

---

ROBIN-M7 対応  
MATLAB/Simulink 実稼動システム  
Pass/ROBIN WPF 版  
型式 PASS-ROBIN-WPF  
Simulink ライブラリ  
取扱説明書

---

2025年 3月 6日

Ver.1.0.0

## 目 次

<b>1. 概要</b> .....	<b>1</b>
1-1 Pass/ROBINについて.....	1
1-2 Pass/ROBINの機能.....	1
1-3 実行環境.....	2
1-4 プロテクトキーについて.....	2
1-5 イーサネット接続について.....	3
1-6 ROBINの電源ON待機時間について.....	3
1-7 実行手順概要.....	4
<b>2. 環境設定</b> .....	<b>5</b>
2-1 システム環境設定.....	5
2-2 MATLAB環境設定.....	6
<b>3. Pass/ROBINライブラリブロック仕様</b> .....	<b>7</b>
3-1 ROBIN_AD.....	8
3-2 ROBIN_DA.....	11
3-3 ROBIN_DI.....	14
3-4 ROBIN_DO.....	17
3-5 ROBIN_PWM.....	20
3-6 ROBIN_CNT.....	31
3-7 ROBIN_SERIAL_CONFIG.....	36
3-8 ROBIN_SERIAL_SEND.....	41
3-9 ROBIN_SERIAL_RECV.....	43
3-10 ROBIN_UDP_CONFIG.....	45
3-11 ROBIN_UDP_SEND.....	47
3-12 ROBIN_UDP_RECV.....	49
3-13 ROBIN_UDP_RECV_ANYIP.....	51
3-14 ROBIN_REALTIME_TRANSFER.....	53
3-15 ROBIN_DATA_SAVE.....	55
3-16 ROBIN_DATA_SAVE_REPEAT.....	58
3-17 ROBIN_EXT_TRIG_START.....	61
3-18 ROBIN_EXT_TRIG_STOP.....	64
<b>4. MATLAB/SimulinkでのARMプログラム作成</b> .....	<b>67</b>
4-1 Simulinkモデルの作成.....	67
4-1-1 ROBINライブラリの起動.....	67
4-1-2 Simulinkモデルの作成、保存.....	68
4-2 Cコード生成、ARMプログラムビルド.....	69
4-2-1 システムターゲットファイルの選択.....	69
4-2-2 テンプレートmakeファイルの選択（通常モード用）.....	70
4-2-3 テンプレートmakeファイルの選択（スタンドアローンモード用）.....	71
4-2-4 Simulinkモデル設定.....	72
4-2-5 実行ハードウェア設定.....	73
4-2-6 インターフェイス設定.....	74
4-2-7 ARMプログラム作成.....	75

## 1. 概要

本書は、Pass/ROBIN WPF 版（以下、Pass/ROBIN と表記）の Simulink ライブラリの取扱説明書です。  
※なお、画面イメージは、Windows10 で起動した MATLAB R2022b のイメージを使用しております。

### 1-1 Pass/ROBINについて

Pass/ROBIN は、ROBIN-M7 用のソフトウェアです。  
MATLAB/Simulink および Simulink Coder に対応しており、Simulink モデルから Simulink Coder で C コードを生成し、ARM 用コンパイラでコンパイル・リンクして ROBIN-M7 用 ARM プログラムを作成します。  
Pass/ROBINでは、Cコード生成からARM用コンパイラでのコンパイル・リンクまでの一連のビルドをMATLAB上で自動的に実行できます。

I/Oは、Simulinkモデル上で入出力位置に弊社が提供するSimulinkライブラリ内のI/Oデバイスブロックを配置、接続することで使用できます。

作成したARMプログラムは、ROBINとイーサネット接続したホストPCから専用アプリケーションによりダウンロードします。

Pass/ROBINにはROBINを専用アプリケーションによって制御する通常モードとROBINが単独で動作するスタンドアロンモードがあります。  
通常モードでは、専用アプリケーションを通じてプログラムの実行開始/停止、データのモニタリング/保存、パラメータ変更等の操作が可能です。  
スタンドアロンモードではデータ保存のみ可能です。

本書では、SimulinkでROBINを使用するためのライブラリを「ROBINライブラリ」とします。

### 1-2 Pass/ROBINの機能

- Simulinkモデルに使用できるROBIN IOのデバイスドライバライブラリ。
- Cコード生成、コンパイル、リンク作業の自動実行。
- データのファイル保存機能（トリガによる取得データ）。

以下は通常モードでのみ可能

- 入出力信号、ブロック出力等の時系列信号のグラフ表示機能および数値モニタ機能。
- 時系列信号のリアルタイム保存機能。
- 制御プログラムの実行時間(ARMでの処理時間)のモニタ機能。
- 制御プログラムの実行・停止機能。
- グラフ表示の間引き機能。

### 1-3 実行環境

Pass/ROBIN は、以下の環境に対応します。

#### ■ ROBIN-M7

#### ■ ホストコンピュータ

OS	Windows 10 (64 ビット) または Windows 11 (64 ビット)
CPU	動作クロック 1GHz 以上 (必須) Intel Core i5 よりも上位グレードの CPU を推奨
メモリ	2GB (必須) 8GB (推奨)
イーサネット 空きポート	1000BASE-T USB2.0 ポート
ソフトウェア	MATLAB R2022b 以降 (64 ビット版) MATLAB Simulink MATLAB Coder Simulink Coder GCC GNU Arm Embedded Toolchain10.3-2021-10

### 1-4 プロテクトキーについて

Pass/ROBINアプリケーションの実行には弊社より提供するUSBプロテクトキーが必要です。

Pass/ROBIN用のUSBプロテクトキーはPass/SEAGULL、Pass/iBIS、Pass/C67の各製品向けに提供しているUSBプロテクトキーと同じ外観ですが、キー内の格納情報が異なるためPass/ROBINアプリケーションはこれらの製品のUSBプロテクトキーでは使用できません。

既にこれらの製品をお持ちでPass/ROBINと同じホストPCでの使用をご希望される場合、複数製品に対応するUSBプロテクトキーを提供いたしますので弊社担当営業までご連絡ください。

なお、PassBOX IIはPass/ROBINと同内容のUSBプロテクトキーのため、Pass/ROBIN用USBプロテクトキーで使用いただけます。

### 1-5 イーサネット接続について

Pass/ROBINは、ホストPCとROBINのto HOSTポートまたはWLAN間をイーサネットで接続し、ROBINをホストPCから制御することで動作します。

イーサネット通信は、TCP/IPのソケット通信です。

ROBINのIPアドレスを変更する場合の設定方法については、インストール説明書（DRA0001988-00-S26630）をご参照ください。（出荷時設定値ではto HOSTポートを使用しIPアドレス192.168.0.1となります）

IPアドレスを変更する場合、ホストPCからサブネットマスク255.255.255.0で識別できる固定アドレスとしてください。

なお、ROBINのto DEVICEポートはARMプログラムでのUDP通信用ポートのため、ホストPCでのROBIN制御には使用できません。

### 1-6 ROBINの電源ON待機時間について

ROBINはイーサネット通信制御をファームウェアで行うため、電源ON時にファームウェアが起動するまでプログラムのダウンロード等の通信ができません。フロントパネルのBOOT LED点灯によりファームウェアが起動したことを確認してから通信を実行してください。

## 1-7 実行手順概要

Pass/ROBINを使用する手順の概要を、図1-1に示します。

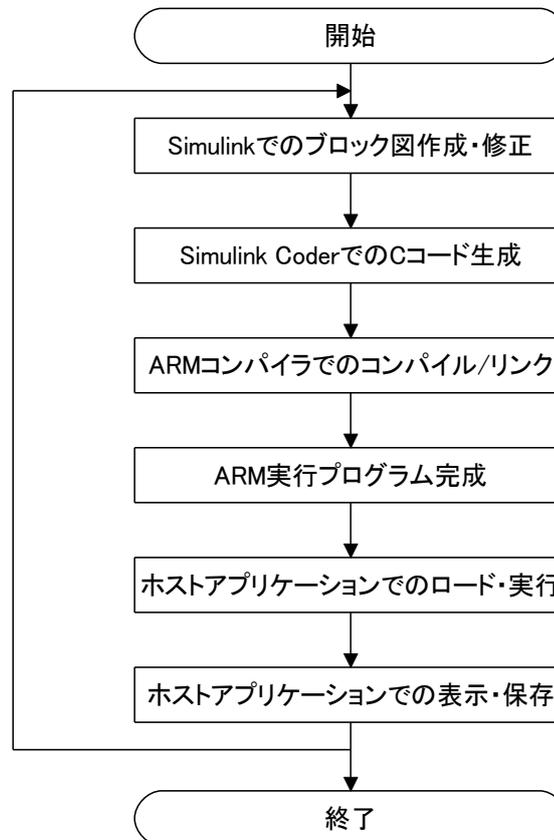


図 1-1 Pass/ROBIN動作フロー

## 2. 環境設定

### 2-1 システム環境設定

Pass/ROBINSimulink ライブラリの使用には、以下の環境変数を設定する必要があります。  
 環境変数の設定はホストアプリケーションのインストール時に行いますので、詳細はインストール説明書  
 (DRA0001988-00-S26630) を参照ください。

表 2-1 Pass/ROBIN 環境変数一覧

変数名	変数説明	デフォルト設定での値
PASS_ROBIN_WPF	Pass/ROBIN インストールフォルダ	C:\WinPass\RobinWpf
ARMGCC_DIR	ARM コンパイラ検索パス	C:\Program Files (x86)\GNU Arm Embedded Toolchain\10.2021.10
ROBINSDKROOT_DIR	SDK インストールフォルダ	C:\WinPass\RobinWpf\SDK
PATH	ARM コンパイラへのパス	C:\Program Files (x86)\GNU Arm Embedded Toolchain\10.2021.10\bin

## 2-2 MATLAB環境設定

MATLABを起動し、ホーム画面の環境一パスの設定でPass/ROBINライブラリにMATLAB検索パスを設定します。パス設定画面で「フォルダーを追加」を選択し、「Pass/ROBINインストールフォルダ¥matlab¥rtw¥c¥pass」を追加します。

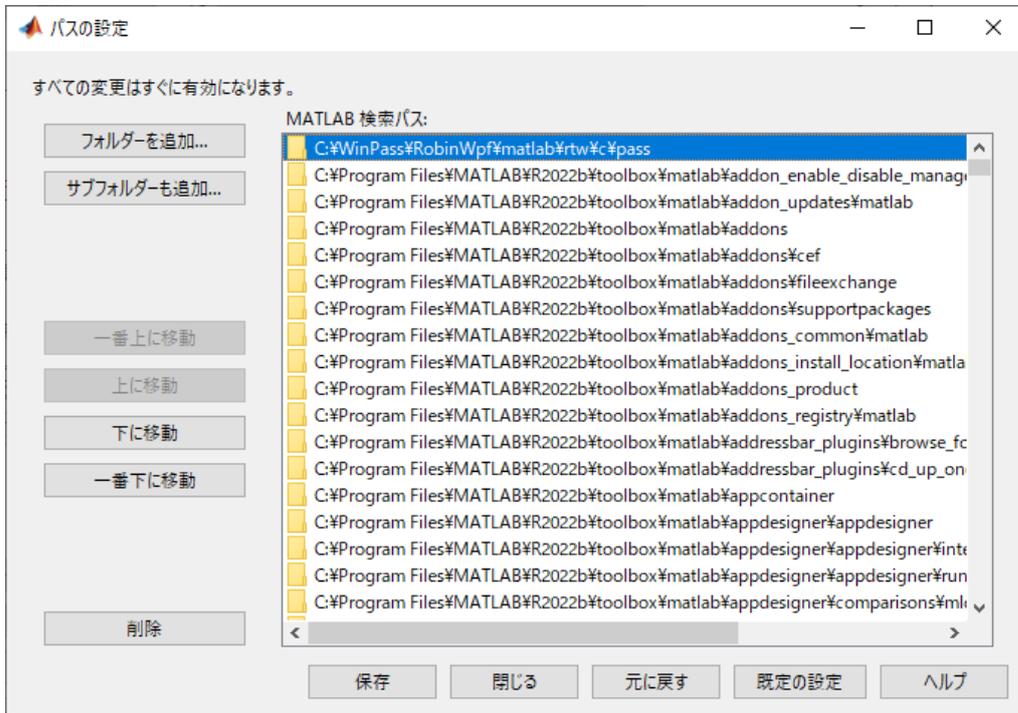


図 2-1 MATLAB 検索パス

なお、弊社他製品（Pass/SEAGULL 等）の検索パスが存在していても Pass/ROBIN の使用に問題はありません。

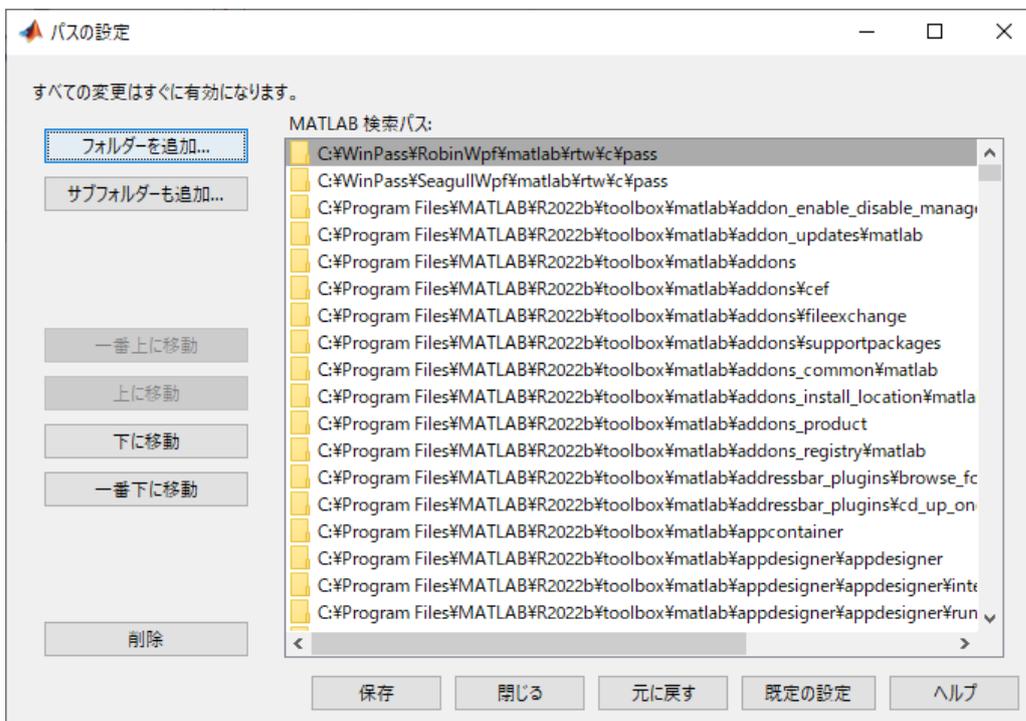


図 2-2 MATLAB 検索パス (Pass/SEAGULL WPF 版との共存例)

設定の保存後、MATLAB コマンドウィンドウに `passRobinWpf` と入力します。  
 パスを正常に登録できている場合、Simulink ライブラリと ROBIN ライブラリ (`robin.slx`) が開きます。

### 3. Pass/ROBINライブラリブロック仕様

Pass/ROBIN では、Simulink から ROBIN を使用するための Simulink ライブラリ”ROBIN”を用意しています。  
 ROBIN ライブラリは、MATLAB コマンドウィンドウで `passRobinWpf` コマンドを入力した際に表示します。

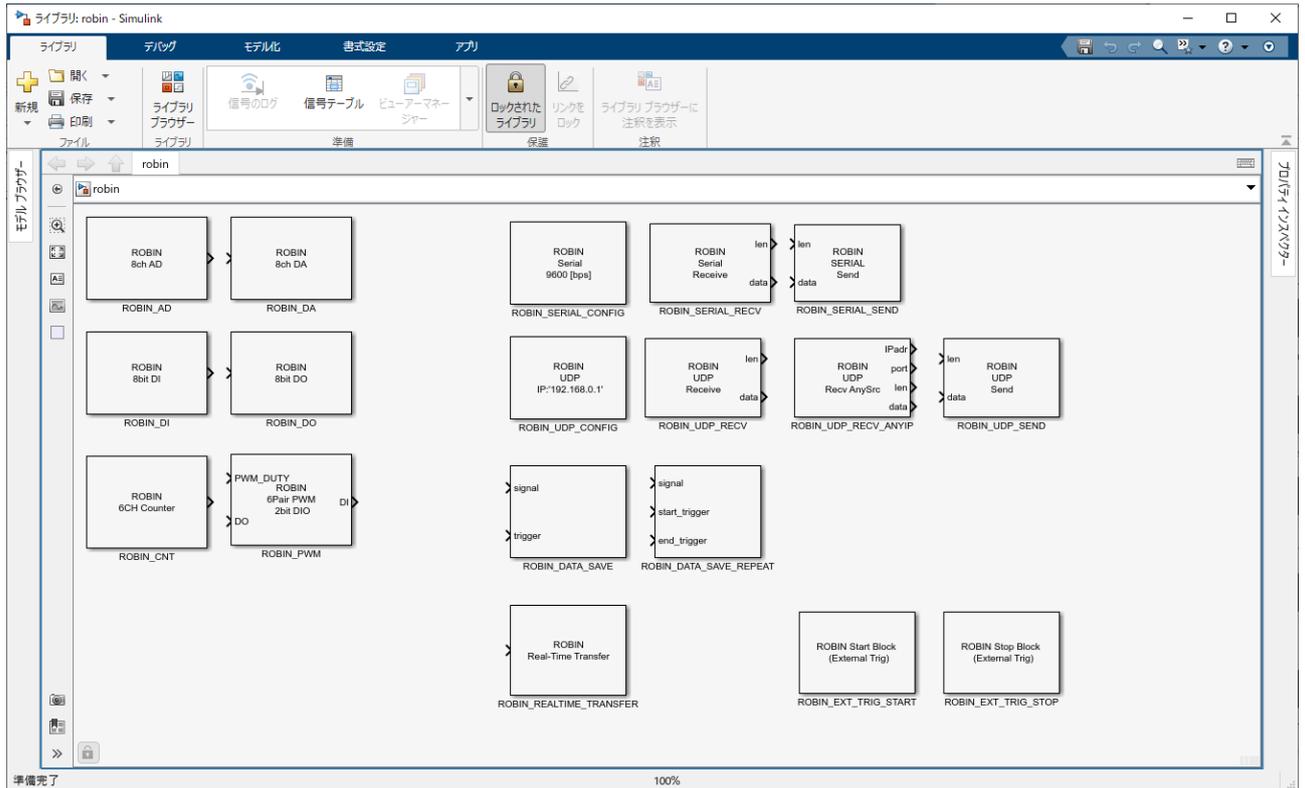
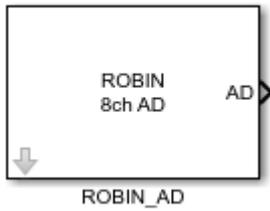


図 3-1 ROBIN ライブラリ

## 3-1 ROBIN\_AD


**【 機能 】**

ROBIN から AD データを取り込みます。

**【 入力端子 】**

なし

**【 出力端子 】**

AD : 浮動小数点数型 8次元ベクトル

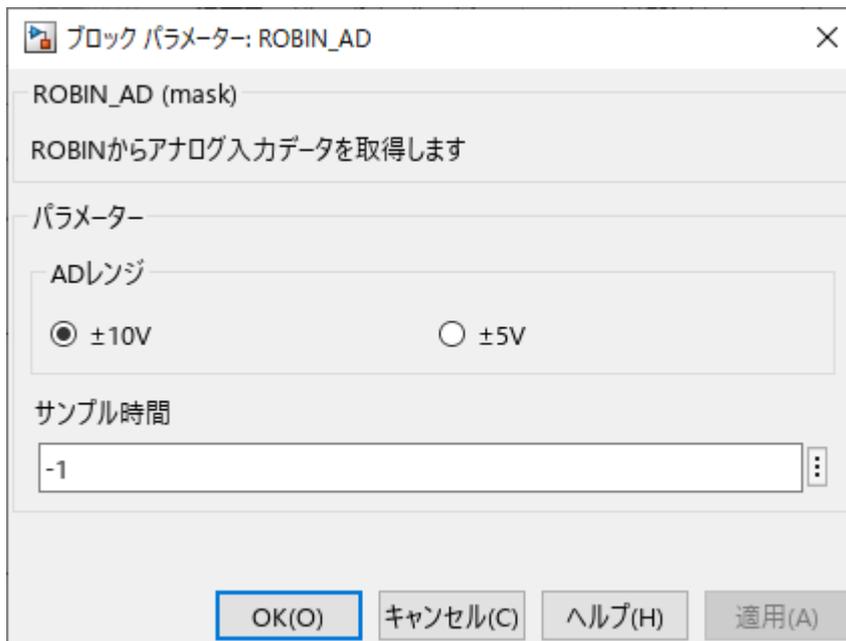
**【 パラメータ 】**


図 3-2 ROBIN\_AD パラメータ

AD レンジ : AD レンジ設定  
 サンプル時間 : ブロックのサンプリング時間

**【 動作説明 】**

サンプル時間毎に ROBIN の ANALOG I/O コネクタへの入力電圧を「AD」端子から浮動小数点数として出力します。

入力電圧は AD レンジで設定したレンジの範囲内としてください。

本ブロックは AD 変換トリガと同一サンプリング周期で AD データを出力します。

サンプル時間の値は、Simulink モデル全体のサンプリング時間の整数倍または-1としてください。

-1 を設定した場合は、Simulink モデル全体のサンプリング時間を継承します。

**【 使用例 】**

サンプルモデル adda.slx での使用例です。

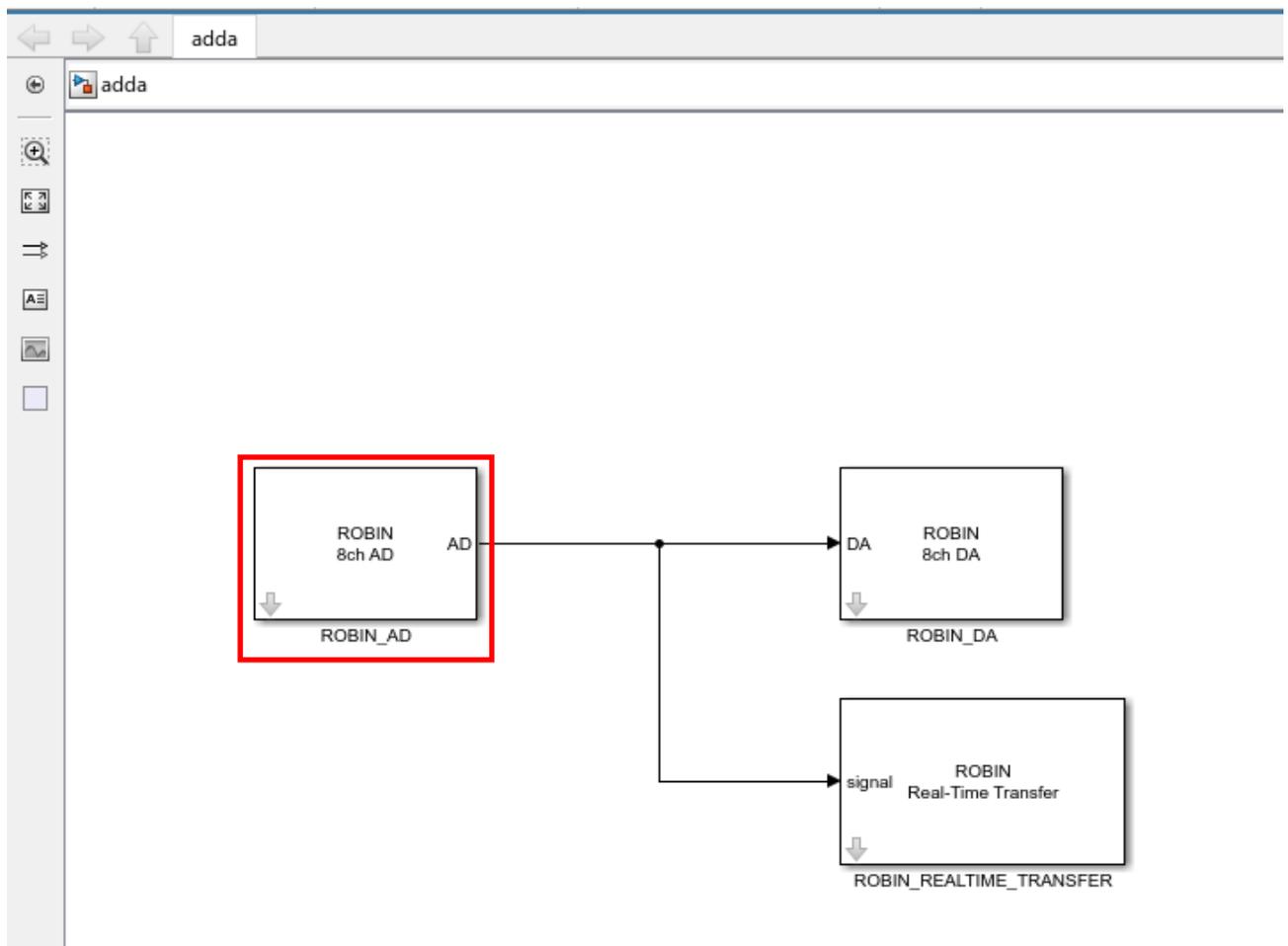


図 3-3 ROBIN\_AD 使用例

**【 端子台展開図 】**

本ブロックに対応する部分の ANALOG I/O の端子台展開図は、以下となります。

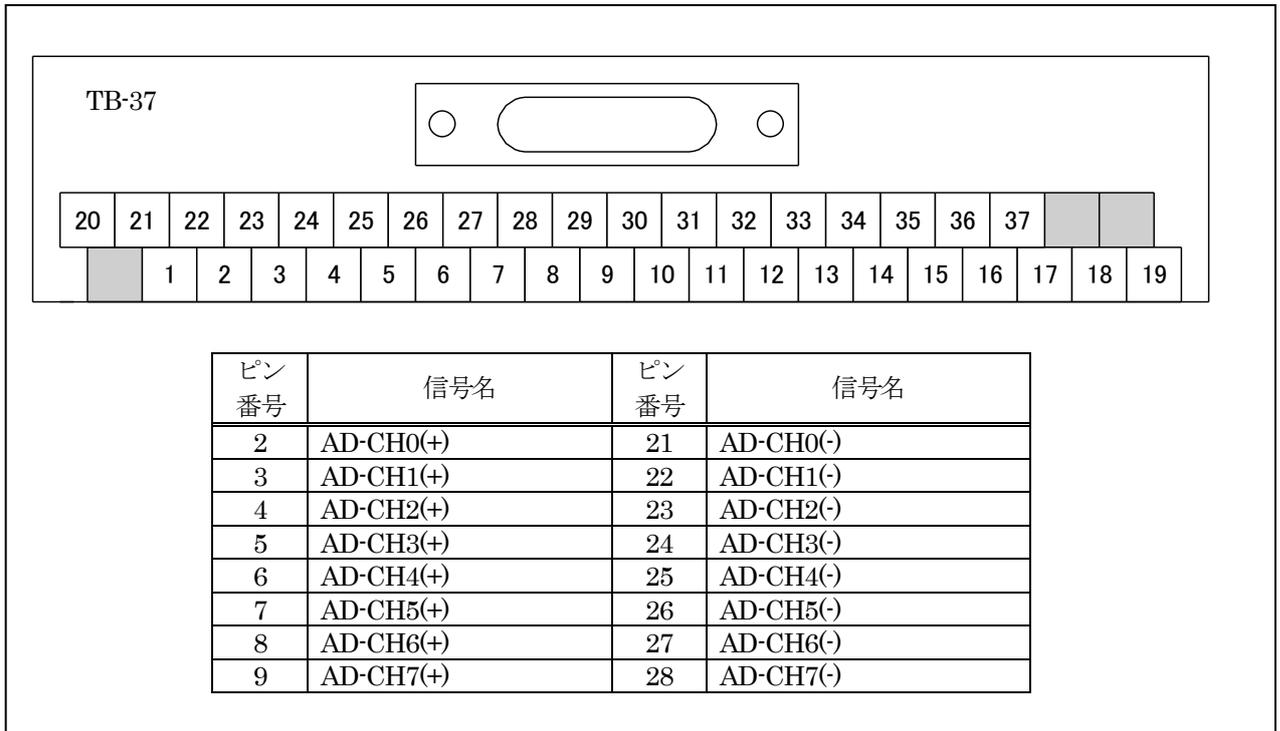
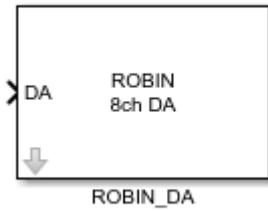


図 3-4 ROBIN\_AD 端子台展開図

## 3-2 ROBIN\_DA


**【 機能 】**

ROBIN から DA データを出力します。

**【 入力端子 】**

DA : 浮動小数点数型 8次元ベクトル

**【 出力端子 】**

なし

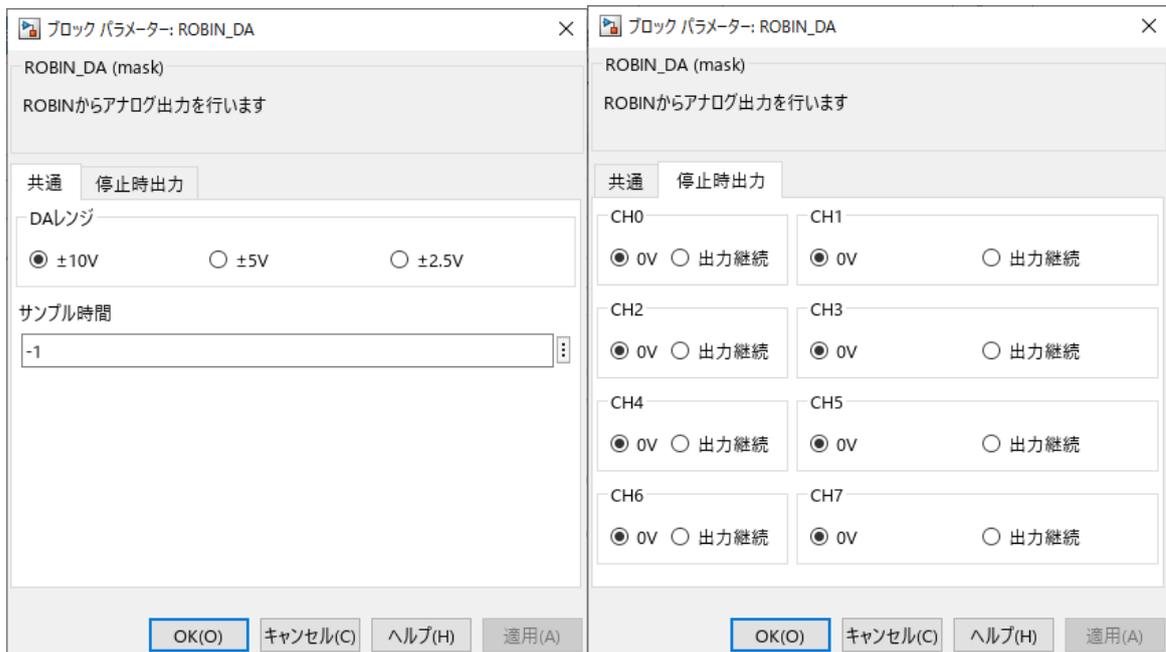
**【 パラメータ 】**


図 3-5 ROBIN\_DA パラメータ (共通)

図 3-6 ROBIN\_DA パラメータ (停止時出力)

- DA レンジ** : DA 出力レンジ  
**サンプル時間** : ブロックのサンプリング時間  
**停止時出力** : 各チャンネルの ARM プログラム停止時出力  
     **0V** : 停止時に DA 出力電圧を 0V に変更します。  
     **出力継続** : 停止時に DA 出力電圧を変更しません。

**【動作説明】**

「DA」端子への入力値を電圧値としてサンプル時間毎に ROBIN の ANALOG I/O コネクタから出力します。入力値と出力電圧の対応を表 3-1 に示します。

表 3-1 ROBIN\_DAブロック入出力対応表

入力値	出力電圧
10.0 以上	10.0V
-10.0 ~ 10.0	入力値 V
-10.0 以下	-10.0V

サンプル時間の値は、Simulink モデル全体のサンプリング時間の整数倍または-1 としてください。-1 を設定した場合は、本ブロックへの入力信号を出力しているブロックのサンプリング時間を継承します。

停止時出力により、ARM プログラム停止時の出力電圧を設定可能です。

**【使用例】**

サンプルモデル `adda.slx` での使用例です。

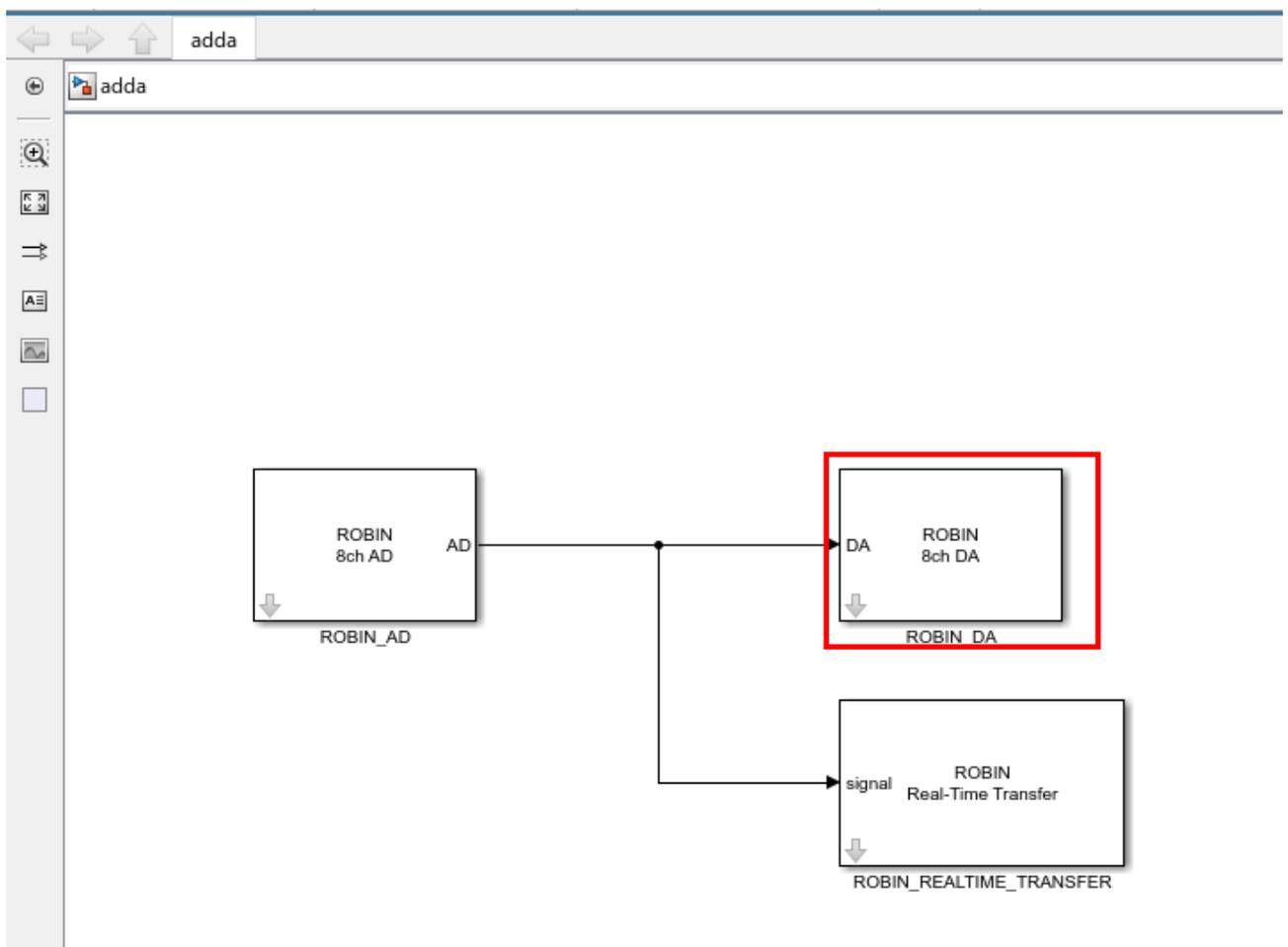


図 3-7 ROBIN\_DA 使用例

## 【 端子台展開図 】

本ブロックに対応する部分の ANALOG I/O の端子台展開図は、以下となります。

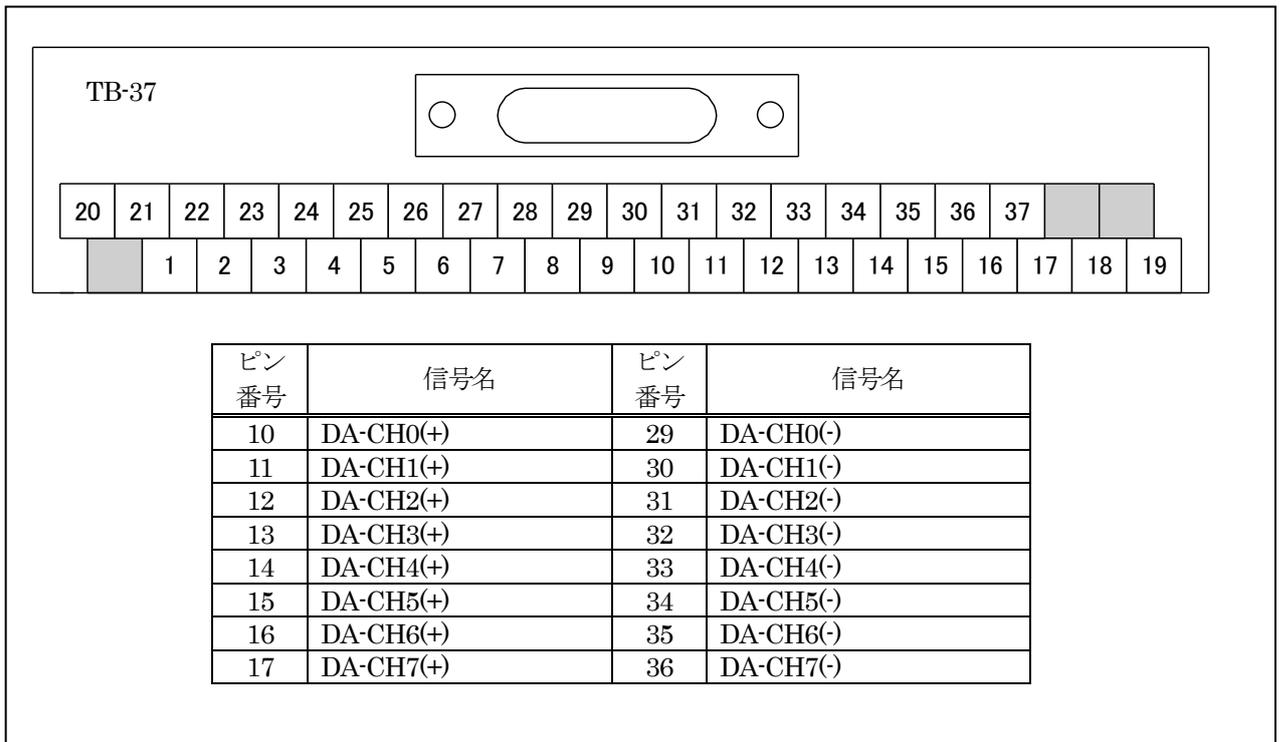
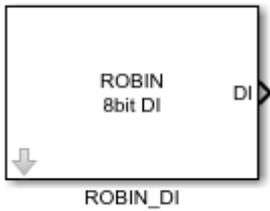


図 3-8 ROBIN\_DA 端子台展開図

## 3-3 ROBIN\_DI


**【 機能 】**

ROBIN の DIGITAL I/O をデジタル入出力モードに設定し、DI データを取り込みます。

**【 入力端子 】**

なし

**【 出力端子 】**

DI : 浮動小数点数型 8次元ベクトル

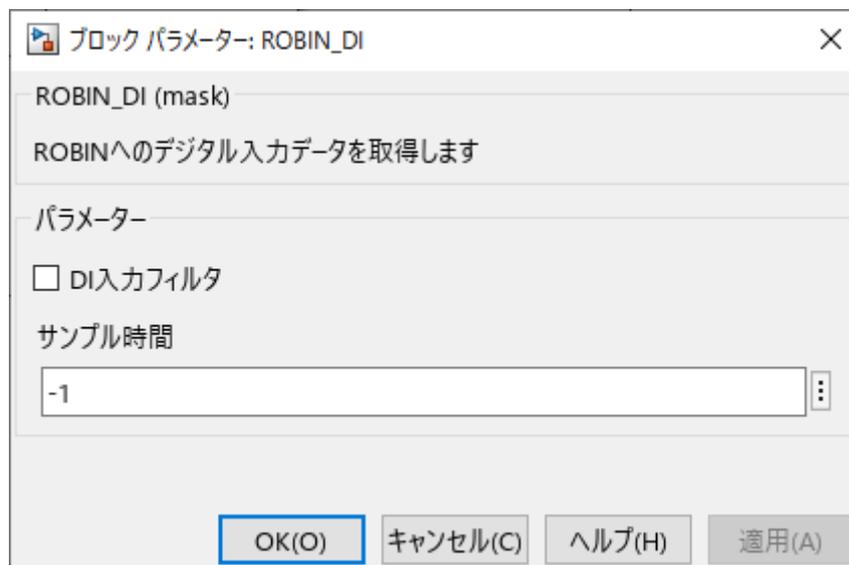
**【 パラメータ 】**


図 3-9 ROBIN\_DI パラメータ

DI フィルタ : DI 入力フィルタ有効/無効  
 サンプル時間 : ブロックのサンプリング時間

**【 動作説明 】**

サンプル時間毎に ROBIN の DIGITAL I/O コネクタへのデジタル入力値を「DI」端子から出力します。  
 出力値は、デジタル入力が ON のとき 1.0、OFF のとき 0.0 となります。  
 DI フィルタが有効の場合、80nsec 以下の入力をフィルタで除去します。

サンプル時間の値は、Simulink モデル全体のサンプリング時間の整数倍または-1としてください。  
 -1を設定した場合は、Simulink モデル全体のサンプリング時間を継承します。

本ブロックと ROBIN\_PWM ブロックは同時に使用できません

**【 使用例 】**

サンプルモデル `dido.slx` での使用例です。

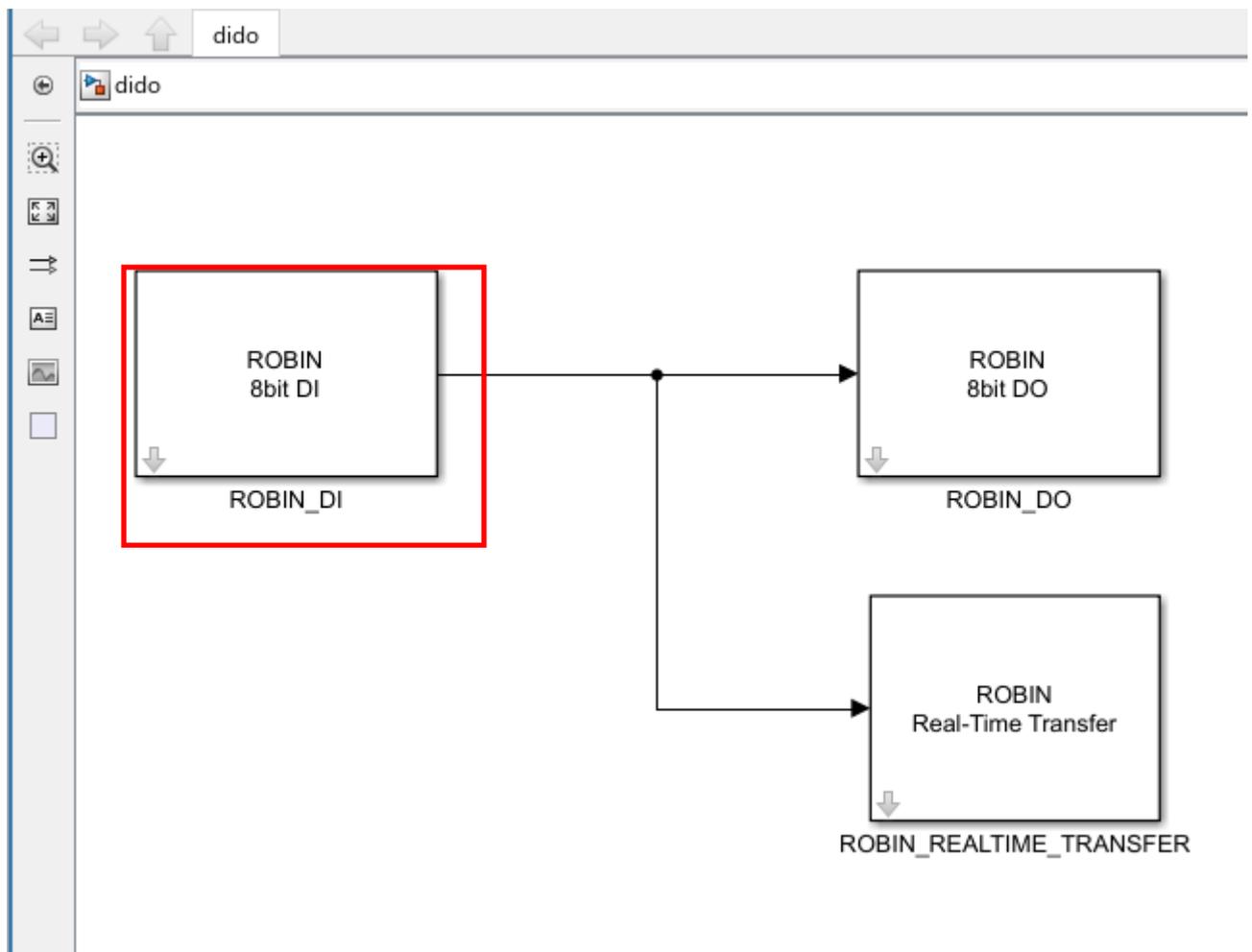


図 3-10 ROBIN\_DI 使用例

**【 端子台展開図 】**

本ブロックに対応する DIGITAL I/O の端子台展開図は、以下となります。

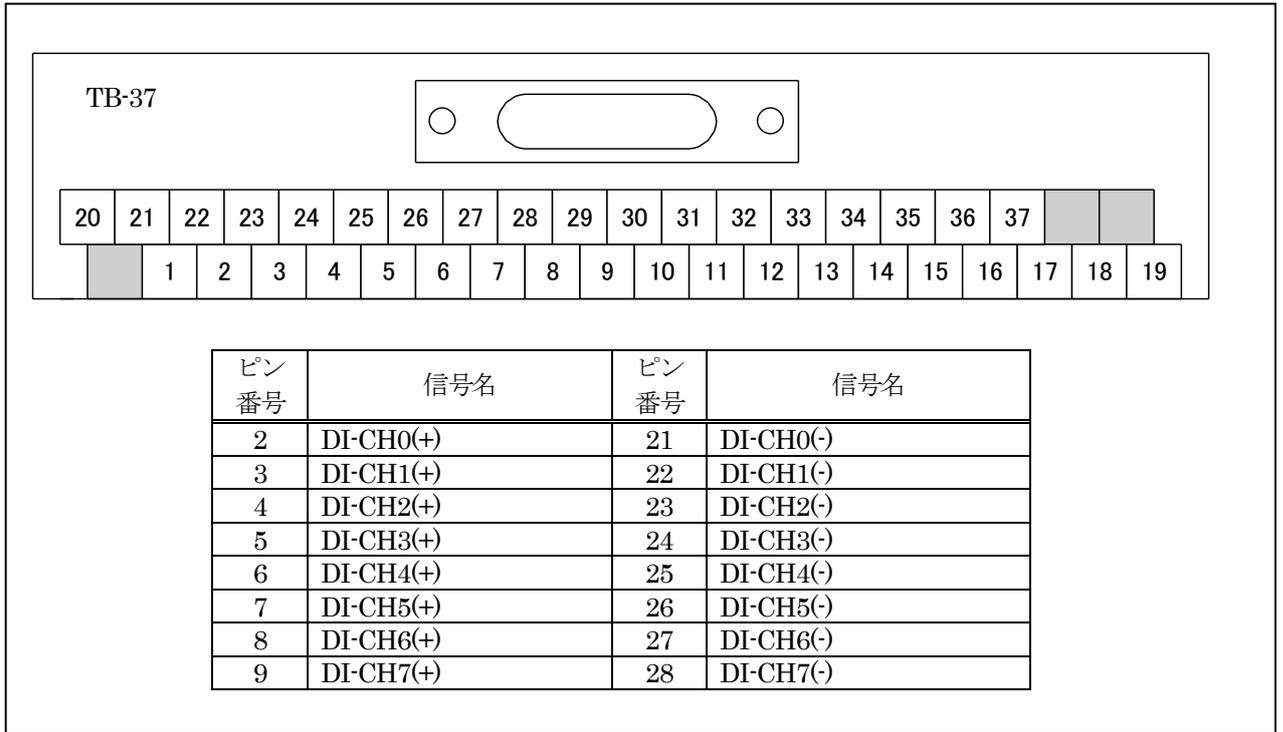
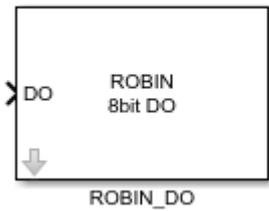


図 3-1 1 ROBIN\_DI 端子台展開図

## 3-4 ROBIN\_DO


**【 機能 】**

ROBIN の DIGITAL I/O をデジタル入出力モードに設定し、DO データを出力します。

**【 入力端子 】**

DO : 浮動小数点数型 8 次元ベクトル

**【 出力端子 】**

なし

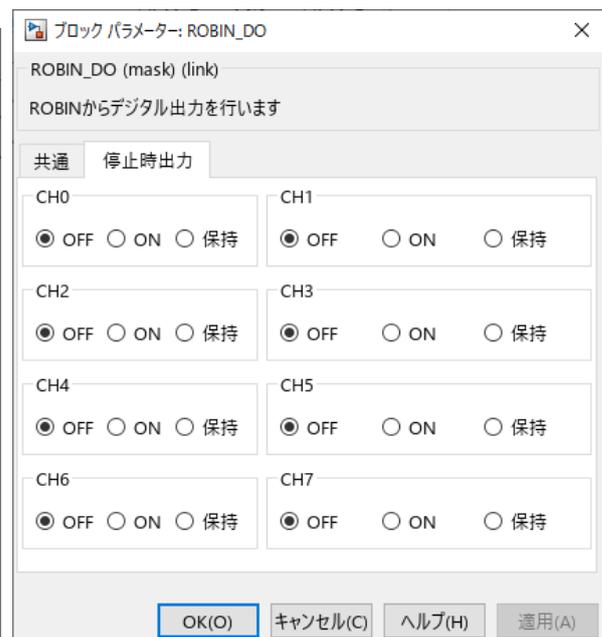
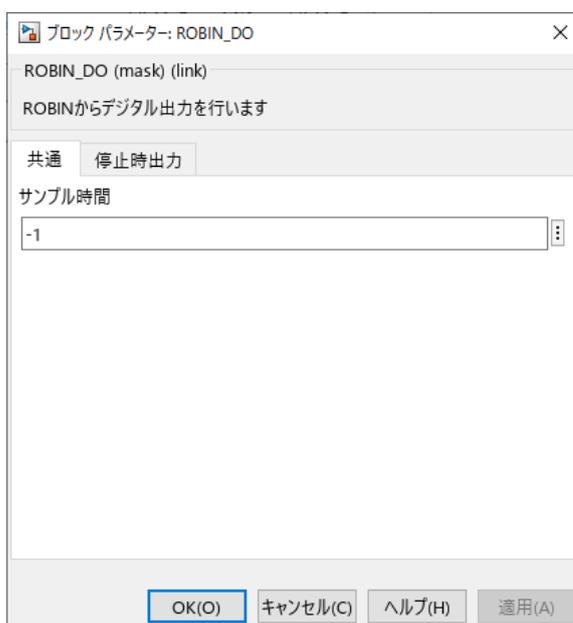
**【 パラメータ 】**


図 3-1 2 ROBIN\_DO パラメータ (共通)    図 3-1 3 ROBIN\_DO パラメータ (停止時出力)

- サンプル時間 : ブロックのサンプリング時間
- 停止時出力 : 各チャンネルの ARM プログラム停止時出力
- OFF* : 停止時に DO 出力を OFF に変更します。
- ON* : 停止時に DO 出力を ON に変更します。
- 保持* : 停止時に DO 出力を変更しません。

**【動作説明】**

「DO」端子への入力を DO データとして取り込み、サンプル時間毎に ROBIN の DIGITAL I/O コネクタから出力します。

入力値と出力の対応を表 3-9 に示します。  
 なお、出力は TTL 出力となります。

表 3-2 ROBIN\_DOブロック入出力対応表

入力値	出力
1.0 未満	OFF (TTL Low)
1.0 以上	ON (TTL High)

サンプル時間の値は、Simulink モデル全体のサンプリング時間の整数倍または-1 としてください。  
 -1 を設定した場合は、本ブロックへの入力信号を出力しているブロックのサンプリング時間を継承します。

停止時出力により、ARM プログラム停止時のデジタル出力値を設定可能です。

本ブロックと ROBIN\_PWM ブロックは同時に使用できません

**【使用例】**

サンプルモデル `dido.slx` での使用例です。

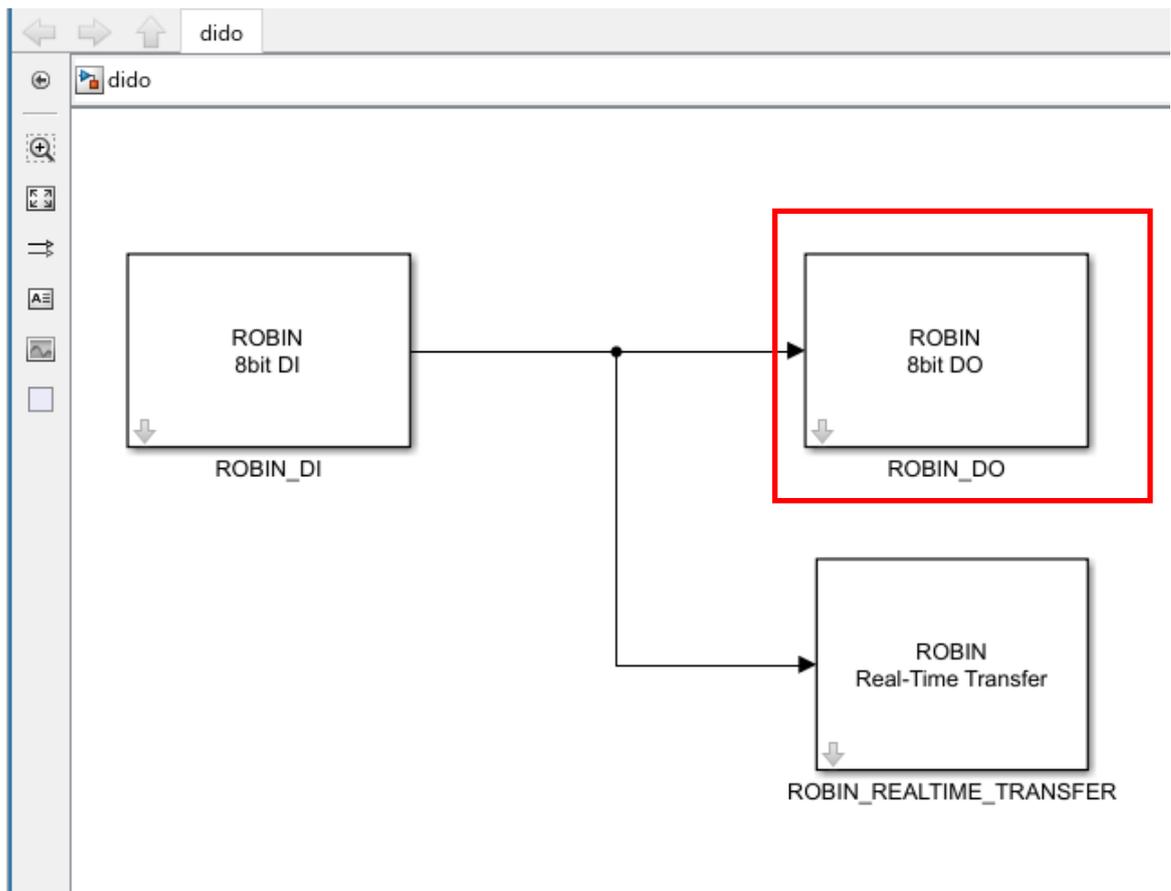


図 3-1 4 ROBIN\_DO 使用例

**【 端子台展開図 】**

本ブロックに対応する DIGITAL I/O の端子台展開図は、以下となります。

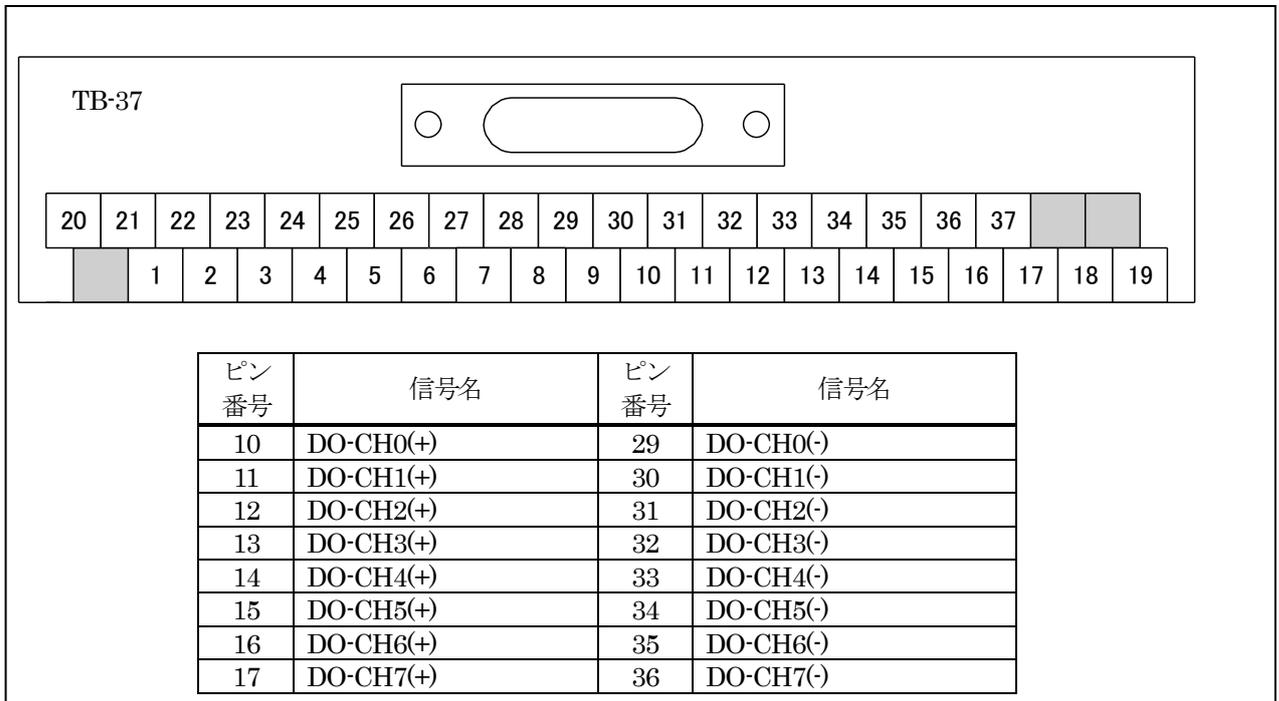
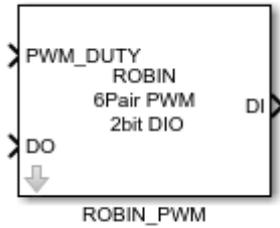


図 3-1 5 ROBIN\_DO 端子台展開図

## 3-5 ROBIN\_PWM


**【 機能 】**

ROBIN の DIGITAL I/O を PWM モードに設定し、入出力を行います。

本ブロックの使用時、ANALOG I/O や COUNTER IN のトリガは PWM CH0/1 のキャリア信号に同期します。PWM CH2/3～CH10/11 も同期する場合は CH0/1 との位相差設定を 0 に設定してください。

IO 同期設定で PWM キャリア波の谷、山のいずれかで他の IO と同期するよう設定可能ですが、PWM 出力波形を非対称（のこぎり波キャリア）とする場合は必ず谷同期に設定してください。

デッドバンド/オーバーラップは、偶数 CH と奇数 CH を同じ極性としている場合のみ使用できます。

本ブロックと ROBIN\_DI ブロック、ROBIN\_DO ブロックは同時に使用できません

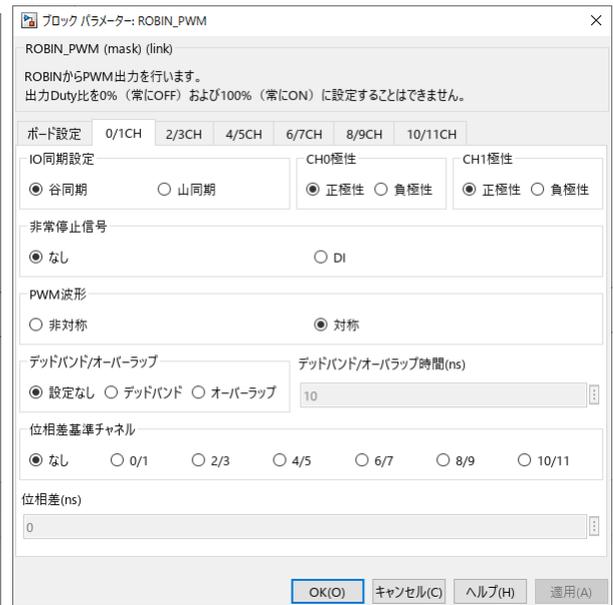
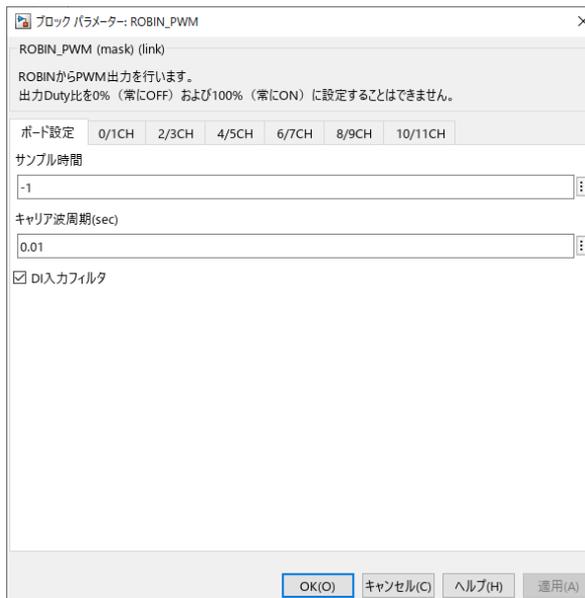
**【 入力端子 】**

PMW\_DUTY : 浮動小数点数型 6 次元ベクトル  
 DO : 浮動小数点数型 2 次元ベクトル

**【 出力端子 】**

DI : 浮動小数点数型 2 次元ベクトル

## 【 パラメータ 】



左 : 図 3-16 ROBIN\_PWM パラメータ (共通) 右 : 図 3-17 ROBIN\_PWM パラメータ (デフォルト時)



サンプル時間 : ブロックのサンプリング時間  
 キャリア波周期 : PWM キャリア波の周期  
 DI フィルタ : DI 入力フィルタ有効/無効

**IO 同期設定** : 他 IO と PWM 波形の同期位置  
 谷同期  
 PWM CH0/1 のキャリア波カウント値 0 で同期します。  
 「基本サンプル時間」とキャリア波周期が異なる場合トリガを間引きます  
 同期タイミングは図 3-19、図 3-20を参照ください。

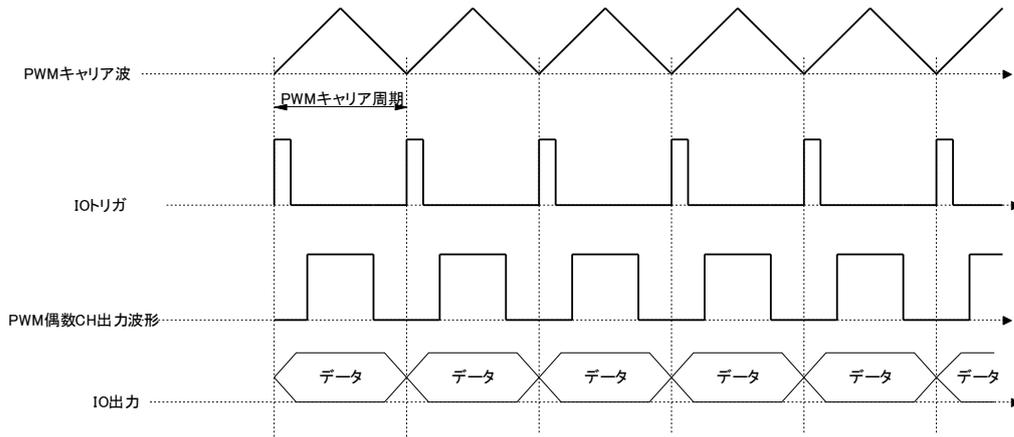


図 3-19 谷同期 同期タイミング(分周比1の場合)

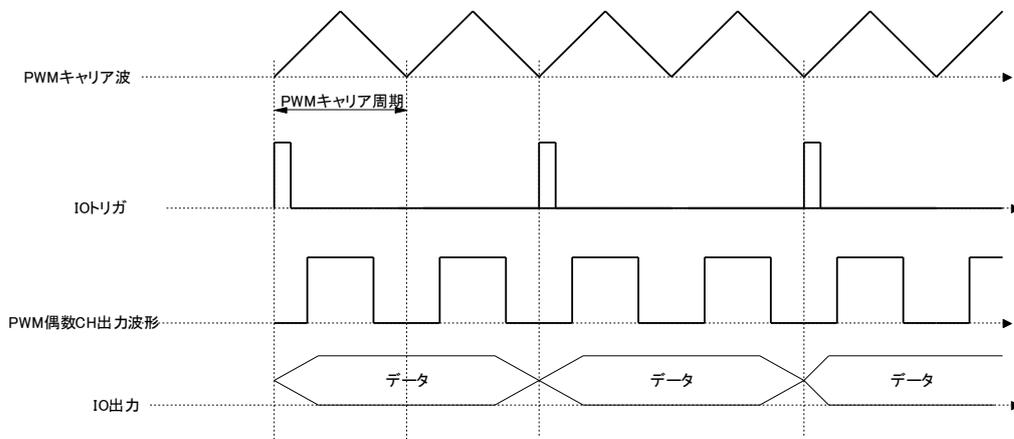


図 3-20 谷同期 同期タイミング(分周比2の場合)

山同期:

PWM CH0/1 のキャリア波カウント値 freq で同期します。

「基本サンプル時間」とキャリア波周期が異なる場合トリガを間引きます  
同期タイミングは図 3-2 1、図 3-2 2を参照ください。

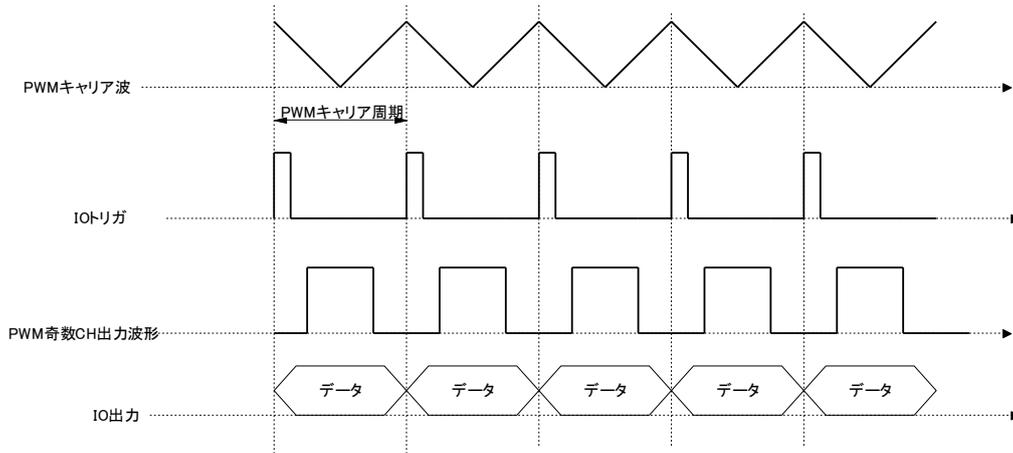


図 3-2 1 山同期 同期タイミング(分周比1の場合)

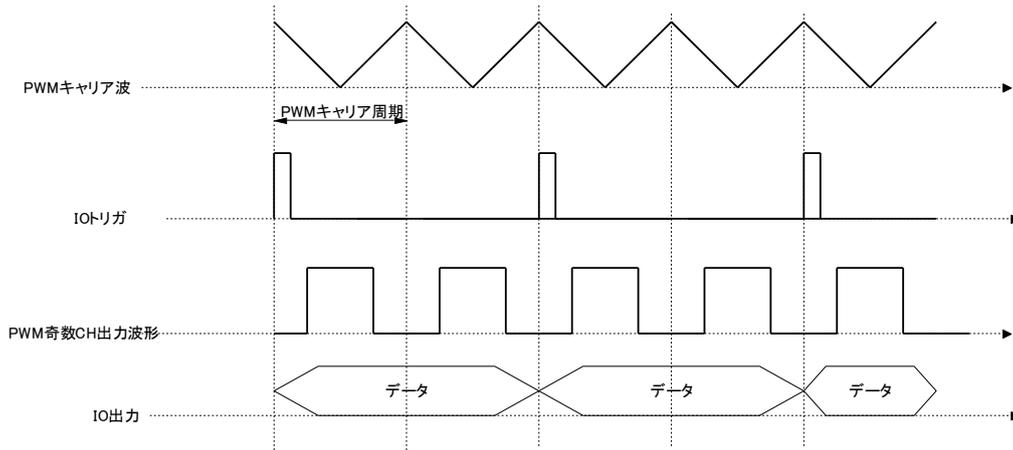


図 3-2 2 山同期 同期タイミング(分周比2の場合)

偶数 CH 極性	:	偶数 CH の PWM 出力信号の極性非常停止信号設定 正極性 : アクティブ High 負極性 : アクティブ Low
奇数 CH 極性	:	奇数 CH の PWM 出力信号の極性非常停止信号設定 正極性 : アクティブ Low 負極性 : アクティブ High
非常停止信号	:	PWM 非常停止信号設定 なし : 非常停止信号を使用しない DI : DI 入力を非常停止信号に使用
非常停止信号検出時動作	:	PWM 非常停止信号発生時動作設定 出力継続 : PWM 出力を継続します (停止しません) 保持 : 停止時の状態を保持します。 Low : 停止時に PWM 出力を Low に変更します High : 停止時に PWM 出力を High に変更します
非常停止 (DI 各チャンネル)	:	非常停止信号の設定 DI CH0 立ち上がり : DI CH0 立ち上がりジ DI CH0 立ち下がり : DI CH0 立ち下がりエッジ DI CH1 立ち上がり : DI CH1 立ち上がりエッジ DI CH1 立ち下がり : DI CH1 立ち下がりエッジ 外部トリガ 立ち上がり : 外部トリガ立ち上がりエッジ 外部トリガ 立ち下がり : 外部トリガ立ち下がりエッジ

## PWM 波形

: PWM 出力波形設定

対称 : 三角波をキャリア波として下図を出力

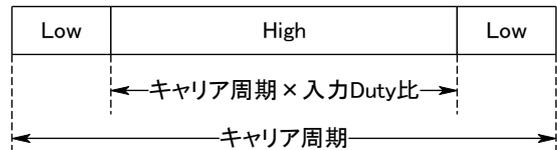


図 3-2 3 偶数チャンネル出力 (三角波キャリア)

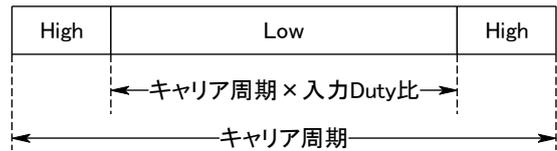


図 3-2 4 奇数チャンネル出力 (三角波キャリア)

非対称 : のこぎり波をキャリア波として下図を出力

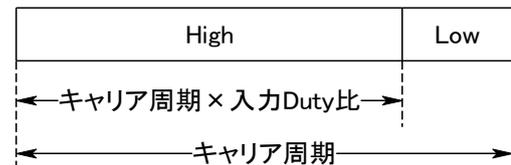


図 3-2 5 偶数チャンネル出力 (のこぎり波キャリア)

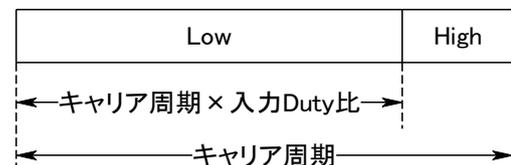


図 3-2 6 奇数チャンネル出力 (のこぎり波キャリア)

デッドバンド/オーバーラップ : デッドバンド/オーバーラップ (下図参照) の設定  
 なし : 設定無効  
 デッドバンド : PWM 対間のデッドバンドを設定  
 オーバーラップ : PWM 対間のオーバーラップを設定

デッドバンド/オーバーラップ時間 : デッドバンド/オーバーラップ時間設定

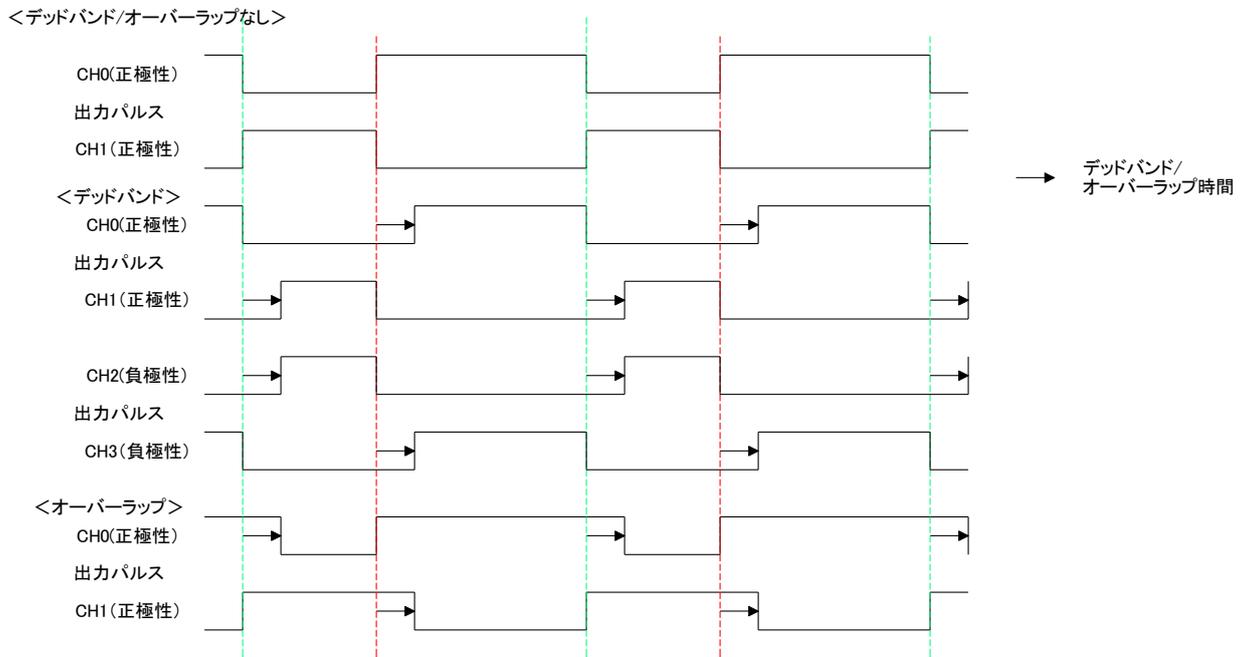


図 3-27 デッドバンド/オーバーラップ動作図

位相差基準チャンネル : 位相差の基準チャンネル設定  
*0/1CH* : 0/1CH を基準とする  
*2/3CH* : 2/3CH を基準とする  
*4/5CH* : 4/5CH を基準とする  
*6/7CH* : 6/7CH を基準とする  
*8/9CH* : 8/9CH を基準とする  
*10/11CH* : 10/11CH を基準とする

位相差 : 位相差時間設定

**【 動作説明 】**

サンプル時間毎に「PWM\_DUTY」端子への入力を PWM デューティ比 (0.0%~100.0%) として取り込み、パラメータによる PWM 設定での出力制御を行います。

なお、キャリア波周期が「4-2-4 Simulink モデル設定」で設定する「基本サンプル周期」の1~1023分周で表現できない場合、分周で表現できるキャリア周期に ARM プログラムで自動的に変更してワーニングを通知します。変更後のキャリア周期はホストアプリ上で確認可能です。

例として、「基本サンプル時間」0.001sec でキャリア周期 0.002sec の場合は 0.001sec、「基本サンプル時間」1.0sec でキャリア周期 0.000001sec の場合は 0.000000977sec、「基本サンプル時間」1/30sec でキャリア周期 0.005sec の場合は 0.0047619sec に変更します。

同期位置で本ブロック以外の IO トリガと本ブロックからの PWM 出力が同期します。

本ブロックでの PWM 制御設定はチャンネル対毎に行います。

なお、同期位置に PWM 山を設定した場合、全チャンネルで PWM 波形を「対称」に設定する必要があります。

「DO」端子への入力を DO から出力します。

本ブロックへの入力値とボードからの出力を表 3-3 に示します。

なお、ROBIN からの出力は TTL 出力となります。

表 3-3 ROBIN\_PWMブロックDO出力対応表

入力値	出力
1.0 未満	OFF (TTL Low)
1.0 以上	ON (TTL High)

DI からの入力値を「DI」端子へ出力します。

ボードへの入力値と本ブロックからの出力を表 3-4 に示します。

DI フィルタが有効の場合、80nsec 以下の入力をフィルタで除去します。

表 3-4 ROBIN\_PWMブロックDI入力対応表

DI 入力	出力値
OFF (TTL Low)	0.0
ON (TTL High)	1.0

サンプル時間の値は、Simulink モデル全体のサンプリング時間の整数倍または1としてください。  
-1を設定した場合は、本ブロックに接続しているブロックのサンプリング時間を継承します。

**【 使用例 】**

サンプルモデル pwmSync.slx での使用例です。

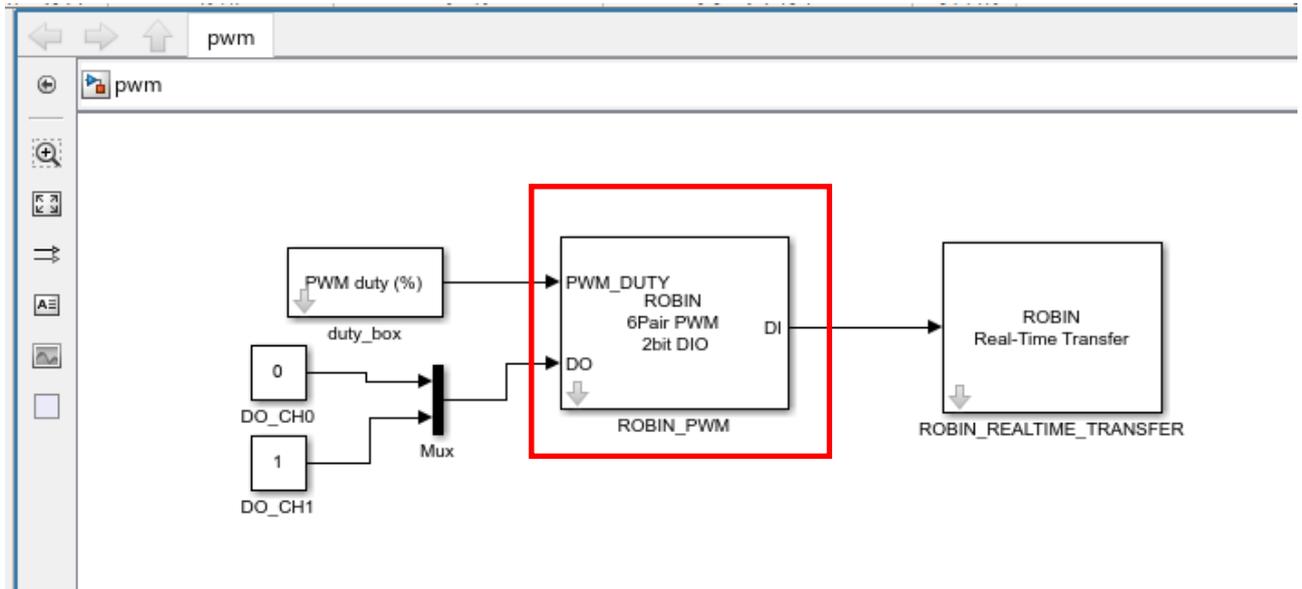


図 3-28 ROBIN\_PWM 使用例

**【 端子台展開図 】**

本ブロックに対応する DIGITAL I/O の端子台展開図は、以下となります。

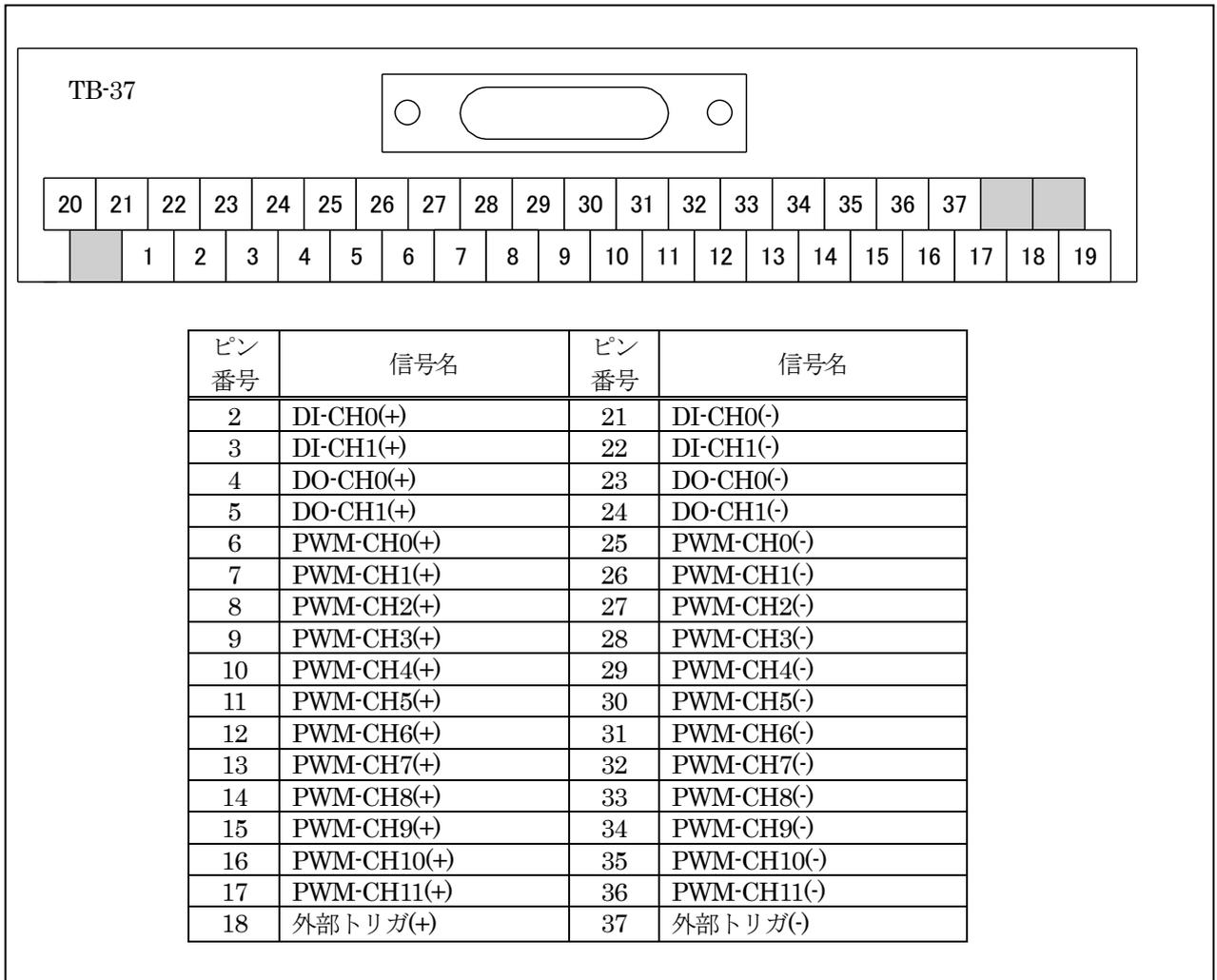
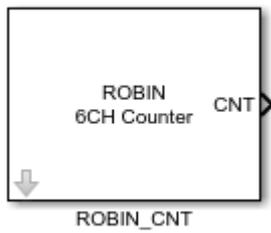


図 3-29 ROBIN\_PWM 端子台展開図

## 3-6 ROBIN\_CNT


**【 機能 】**

ROBIN からパルスカウンタデータを取り込みます。

**【 入力端子 】**

なし

**【 出力端子 】**

CNT : 6次元ベクトル (型は出力形式パラメータで指定)

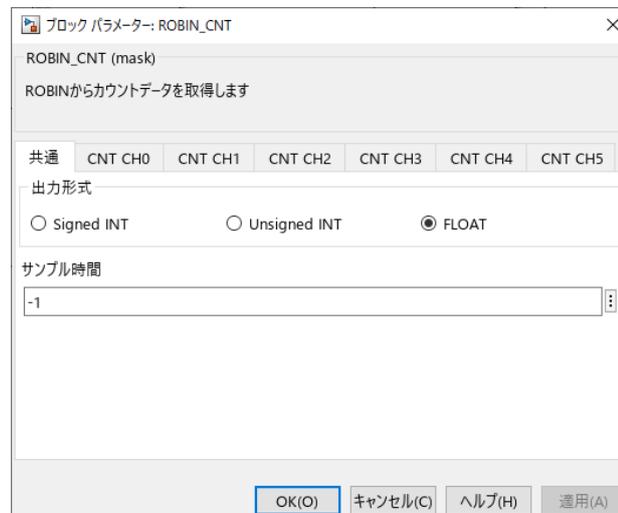
**【 パラメータ 】**


図 3-30 ROBIN\_CNT パラメータ (共通)

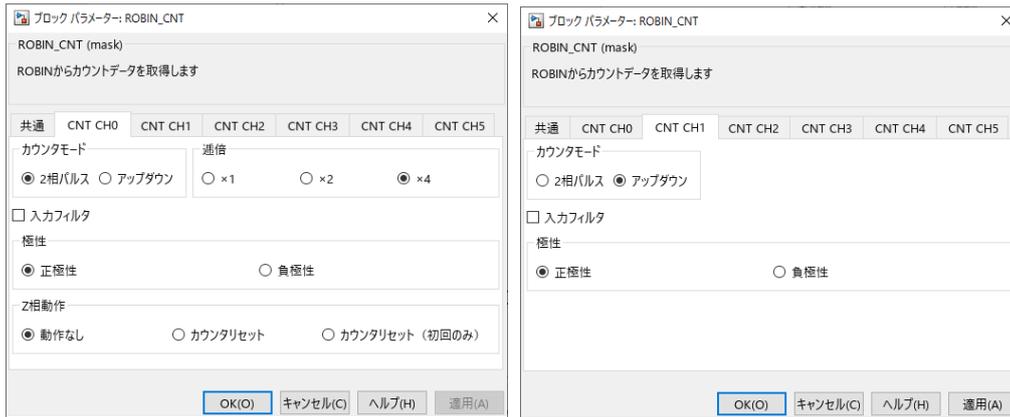


図 3-3 1 ROBIN\_CNT パラメータ (2相パルス)

図 3-3 2 ROBIN\_CNT パラメータ (UDC)

## 共通設定項目

サンプル時間	:	ブロックのサンプリング時間
出力形式	:	カウンタ値出力形式設定
		<i>signed int</i> : 32bit 符号つき整数
		<i>unsigned int</i> : 32bit 符号なし整数
		<i>float</i> : 浮動小数点数

## 各チャンネル設定項目 (モード共通)

カウンタモード	:	カウンタ動作モード設定
		アップダウン : アップダウンカウンタ
		2相パルス : 2相パルスカウンタ
入力フィルタ	:	カウンタ入力フィルタ有効/無効
極性	:	カウンタ入力極性設定
		正極性 : 入力なし時 Low
		負極性 : 入力なし時 High

## 各チャンネル設定項目 (2相パルスカウンタモード)

逓倍	:	逓倍設定
		×1 : 1 逓倍
		×2 : 2 逓倍
		×4 : 4 逓倍
Z相動作	:	Z相信号入力時動作設定
		動作なし : Z相信号無効
		カウンタリセット : Z相信号で常にカウンタリセット
		カウンタリセット (初回) : 初回のZ相信号のみカウンタリセット

**【動作説明】**

サンプル時間毎に ROBIN の COUNTER\_IN に入力されたパルスのカウントデータを「CNT」端子から出力します。

カウンタモードをアップダウンに設定した場合、カウンタデータは UP 入力でカウントアップ、DOWN 入力でカウントダウン、CLR 入力でクリアします。

カウンタモードを 2 相パルスに設定した場合、カウンタデータは A 相入力が B 相入力より進んでいる場合カウントアップ、B 相入力より遅れている場合カウントダウンし、Z 相入力で Z 相動作に対応したクリア動作を行います。

IEEE 754 浮動小数点数の仕様上、絶対値が  $2^{24}$  (16,777,216) を超える整数には単精度浮動小数点数では正しく表現できない数がありますので、±16,777,216 の範囲外となるカウントデータを使用する場合は倍精度で演算を行うテンプレート make ファイルを使用してください。

本ブロックではカウンタデータをサイクリックで扱うため、出力形式が unsigned int の場合はカウント値 0 とカウント値 4,294,967,295 の間、出力形式が signed int および float の場合はカウント値 2,147,483,647 とカウント値 -2,147,483,648 の間で値のオーバーフローが発生します。

サンプル時間の値は、Simulink モデル全体のサンプリング時間の整数倍または 1 としてください。

-1 を設定した場合は、Simulink モデル全体のサンプリング時間を継承します。

なお、本ブロックを使用したモデルのビルド時に「m7\_CntGetData の引数の符号が整合しない」というワーニングが表示されますが、本ブロックでは出力形式の設定によってカウントデータを符号付きで取り扱う必要があるためこのワーニングは無視してください。

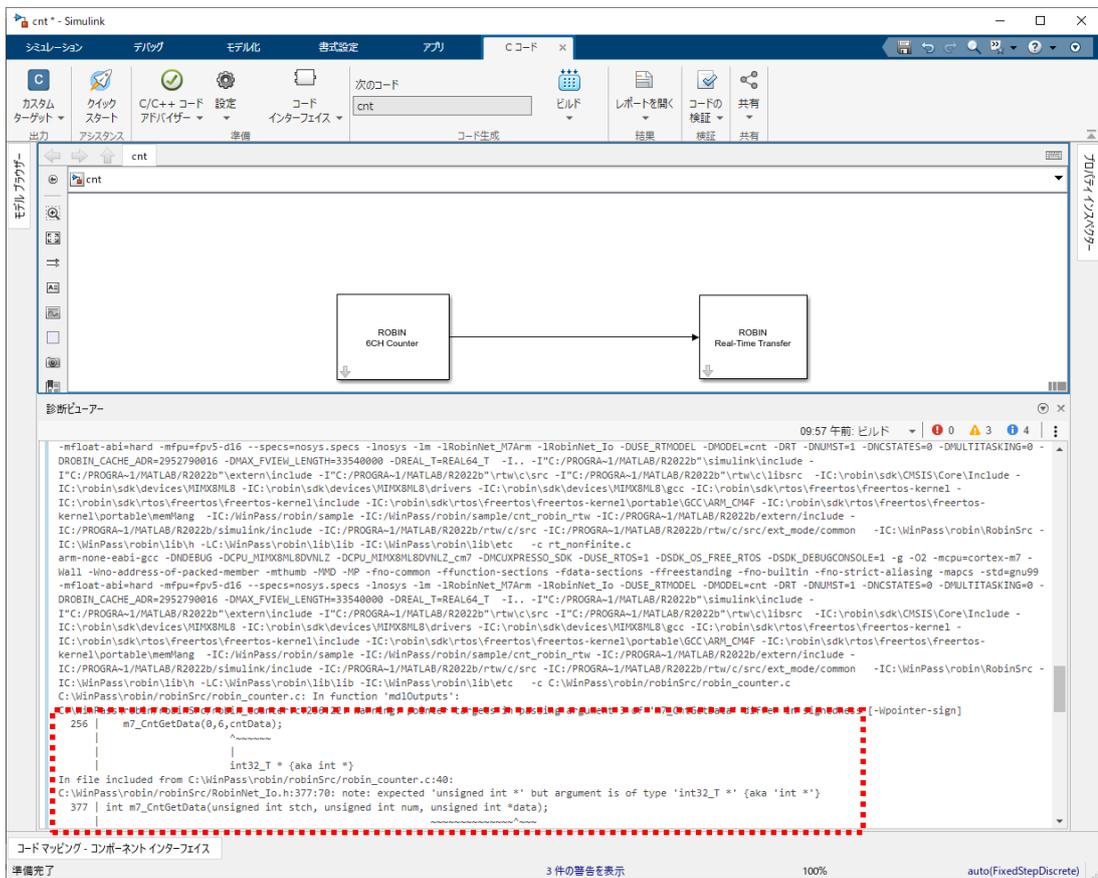


図 3-33 ROBIN\_CNT ワーニング

**【 使用例】**

サンプルモデル cnt.slx での使用例です。

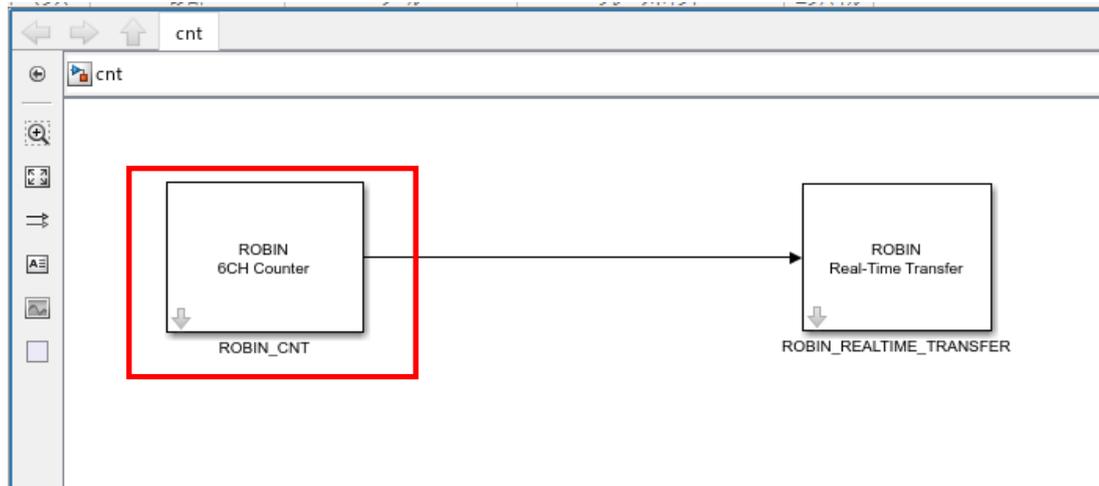


図 3-3 4 ROBIN\_CNT 使用例

**【 端子台展開図 】**

本ブロックに対応する COUNTER IN の端子台展開図は、以下となります。

展開図は 2 相パルスモードの信号で記載しておりますので、アップダウンカウンタモードで使用する場合は PHASE\_A を UP、PHASE\_B を DOWN、PHASE\_Z を CLR に 読み替えてください。

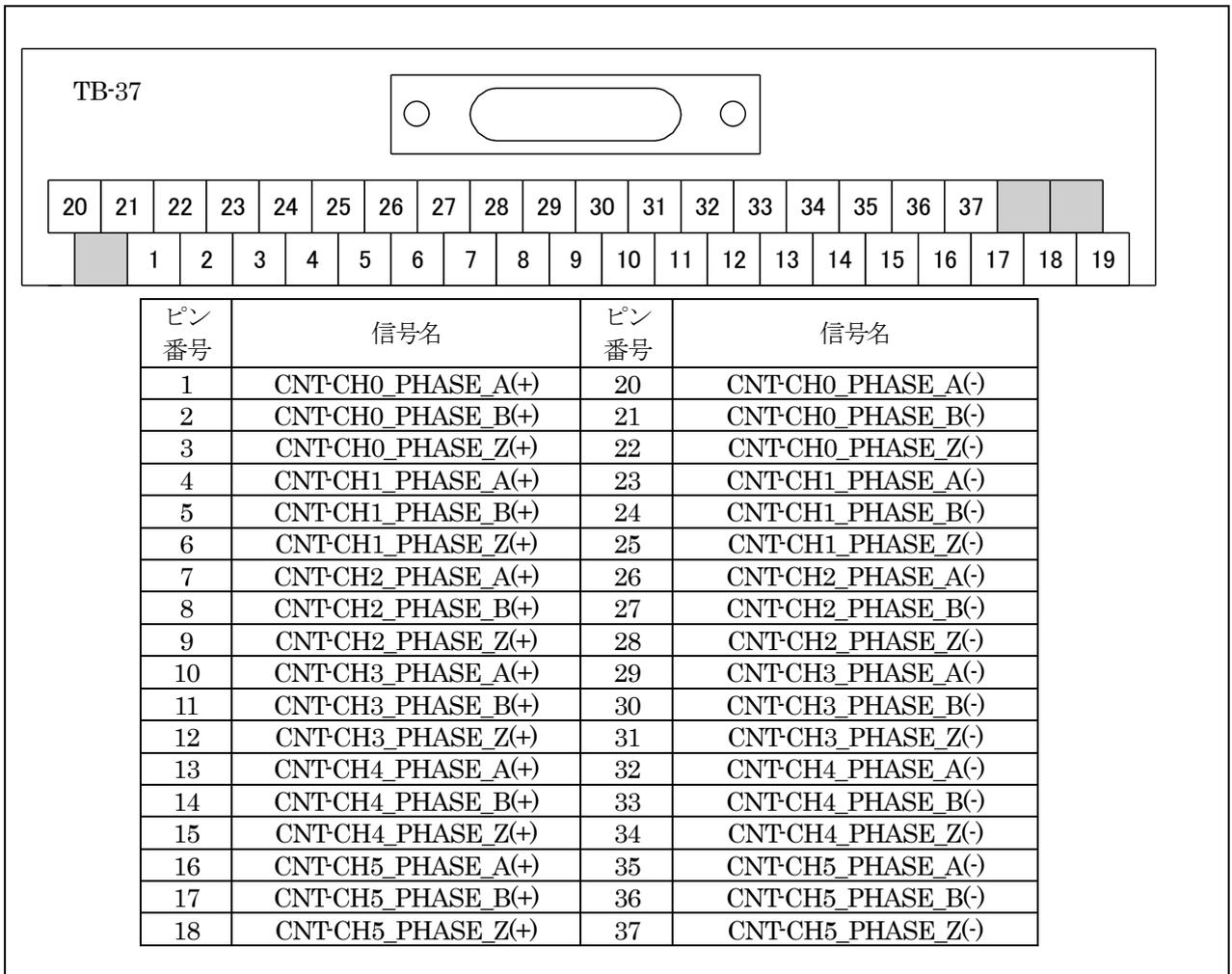
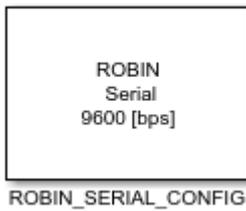


図 3-35 ROBIN\_CNT 端子台展開図

### 3-7 ROBIN\_SERIAL\_CONFIG



#### 【 機能 】

ROBIN のシリアル通信設定を行います。

#### 【 入力端子 】

なし

#### 【 出力端子 】

なし

#### 【 パラメータ 】

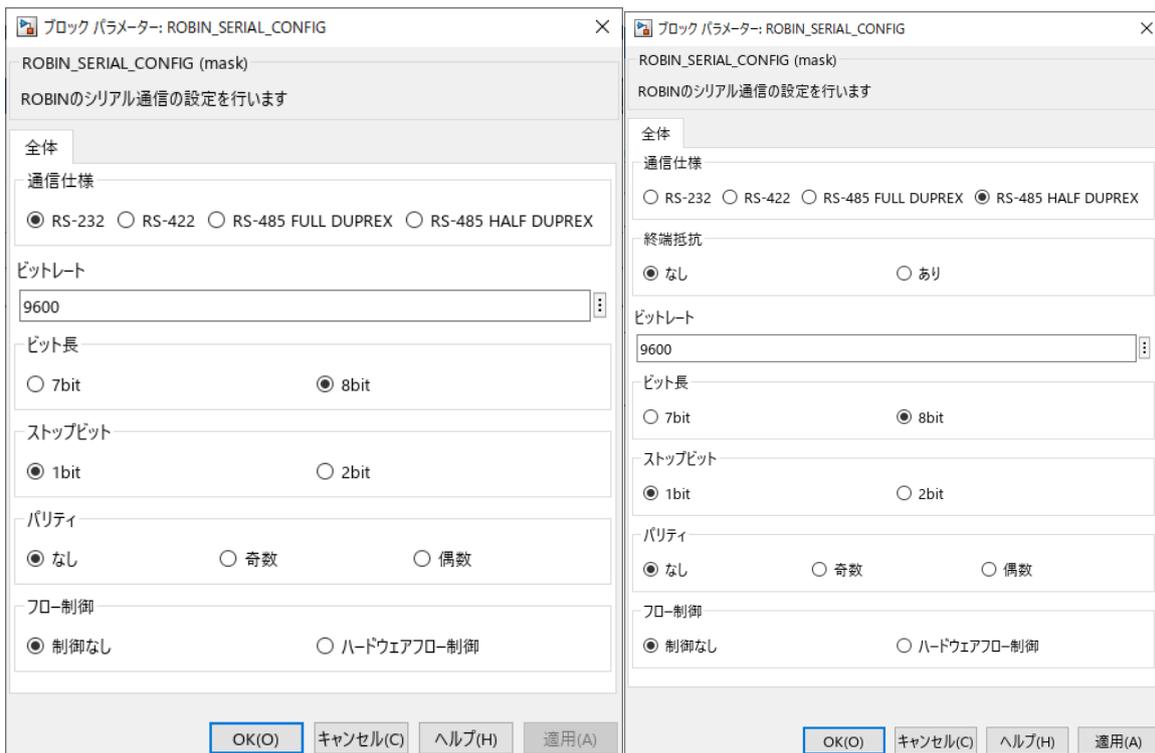


図 3-3 6 ROBIN\_SERIAL\_CONFIG パラメータ (232)

図 3-3 7 ROBIN\_SERIAL\_CONFIG パラメータ (422/485)

通信仕様	:	RS-232 / RS-422 / RS-485 全二重 / RS-485 半二重から選択。
終端抵抗	:	有無を選択 (RS-232 の場合「なし」固定)
ビットレート	:	ビットレートを 300bps ~ 1500000bps で指定。 (RS-232 の場合上限 500000 bps)
ビット長	:	7bit/8bit から選択。
ストップビット	:	1bit/2bit から選択。
パリティ	:	なし/奇数パリティ/偶数パリティから選択。
フロー制御	:	制御有無を選択。

【 動作説明 】

ROBIN の SERIAL ポートが本ブロックでのパラメータ設定通りに動作するよう設定します。  
設定後、シリアル通信の送受信が可能になります。

【 使用例 】

サンプルモデル rs\_echo での使用例です。

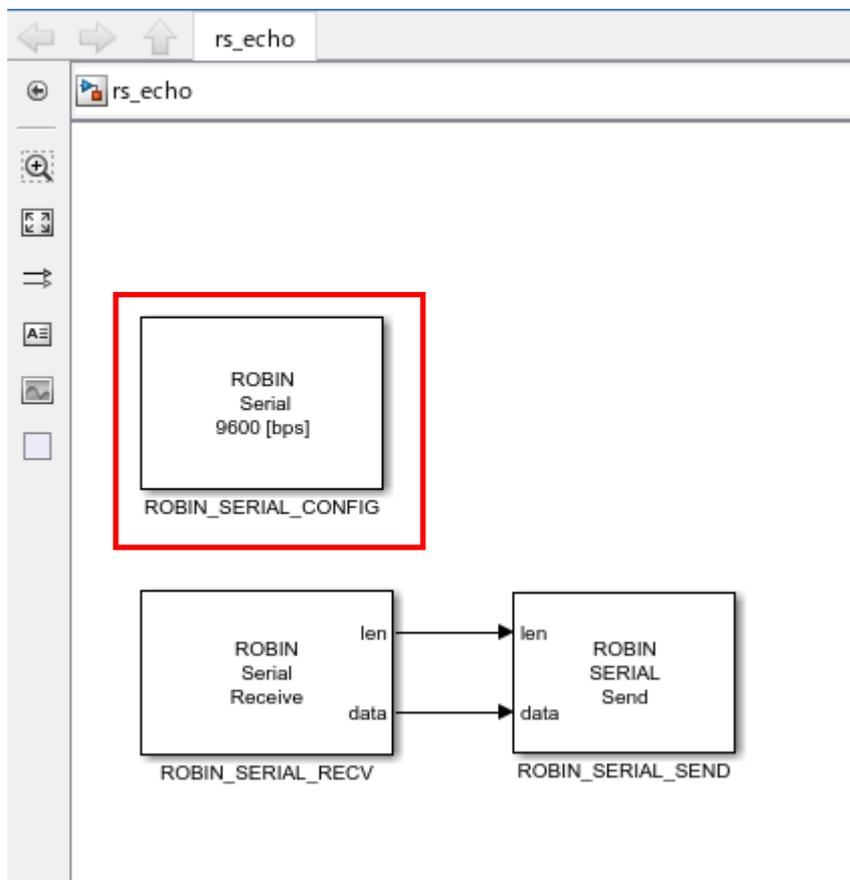


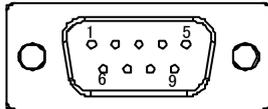
図 3-38 ROBIN\_SERIAL\_CONFIG 使用例

**【 コネクタピンアサイン 】**

本ブロックに対応する SERIAL のコネクタピンアサインは以下となります。  
通信仕様によりピンアサインが異なりますのでご注意ください。

なお、ROBIN で使用している「i.MX8M Plus」マイコンのデータシートでは RS-232C のハードウェアフロー制御出力を CTS 信号、入力を RTS 信号としていますが、本書記載のピンアサインは一般的な RS-232C と合わせて出力を RTS 信号、入力を CTS 信号としています。

[RS-232C]



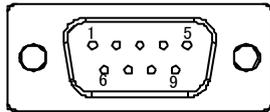
ピン番号	信号名	ピン番号	信号名
1	N.C.	6	N.C.
2	RxD	7	RTS
3	TxD	8	CTS
4	N.C.	9	N.C.
5	GND		

信号説明

信号名	入出力	信号説明
RXD	入力	RS232C データ受信
TXD	出力	RS232C データ送信
RTS	出力	RS232C 送信リクエスト
CTS	入力	RS232C 送信許可
GND	—	グラウンド
N.C.	—	No Connect (未使用)

図 3-39 ROBIN\_SERIAL\_CONFIG コネクタピンアサイン (RS-232)

[RS-422、RS-485FULL DUPLEX]



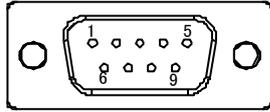
ピン番号	信号名	ピン番号	信号名
1	N.C.	6	N.C.
2	RxD+	7	TxD-
3	TxD+	8	RxD-
4	N.C.	9	N.C.
5	GND		

信号説明

信号名	入出力	信号説明
RXD+	入力	RS422/RS485 データ受信+側
RXD-	入力	RS422/RS485 データ受信-側
TXD+	出力	RS422/RS485 データ送信+側
TXD-	出力	RS422/RS485 データ送信-側
GND	—	グラウンド
N.C.	—	No Connect (未使用)

図 3-40 ROBIN\_SERIAL\_CONFIG コネクタピンアサイン (RS-422、RS-485 全二重)

## [RS-485HALF DUPLEX]



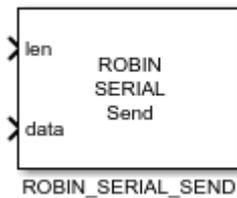
ピン番号	信号名	ピン番号	信号名
1	N.C.	6	N.C.
2	Reserved	7	RxD-/TxD-
3	RxD+/TxD+	8	Reserved
4	N.C.	9	N.C.
5	GND		

## 信号説明

信号名	入出力	信号説明
RXD+	入力	RS422/RS485 データ受信+側
RXD-	入力	RS422/RS485 データ受信-側
TXD+	出力	RS422/RS485 データ送信+側
TXD-	出力	RS422/RS485 データ送信-側
GND	—	グラウンド
N.C.	—	No Connect (未使用)
Reserved		予約 (未使用)

図 3-4 1 ROBIN\_SERIAL\_CONFIG コネクタピンアサイン (RS-485 半二重)

### 3-8 ROBIN\_SERIAL\_SEND



#### 【 機能 】

ROBIN からシリアル通信データを送信します。

#### 【 入力端子 】

len : 浮動小数点数型スカラー  
 data : 8bit 符号なし整数型 1024 次元ベクトル

#### 【 出力端子 】

なし

#### 【 パラメータ 】



図 3-4 2 ROBIN\_SERIAL\_SEND パラメータ

サンプル時間 : ブロックのサンプリング時間

#### 【 動作説明 】

サンプル時間毎に「Data」端子への入力データ配列の先頭から「Len」端子への入力値個のデータを ROBIN の SERIAL からシリアル通信で出力します。

本ブロックは、ROBIN\_SERIAL\_CONFIG ブロックと共に使用します。

サンプル時間の値は、Simulink モデル全体のサンプリング時間の整数倍または-1 としてください。

-1 を設定した場合は、Simulink モデル全体のサンプリング時間を継承します。

本ブロックではシリアル通信の出力完了監視を行っており、ROBIN\_SERIAL\_CONFIG ブロックで設定したビットレートが低いほど処理に時間がかかる仕様であることにご注意ください。

**【 使用例 】**

サンプルモデル rs\_echo での使用例です。

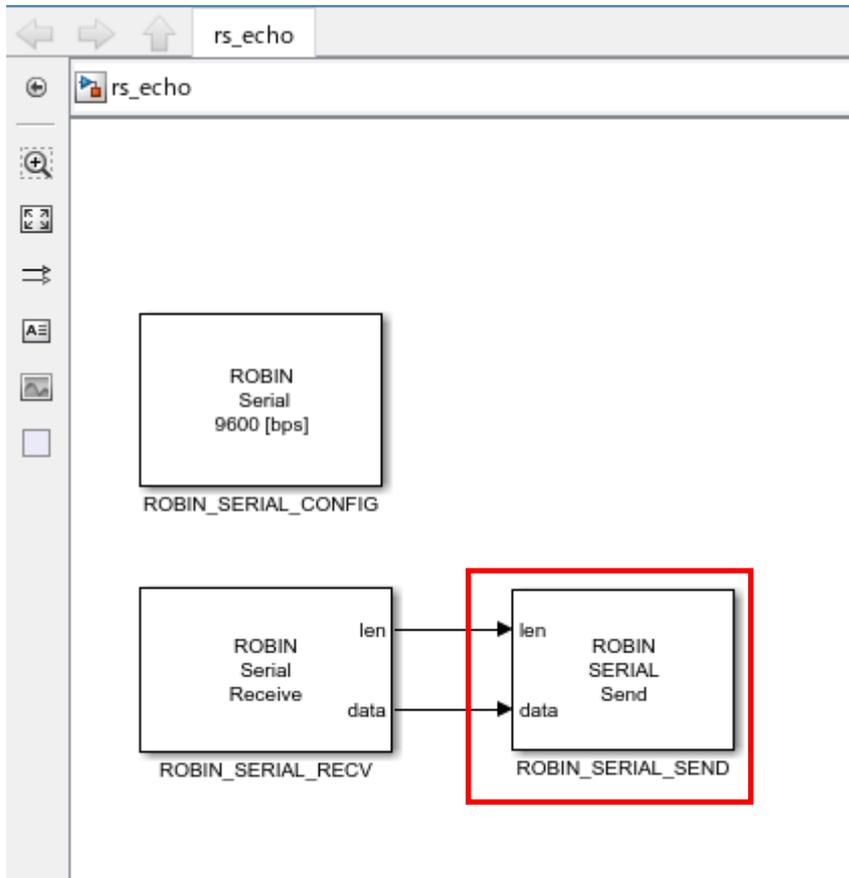
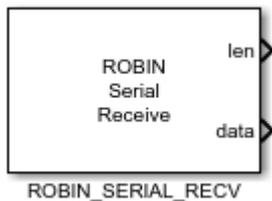


図 3-4 3 ROBIN\_SERIAL\_SEND 使用例

**【 端子台展開図 】**

ROBIN\_SERIAL\_CONFIG と同じです。

### 3-9 ROBIN\_SERIAL\_RECV



#### 【 機能 】

ROBIN で受信したシリアル通信データを取得します。

#### 【 入力端子 】

なし

#### 【 出力端子 】

len : 浮動小数点数型スカラー  
 data : 8bit 符号なし整数型 1024 次元ベクトル

#### 【 パラメータ 】



図 3-4 4 ROBIN\_SERIAL\_RECV パラメータ

サンプル時間 : ブロックのサンプリング時間

#### 【 動作説明 】

サンプル時間毎に ROBIN の SERIAL で受信したシリアル通信データの長さを「Len」端子、受信データを「Data」端子から出力します。  
 サンプル時間毎に受信できる最大データ長は 1024byte となります。

本ブロックは、ROBIN\_SERIAL\_CONFIG ブロックと共に使用します。

サンプル時間の値は、Simulink モデル全体のサンプリング時間の整数倍または-1 としてください。  
 -1 を設定した場合は、Simulink モデル全体のサンプリング時間を継承します。

**【 使用例 】**

サンプルモデル rs\_echo での使用例です。

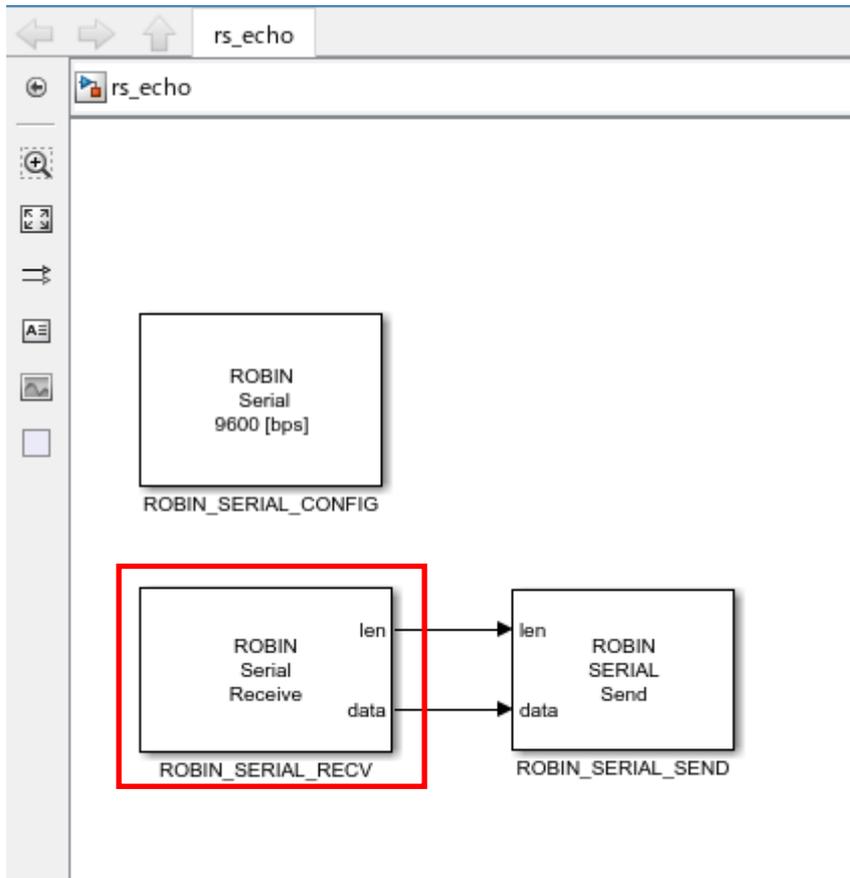
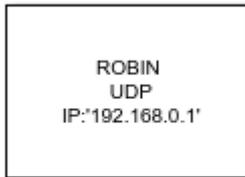


図 3-4 5 ROBIN\_SERIAL\_RECV 使用例

**【 端子台展開図 】**

ROBIN\_SERIAL\_CONFIG と同じです。

## 3-10 ROBIN\_UDP\_CONFIG



ROBIN\_UDP\_CONFIG

**【 機能 】**

ROBIN の UDP 通信設定を行います。

**【 入力端子 】**

なし

**【 出力端子 】**

なし

**【 パラメータ 】**


図 3-4 6 ROBIN\_UDP\_CONFIG パラメータ

<i>IPアドレス</i>	:	IPアドレス
<i>サブネットマスク</i>	:	IPアドレスサブネットマスク
<i>デフォルトゲートウェイ</i>	:	デフォルトゲートウェイ
<i>ポート番号指定</i>	:	ポート番号の固定設定
<i>ポート番号</i>	:	ポート番号 (固定設定としない場合、非表示)

**【 動作説明 】**

ROBIN の to DEVICE ポートがパラメータ設定通りに動作するようネットワークの設定を行います。設定後に to DEVICE ポートで UDP 通信の送受信が可能になります。

本ブロックで設定するネットワーク設定は to HOST ポートでのホスト PC との通信には影響しません。

**【 使用例 】**

サンプルモデル `udp_echo` での使用例です。

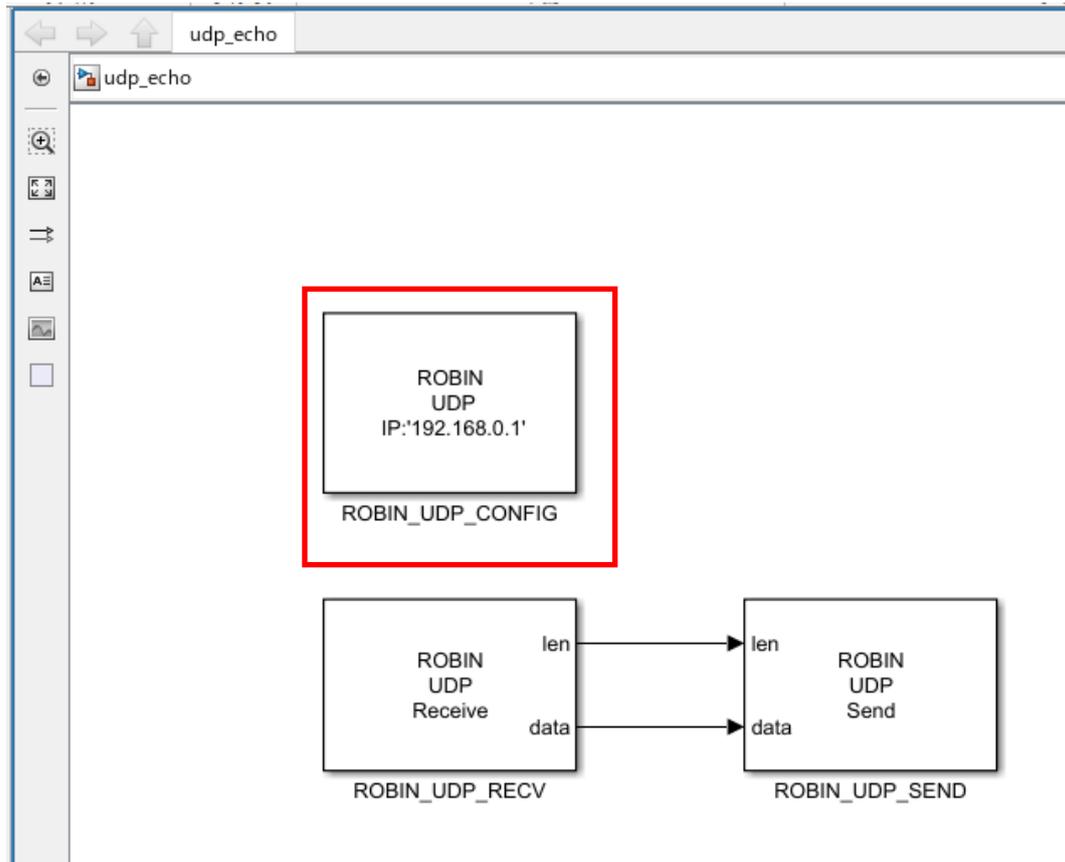
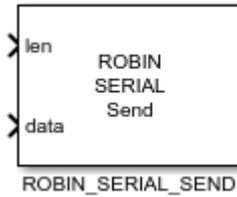


図 3-4 7 ROBIN\_UDP\_CONFIG 使用例

## 3-11 ROBIN\_UDP\_SEND


**【 機能 】**

ROBIN から指定の IP アドレスに UDP 通信データを送信します。

**【 入力端子 】**

len : 浮動小数点数型スカラー  
 data : 8bit 符号なし整数型 可変次元ベクトル

**【 出力端子 】**

なし

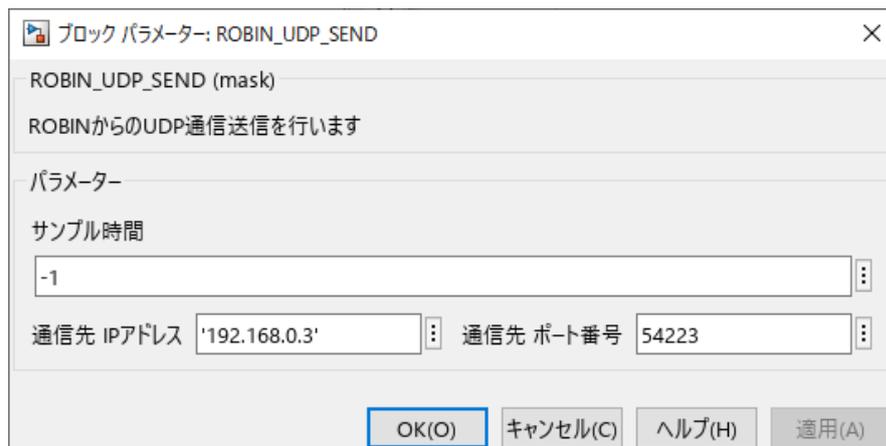
**【 パラメータ 】**


図 3-4 8 ROBIN\_UDP\_SEND パラメータ

サンプル時間 : ブロックのサンプリング時間  
 通信先 IP アドレス : UDP 送信相手の IP アドレス  
 通信先ポート番号 : UDP 送信相手のポート番号

**【 動作説明 】**

サンプル時間毎に「data」端子への入力データ配列の先頭から「len」端子への入力値個のデータを ROBIN の to DEVICE ポートから通信先 IP アドレス、通信先ポート番号で指定した通信相手に送信します。本ブロックは、ROBIN\_UDP\_CONFIG ブロックと共に使用します。

サンプル時間の値は、Simulink モデル全体のサンプリング時間の整数倍または-1 としてください。  
-1 を設定した場合は、Simulink モデル全体のサンプリング時間を継承します。

本ブロックでは UDP 通信の送信完了監視を行っており、送信データ数が多い場合、「基本サンプル時間」の設定によっては周期超過のエラーが発生することにご注意ください。

ROBIN の UDP には、初回の UDP データ送信開始から ARP による MAC アドレス解決が行われるまでの間（約 1 秒）に 32,768 バイトを超えるデータを送信することができない制約がありますが、この制約による UDP 送信失敗の発生は ROBIN の ARM プログラム、UDP 送信先のデバイスのいずれでも認識できません。

データの送信漏れのない確実な UDP 通信を行うには、Enabled Subsystem 等を利用して初回送信後約 1 秒間本ブロックを実行しないか、初回の通信データサイズを 32768byte 以下として通信相手とのハンドシェイクを行った後に 2 回目以降の送信を行うようにしてください。

**【 使用例 】**

サンプルモデル udp\_echo での使用例です。

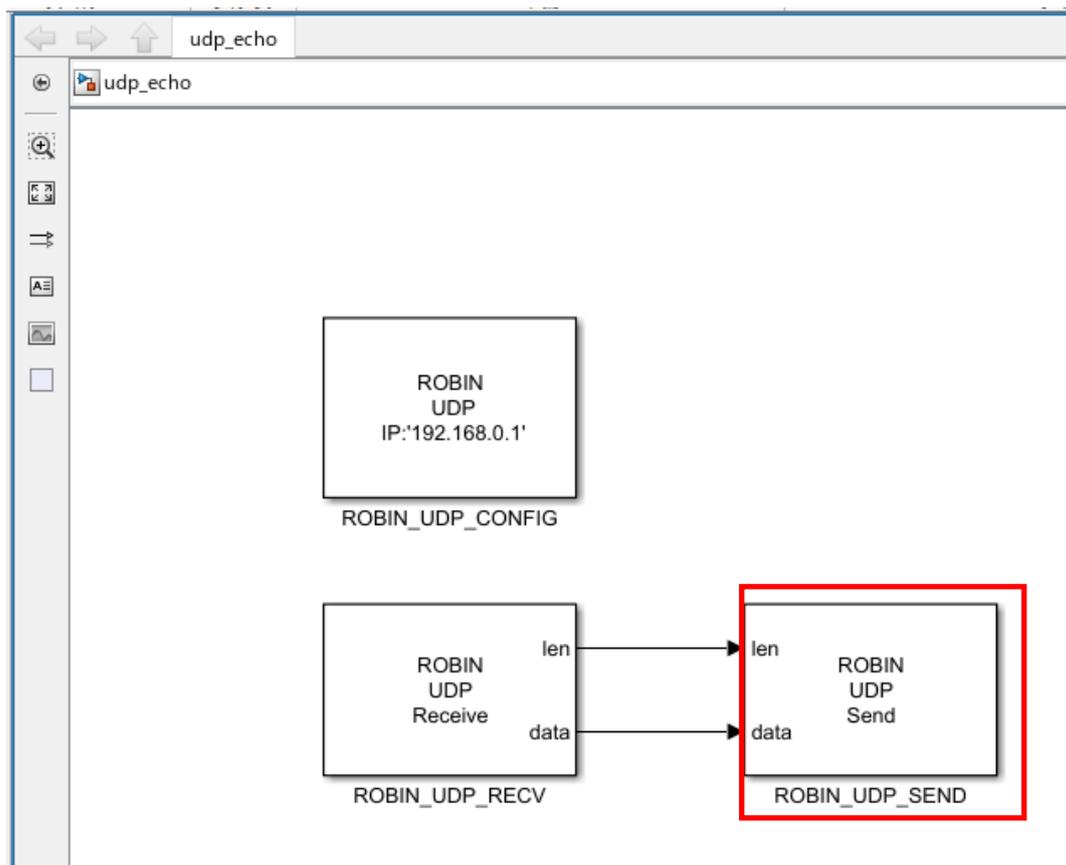
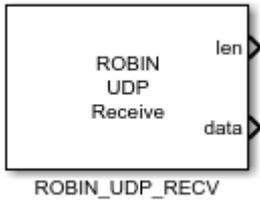


図 3-4 9 ROBIN\_UDP\_SEND 使用例

### 3-12 ROBIN\_UDP\_RECV



#### 【 機能 】

ROBIN で指定の IP アドレスから受信した UDP 通信データを取得します。

#### 【 入力端子 】

なし

#### 【 出力端子 】

len : 浮動小数点数型スカラー  
 data : 8bit 符号なし整数型 8872 次元ベクトル

#### 【 パラメータ 】



図 3-50 ROBIN\_UDP\_RECV パラメータ

サンプル時間 : ブロックのサンプリング時間  
 受信 IP アドレス : UDP 送信元の IP アドレス

**【 動作説明 】**

サンプル時間毎に ROBIN の to DEVICE ポートで受信した受信 IP アドレスから送信された UDP 通信データ長を「len」端子、データを「data」端子から出力します。

UDP は最大 65,507byte のデータを「UDP データグラム」としてまとめて通信を行うプロトコルですが、ROBIN は 8,872byte 以下の UDP データグラムのみを受信可能な仕様となっております。

8,873byte 以上の UDP データグラムは受信時に破棄されますが、送信元、ROBIN のどちらともデータグラムの破棄はユーザーに通知されません。

サンプル時間内に受信 IP アドレスから UDP データグラムを受信していない場合 (8,873byte 以上の UDP データグラムを破棄した場合も含みます)、「len」端子の出力値は 0 となります。

サンプル時間内に受信 IP アドレスからの UDP データグラムを複数回受信した場合、サンプル時間毎に 1 データグラムずつ受信順に「len」端子、「data」端子から出力します。

本ブロックは、ROBIN\_UDP\_CONFIG ブロックと共に使用します。

本ブロックと ROBIN\_UDP\_RECV\_ANYIP ブロックを同時に使用することはできません。

サンプル時間の値は、Simulink モデル全体のサンプリング時間の整数倍または-1 としてください。

-1 を設定した場合は、Simulink モデル全体のサンプリング時間を継承します。

**【 使用例 】**

サンプルモデル udp\_echo での使用例です。

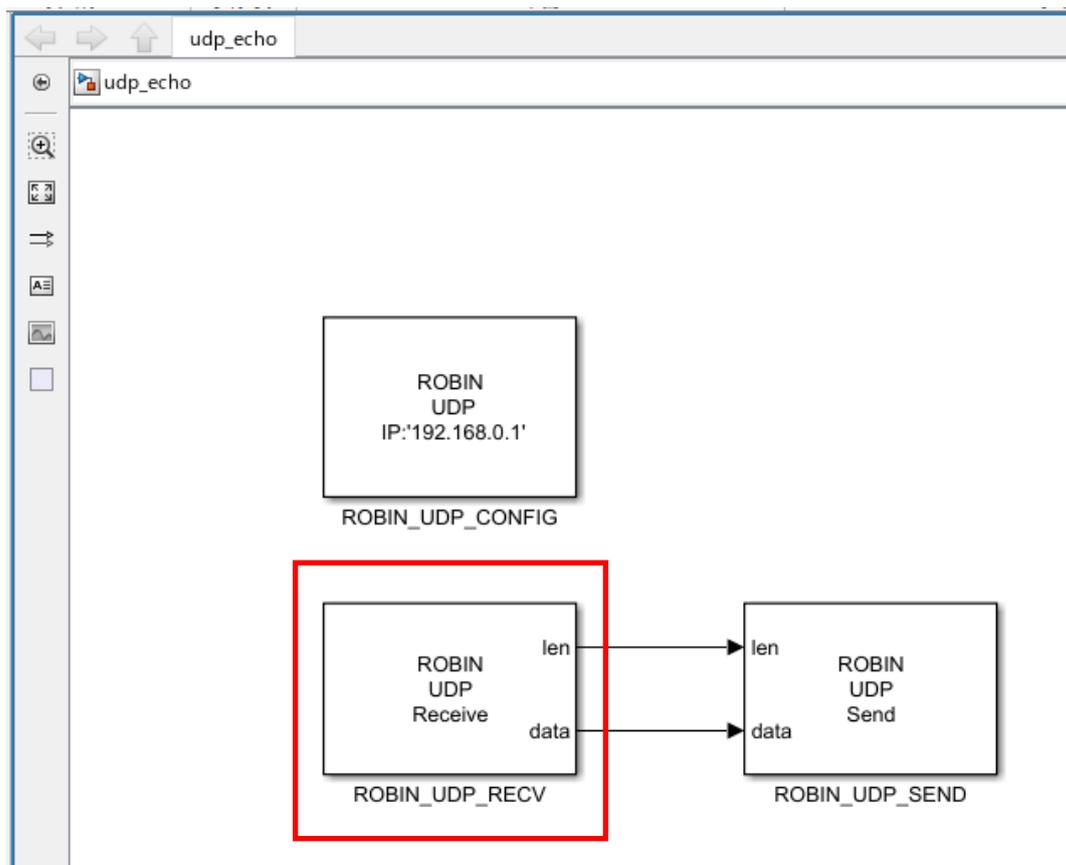
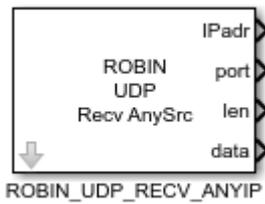


図 3-5 1 ROBIN\_UDP\_RECV 使用例

## 3-13 ROBIN\_UDP\_RECV\_ANYIP


**【 機能 】**

ROBIN で受信した UDP 通信データを取得します。

**【 入力端子 】**

なし

**【 出力端子 】**

IPadr	:	8bit 符号なし整数型 4 次元ベクトル
port	:	16bit 符号なし整数型スカラー
len	:	浮動小数点数型スカラー
data	:	8bit 符号なし整数型 8872 次元ベクトル

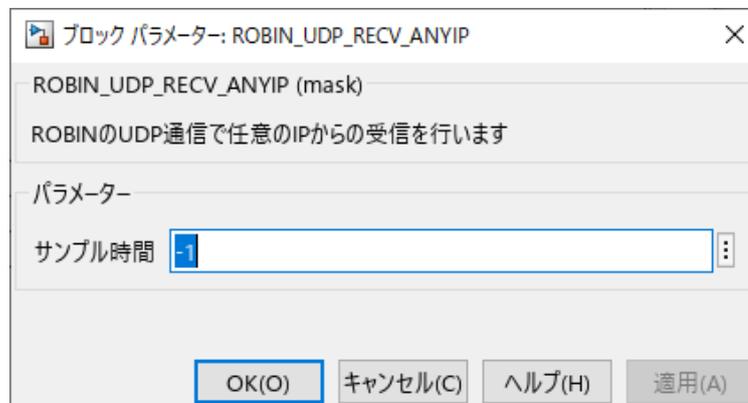
**【 パラメータ 】**


図 3-5 2 ROBIN\_UDP\_RECV\_ANYIP パラメータ

サンプル時間 : ブロックのサンプリング時間

**【 動作説明 】**

サンプル時間毎に ROBIN の to DEVICE ポートで受信した UDP 通信データの送信元 IP アドレスを「IPadr」端子、送信元ポート番号を「port」端子、受信したデータ長を「len」端子、受信データを「data」端子から出力します。

UDP は最大 65,507byte のデータを「UDP データグラム」としてまとめて通信を行うプロトコルですが、ROBIN は 8,872byte 以下の UDP データグラムのみを受信可能な仕様となっております。

8,873byte 以上の UDP データグラムは受信時に破棄されますが、送信元、ROBIN のどちらともデータグラムの破棄はユーザに通知されません。

サンプル時間内に受信 IP アドレスから UDP データグラムを受信していない場合 (8,873byte 以上の UDP データグラムを破棄した場合も含みます)、「len」端子の出力値は 0 となります。

サンプル時間内に UDP 通信データを複数回受信した場合、サンプル時間毎に 1 データずつ受信順に出力します。

サンプル時間毎に受信できる最大データ長は 8872byte となります。

本ブロックは、ROBIN\_UDP\_CONFIG ブロックと共に使用します。

本ブロックと ROBIN\_UDP\_RECV ブロックを同時に使用することはできません。

サンプル時間の値は、Simulink モデル全体のサンプリング時間の整数倍または-1 としてください。

-1 を設定した場合は、Simulink モデル全体のサンプリング時間を継承します。

**【 使用例 】**

任意の IP アドレスからの UDP 受信をホストに通知するモデルでの使用例を以下に示します。

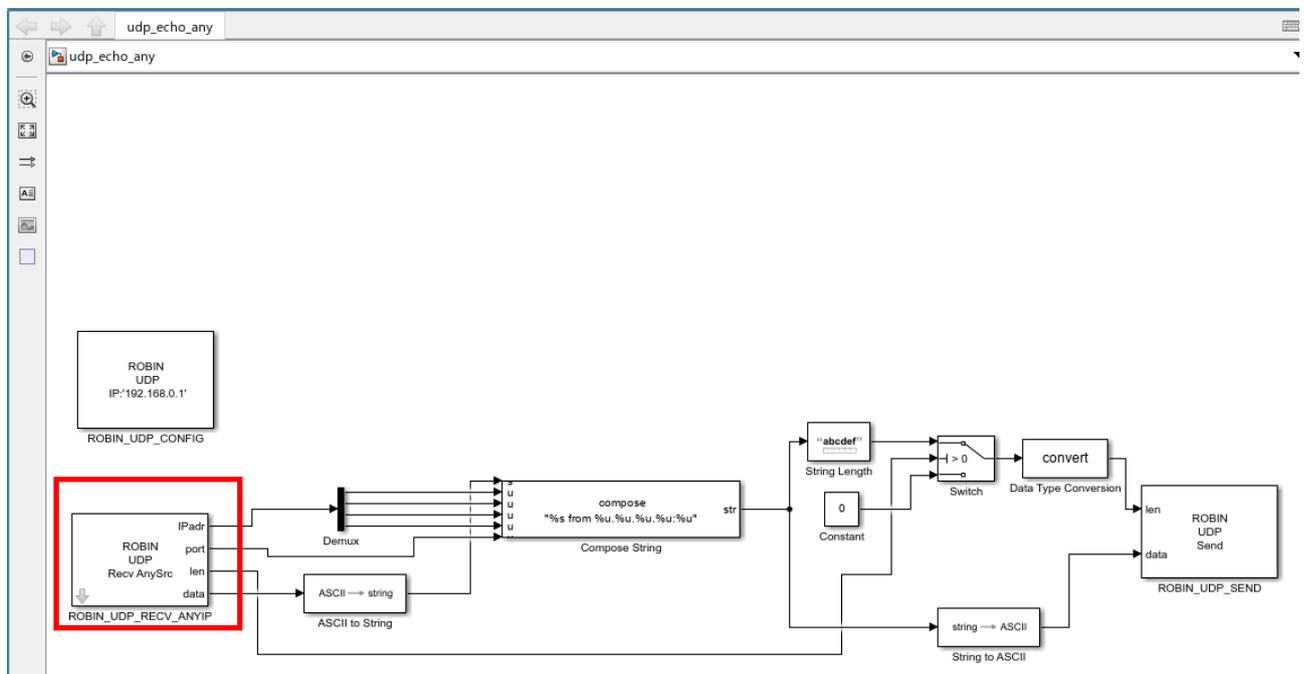
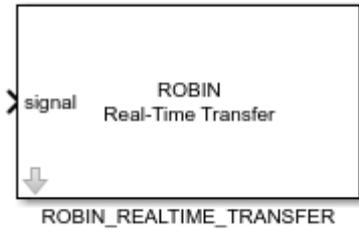


図 3-53 ROBIN\_UDP\_RECV\_ANYIP 使用例

## 3-14 ROBIN\_REALTIME\_TRANSFER


**【 機能 】**

ホストアプリケーションにグラフ表示とリアルタイム保存を行うデータを送信します。

**【 入力端子 】**

signal : 浮動小数点数型可変次元ベクトル

**【 出力端子 】**

なし

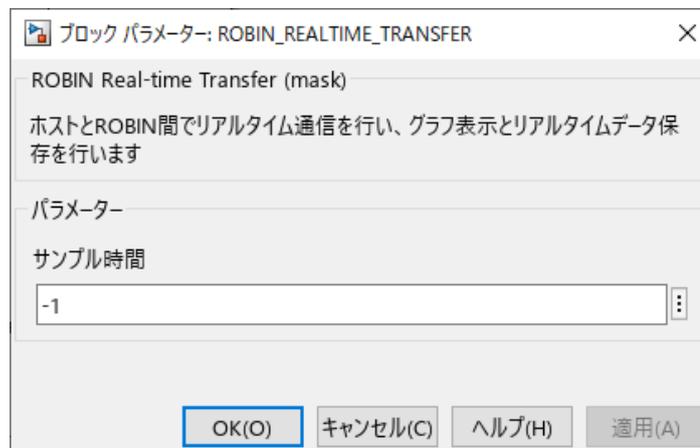
**【 パラメータ 】**


図 3-5 4 SGL\_REALTIME\_TRANSFER パラメータ

サンプル時間 : ブロックのサンプリング時間

**【 動作説明 】**

サンプル時間毎に「Signal 端子」から取り込んだ信号値をホストアプリケーションに送信できる状態とします。ホストアプリケーションで信号を登録することによりグラフ表示とリアルタイムデータ保存が可能です。グラフ表示は 4096 点、リアルタイム保存データは 1 ファイルあたり 8192 点となります。詳細はホストアプリケーション取扱説明書 (DRA0001988-00-S26632) を参照ください。

本ブロックはホストアプリケーションとリアルタイムで通信を行うブロックのため、スタンドアローンモードでは動作しません。

本ブロックへの入力信号数に制限はありませんが、ホストアプリケーションで登録してグラフ表示とリアルタイムデータ保存を行う信号数 (以下、登録信号数) はホストアプリケーションの仕様により 64 信号までとなります。

ただし、ROBIN\_DATA\_SAVE ブロックまたは ROBIN\_DATA\_SAVE\_REPEAT ブロックで非常に多数のデータを保存する設定としている場合、ROBIN のメモリ上に確保したヒープ領域の容量の制約により 64 信号を登録できない場合もあります

本ブロックへの複数信号の入力には、Simulink の信号結合ブロック (MUX ブロックまたは Bus Creator ブロック) を使用してください。

アプリケーションでの表示信号の指定は、ブロックへの入力チャンネル番号で行います。(信号結合ブロックの上から順に 0、1、2、... となります)

サンプル時間の値は、Simulink モデル全体のサンプリング時間の整数倍または -1 としてください。-1 を設定した場合は、Simulink モデル全体のサンプリング時間を継承します。

**【 使用例 】**

サンプルモデル sineWave.slx での使用例です。

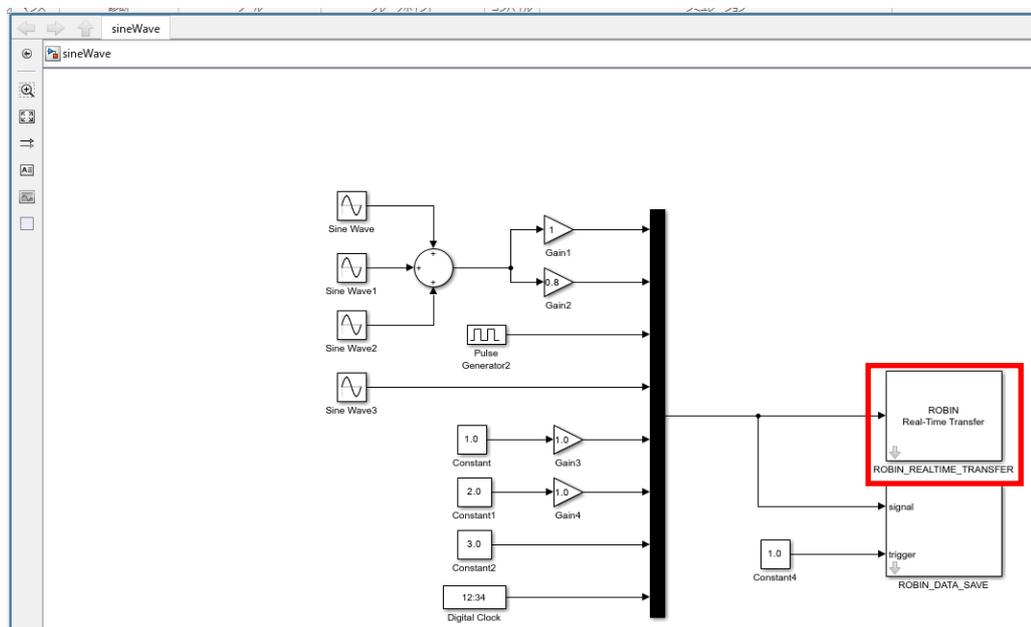
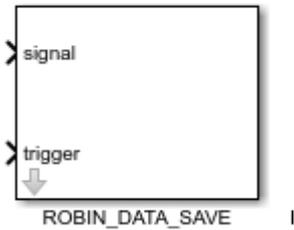


図 3-55 ROBIN\_REALTIME\_TRANSFER 使用例

## 3-15 ROBIN\_DATA\_SAVE


**【 機能 】**

トリガ入力でデータを保存します。

**【 入力端子 】**

signal : 浮動小数点数型可変次元ベクトル  
 trigger : 浮動小数点数型スカラー

**【 出力端子 】**

なし

**【 パラメータ 】**


図 3-5 6 ROBIN\_DATA\_SAVE パラメータ

トリガ待ち設定 : トリガ待ちの開始設定  
 データ点数 : 保存するデータ点数  
 プリトリガデータ点数 : 保存するトリガ発生前のデータ点数  
 サンプル時間 : ブロックのサンプリング時間

**【 動作説明 】**

「trigger」端子への入力値が 1 以上の時をトリガとして、サンプル時間毎に「signal」端子から取り込んだ信号値を ROBIN の内部メモリにデータ点数保存します。  
 通常はトリガ発生後の「データ点数-プリトリガデータ点数」点とトリガ発生前の「プリトリガデータ点数」点を保存しますが、トリガ待ちの開始から「プリトリガデータ点数」を保存する前にトリガが発生した場合はトリガ待ちの開始から「データ点数」点の保存となります。

トリガ値の確認はトリガ待ち設定により行います。ARM プログラム実行開始直後からデータの保存を行う場合、「常時トリガ待ち」に設定してください。

なお、本ブロックをトリガ待ち設定「常時トリガ待ち」で使用する場合、ARM プログラム実行中に 1 回のみデータを保存する仕様となっております。

複数回トリガでデータを保存する場合はトリガ待ち設定を「ホストアプリケーション指示」に設定するか、後述の ROBIN\_DATA\_SAVE\_REPEAT ブロックを使用してください。

本ブロックで保存できるデータ点数は、ご使用のテンプレート make ファイルにより変動します。

テンプレート make ファイルについては「4-2-2 テンプレート make ファイルの選択 (通常モード用)」および「4-2-3 テンプレート make ファイルの選択 (スタンドアロンモード用)」を参照ください。

なお、本ブロックと ROBIN\_REALTIME\_TRANSFER ブロックは同じメモリ空間にデータ保存するため、ROBIN\_REALTIME\_TRANSFER ブロックを使用する場合、登録 1 信号毎に 24,576 点分のメモリを使用することにご注意ください。

表 3-5 ROBIN\_DATA\_SAVE 最大保存点数

テンプレート make ファイル	最大保存点数
robin itc real64	33,540,000
robin itc real32	67,090,000
robin ddr real64	16,760,000
robin ddr real32	33,540,000
robin bss real64	8,370,000
robin bss real32	16,760,000
robin itc real64 sa	33,540,000
robin itc real32 sa	67,090,000
robin ddr real64 sa	16,760,000
robin ddr real32 sa	33,540,000
robin bss real64 sa	8,370,000
robin bss real32 sa	16,760,000

本ブロックへの複数信号の入力には、Simulink の信号結合ブロック (MUX ブロックまたは Bus Creator ブロック) を使用してください。

アプリケーションでの信号の指定は、ブロックへの入力チャンネル番号で行います。

(信号結合ブロックの上から順に 0、1、2、... となります)

サンプル時間の値は、Simulink モデル全体のサンプリング時間の整数倍または-1 としてください。

-1 を設定した場合は、Simulink モデル全体のサンプリング時間を継承します。

**【 使用例 】**

サンプルモデル sineWave.slx での使用例です。

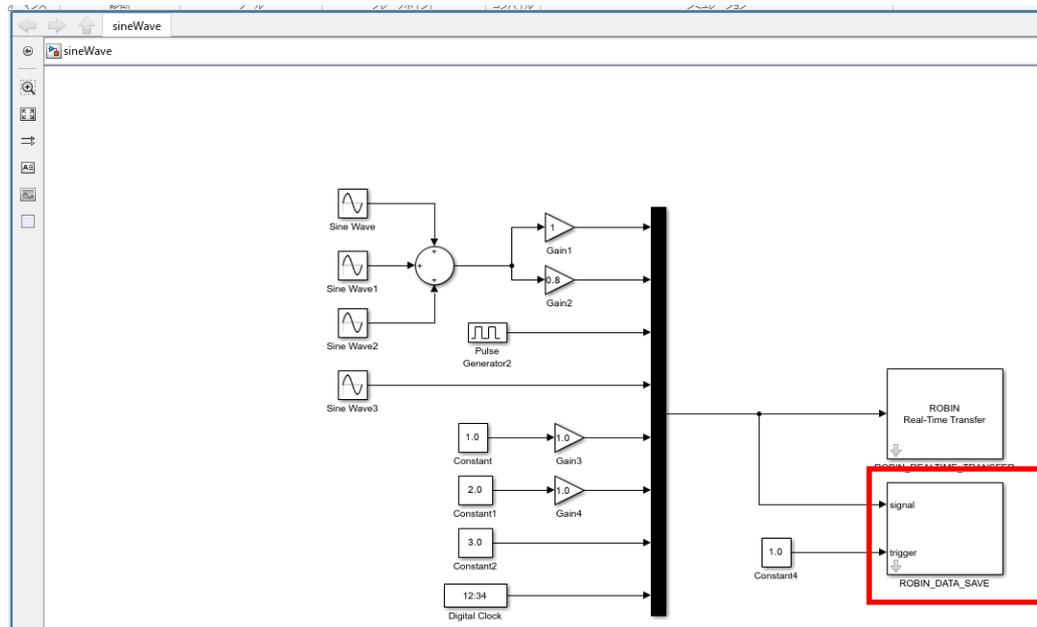
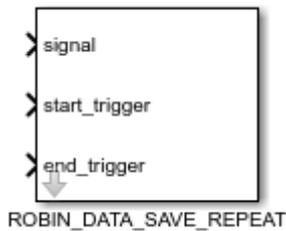


図 3-57 SGL\_DATA\_SAVE 使用例

## 3-16 ROBIN\_DATA\_SAVE\_REPEAT


**【 機能 】**

トリガ入力でデータの繰返保存を行います。

**【 入力端子 】**

signal : 浮動小数点数型可変次元ベクトル  
 start\_trigger : 浮動小数点数型スカラー  
 end\_trigger : 浮動小数点数型スカラー

**【 出力端子 】**

なし

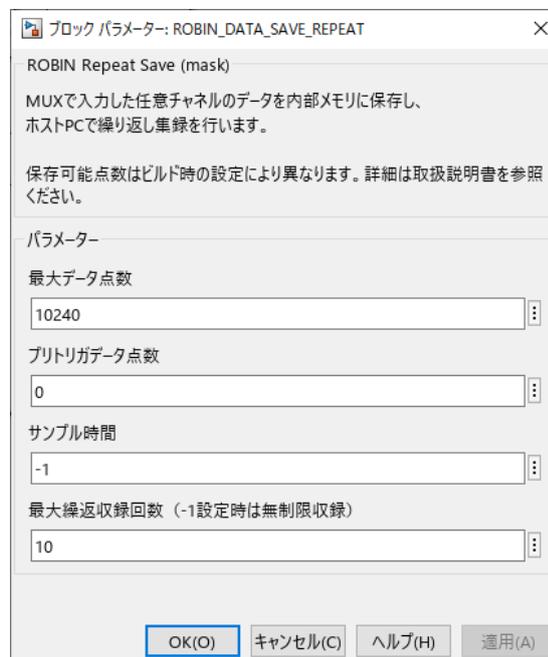
**【 パラメータ 】**


図 3-58 ROBIN\_DATA\_SAVE\_REPEAT パラメータ

最大データ点数 : 保存するデータ点数  
 プリトリガデータ点数 : 保存するトリガ発生前のデータ点数  
 サンプル時間 : ブロックのサンプリング時間  
 最大繰返収録回数 : 繰返保存を行う回数設定

**【動作説明】**

「start\_trigger」端子への入力値が1以上の時をトリガとして、サンプル時間毎に「signal」端子から取り込んだ信号値を ROBIN の内部メモリにデータ点数保存する処理を繰返保存回数繰り返します。ただし、繰返保存回数を-1 に設定している場合はホストから停止指示があるまで無限に繰り返します

通常はトリガ発生後の「データ点数-プリトリガデータ点数」点とトリガ発生前の「プリトリガデータ点数」点を保存しますが、トリガ待ちの開始から「プリトリガデータ点数」を保存する前にトリガが発生した場合はトリガ待ちの開始から「データ点数」点の保存となります。プリトリガを設定している場合、2 回目以降の保存において 1 回前の繰り返し保存で保存済みのデータはプリトリガデータに含まれないことにご注意ください。

トリガ待ちの開始以降に「end\_trigger」端子からの入力値が 1 以上の場合、データ点数の設定にかかわらず保存処理を中断し、次の繰返集録のトリガ待ちを開始します。なお、トリガ待ちが有効な状態で「start\_trigger」端子と「end\_trigger」端子が同時に 1.0 以上とならないようにモデリングを行ってください。

本ブロックで保存できるデータ点数は、ご使用のテンプレート make ファイルにより変動します。テンプレート make ファイルについては「4-2-2 テンプレート make ファイルの選択 (通常モード用)」および「4-2-3 テンプレート make ファイルの選択 (スタンドアロンモード用)」を参照ください。

なお、本ブロックと ROBIN\_REALTIME\_TRANSFER ブロックは同じメモリ空間にデータ保存するため、ROBIN\_REALTIME\_TRANSFER ブロックを使用する場合、登録 1 信号毎に 24,576 点分のメモリを使用することにご注意ください。

表 3-6 ROBIN\_DATA\_SAVE 最大保存点数

テンプレート make ファイル	最大保存点数
robin itc real64	33,540,000
robin itc real32	67,090,000
robin ddr real64	16,760,000
robin ddr real32	33,540,000
robin bss real64	8,370,000
robin bss real32	16,760,000
robin itc real64 sa	33,540,000
robin itc real32 sa	67,090,000
robin ddr real64 sa	16,760,000
robin ddr real32 sa	33,540,000
robin bss real64 sa	8,370,000
robin bss real32 sa	16,760,000

本ブロックへの複数信号の入力には、Simulink の信号結合ブロック (MUX ブロックまたは Bus Creator ブロック) を使用してください。

アプリケーションでの信号の指定は、ブロックへの入力チャンネル番号で行います。(信号結合ブロックの上から順に 0、1、2、... となります)

サンプル時間の値は、Simulink モデル全体のサンプリング時間の整数倍または-1 としてください。-1 を設定した場合は、Simulink モデル全体のサンプリング時間を継承します。

**【 使用例 】**

サンプルモデル sineWave\_repeat.slx での使用例です。

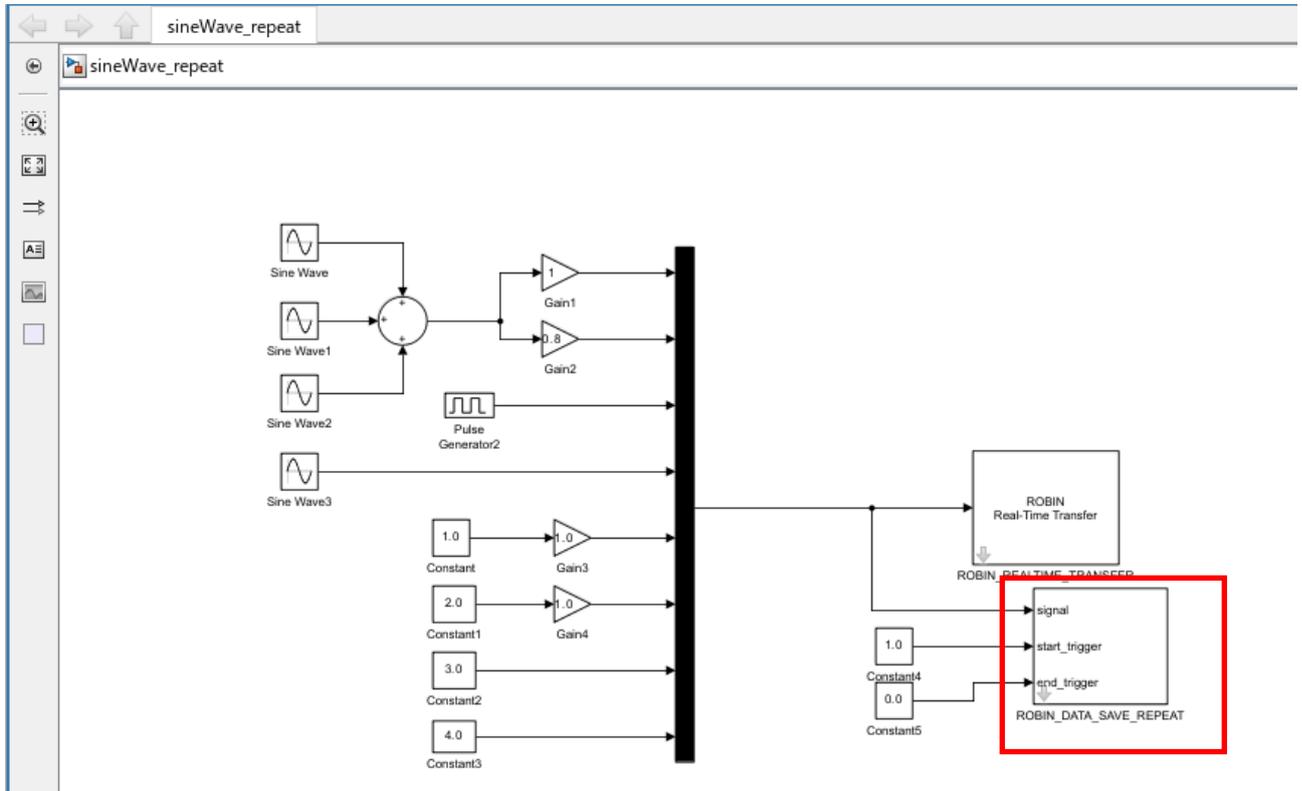
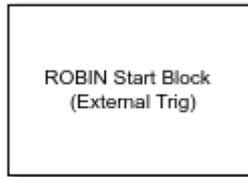


図 3-5 9 ROBIN\_DATA\_SAVE\_REPEAT 使用例

### 3-17 ROBIN\_EXT\_TRIG\_START



ROBIN\_EXT\_TRIG\_START

**【 機能 】**

外部制御信号による ARM プログラムの開始設定を行います。

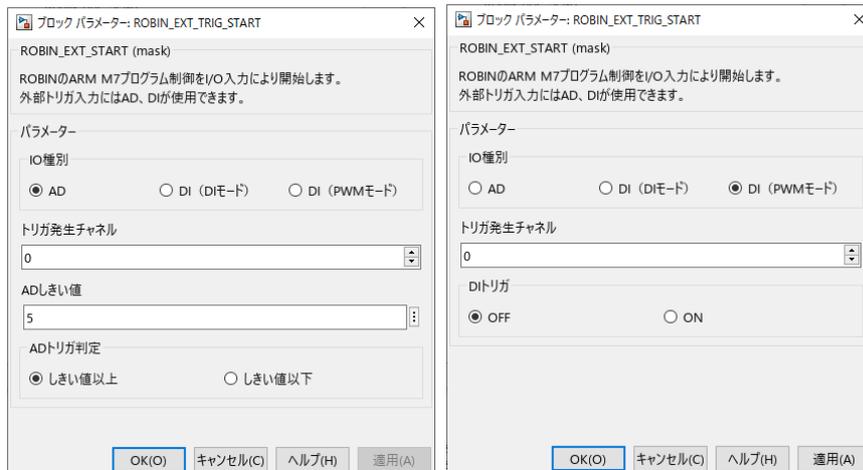
**【 入力端子 】**

なし

**【 出力端子 】**

なし

## 【 パラメータ 】



左：図 3-6 0 ROBIN\_EXT\_TRIG\_START パラメータ (AD)

右：図 3-6 1 ROBIN\_EXT\_TRIG\_START パラメータ (DI)

- IO 種別** : 外部制御信号として使用する IO
- AD* : ANALOG I/O ADCH0~CH7
  - DI (DI モード)* : DIGITAL I/O DICH0~CH7
  - DI (PWM モード)* : DIGITAL I/O PWMDICH0~CH1
- トリガ発生チャネル** : 外部制御信号として使用する I/O チャネル
- AD しきい値** : AD を使用する場合のしきい値
- AD トリガ判定** : AD を使用する場合のトリガ判定方法
- しきい値以上* : AD 入力電圧 > AD しきい値の時制御開始
  - しきい値以下* : AD 入力電圧 < AD しきい値の時制御開始
- DI トリガ** : DI を使用する場合のトリガ判定方法
- ON* : DI が ON の時制御開始
  - OFF* : DI が OFF の時制御開始

## 【 動作説明 】

ARM プログラムの実行停止中に **IO 種別** のトリガ発生チャネルをポーリングします。  
**AD しきい値** と **AD トリガ判定** (**IO 種別** が AD の場合) または **DI トリガ** (**IO 種別** が DI の場合) の条件をみたすことで、ホストアプリケーションからの ARM プログラム実行開始指示によらず ARM プログラムの実行を開始します。

**IO 種別** が AD の場合、AD レンジは ROBIN\_AD ブロックで設定したレンジを使用します。  
 ただし、ROBIN\_AD ブロックを使用していないモデルでは ±10V レンジで動作します。

**IO 種別** が DI (DI モード) の場合、ROBIN\_PWM ブロックは使用できず、ROBIN\_EXT\_TRIG\_STOP ブロックの **IO 種別** を DI (PWM モード) に設定することもできません。

**IO 種別** が DI (PWM モード) の場合、ROBIN\_DI ブロックと ROBIN\_DO ブロックは使用できず、ROBIN\_EXT\_TRIG\_STOP ブロックの **IO 種別** を DI (DI モード) に設定することもできません。

本ブロックと ROBIN\_EXT\_TRIG\_STOP ブロックの両方の条件を満たす外部制御信号入力が行われると ARM プログラムを制御できなくなるため、併用時は同時に条件を満たさないようご注意ください

**【 使用例 】**

サンプルモデル sineWave\_ext.slx での使用例です。

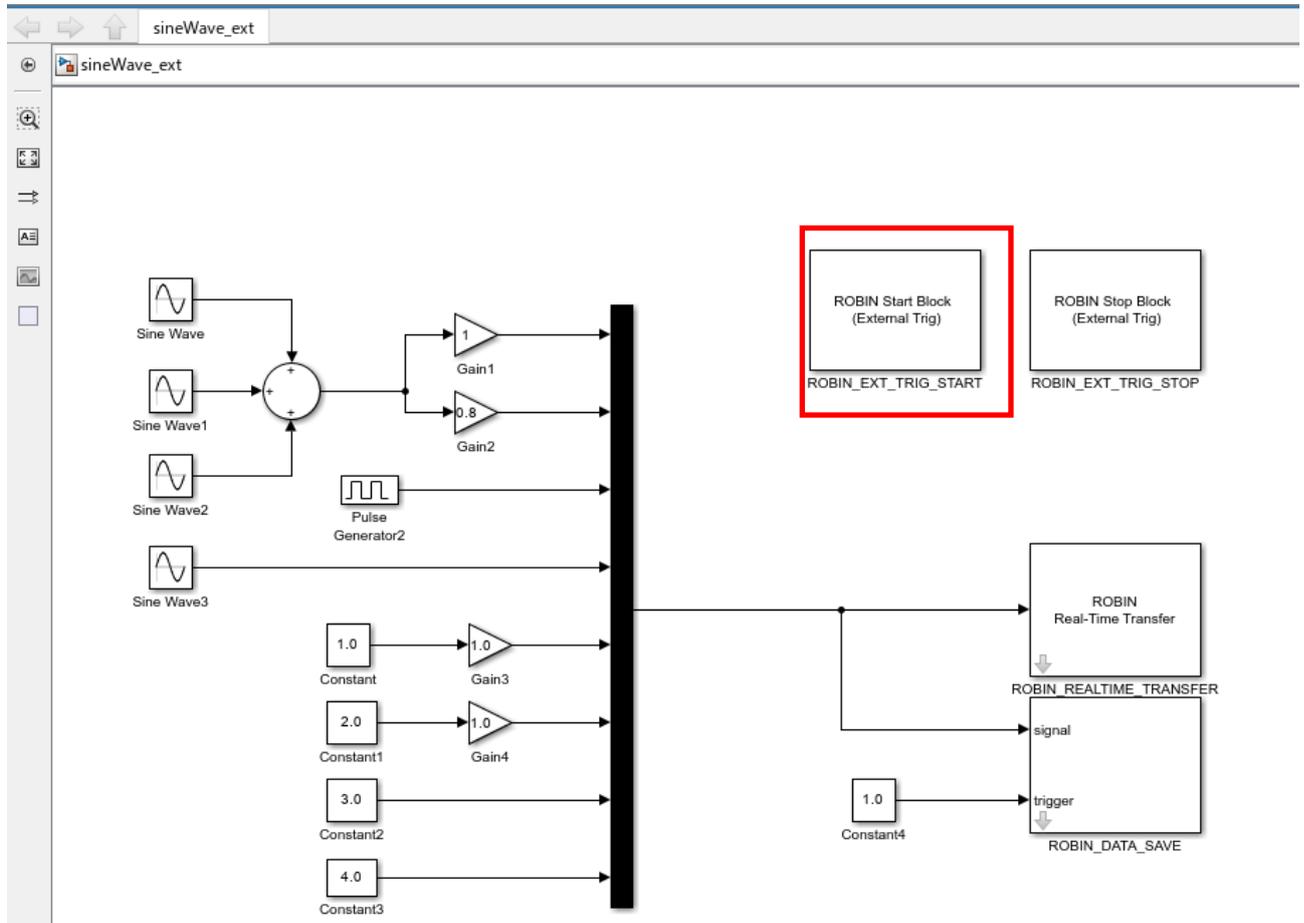
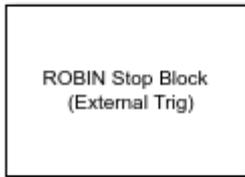


図 3-6 2 ROBINEXT\_TRIG\_START 使用例

## 3-18 ROBIN\_EXT\_TRIG\_STOP



ROBIN\_EXT\_TRIG\_STOP

## 【 機能 】

外部制御信号による ARM プログラムの開始設定を行います。

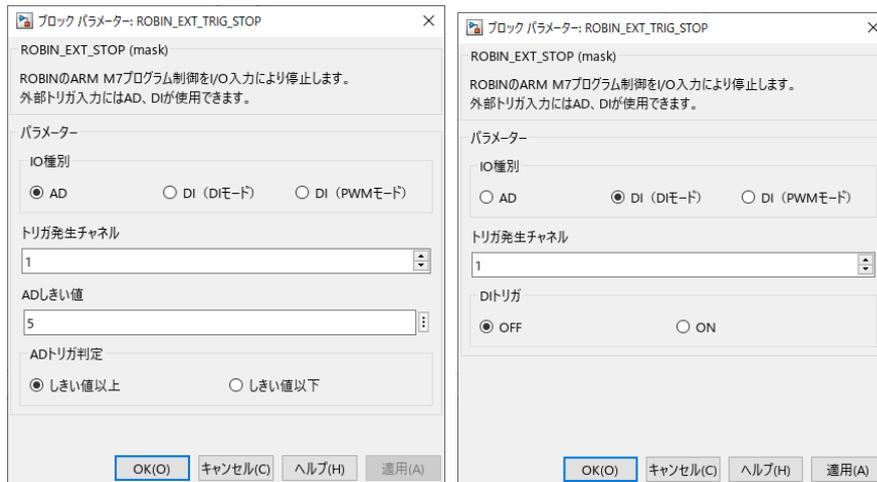
## 【 入力端子 】

なし

## 【 出力端子 】

なし

## 【 パラメータ 】



左：図 3-6 3 ROBIN\_EXT\_TRIG\_STOP パラメータ (AD)

右：図 3-6 4 ROBIN\_EXT\_TRIG\_STOP パラメータ (DI)

- IO 種別* : 外部制御信号として使用する IO
- AD* : ANALOG I/O ADCH0～CH7
  - DI (DI モード)* : DIGITAL I/O DICH0～CH7
  - DI (PWM モード)* : DIGITAL I/O PWMDICH0～CH1
- トリガ発生チャネル* : 外部制御信号として使用する I/O チャンネル
- AD しきい値* : AD を使用する場合のしきい値
- AD トリガ判定* : AD を使用する場合のトリガ判定方法
- しきい値以上* : AD 入力電圧 > AD しきい値の時制御開始
  - しきい値以下* : AD 入力電圧 < AD しきい値の時制御開始
- DI トリガ* : DI を使用する場合のトリガ判定方法
- ON* : DI が ON の時制御開始
  - OFF* : DI が OFF の時制御開始

## 【 動作説明 】

ARM プログラムの実行中に *IO 種別* のトリガ発生チャネルのデータを取得し、*AD しきい値* と *AD トリガ判定* (*IO 種別* が AD の場合) または *DI トリガ* (*IO 種別* が DI の場合) の条件をみたすことで、ホストアプリケーションからの ARM プログラム実行停止指示によらず ARM プログラムの実行を停止します。

*IO 種別* が AD の場合、AD レンジは ROBIN\_AD ブロックで設定したレンジを使用します。ただし、ROBIN\_AD ブロックを使用していないモデルでは ±10V レンジで動作します。

*IO 種別* が DI (DI モード) の場合、ROBIN\_PWM ブロックは使用できず、ROBIN\_EXT\_TRIG\_START ブロックの *IO 種別* を DI (PWM モード) に設定することもできません。

*IO 種別* が DI (PWM モード) の場合、ROBIN\_DI ブロックと ROBIN\_DO ブロックは使用できず、ROBIN\_EXT\_TRIG\_START ブロックの *IO 種別* を DI (DI モード) に設定することもできません。

本ブロックと ROBIN\_EXT\_TRIG\_START ブロックの両方の条件を満たす外部制御信号入力が行われると ARM プログラムを制御できなくなるため、併用時は同時に条件を満たさないようご注意ください

**【 使用例 】**

サンプルモデル sineWave\_ext.slx での使用例です。

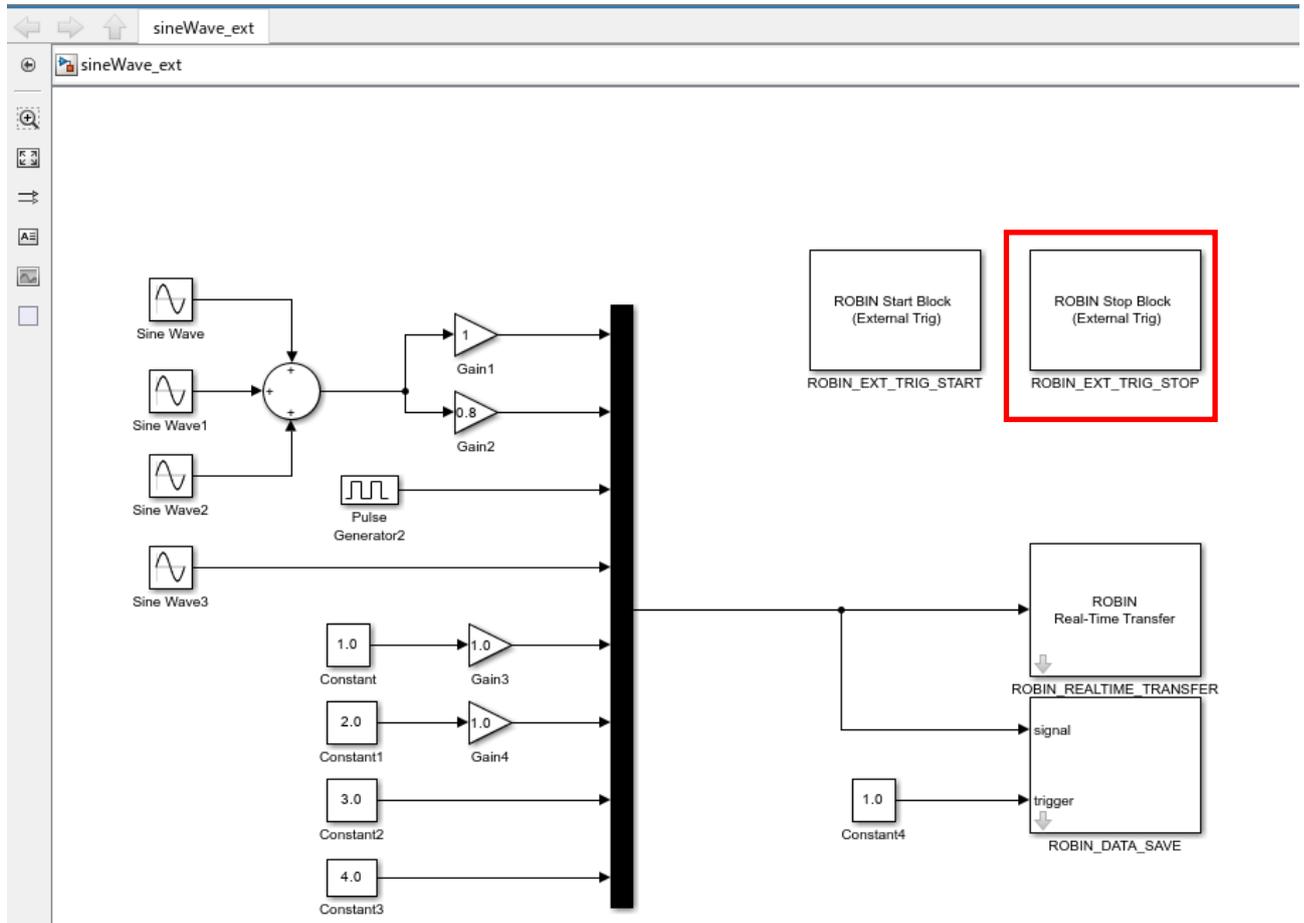


図 3-6 5 ROBINEXT\_TRIG\_STOP 使用例

## 4. MATLAB/SimulinkでのARMプログラム作成

本章では、ROBIN ライブラリを使用した Simulink モデルから ARM プログラムを作成する手順を示します。

### 4-1 Simulinkモデルの作成

#### 4-1-1 ROBINライブラリの起動

MATLABを起動し、MATLABコマンドウィンドウに”passRobinWpf”と入力してROBINライブラリを呼び出します。

ROBINライブラリはMATLAB R2022b以降でのみ使用できる仕様としているため、MATLAB R2022a以前では起動できません。

また、32bit版のMATLABでは使用できません。

MathWorks社はMATLAB/Simulinkのバージョンアップ時に予告なく仕様の大きな変更を行う場合があるため、弊社で動作を保証しているバージョンよりも新しいMATLABでの動作保証はいたしかねます。

ROBINライブラリの詳細については「3. Pass/ROBINライブラリブロック仕様」を参照ください。

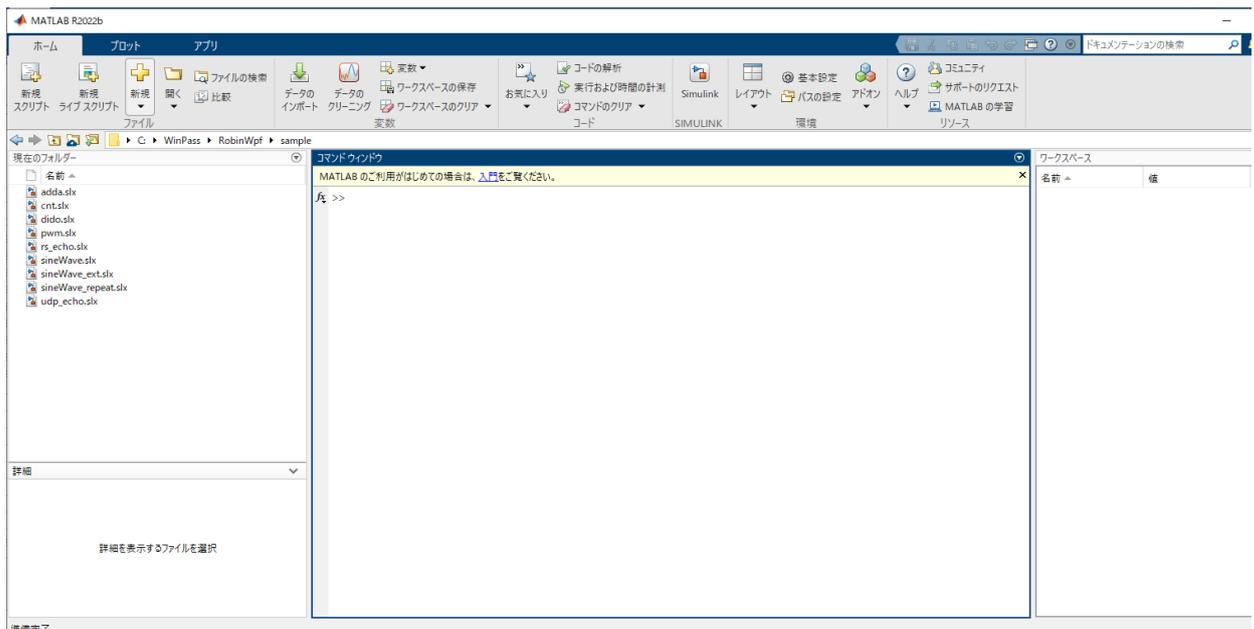


図 4-1 MATLAB画面

#### 4-1-2 Simulinkモデルの作成、保存

Simulinkライブラリブラウザで「ファイル」メニューから「新規作成」→「モデル」を選択し、新規Simulinkモデルを開きます。

SimulinkモデルにROBINライブラリおよびSimulinkライブラリのブロック配置と結線を行います。図 4-2 に作成例を示します。

Pass/ROBINのsampleフォルダ内にROBINライブラリを使用した簡単なサンプルSimulinkモデルを用意していますのでご活用ください。

Simulinkモデルの保存時に、SLX形式とMDL形式を選択できますが、どちらの形式で保存しても問題ありません。

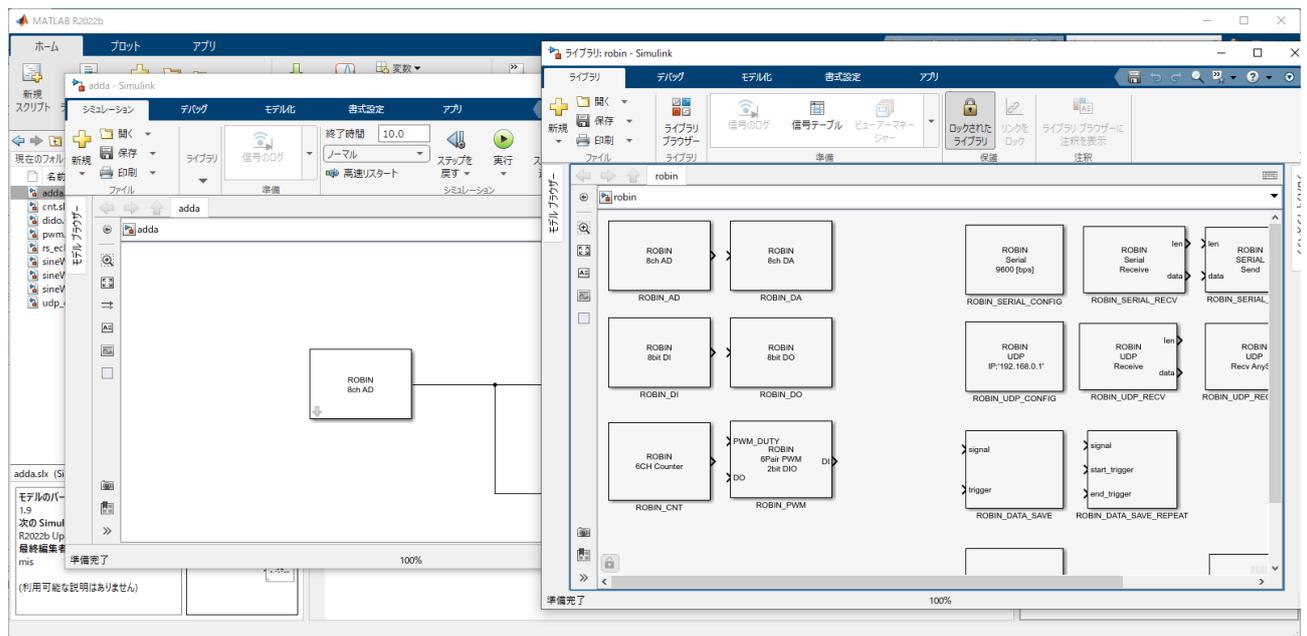


図 4-2 Simulinkモデルの例

## 4-2 Cコード生成、ARMプログラムビルド

### 4-2-1 システムターゲットファイルの選択

ROBIN用ARMプログラムを作成するための設定変更を行います。

Simulinkモデルの「シミュレーション」メニューから「モデルコンフィギュレーションパラメータ」をクリックします。画面左側の「選択」ツリーから「コード生成」を選択すると、**図 4-3**の画面になります。

Pass/ROBIN用ARMプログラムを作成するために、システムターゲットファイルをrobin.tlcとしてください。

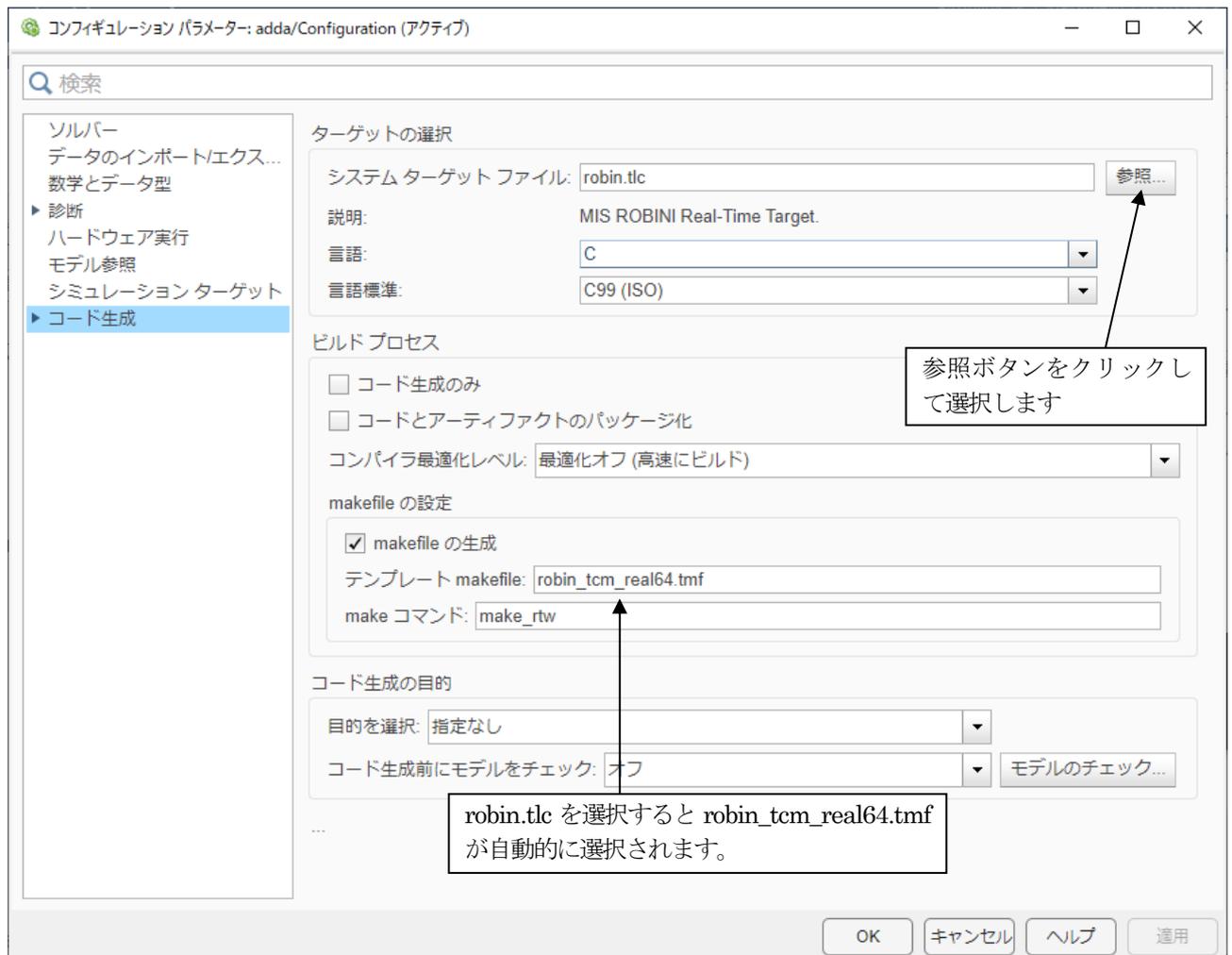


図 4-3 コード生成設定

#### 4-2-2 テンプレートmakeファイルの選択(通常モード用)

ホストアプリケーションからARMプログラムを制御する通常モード用の「テンプレートmakefile」として、表 4-1 に示すファイルを用意しています。

表 4-1 テンプレートmakeファイル(通常モード用)一覧

テンプレート makefile	プログラム格納位置	.bss 領域格納位置	データ保存領域サイズ	浮動小数点精度
robin_tcm_real32.tmf	ITCM	DTCM	256MB (DDR 全域)	単精度
robin_tcm_real64.tmf	ITCM	DTCM	256MB (DDR 全域)	倍精度
robin_ddr_real32.tmf	DDR	DTCM	128MB (DDR の 1/2)	単精度
robin_ddr_real64.tmf	DDR	DTCM	128MB (DDR の 1/2)	倍精度
robin_bss_real32.tmf	DDR	DDR	64MB (DDR の 1/4)	単精度
robin_bss_real64.tmf	DDR	DDR	64MB (DDR の 1/4)	倍精度

なお、32bit 単精度浮動小数点数を使用するテンプレートmakeファイルで作成したARMプログラムはSimulink モデルのPC上でのシミュレーション実行 (64bit 倍精度浮動小数点数) 結果との間に大きな演算誤差が発生する場合がありますため、通常は倍精度で演算を行うテンプレートmakeファイルの使用を推奨いたします。

表 4-2 テンプレートmakeファイル(通常モード用)使用目的

テンプレート makefile	使用目的
robin_tcm_real32.tmf	通常使用するテンプレート make ファイル (単精度)
robin_tcm_real64.tmf	通常使用するテンプレート make ファイル (倍精度)
robin_ddr_real32.tmf	From Workspace 等の静的メモリを多量に必要とするブロックの使用 (単精度)
robin_ddr_real64.tmf	From Workspace 等の静的メモリを多量に必要とするブロックの使用 (単精度)
robin_bss_real32.tmf	非常に多数のブロックを含み.bss 領域が DTCM に格納できないモデル
robin_bss_real64.tmf	非常に多数のブロックを含み.bss 領域が DTCM に格納できないモデル

#### 4-2-3 テンプレートmakeファイルの選択(スタンドアローンモード用)

スタンドアローンモード用の「テンプレートmakefile」として、表 4-3に示すファイルを用意しています。使用用途は\_saがつかない通常モード用のテンプレートmakeファイルと同じになります。スタンドアローンモードで保存したデータの取得を行うためには、SGL\_DATA\_SAVEブロックのトリガ待ち設定を「常時トリガ待ち」に設定する必要があることにご注意ください。

なお、スタンドアローンモードは、通常モードで開発し十分に動作検証を行ったARMプログラムをROBIN単独で動作させるモードとなりますので、スタンドアローンモードでのARMプログラム開発は行わないでください。

表 4-3 テンプレートmakeファイル(スタンドアローンモード用)一覧

テンプレート makefile	プログラム格納位置	.bss 領域格納位置	データ保存領域サイズ	浮動小数点精度
robin_tcm_real32_sa.tmf	ITCM	DTCM	256MB (DDR 全域)	単精度
robin_tcm_real64_sa.tmf	ITCM	DTCM	256MB (DDR 全域)	倍精度
robin_ddr_real32_sa.tmf	DDR	DTCM	128MB (DDR の 1/2)	単精度
robin_ddr_real64_sa.tmf	DDR	DTCM	128MB (DDR の 1/2)	倍精度
robin_bss_real32_sa.tmf	DDR	DDR	64MB (DDR の 1/4)	単精度
robin_bss_real64_sa.tmf	DDR	DDR	64MB (DDR の 1/4)	倍精度

#### 4-2-4 Simulinkモデル設定

Simulink Coderでコードを生成するためのSimulinkモデル設定を行います。

Simulinkモデルの「シミュレーション」メニューから「モデルコンフィギュレーションパラメータ」をクリックします。画面左側の「選択」ツリーから「ソルバー」を選択すると、**図 4-4**の画面になります。

ソルバーオプションタイプを固定ステップに設定します。可変ステップに設定した場合、Cコードを生成できません。

モデルの動作速度を、固定ステップ（基本サンプル時間）にサンプリングクロック周期で設定します。サンプリングクロック周期は0.000002から10.0までの範囲で設定してください。範囲外のクロック周期を指定した場合は、ARMプログラム実行時に0.01に自動で変更します。

**図 4-4**の例では0.01を設定していますので、モデルは100Hzで動作します。

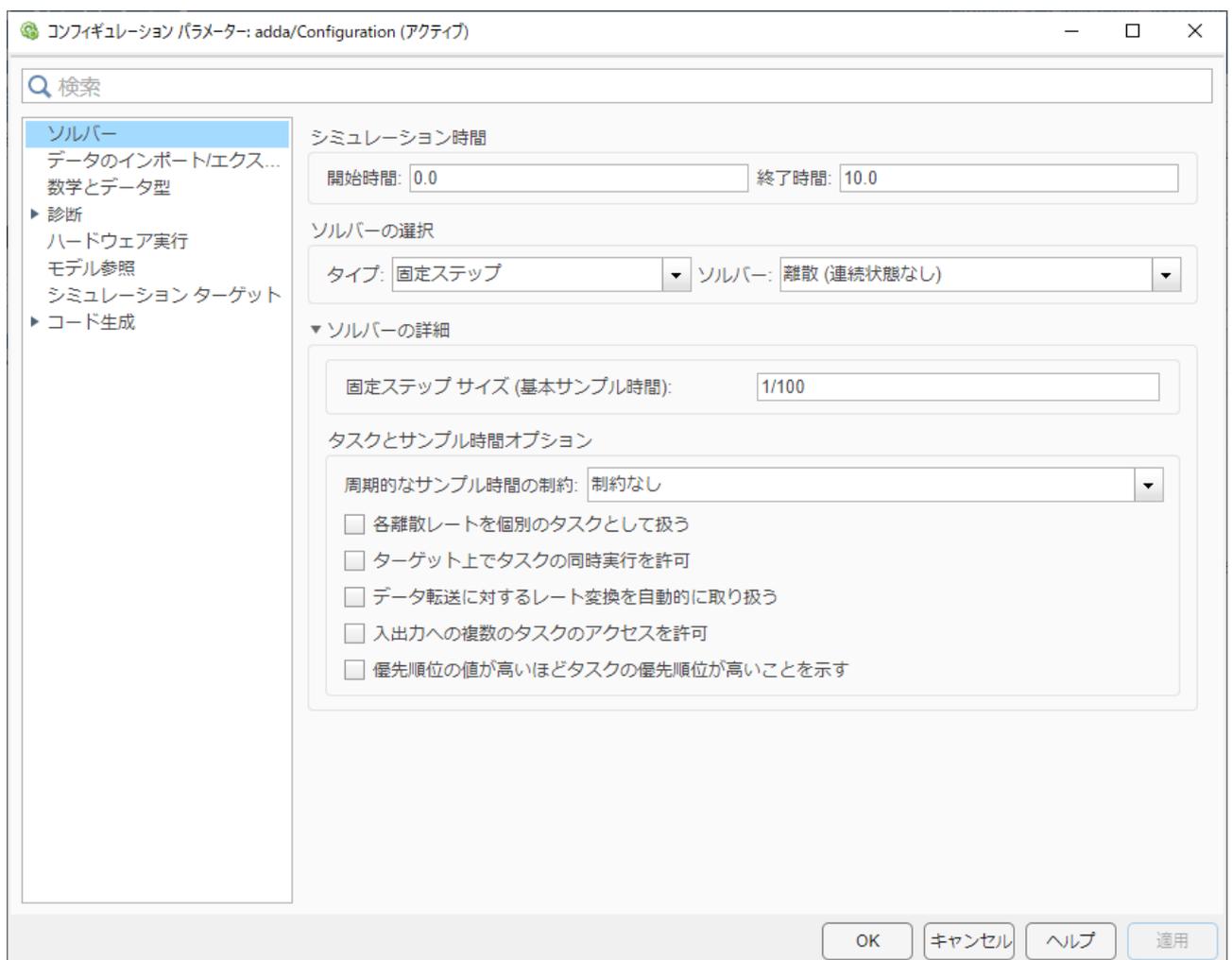


図 4-4 ソルバー設定

#### 4-2-5 実行ハードウェア設定

Simulink Coderの生成するCコードの対象ハードウェア設定を行います。

Simulinkモデルの「シミュレーション」メニューから「モデルコンフィギュレーションパラメータ」をクリックします。画面左側の「選択」ツリーから「ハードウェア実行」を選択すると、図 4-5の画面になります。

ROBINのARMプログラムは「i.MX8M Plus」マイコンのARM Cortex-Mコアで動作するため、ベンダー「ARM Compatible」、デバイスタイプ「ARM Cortex-M」に設定します。

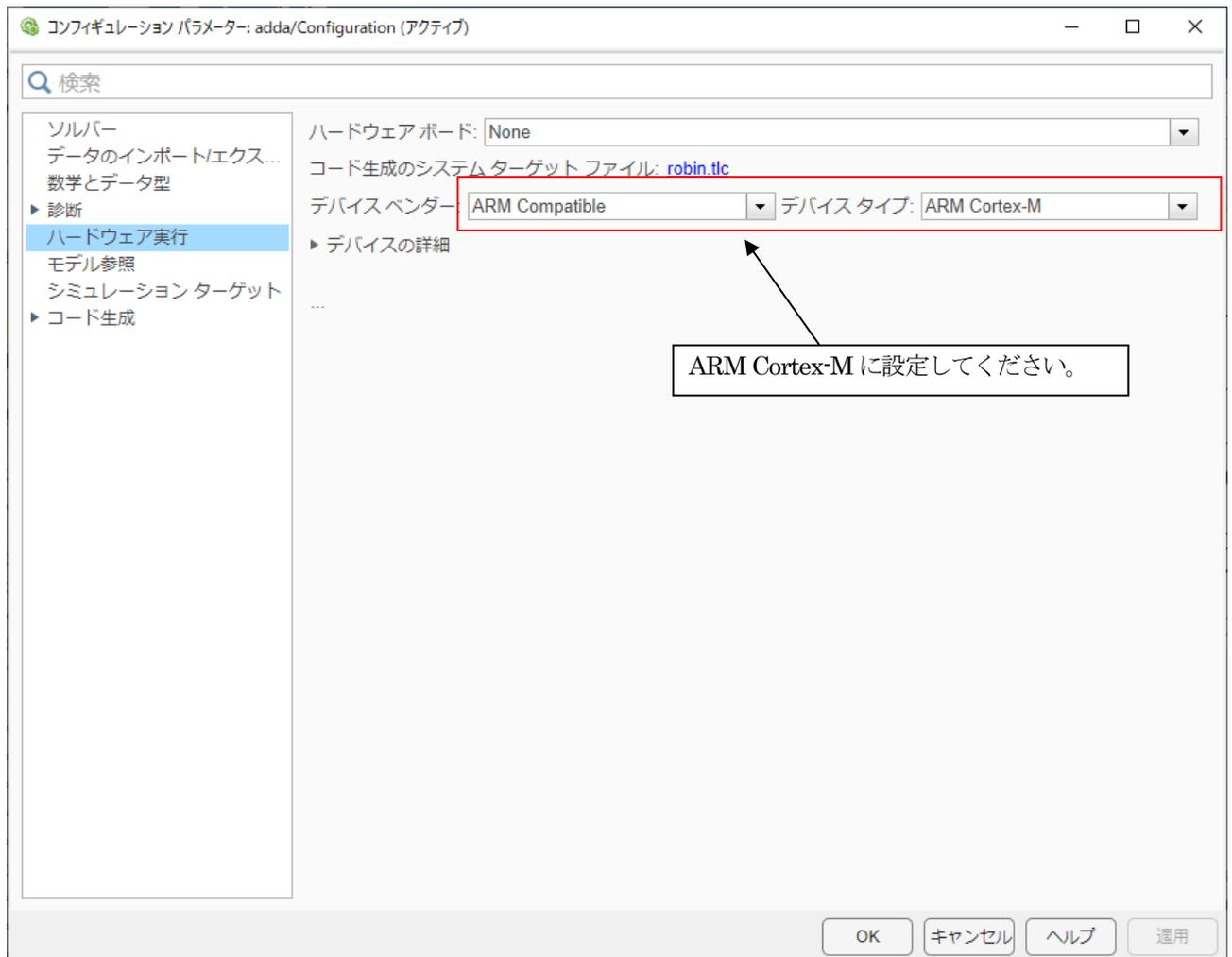


図 4-5 ハードウェア設定

#### 4-2-6 インターフェイス設定

Simulink Coderのインターフェイス設定を行います。

Simulinkモデルの「シミュレーション」メニューから「モデルコンフィギュレーションパラメータ」をクリックします。画面左側の「選択」ツリーから「コード生成」→「インターフェイス」を選択すると、**図 4-6**の画面になります。

Pass/ROBINはC APIインターフェイスを使用することを前提としていますので、「C API」の信号とパラメータを生成する設定としてください。

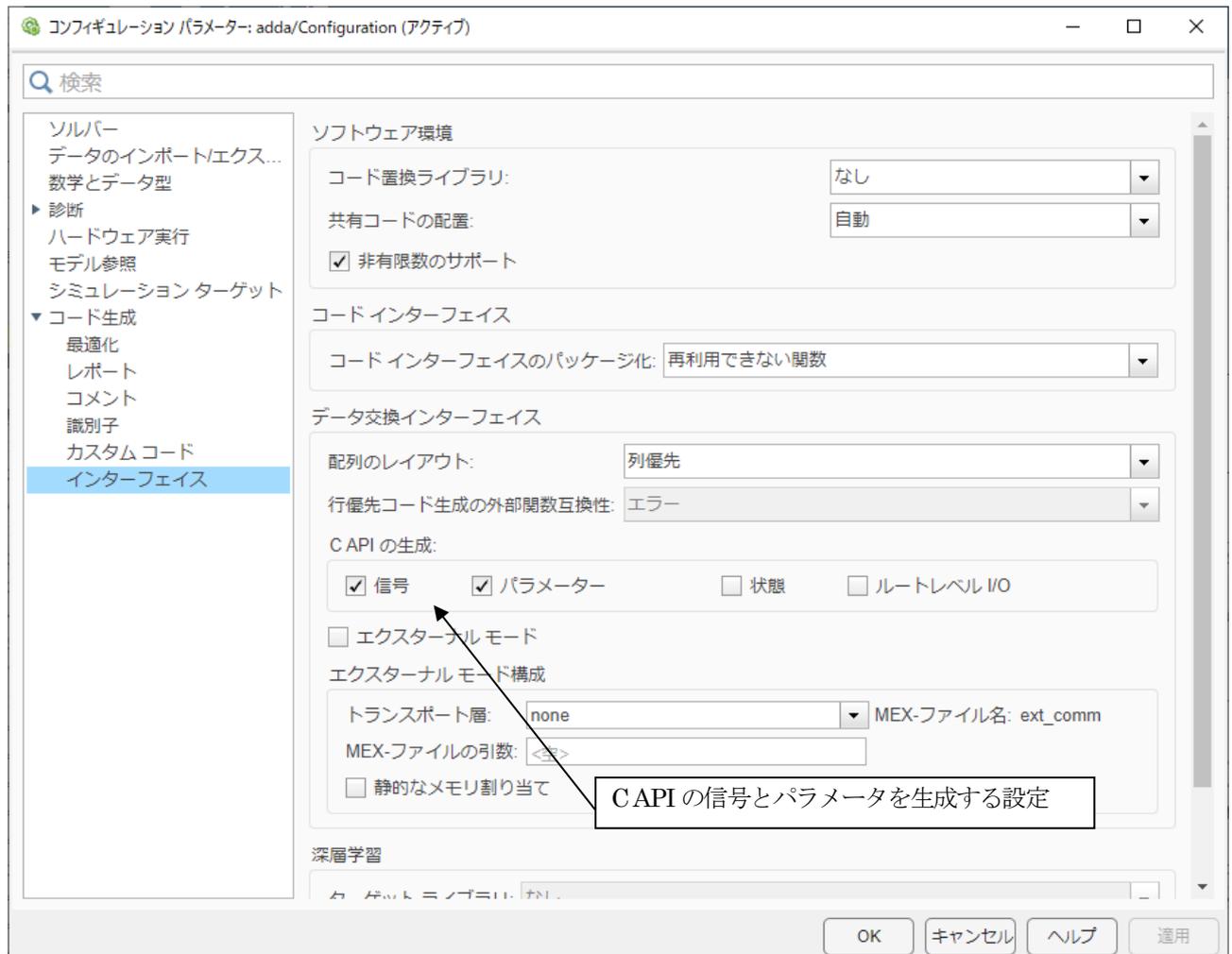


図 4-6 インターフェイス設定

## 4-2-7 ARMプログラム作成

Simulink Coder による C コード生成と、ARM コンパイラによる ARM プログラム作成を行います。

Simulinkモデルで「Ctrl+B」ショートカットキーを使用すると、Simulinkモデルからのビルド（Cコード生成とARMコンパイラによるコンパイル、リンク）を自動的にを行います。

ショートカットキーを使用しない場合は、Simulinkモデル上部の「アプリ」タブで「Simulink Coder」をクリックして追加される「Cコード」タブで「ビルド」ボタンをクリックします。

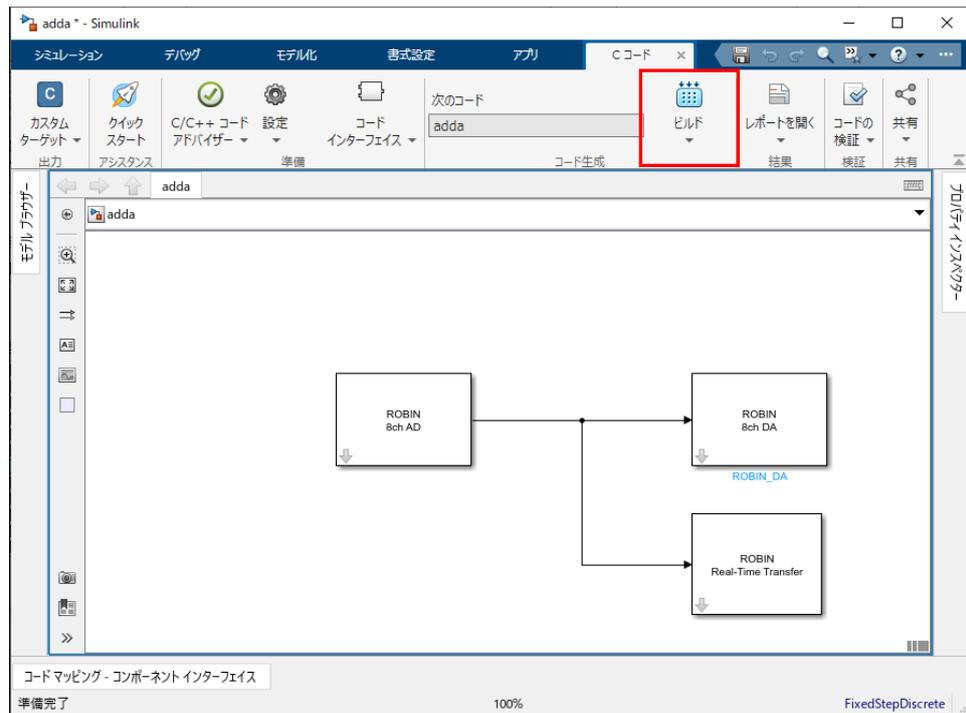


図 4-7 ビルドボタン

ビルド結果はSimulinkモデルの「ツール表示」メニューから開く「診断ビューアー」に表示します。ビルド成功時の表示例を図 4-8 に示します。

Simulinkの仕様上、成功の表示が「ビルド概要」で行われないことにご注意ください。

Pass/ROBINが作成するARMプログラムファイル名は以下となります。

モデル名\_robin\_rtw¥モデル名.elf

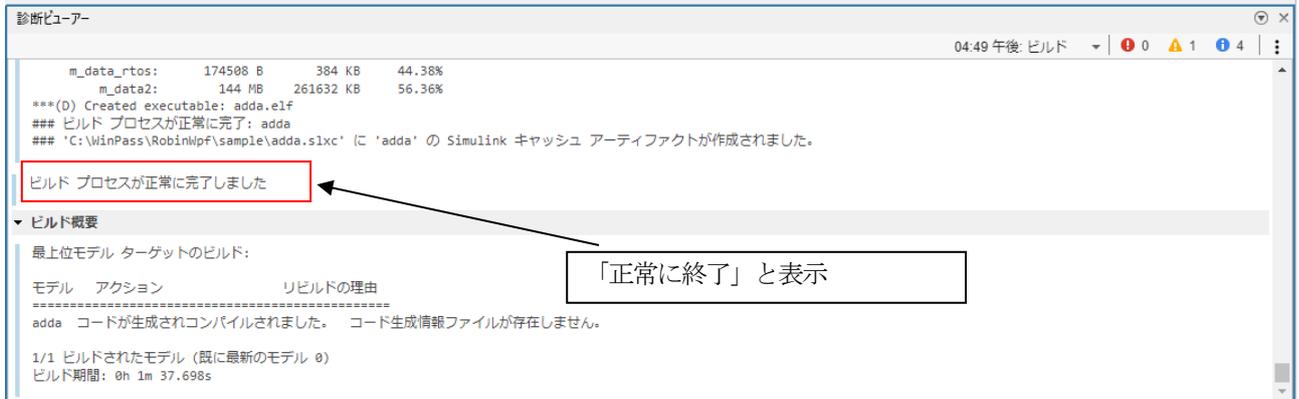


図 4-8 ビルド成功時表示例

なお、Simulinkモデル内でデータフローの先頭となるブロックにサンプリング周期を明示的に設定していない場合下記のワーニングを表示しますが、これは無視しても問題ありません。



図 4-9 サンプル時間ワーニング