phase brushless DC motor driver expansion board based on L6230 for STM32 Nucleo
X-NUCLEO-IHM08M1	Low-Voltage BLDC motor driver expansion board based on STL220N6F7 for STM32 Nucleo
X-NUCLEO-IHM11M1	Low voltage three-phase brushless DC motor driver expansion board based on STSPIN230 for STM32 Nucleo

life.augmented

NUCLEO和EVB的功率板及控制板组合

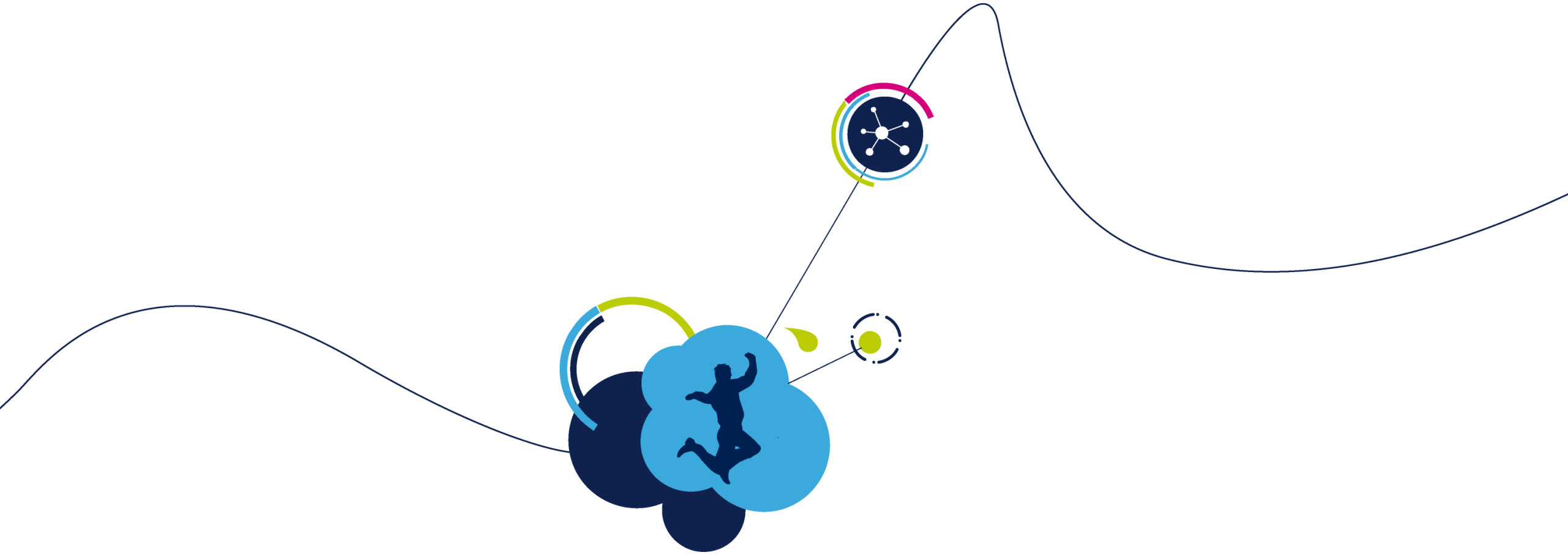
5.0 支持的NUCLEO功率板和控制板的组合

	NUCLEO-F302R8	NUCLEO-F303RE	NUCLEO-F072RB	NUCLEO-F446RE	NUCLEO-F030R8
X-NUCLEO-IHM07M1	√	√	√	√	√
X-NUCLEO-IHM08M1	√	√	√	√	√
X-NUCLEO-IHM11M1	√	√	√	√	√

5.0 支持的EVB功率板和控制板的组合

	STM32072B-EVAL	STM32303E-EVAL	STM3240G-EVAL	STM3241G-EVAL	STM32446E-EVAL	STEVAL-IHM039V1
STEVAL-IHM023V3	√	√	√	√	√	√
STEVAL-IHM028V2	√	√	√	√	√	√
STEVAL-IHM045V1	√	√	√	√	√	√
STEVAL-IPM05F	√	√	√	√	√	√
STEVAL-IPM07F	√	√	√	√	√	√
STEVAL-IPM10B	√	√	√	√	√	√
STEVAL-IPM10F	√	√	√	√	√	√
STEVAL-IPM15B	√	√	√	√	√	√

life.augmented



6 .实验2：基于MC SDK5.0 API，速度控制与电机启动停止

life.augmented

用户代码在main.c中

- /* USER CODE BEGIN x */

  用户代码编写区域

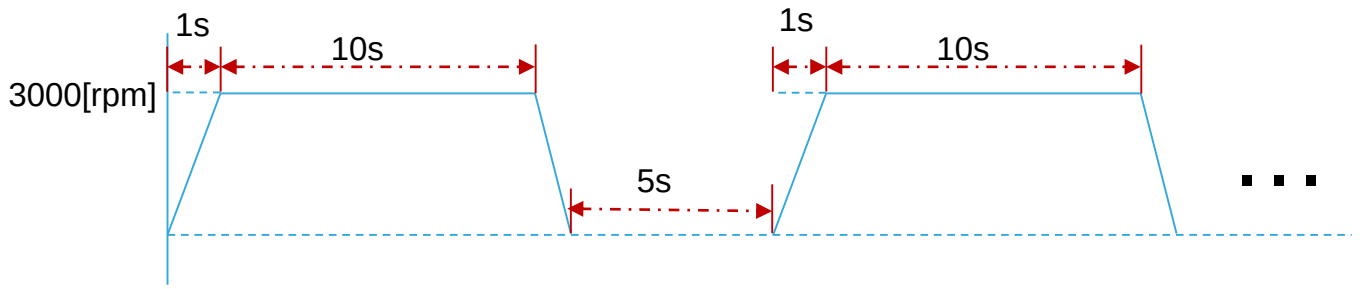
  …

- /* USER CODE END x */

(x = 0,1,2,3….)

Task1

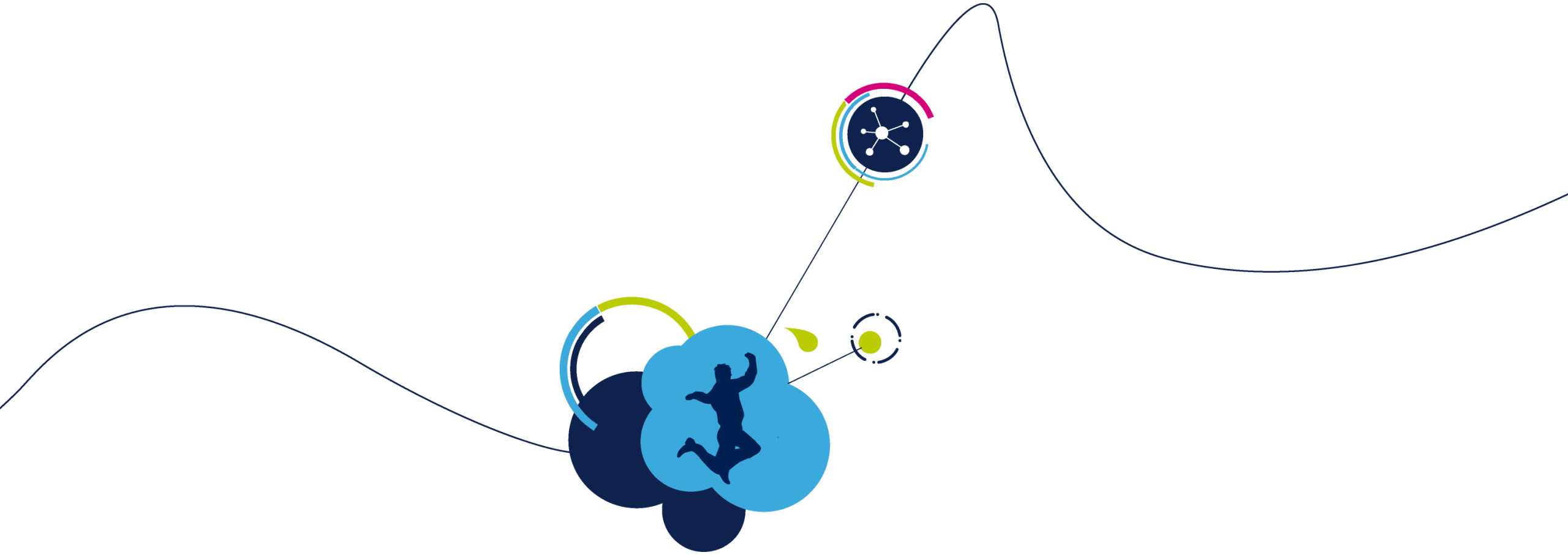
1. 电机以3000rpm运行
2. 程序启动电机运行，运行速度为3000rpm，在10s后停止转动
3. 停止5s后电机重新运行，速度依然是3000rpm
4. 以上过程重复操作



Task1

1. 电机以3000rpm运行
   - MC_ProgramSpeedRampMotor1(3000/6,1000);
   - 1000ms内加速到3000RPM，3000/6：
   - 以0.1HZ作为单位的速度指令
2. 程序启动电机运行，运行速度为3000rpm，在10s后停止转动
3. 停止5s后电机重新运行，速度依然是3000rpm
4. 以上过程重复操作
   - MC_StartMotor1();-- 运行电机
   - MC_StopMotor1();--停止电机
   - 5s的操作使用定时机制产生，这里使用SysTick的500us作为延迟基础，也可以使用Timer计时，软件定时等都可以；
   - Stm32f3xx_hal.c->HAL_GetTick()

life.augmented



7. 实验3：基于MC SDK5.0 PI 组件接口函数做在线参数修改

life.augmented

Task2

1. 程序中修改速度PI参数
2. 修改为原始值的2倍
3. Workbench中观察参数修改后的速度曲线
4. 修改为原始值的0.5倍，再次观察速度曲线

life.augmented

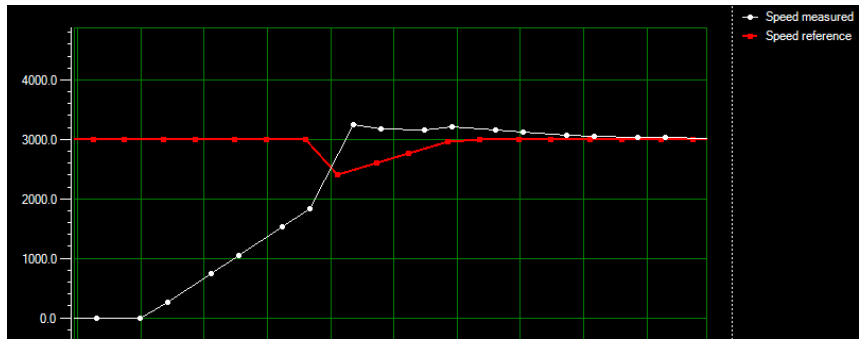
Task2

1. 程序中修改速度PI参数

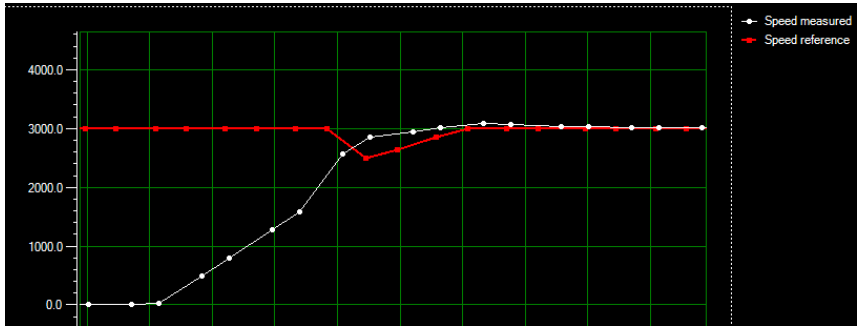
2. 修改为原始值的2倍
   - static int16_t Speed_Kp,Speed_Ki;
   - pMctHdl = GetMCT(M1);
   - Speed_Kp = PID_GetKP(pMctHdl->pPIDSpeed);
   - Speed_Ki = PID_GetKI(pMctHdl->pPIDSpeed);
   - PID_SetKP(pMctHdl->pPIDSpeed,Speed_Kp*2);
   - PID_SetKI(pMctHdl->pPIDSpeed,Speed_Ki*2);

life.augmented

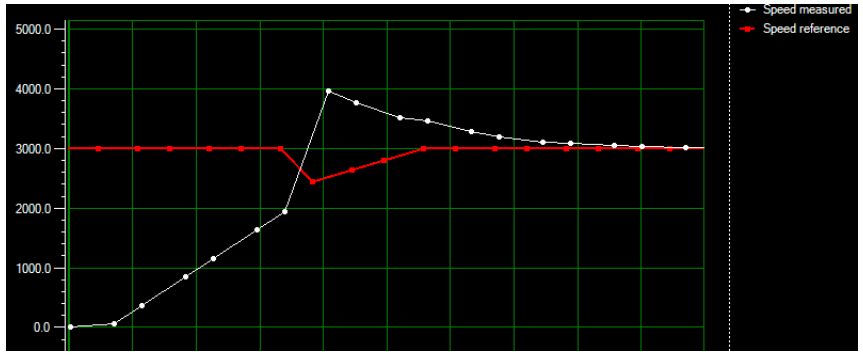
Task2



原始PID数据

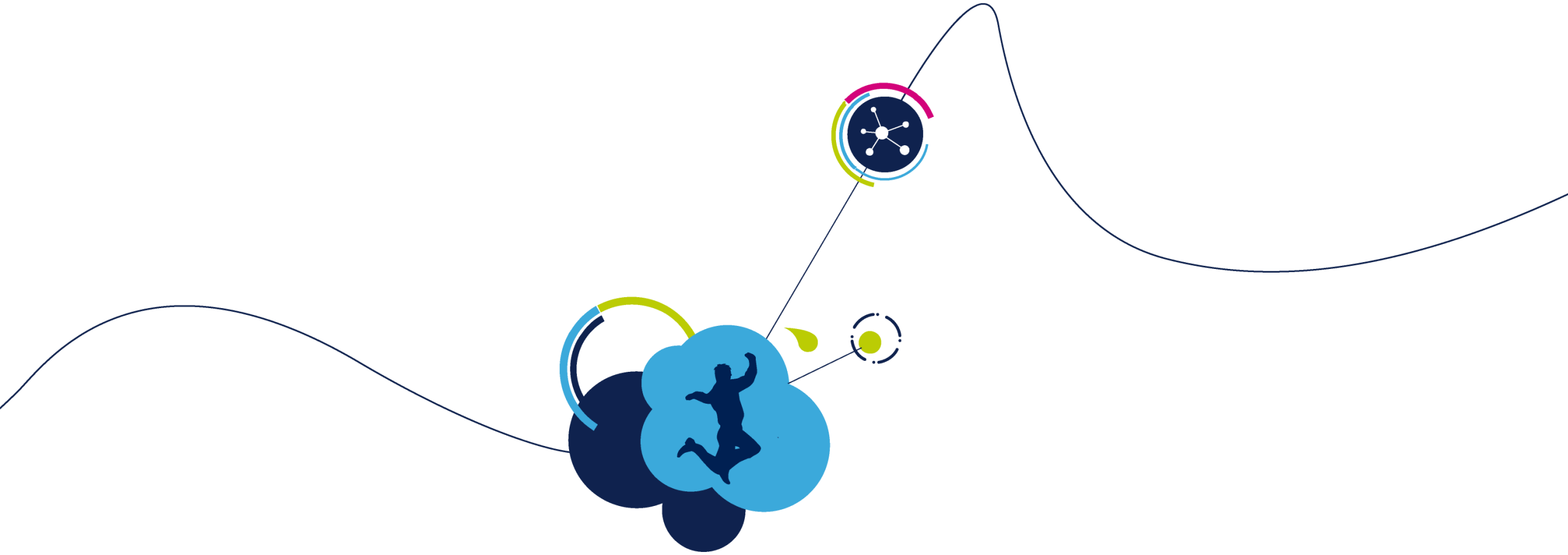


2倍PID数据



0.5倍PID数据

life.augmented

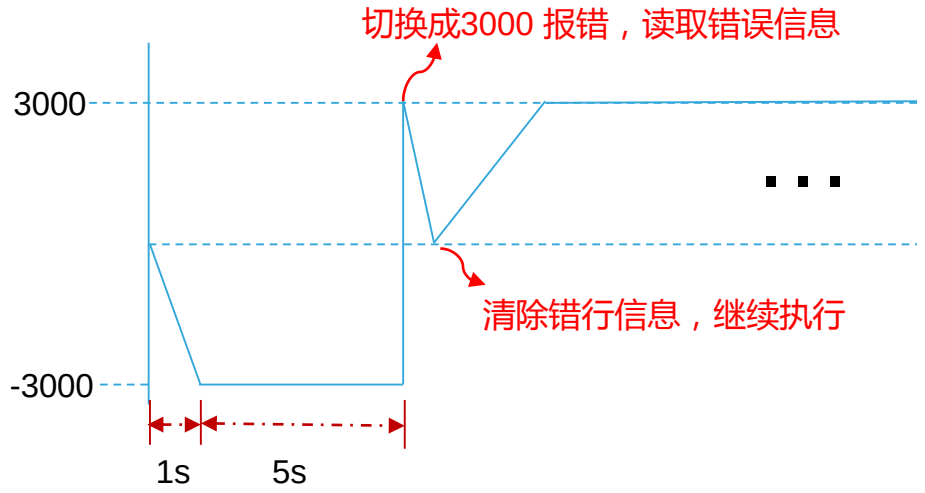


8.实验4：基于MC SDK5.0 状态的切换

life.augmented

Task3

1. 速度反转-3000rpm
2. 速度立刻正转3000rpm
3. 产生状态报错，需要程序返回报错信息
4. 清除报错信息，返回到IDLE状态
5. 然后继续执行正转3000rpm速度指令



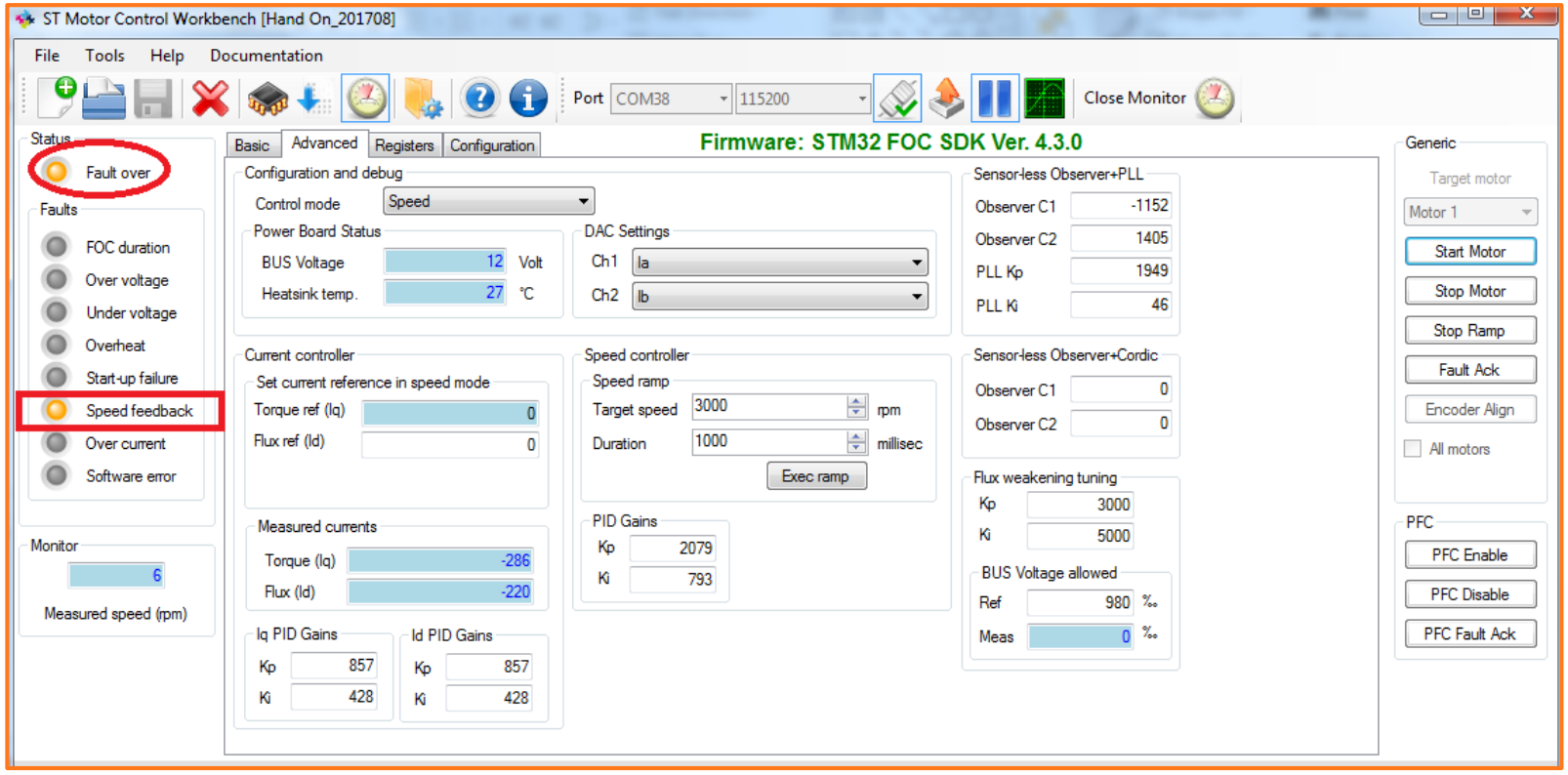
Task3

1. 速度正转3000rpm
   - MC_ProgramSpeedRampMotor1(3000/6,1000);--正转3000RPM
2. 速度立刻反转-3000rpm
   - MC_ProgramSpeedRampMotor1(-3000/6,1000);--反转3000RPM

life.augmented

Task3

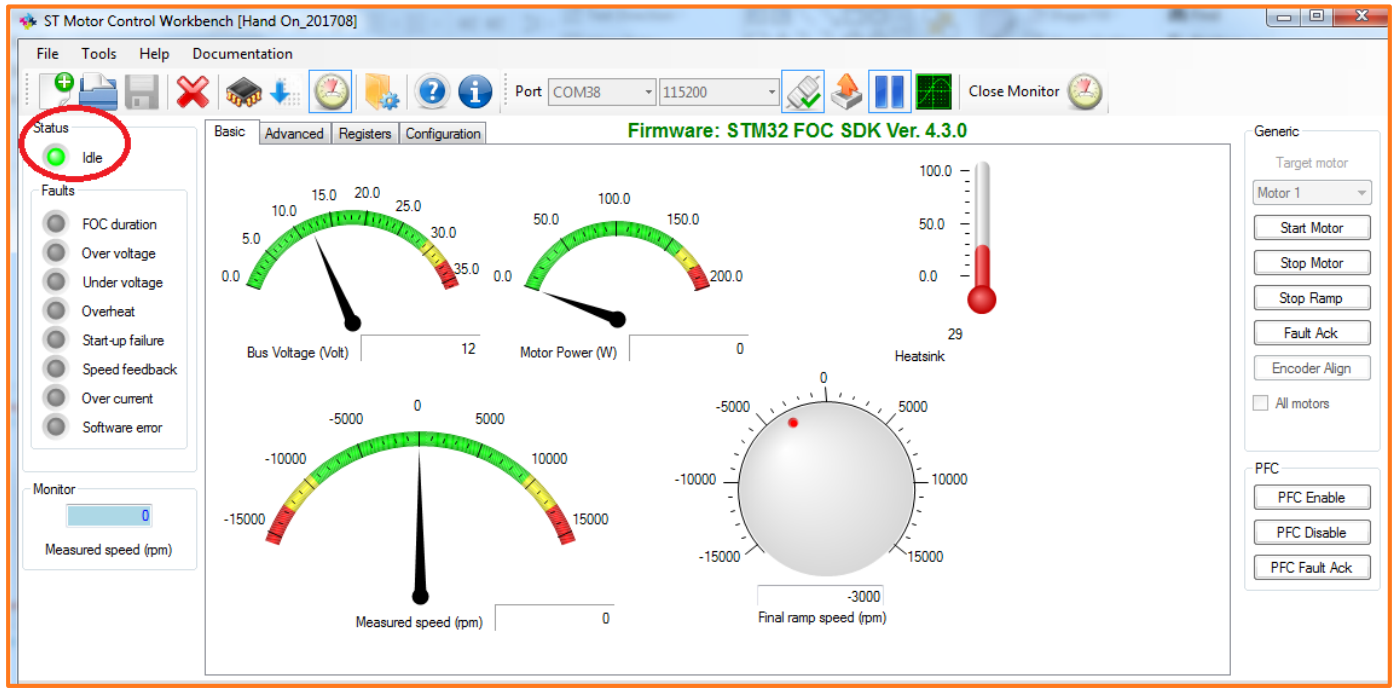
3. 产生状态报错，需要程序返回报错信息
   - State t sts_motor1 = MCI_GetSTMStateMotor1();
   - STM_GetFaultState(&STM[M1]);



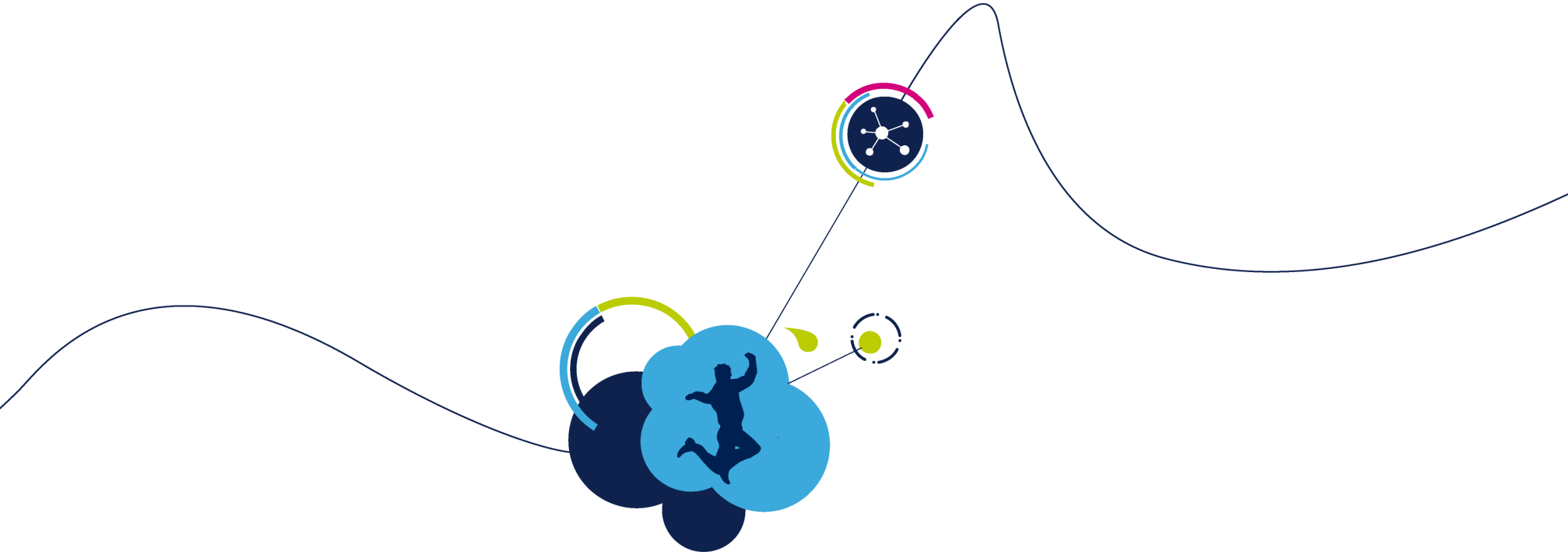
life.augmented

Task3

- 4. 清除报错信息，返回到IDLE状态
- 5. 然后继续执行反转-3000rpm速度指令
  - MC_AcknowledgeFaultMotor1();
  - MC_ProgramSpeedRampMotor1(MC_GetLastRampFinalSpeedMotor1(), 1000);



life.augmented



9.实验5：开放性试验

life.augmented

开放性试验

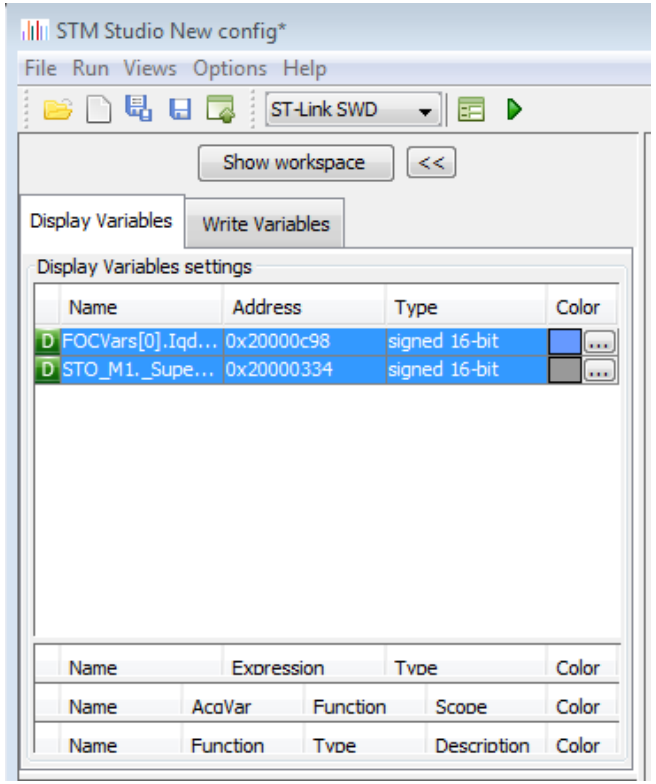
1. 程序控制Idref，Iqref
2. 使用STMStudio对变量改变进行观测
3. 旋钮ADC采样（PB1）对应到Iqref的变化上(Torque Mode)
4. 观察速度以及Studio上的变量变化
5. 可以加入自己想要得到的信号变量

life.augmented

开放性试验

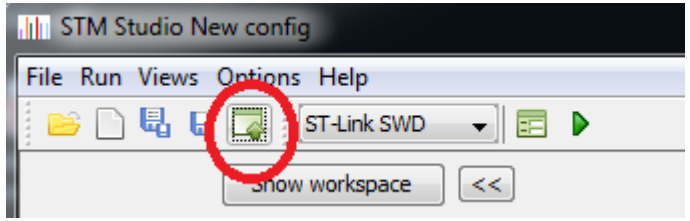
1. 程序控制Idref，Iqref
   - MC_ProgramTorqueRampMotor1(Tuning_Iqref,1000);

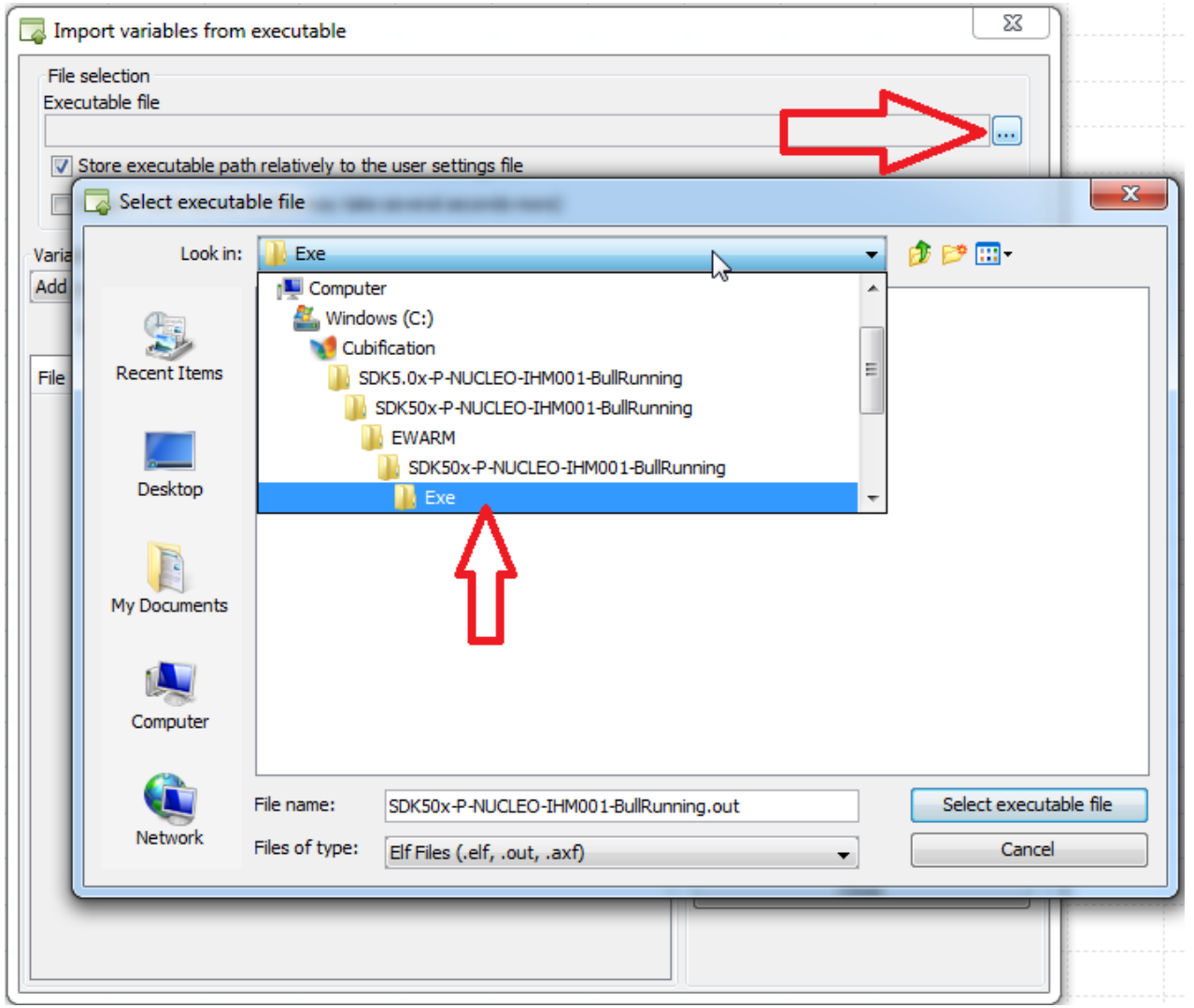
- 2. 使用STMStudio对变量改变进行观测



STMStudio

- STMStudio添加.out文件装载
  - SDK50x-P-NUCLEO-IHM001-BullRunning.out





开放性试验

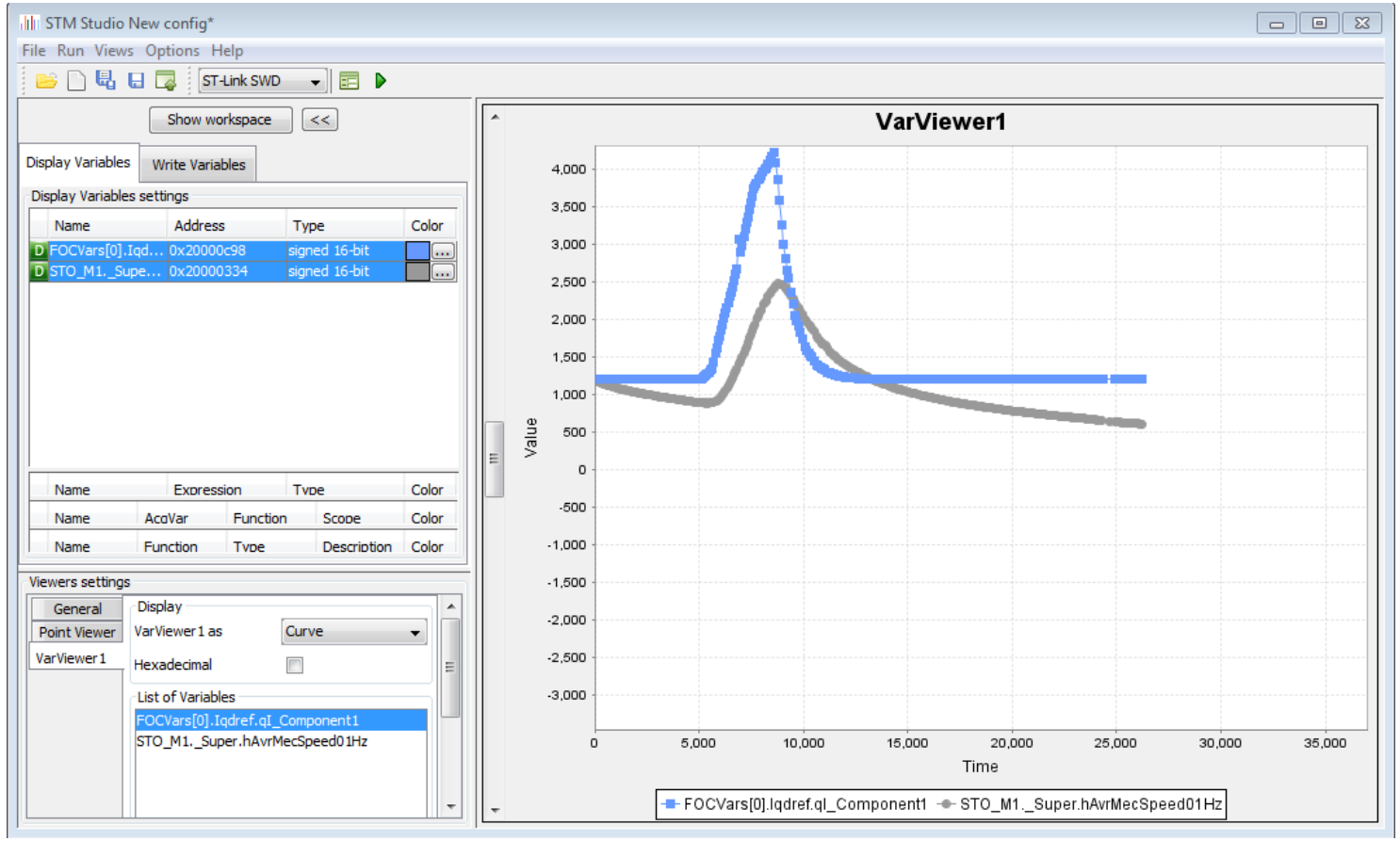
- 3. 旋钮ADC采样（PB1）对应到Iqref的变化上
  - MC_ProgramRegularConversion(ADC_Channel_12, ADC_SampleTime_601Cycles5);
  - MC_GetRegularConversionValue();

```
void Open_Task( void )
{
  pMciHdl = GetMCI( M1 );

  /* Do regular adc convert*/
  if( MC_GetRegularConversionState() == UDRC_STATE_IDLE )
  {
    MC_ProgramRegularConversion( ADC_CHANNEL_12, ADC_SAMPLETIME_601CYCLES_5 );
  }
  else if( MC_GetRegularConversionState() == UDRC_STATE_EOC )
  {
    /* Get ADC value*/
    Tuning_ADC = MC_GetRegularConversionValue();
  }

  if(Delay_Count == 0)
  {
    Delay_Count = HAL_GetTick();
  }
  else if(HAL_GetTick() > (Delay_Count + 1))
  {
    Delay_Count = 0;

    /* Get new Iqref*/
    Tuning_Iqref = ( Tuning_ADC + 1 ) * MAX_TUNING_IQ / 65536;

    if( Tuning_Iqref < MIN_TUNING_IQREF )
    {
      Tuning_Iqref = MIN_TUNING_IQREF;
    }
    /* Set new Iqref ramp*/
    MC_ProgramTorqueRampMotor1( Tuning_Iqref, 1000 );
    MC_StartMotor1();
  }
}
```

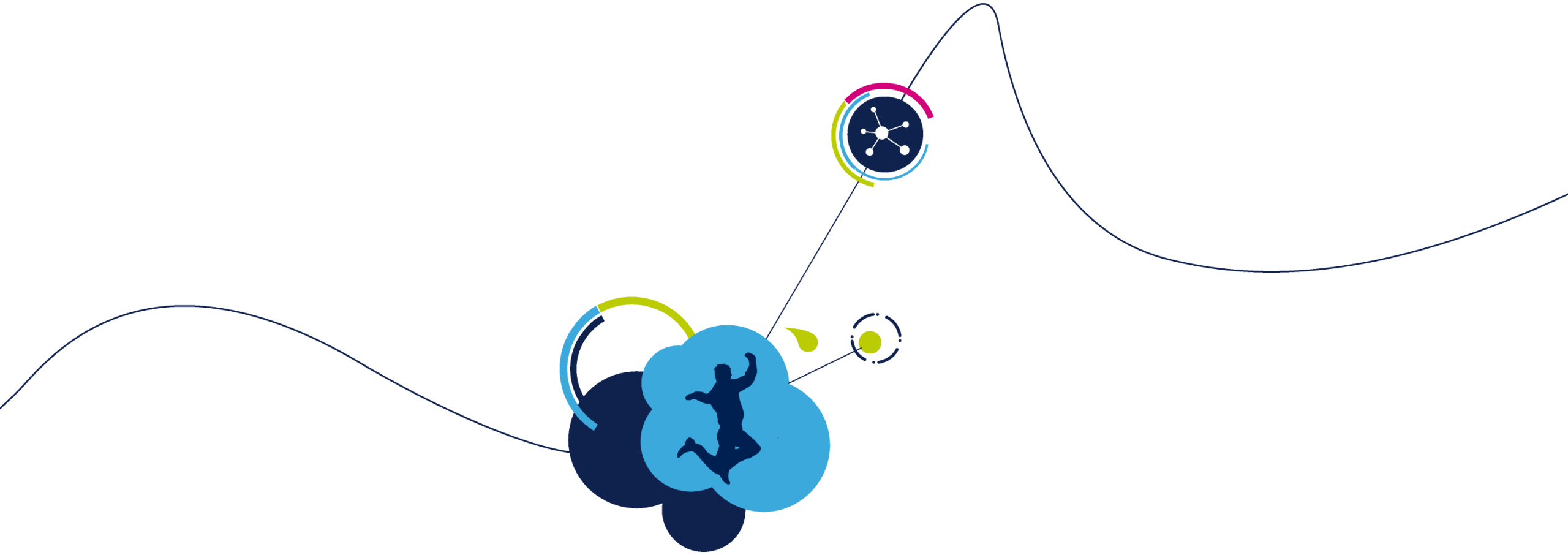
life.augmented

开放性试验

- 旋转Demo板上旋钮改变Iqref，察看速度和Iqref数据



life.augmented



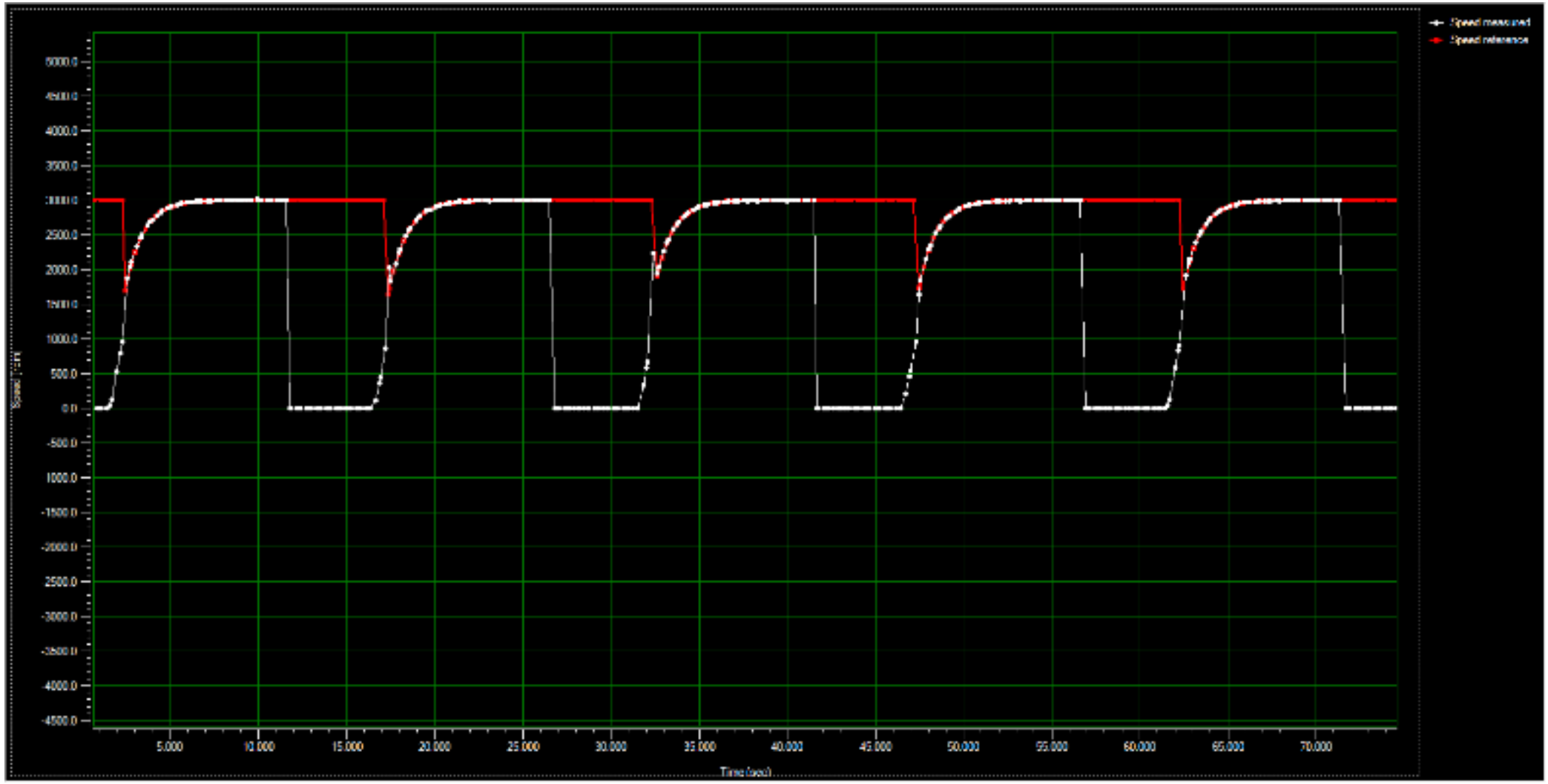
10.实验总结及问答

life.augmented

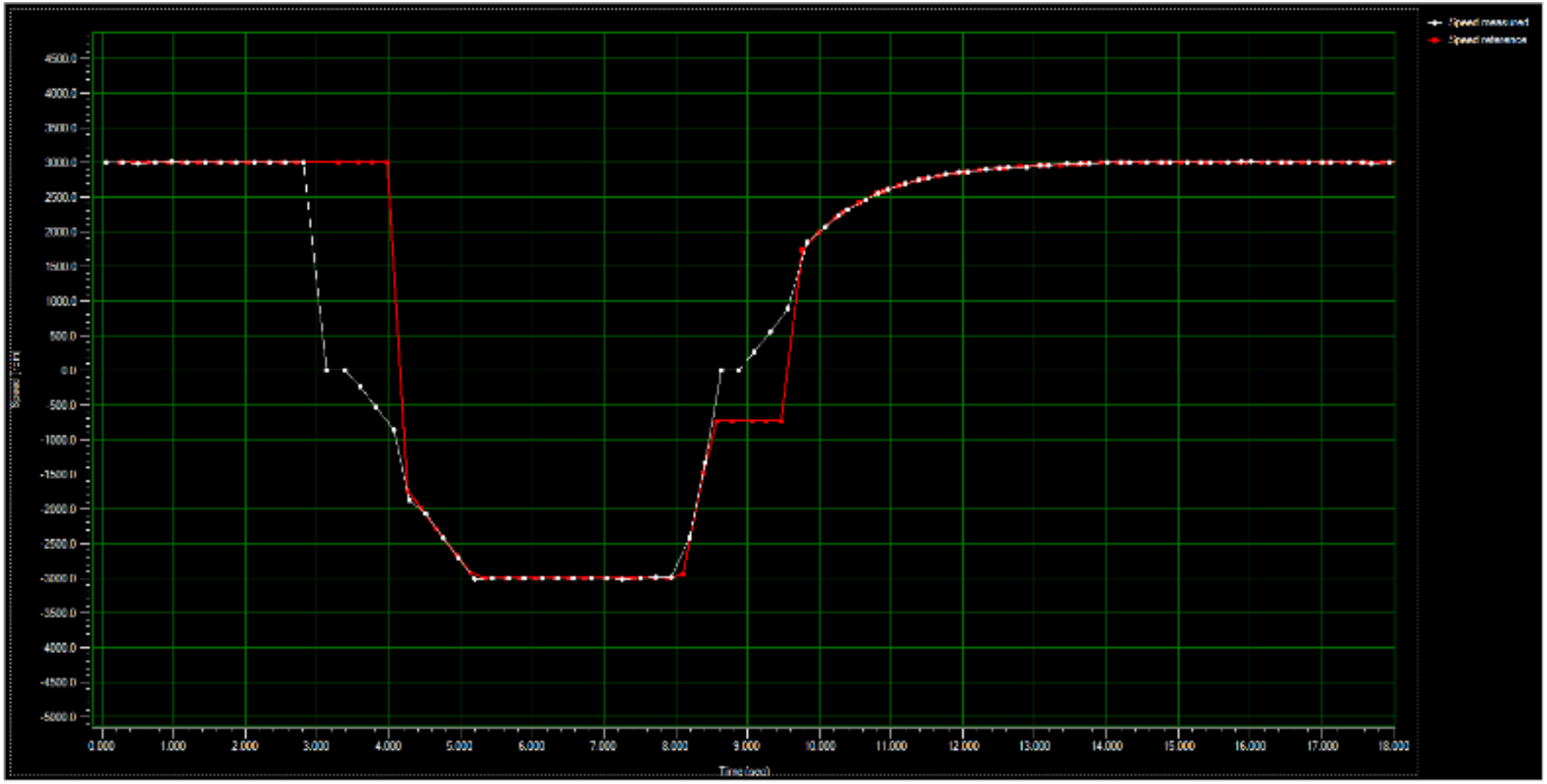
实验总结

项目	目的	说明
Task1	熟悉调用MCI实现电机基础操作	启动电机 -- MC_StartMotor1()
		停止电机 -- MC_StopMotor1()
		速度指令 -- MC_ProgramSpeedRampMotor1(3000 / 6, 1000)
	了解速度参数设定	参考速度指令是以0.1HZ作为单位
Task2	取得MCT句柄	pMctHdl = GetMCT(M1)
	知道如何找到相关操作函数	比如PID设定在pid_regulator.c中
	调用控制函数以及参数设定	取得当前数据 -- PID_GetKP(pMctHdl->pPIDSpeed)
		修改数据 -- PID_SetKP(pMctHdl->pPIDSpeed,Speed_Kp*2)
	Workbench观察速度曲线	操作在Workbench中
Task3	电机反转指令	MC_ProgramSpeedRampMotor1(-3000 / 6, 1000)
	调用函数知道当前电机运行状态	State_t sts_motor1 = MCI_GetSTMStateMotor1();
	得到报错信息	STM_GetFaultState(&STM[M1])
	清除报错，电机回到IDLE状态	MC_AcknowledgeFaultMotor1()
	Workbench做状态监控	注意Workbench状态变化
Open	使用力矩模式控制电机	MC_ProgramTorqueRampMotor1(Tuning_Iqref, 1000)
	旋钮ADC通道采样	MC_ProgramRegularConversion(ADC_CHANNEL_12, ADC_SAMPLETIME_601CYCLES_5)
	熟悉ST Studio使用	软件操作

Task1



Task3



重要通知 – 请仔细阅读

意法半导体公司及其子公司（“ST”）保留随时对ST 产品和/ 或本文档进行变更、更正、增强、修改和改进的权利，恕不另行通知。买方在
订货之前应获取关于ST 产品的最新信息。ST 产品的销售依照订单确认时的相关ST 销售条款。

买方自行负责对ST 产品的选择和使用， ST 概不承担与应用协助或买方产品设计相关的任何责任。

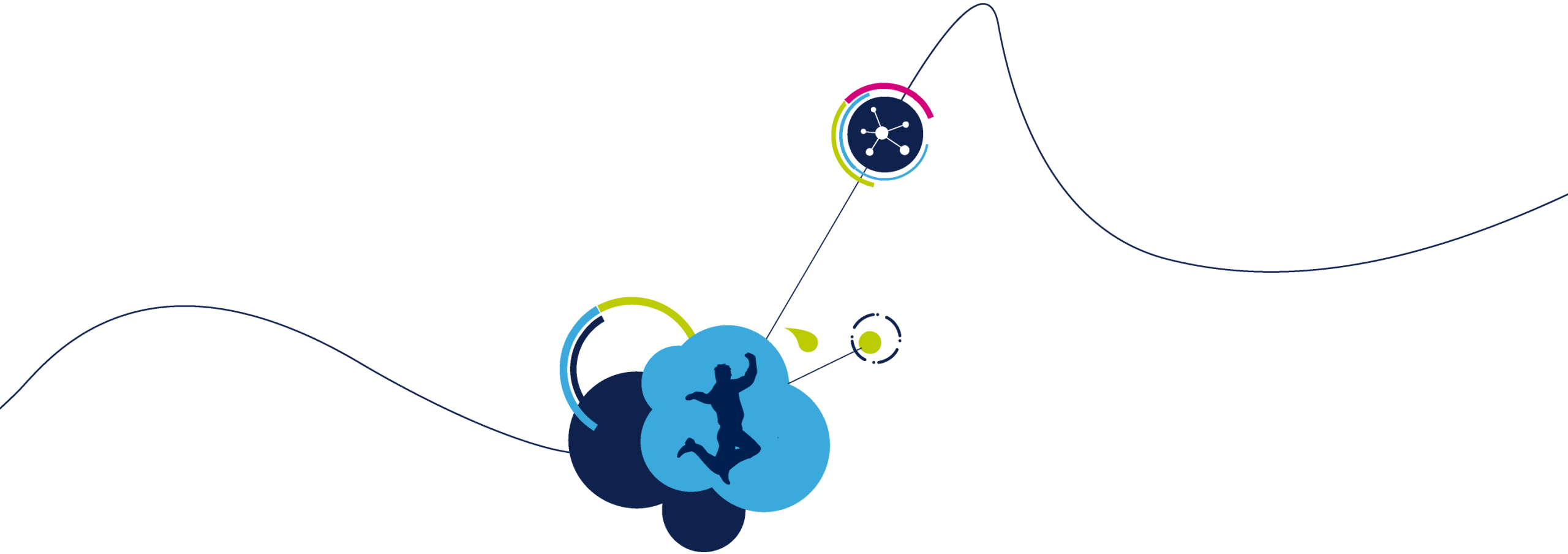
ST 不对任何知识产权进行任何明示或默示的授权或许可。

转售的ST 产品如有不同于此处提供的信息的规定，将导致ST 针对该产品授予的任何保证失效。

ST 和ST 徽标是ST 的商标。所有其他产品或服务名称均为其各自所有者的财产。

本文档中的信息取代本文档所有早期版本中提供的信息。

life.augmented



谢谢大家!

life.augmented