

SimCoder Module

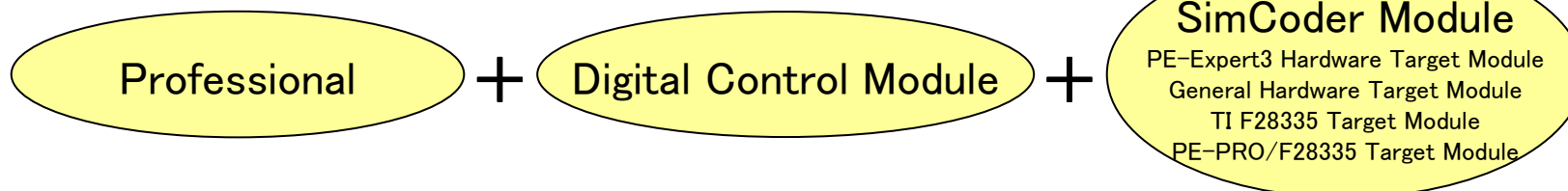
SimCoder Moduleとは

インバータのコントローラに対し、PSIMで作成した制御部と同じ動作をするCコードを自動で生成します

特徴

- 制御部のCコード自動生成が可能なので、実機検証への移行が効率的になります
- 初めて制御プログラミングを行う際の技術習得用サポートツールとしてもご利用いただけます
- コードを分かり易いPSIM回路で管理することができるため、変更や履歴管理が容易になります

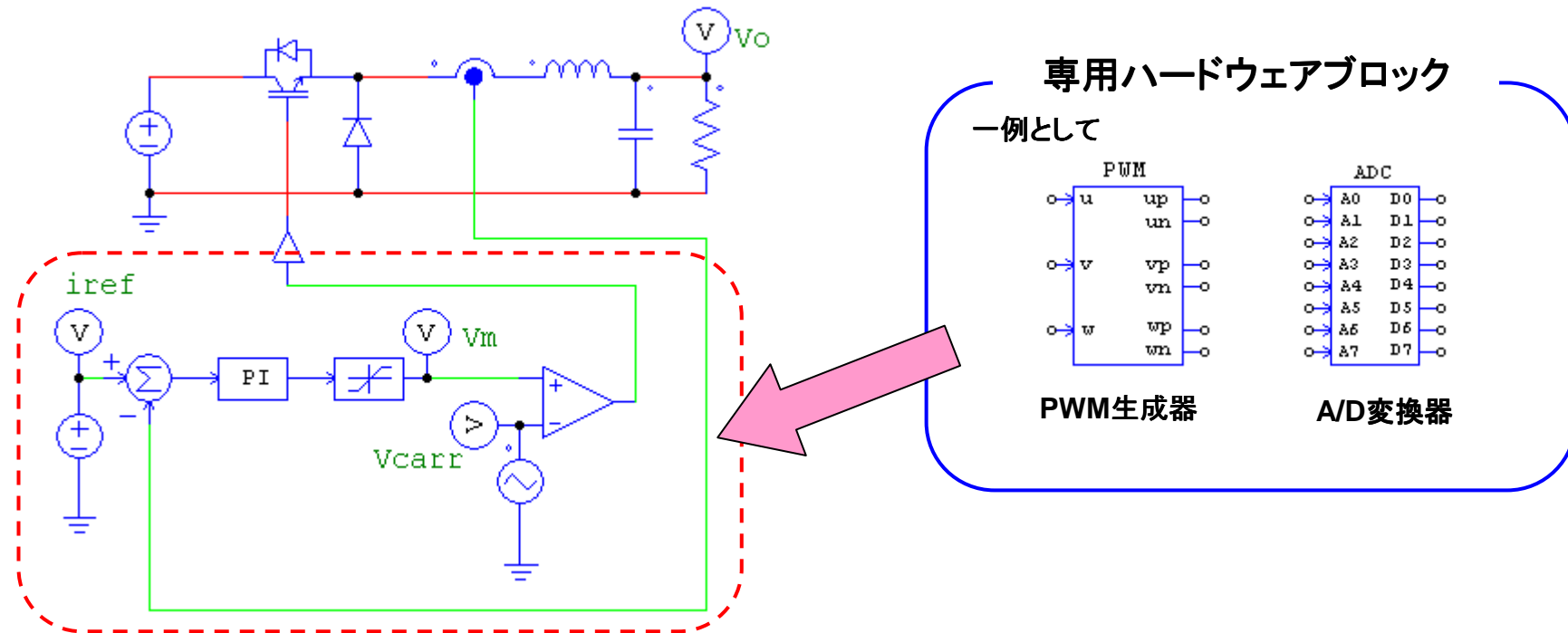
必要構成



SimCoder Module

ーメリットー

- PSIMの制御回路に専用ハードウェアブロックを組み込むだけで簡単にCコードが生成できます



専用ハードウェアブロックを組み込んでも
通常通りのシミュレーションを行なうことができます

SimCoder Module —対象のハードウェア—

以下のハードウェアを対象にCコードの生成を行なうことができます

Mywayプラス製
デジタルコントロールシステム
PE-Expert3



Mywayプラス製デジタル制御ボード
PE-PRO/F28335



テキサスインスツルメンツ社製 DSC
TMS320F28335



汎用DSP

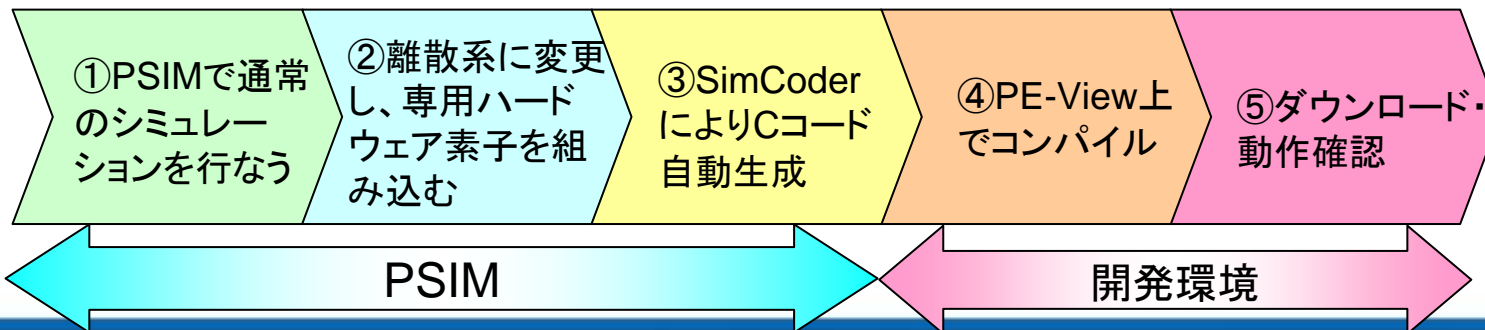
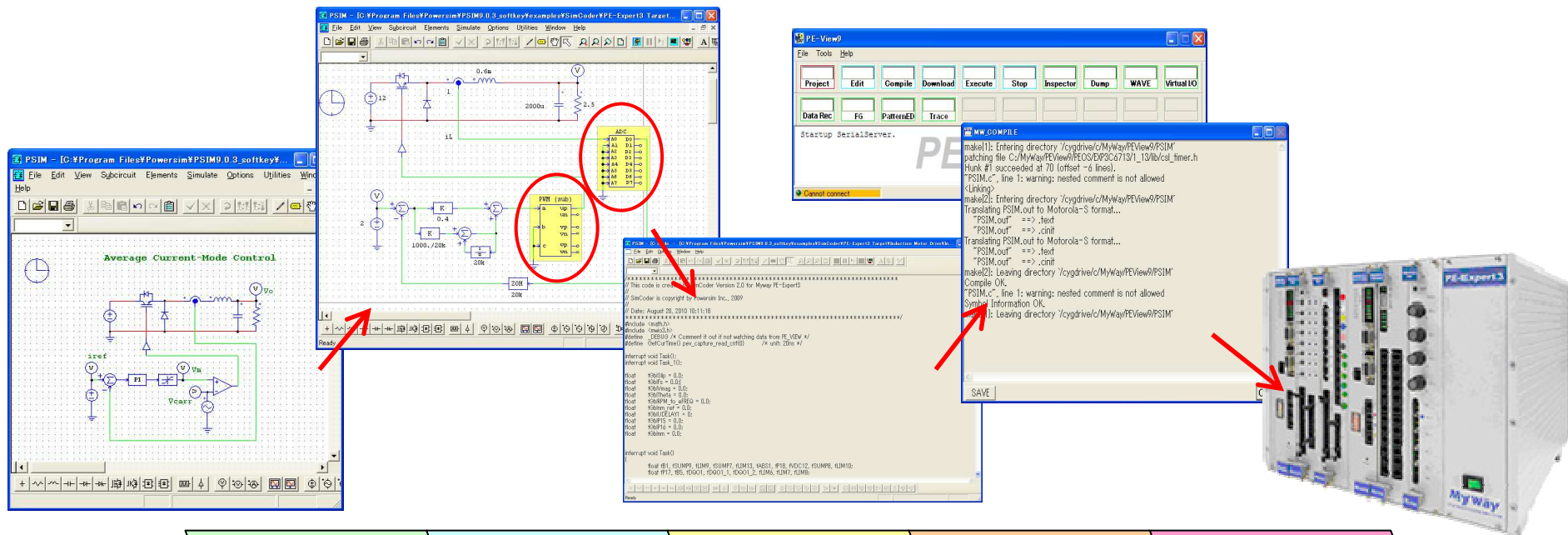


Mywayプラス開発ツールについてはこちら <http://www.myway.co.jp/>

SimCoder Module

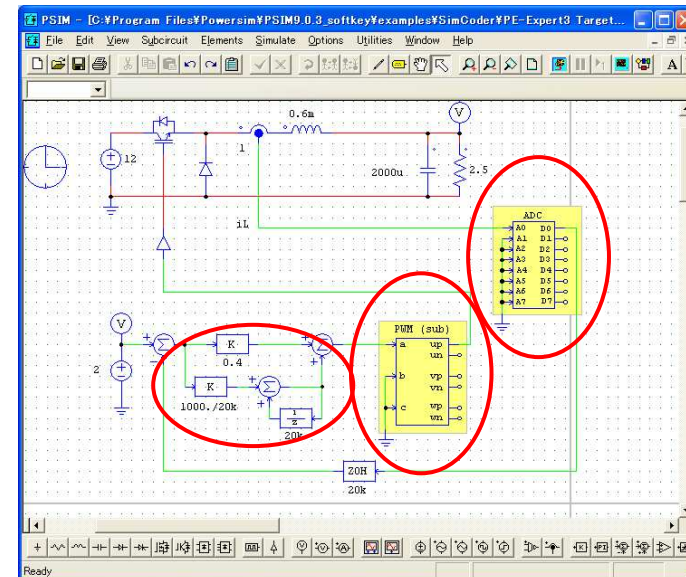
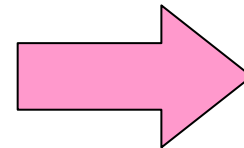
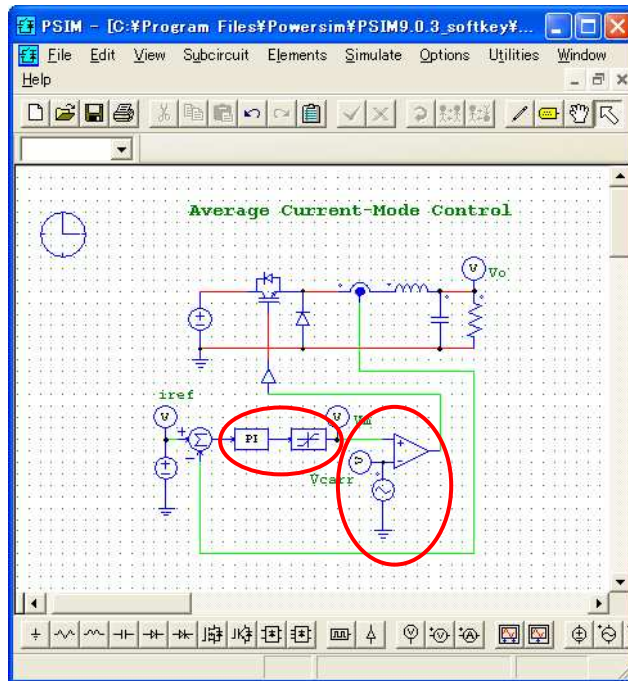
—コード生成・プログラム開発までの流れ—

例として対象ハードウェアをPE-Expertの時の使い方の流れを紹介します



SimCoder Module

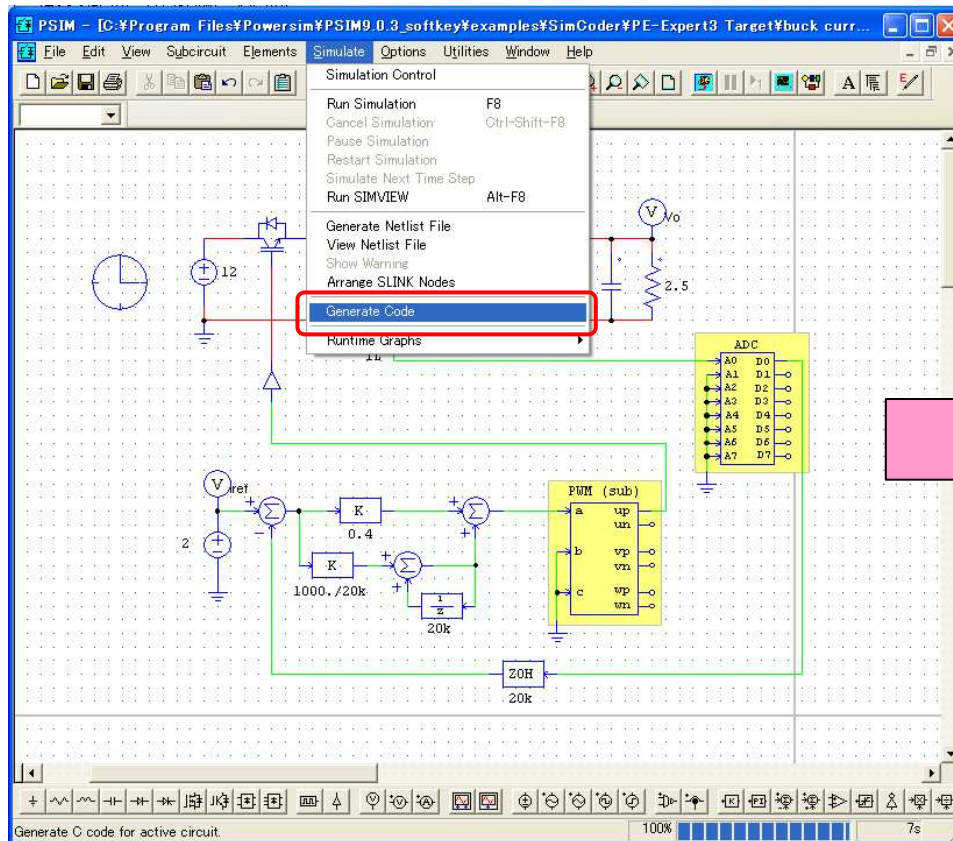
—Cコード生成のための回路の変更点—



- 連続領域(s領域)を離散領域(z領域)に変更
- センサ入力部にA/D変換器を追加
- ゲート出力部のPWM生成器をハードウェア素子に変更

SimCoder Module

—SimCoderによるCコード自動生成—



```
PSIM - [C:\Program Files\Powersim\PSIM9.0.3\softkey\examples\SimCoder\PE-Expert3 Target\buck curr...
File Edit Options Window Help
Simulation Control
Run Simulation F8
Cancel Simulation Ctrl-Shift-F8
Pause Simulation
Restart Simulation
Simulate Next Time Step
Run SIMVIEW Alt-F8
Generate Netlist File
View Netlist File
Show Warning
Arrange SLINK Nodes
Generate Code
Runtime Graphs

*****
// This code is created by SimCoder Version 2.0 for Myway PE-Expert3
// SimCoder is copyright by Powersim Inc., 2009
// Date: August 20, 2010 16:45:17
*****
#include <math.h>
#include <mwio3.h>
#define _DEBUG /* Comment it out if not watching data from PE_VIEW */
#define GetCurTime() pev_capture_read_cnt(0) /* unit: 20ns */

interrupt void Task();

float fGbliref = 0.0;
float fGbluDELAY1 = 0;

interrupt void Task()
{
    float fVDC2, fPEV_ADC1, fPEV_ADC1_1, fPEV_ADC1_2, fPEV_ADC1_3, fZOH3, fSUM1, fP2;
    float fSUMP3, fUDELAY1, fP1, fSUMP1, fPEV_PWM_11Div0, fPEV_PWM_11Div1, fPEV_PWM_11Div5;
    float fPEV_PWM_11Div2, fPEV_PWM_11Div6, fPEV_PWM_11Div3, fPEV_PWM_11Div7;
    pev_ad_start(0, 0);

    fUDELAY1 = fGbluDELAY1;

    fVDC2 = 2;

#ifdef _DEBUG

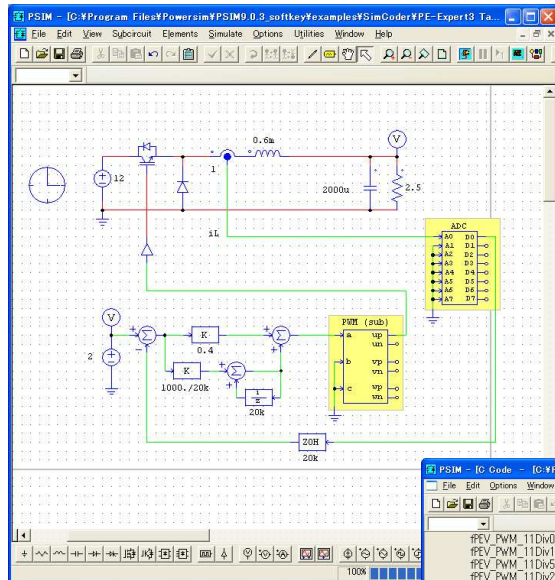
```

“Generate Code”を選択するだけで
瞬時にCコードが自動生成されます

SimCoder Module

—その他の機能・関数化—

サブ回路に演算をまとめることで、その部分を関数としてコードに出力することができます



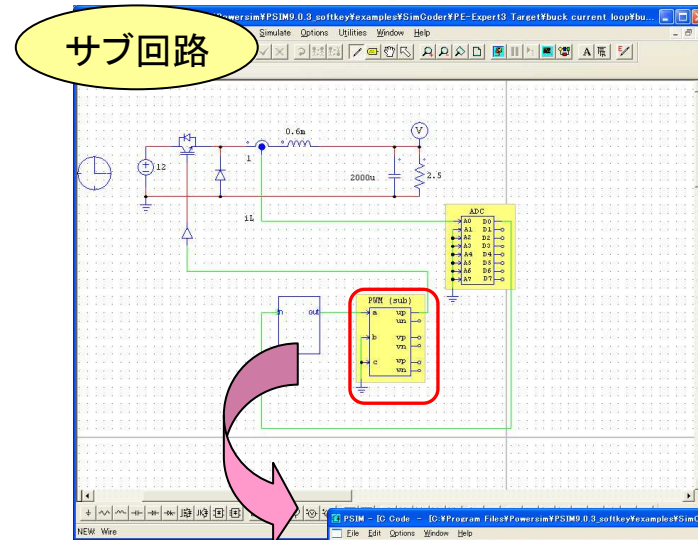
```

fPEV_PWM_11Div0 = (10./2.0+0);
fPEV_PWM_11Div1 = fSUMP1 - fPEV_PWM_11Div0;
fPEV_PWM_11Div5 = fPEV_PWM_11Div1 * (2.0/(10));
fPEV_PWM_11Div2 = 0 - fPEV_PWM_11Div0;
fPEV_PWM_11Div6 = fPEV_PWM_11Div2 * (2.0/(10));
fPEV_PWM_11Div3 = 0 - fPEV_PWM_11Div0;
fPEV_PWM_11Div7 = fPEV_PWM_11Div3 * (2.0/(10));
pev_inverter_set_uw(0, 0, fPEV_PWM_11Div5, fPEV_PWM_11Div6, fPEV_PWM_11Div7);
DEBUG
#endif
watch_data_Bch0;
#endif

void Initialize(void)
{
    pev_init(0);
    pev_capture_set_freerun_period(0, 0xfffffff);
    pev_capture_start_freerun(0);
    pev_ad_set_range(0, 0, 5, 1, 1, 1);
    pev_ad_set_range(0, 1, 1, 1, 1, 1);

    pev_inverter_init(0, 0, 20000, (4e-6)*1E9);
    pev_inverter_set_uw(0, 0, (0-(10.0/2.0+0))*1.0/(10.0/2.0), (0-(10.0/2.0+0))*1.0/(10.0/2.0),
    pev_inverter_start_pwm(0, 0);

    timer0_init(50);
    timer0_init_vector(Task);
    timer0_enable_int();
    timer0_start();
}
    
```



サブ回路の追記

```

#endif
void tasktest(float fln0, float *fOut0)
{
    float ftest_VDC2, ftest_ZOH3, ftest_SUM1, ftest_P1, ftest_P2, ftest_SUMP3, ftest_UDELAY1;
    ftest_UDELAY1 = ftest_UDELAY1;

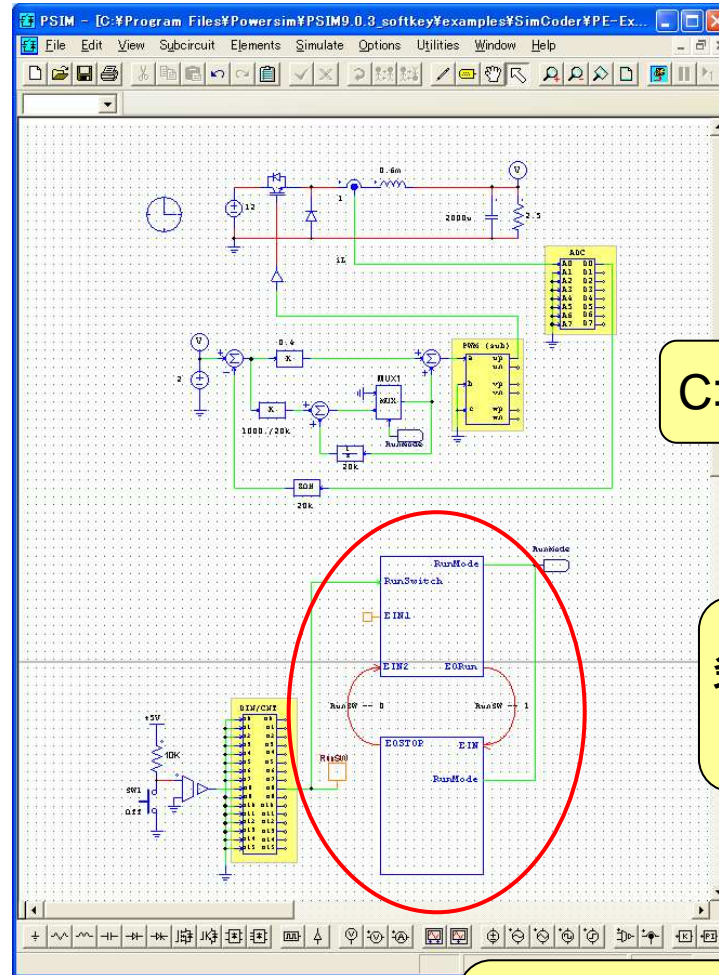
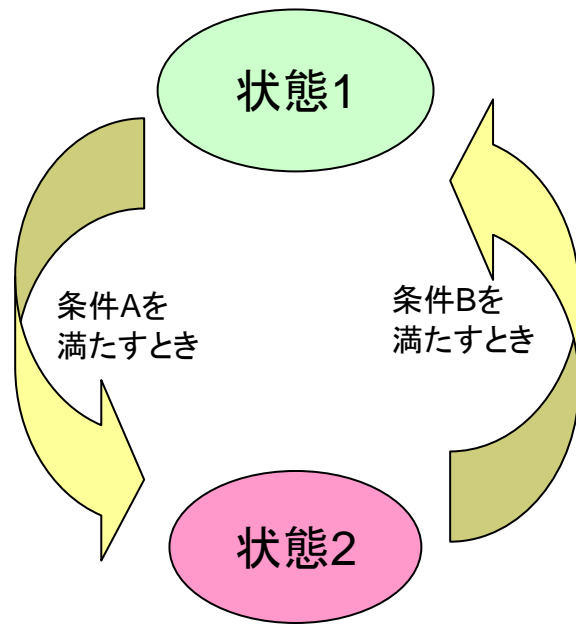
    ftest_VDC2 = 2;
    ftest_ZOH3 = fln0;
    ftest_SUM1 = ftest_VDC2 - ftest_ZOH3;
    ftest_P1 = ftest_SUM1 * 0.4;
    ftest_P2 = ftest_SUM1 * (1000./20000);
    ftest_SUMP3 = ftest_P2 + ftest_UDELAY1;
    *fOut0 = ftest_P1 + ftest_SUMP3;
    DEBUG
    ftest_infer = ftest_VDC2;
}
#endif
fctest_UDELAY1 = ftest_SUMP3;

void Initialize(void)
{
    pev_init(0);
    pev_capture_set_freerun_period(0, 0xfffffff);
    pev_capture_start_freerun(0);
    pev_ad_set_range(0, 0, 5, 1, 1, 1);
    pev_ad_set_range(0, 1, 1, 1, 1, 1);

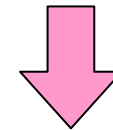
    pev_inverter_init(0, 0, 20000, (4e-6)*1E9);
}
    
```

SimCoder Module —その他の機能・関数化—

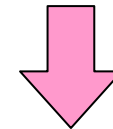
ある状態が条件を満たしたときに
別の状態に遷移します



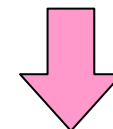
サブ回路で
状態を実現



Cコードで条件を記入



条件が成立すると
状態を移行



システムが状態遷移することを
簡単に示すことができます

SimCoder Module

—使用できる素子—覧—

PE-Expert3	PE-PRO/F28335	TMS320 F28335	汎用DSP
PWM生成	PWM生成	PWM生成	PWM生成
A/D変換器	空間ベクトルPWM生成	A/D変換器	空間ベクトルPWM生成
デジタル入力	A/D変換器	デジタル入出力	A/D変換器
デジタル出力	D/A変換器	エンコーダ	D/A変換器
エンコーダ	デジタル入力	Up/Downカウンタ	デジタル入出力
Up/Downカウンタ	デジタル出力	キャプチャ	エンコーダ
キャプチャ	エンコーダ	トリップゾーン	キャプチャ
	トリップゾーン	DSPクロック	
		ハードウェア基板設定	