モデルI/Fガイドライン
開発経過報告書

令和３年度「無人自動運転等の先進ＭＡＡＳ実装 加速化推進事業費補助金」成果報告書

日付 2022.03.07 （目次誤記訂正 2022.04.05）
ルネサスエレクトロニクス株式会社

目次

1. 背景

1. 背景

現代の自動車開発では、「自動運転」、「コネクティッド」、「コックピット連携」など、「走る・曲がる・止まる」以外の開発が必要となっている。このことより、機能の複雑化によって検証が肥大化するなどの課題が生じている。
解決のため、開発プロセスを跨いだ連携が求められており、経済産業省にて「自動車産業におけるモデルのあり方に関する研究会」が発足した。これまでに、開発プロセス間のモデル連携とモデルベース開発の今後の活用を考慮した、モデル流通のための『プラントモデルI/Fガイドライン』や『ガイドライン準拠モデル』が公開されている。しかしながら、制御SW開発を対象したモデル流通は議論されていない。

一方、ルネサスエレクトロニクスでは、OEMが求めるSWの仮想開発環境を実現するため、ゲートウェイデバイス（CoGW）、自動運転デバイスのためのモデルベース開発環境構築、およびその環境を活用したアプリケーションSWの開発・評価を進めている。

ただし、経済産業省で議論されているモデルとルネサスエレクトロニクスが想定するモデルでは抽象度が異なる。

したがって、それぞれを連携するための I/F の定義、および I/F を活用するためのガイドラインが必要となった。

具体的には、制御の機能・ソフトウェア・ハードウェア全体を『制御装置』とおき、開発上流から自動車の制御装置がどのように連携されていくか想定した上で、『それぞれの開発段階で抽象度の違うモデル同士を接続する際に、どのような I/F を設定すれば連携できるか』を見える化することが求められる。

2. 目的

2. 目的

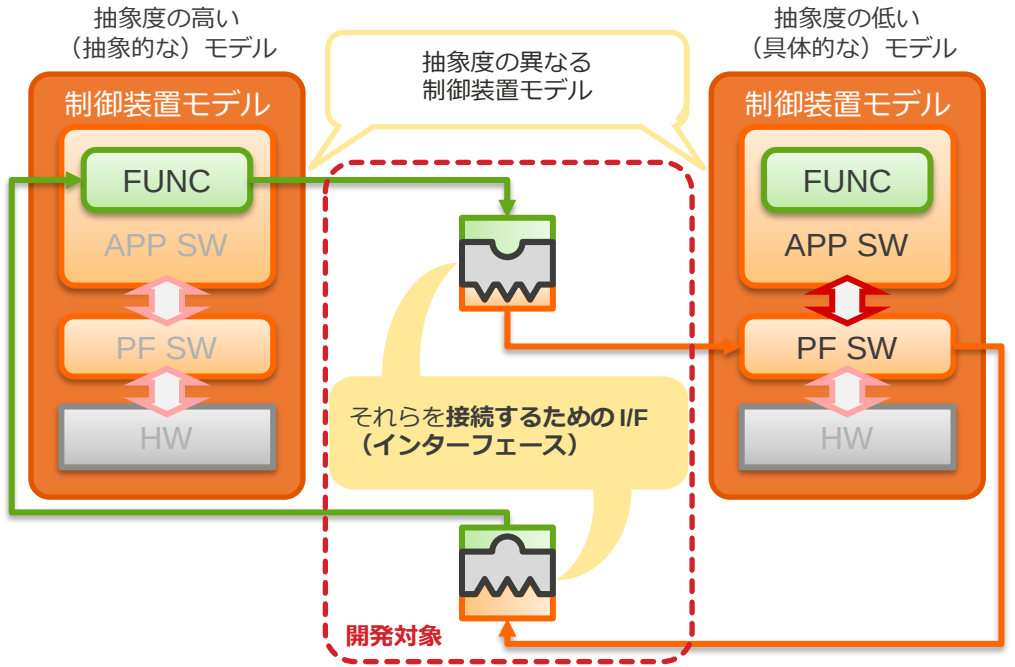
- **自動車産業におけるモデル流通を促進するため、抽象度の異なるモデルを接続する際に用いる I/F を定義し、I/F を活用するためのガイドラインを開発する。**

– 実施事項

CoGWおよび自動運転におけるユースケースを想定したアプリケーションSWをMBD環境上で実行することでガイドラインの有効性を確認する。

– 前提条件

交付規程の補助対象要件(1)②に適合するモデル（METIガイドライン準拠モデル）を、
前述のMBD環境に接続する外部モデルの一部として使用する。



3. ガイドラインの構成・対象

3. ガイドラインの構成・対象

- 制御開発プロセスの各設計フェーズ間で抽象度の異なるモデルを接続することに関して、プロセス範囲の設定、モデル種類の定義、モデル接続方法とその方法を実際に用いた事例の提示を行う。

－本ガイドラインの構成

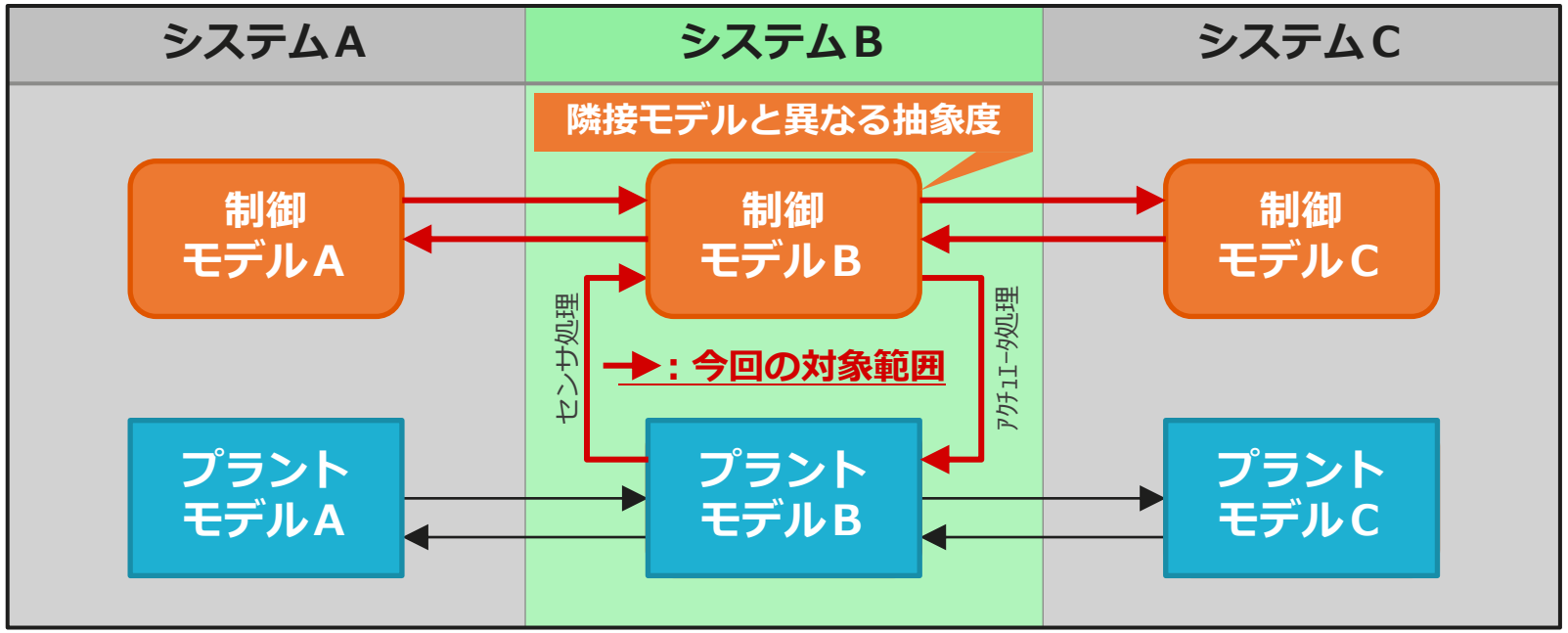
本ガイドラインの目的にしたがって、以下の構成で MILS⇔SILS⇔SPILS を通じたモデル接続に関して解説を行う。

制御開発プロセスと 設計フェーズの想定	第４章：ガイドラインの定義 －制御開発プロセスの想定
制御装置モデルの定義 （抽象度の異なるモデルの定義）	第４章：ガイドラインの定義 －制御装置モデルの定義／－制御装置モデルに対する抽象度の定義
モデル接続方法の提示	第５章：I/F項目の設定方法
事例によるモデル接続の解説	第６章：モデル接続の具体事例

3. ガイドラインの構成・対象

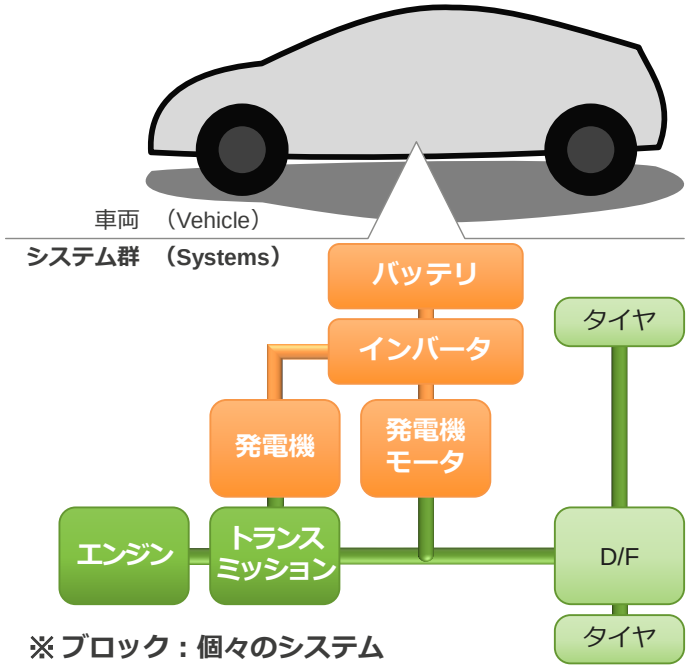
- 制御開発において、開発対象システムのモデルは制御モデルとプラントモデルから成り立つ。
- 本ガイドラインでは、異なる抽象度の制御モデル間における、入出力の接続を対象とする。

–システム群の構造からみた対象範囲



（補足）システム

- 本書では、自動車を構成する最上位の部品をシステムとする。



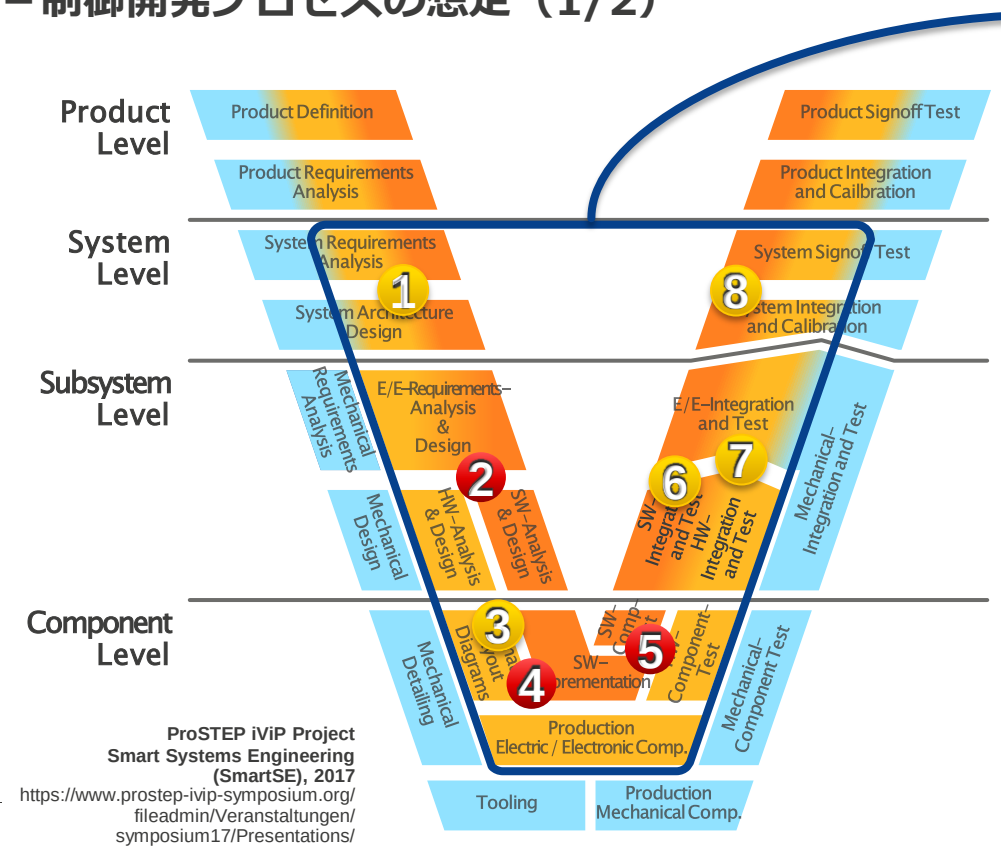
※ブロック：個々のシステム

4.　ガイドラインの定義

4. ガイドラインの定義

- 本ガイドラインでは、下図のソフトウェア開発（①⇒⑧）のうち、**②制御機能開発**、**④SW設計とコード生成**、**⑤SILS/SPILS検証** で扱われる、モデル接続を取り扱う。

－制御開発プロセスの想定（1/2）



（参考）
ProSTEP iViP Project Smart Systems Engineering (SmartSE), 2017
URL: https://www.prostep-ivip-symposium.org/fileadmin/Veranstaltungen/symposium17/Presentations/Presentation_Rude-BMW_Nalbant-PROSTEP.pdf

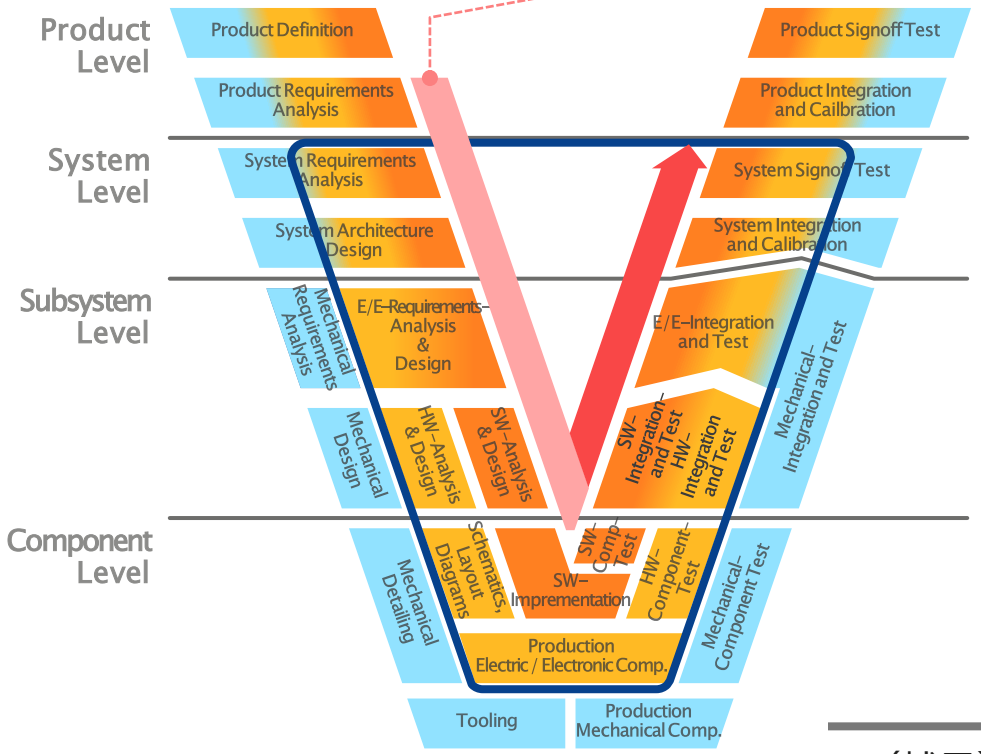
ソフトウェア開発

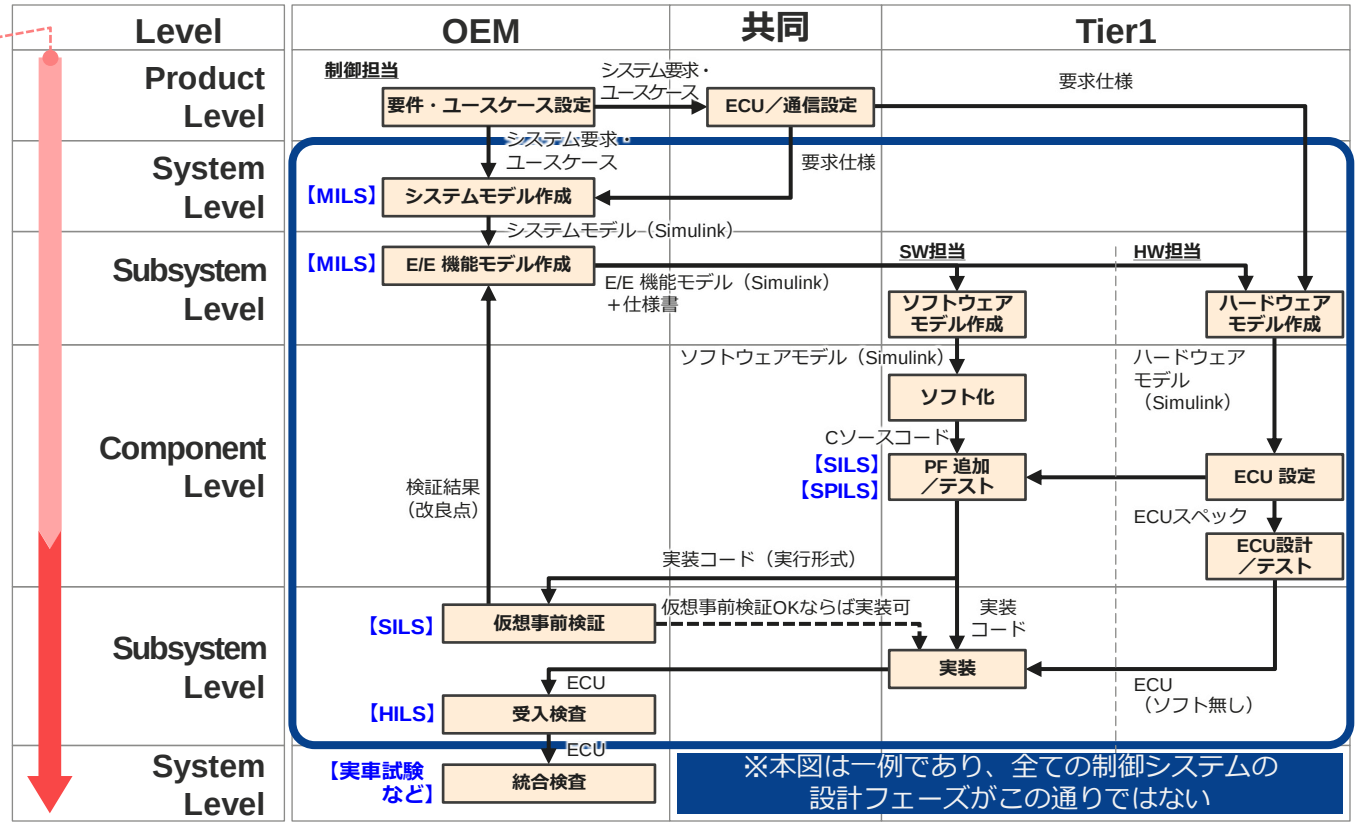
	モデル抽象度が異なる場合の事例
① System Modeling & Analysis	EVの電力消費量のような車両全体の機能・性能について、複数の設計案を定量的に分析して検討する。
② **Function / Controller Design & Simulation**	車両全体の機能・性能要件の実現に向け、モータやバッテリなどのシステムの詳細機能とSW・HW等への割当を検討し、仕様を確定する。
③ Functional Evaluation with RCP	エンジンなどの既存システムについて、制御機能の追加・変更の際、制御モデルを実時間駆動させて、実物に対する動作の検討をする。
④ **Software-Design & Code Generation**	制御ECU上で動作するSWを開発するために、要件定義となる上位モデルからSWを自動生成する。また、HWに依存する部分や詳細化の必要な部分については、追加のコーディングをする。
⑤ **SILS / SPILS (PILS)**	モータやバッテリなどの各制御ユニット向けのオブジェクトファイルを、実装前に仮想検証して、仕様に合った振る舞いであるか検討する。
⑥ Virtual “non-realtime” ECU Validation	機能レベルの動作が確保されていることをシミュレータ及び実機で確認する。
⑦ Virtual “realtime” ECU Validation	機能に加えてタイミングが確保されていることをシミュレータ及び実機で確認する。
⑧ HILS	試作された複数の制御ユニットについて、例えばハイブリッド制御のような連携制御の検証を、実車両の完成を待たずに実施する。

4. ガイドラインの定義

- 制御開発における各設計フェーズを例示すると、下図のように表現可能である。
  制御装置の設計が具体化されることに応じて、検討に使用される制御モデルの抽象度も変化する。

– 制御開発プロセスの想定（2/2）



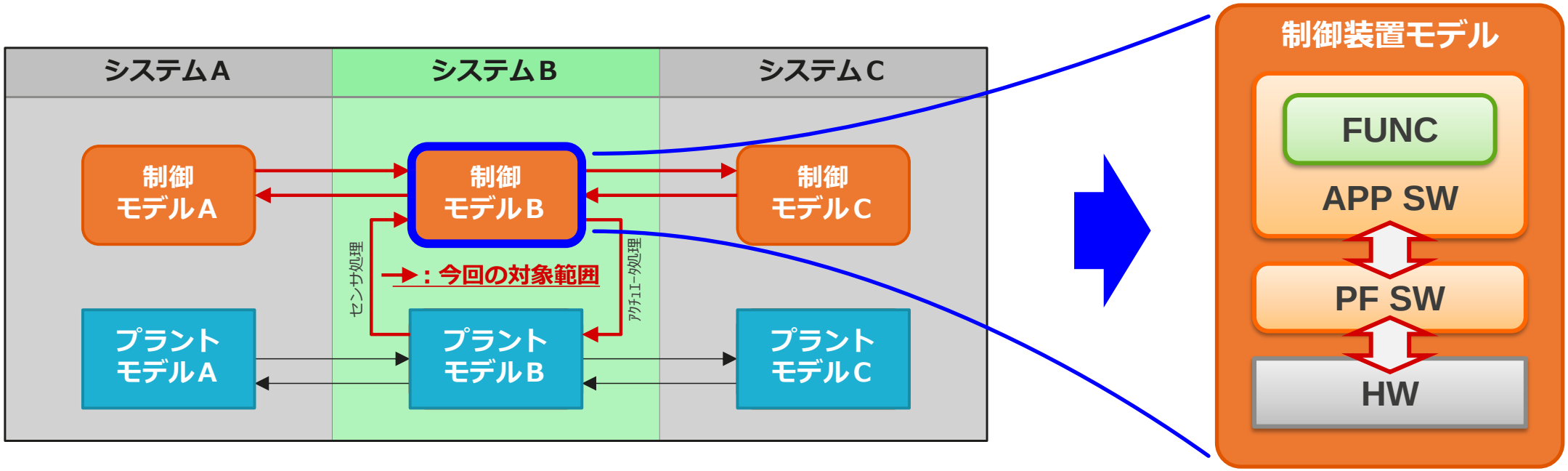


- （補足）ガイドライン対象者：枠内業務に携わる制御系開発者やソフトウェア開発者

4. ガイドラインの定義

- 本ガイドラインでは、開発プロセスの各設計フェーズ間で抽象度の異なるモデルの接続を考慮するため、『制御モデル』を『制御装置モデル』に再定義する。

－制御装置モデルの定義（1/2）



- 制御装置モデルでは、その内部を『FUNC』、『APP SW』、『PF SW』、『HW』に分割する。
- 左図の制御モデルを機能設計時の最も抽象度の高いモデルとすると、右図の制御装置モデル内『FUNC』に相当する。

4. ガイドラインの定義

- 制御装置モデルは、実際の制御装置（ECU）の構造を参照してモデル構造を定義する。

– 制御装置モデルの定義（2/2）

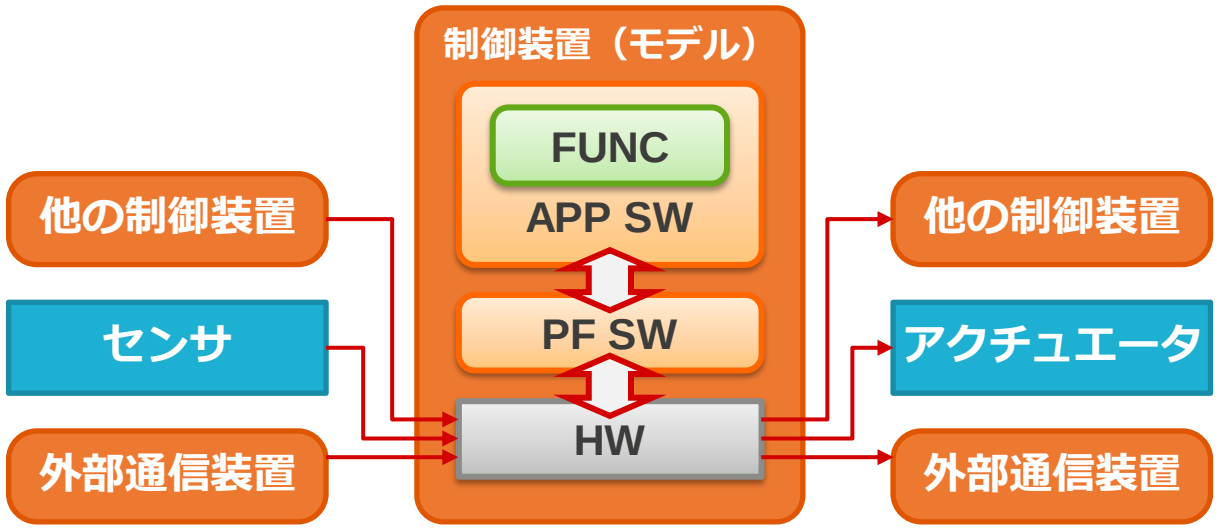
制御装置は、他の制御装置（ECU）との通信やセンサーアクチュエータ、コネクティッドなどの外部装置と電気的に接続されている。

また、内部ではそれぞれのソフトウェアの処理によって制御機能を実現している。

ここで、
制御装置の構成を電気的に処理するHW部（HW）、ソフトウェア的に処理するアプリケーション層（APP SW）およびプラットフォーム層（PF SW）として定義する。

- **制御装置モデルと外部のI/F項目の設定：**

  対象の制御装置モデルにおいて、
  シミュレーション実行時に必要な情報をやり取りするためのI/F項目を設定する。

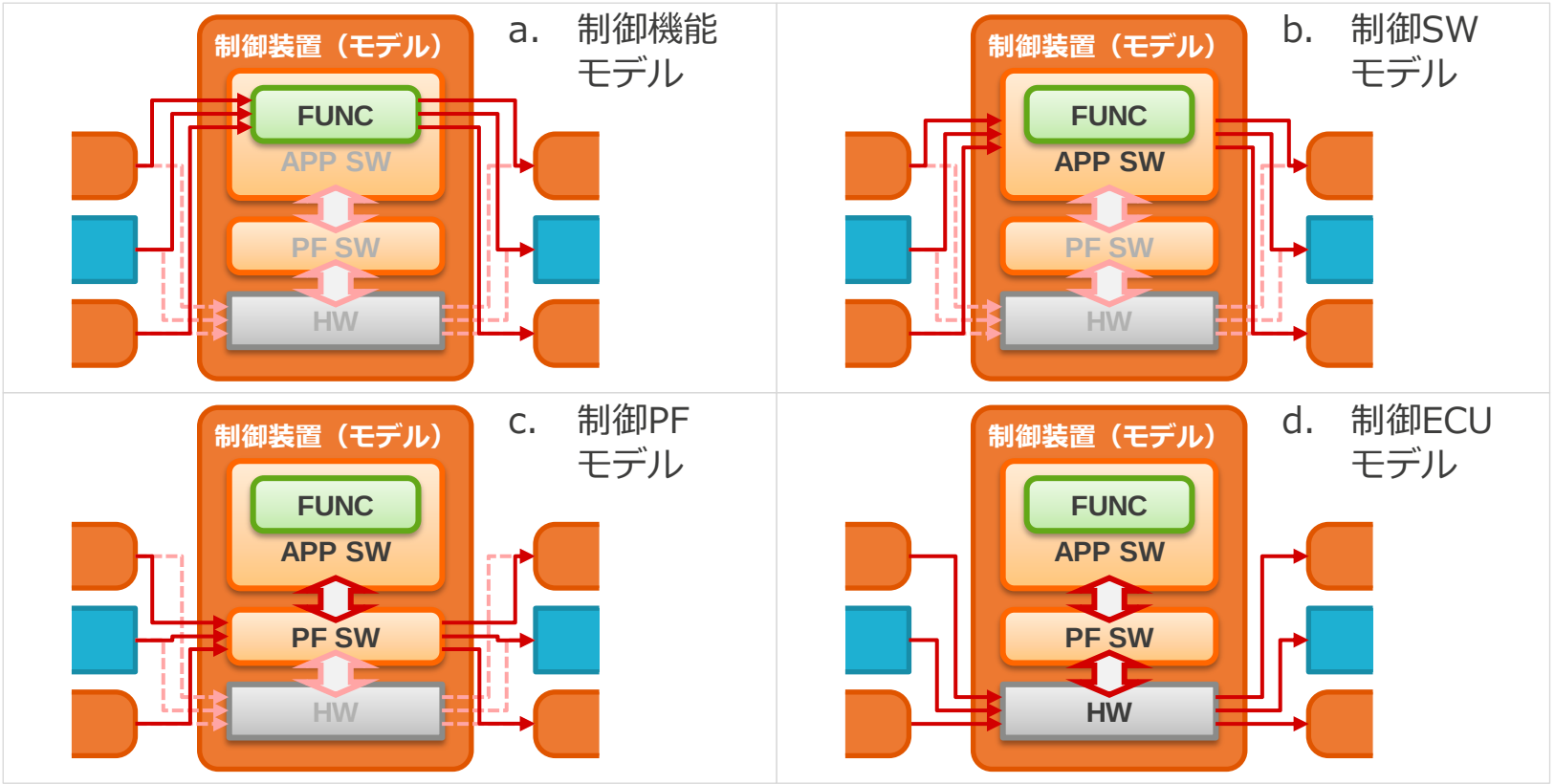


4. ガイドラインの定義

- 制御装置モデルに対して、用途に応じた異なる抽象度を定義する。

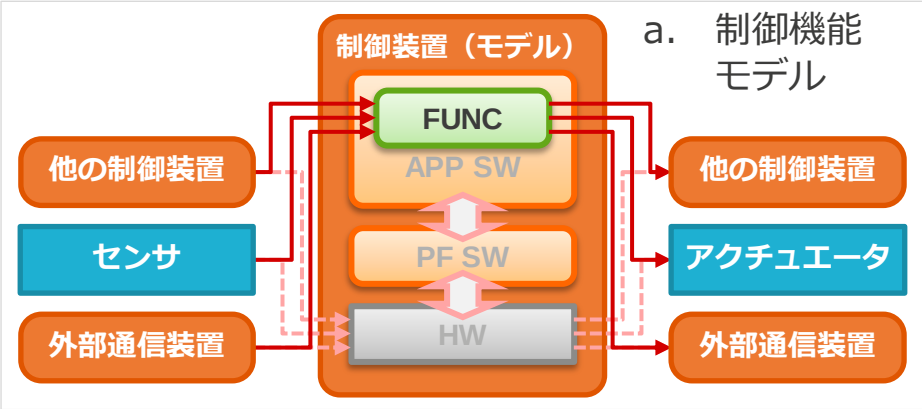
－制御装置モデルに対する抽象度の定義

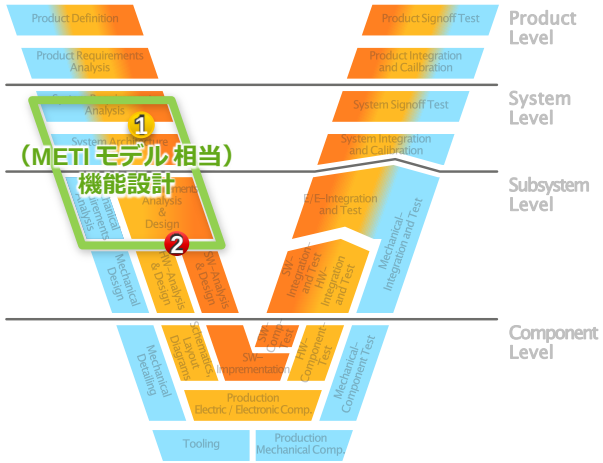
- 制御装置モデルに４つの抽象度
  （a. 制御機能モデル
  b. 制御SWモデル
  c. 制御PFモデル
  d. 制御ECUモデル）
  を定義する。
- また、本ガイドラインで想定する開発プロセス範囲では、
  a. 制御機能モデル
  ～ c. 制御PFモデル
  までが対象となる。



4. ガイドラインの定義

– 制御装置モデルに対する抽象度の定義（補足：1/4） | **– a. 制御機能モデル**における制御装置モデル





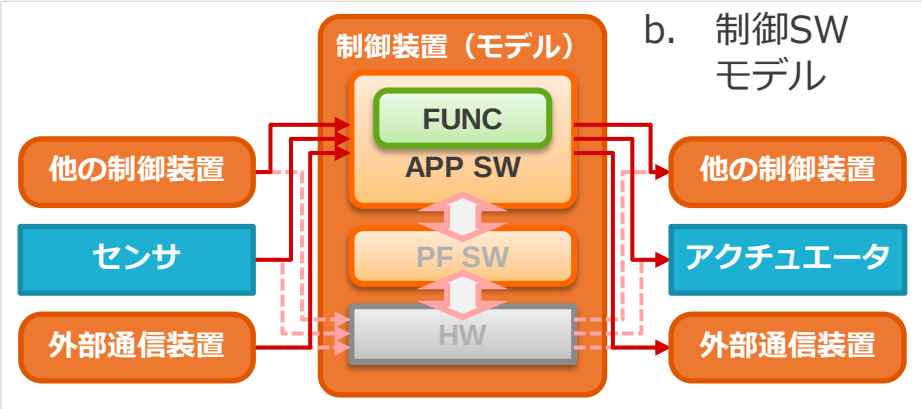
- **ブロック図を使用して表現した制御機能を、実行可能としたモデル。**
- 「FUNC」は、**機能検討の際**には主に、**連続量**で取り扱われる。
  また、**制御論理を検討する際**にはコンピュータ実装を考慮して**離散量**で扱われる。
- 「APP SW」、「PF SW」、「HW」は省略もしくは極端に簡略化する。

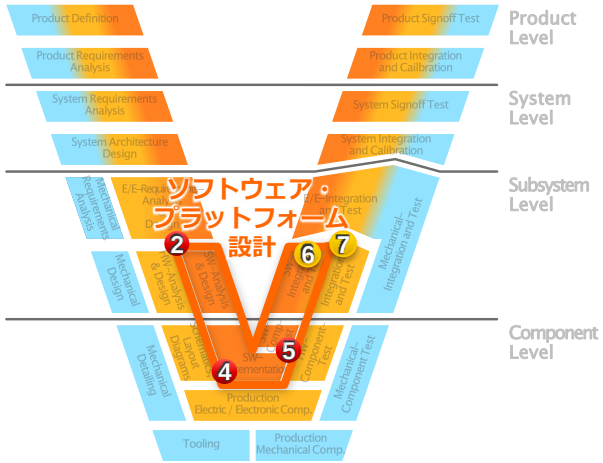
入出力I/F：

– 制御装置モデルは、その外部の機器に対して物理値・論理値をやりとりする。
– 物理値・論理値：人が見て理解できる値。

4. ガイドラインの定義

–制御装置モデルに対する抽象度の定義（補足：2/4）｜–b. 制御SWモデルにおける制御装置モデル





- **ソースコードのアプリケーション部分をシミュレータ上で実行可能としたモデル。**
- 「HW」は含まれず、「APP SW」のモデルを介して「FUNC」とやりとりする。
- 「PF SW」はPCに準備された汎用的なものを使用する。

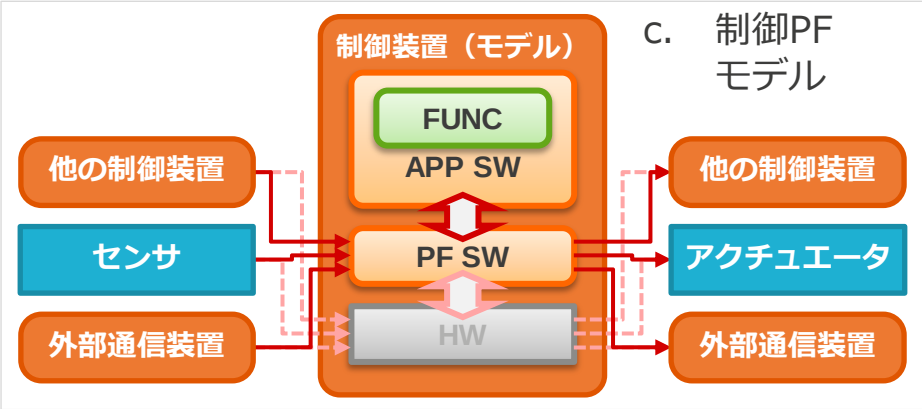
入出力I/F：

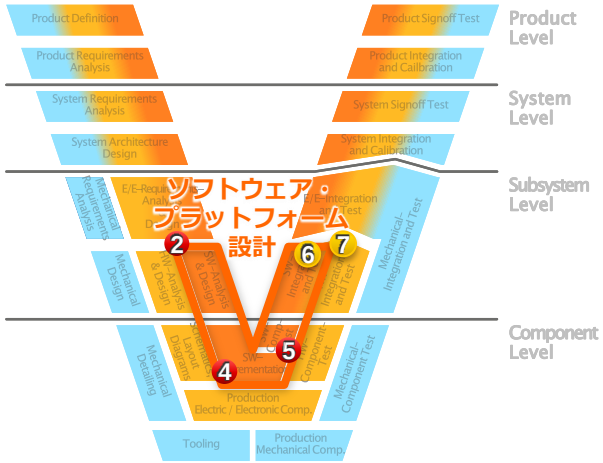
– 制御装置モデルは、その外部の機器に対してSW値をやりとりする。

– SW値：ECU内部に存在するCPUが解釈する値。物理値をバイナリデータに変換したもの。

4. ガイドラインの定義

－制御装置モデルに対する抽象度の定義（補足：3/4）｜－c. 制御PFモデルにおける制御装置モデル





- **ソースコードのアプリケーション部分に加え、使用するマイクロコントローラ（μC）や各種デバイスに合わせたプラットフォーム部分も含めてシミュレータ上で実行可能としたモデル。**
- 「HW」は含まれず、「PF SW」と「APP SW」のモデルを介して「FUNC」とやりとりをする。
- 「PF SW」には、組み込みμCのCPUコア向けにコンパイルした実行形式ファイルを使用する。

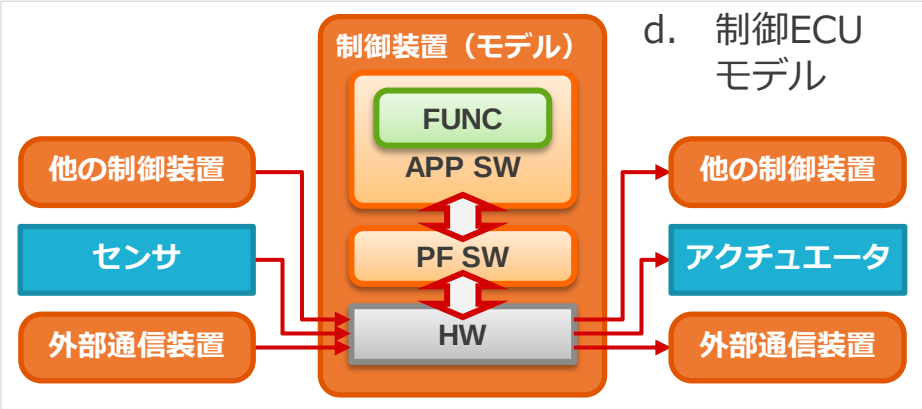
入出力I/F：

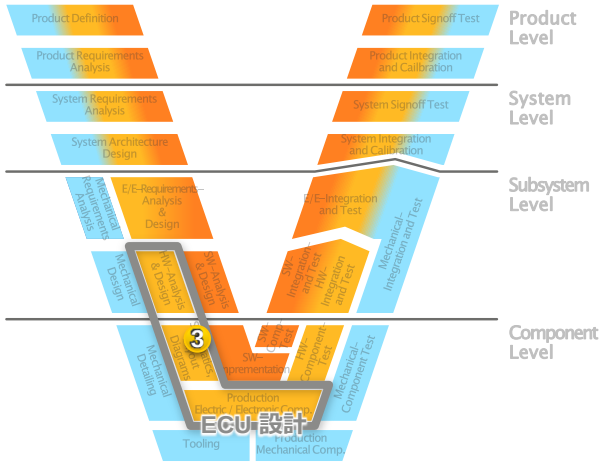
– 制御装置モデルは、その外部の機器に対してPF値をやりとりする。

– PF値：SW値にPFで使用するを追加した値。

4. ガイドラインの定義

－制御装置モデルに対する抽象度の定義（補足：4/4）｜－d. 制御ECUモデルにおける制御装置モデル





- **マイコン用にコンパイルしたソフトウェア全体の実装コードをシミュレータ上で実行可能としたモデル。**
- 全てのモデルを含み、マイコン全体をシミュレートする。

入出力I/F：

- 制御装置モデルは、その外部の機器に対して電気値をやりとりする。
- 電気値：ECUに使用されるハードウェアが解釈する値。PF値を電気信号に変換した値。

5.　I/F項目の設定方法

5. I/F項目の設定方法

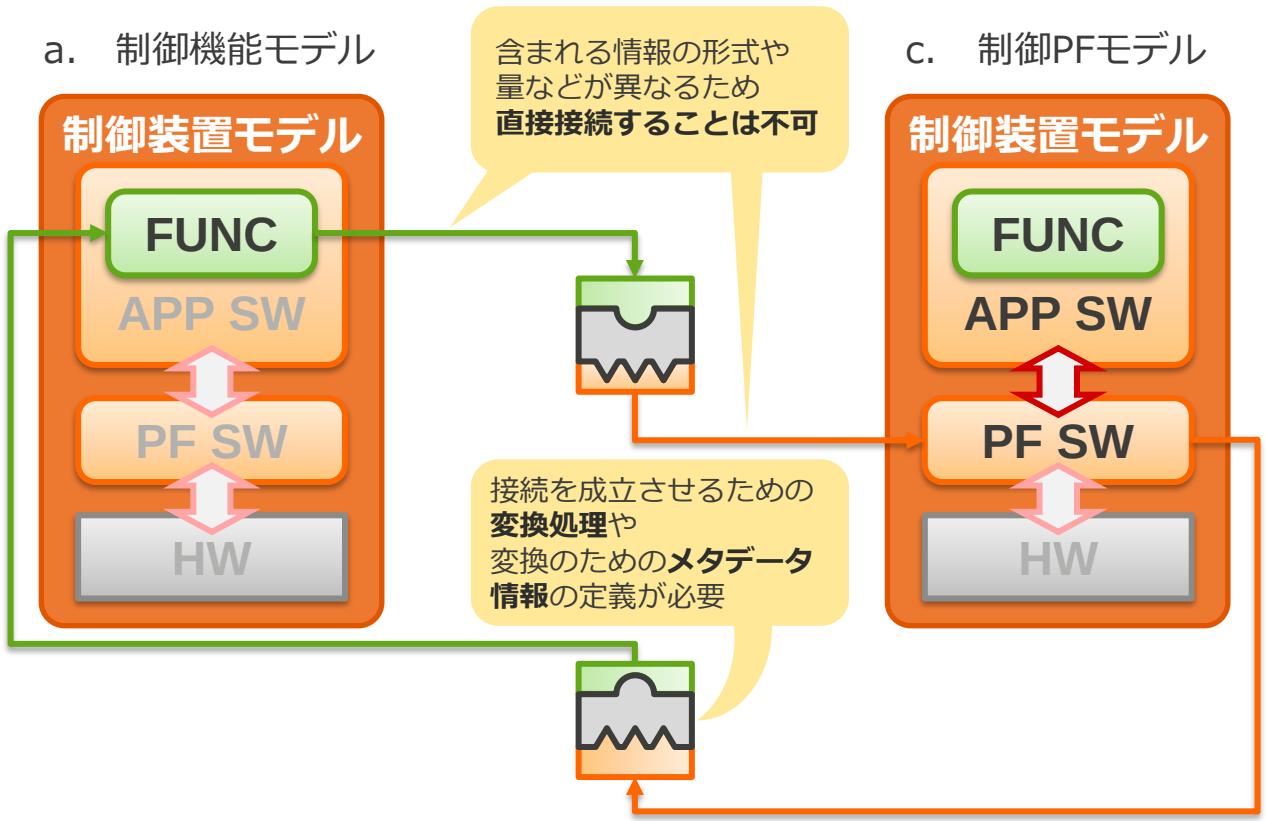
- 各制御開発プロセス間で扱われる異なる抽象度のモデルを接続する場合、信号値の受け渡しには変換や付加を要する。これに際し、各抽象度モデルが扱う信号のメタデータをあらかじめ設定する必要がある。

－異なる抽象度のモデル接続に関する留意点（1/4）

制御装置モデルのI/F項目について

- モデルのI/Fは、モデルの入出力信号である。
- よって、その信号値が何を意味するか定義された情報（＝メタデータ）がなければ、信号値を理解したり、異なる抽象度のモデルを接続する際に信号値の変換が不可能となる。
- モデルのI/Fを理解したり、モデルを接続させるためには、それぞれのモデル抽象度に合致したメタデータ項目を設定する必要がある。

例）制御機能モデルと制御PFモデル間の接続



5. I/F項目の設定方法

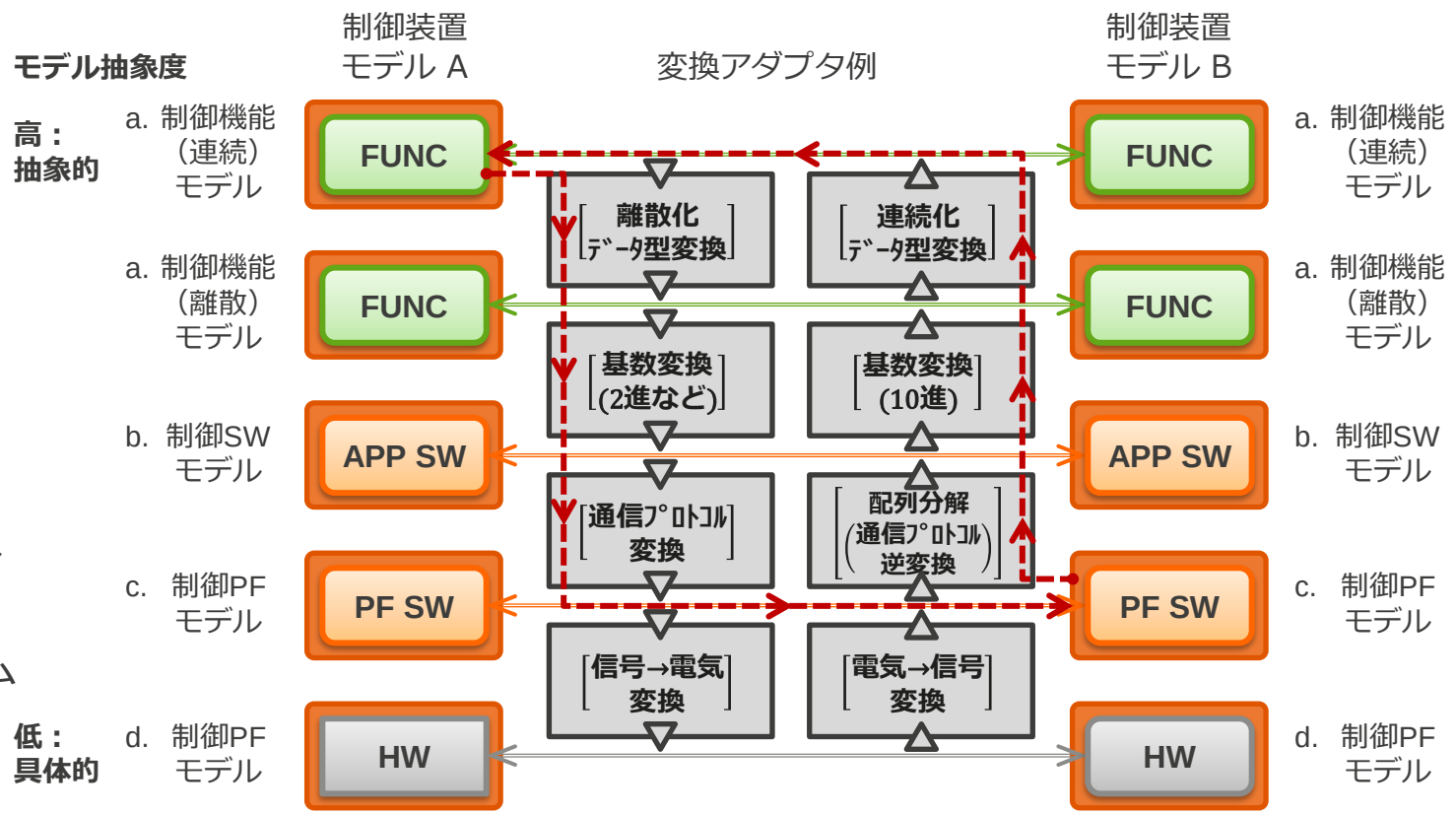
- さらに具体的な表現として、異なる抽象度のモデルを接続することに際し、下図に例示されるようなデータ変換アダプタの考慮と抽象度に応じた選択を要する。

–異なる抽象度のモデル接続に関する留意点（2/4）

モデル接続のための変換アダプタについて

- モデル接続のための入出力信号の変換について、本ガイドラインでは、抽象度違いのモデル同士の間に挿入して作用する、データ変換アダプタを提案する。
- 抽象度の違いは右図のようなモデル同士の差分で表現できる。差分解消のための入出力変換は、事前に接続する各モデルの抽象度を比較し、接続が成立する変換アダプタを選択する。
- 右図では、 一つの例としてプラットフォーム設計の際に考慮される通信仕様に合わせ、赤点線の順に変換アダプタを選択している。

例）制御機能（連続）モデルと制御PFモデル間の接続



5. I/F項目の設定方法

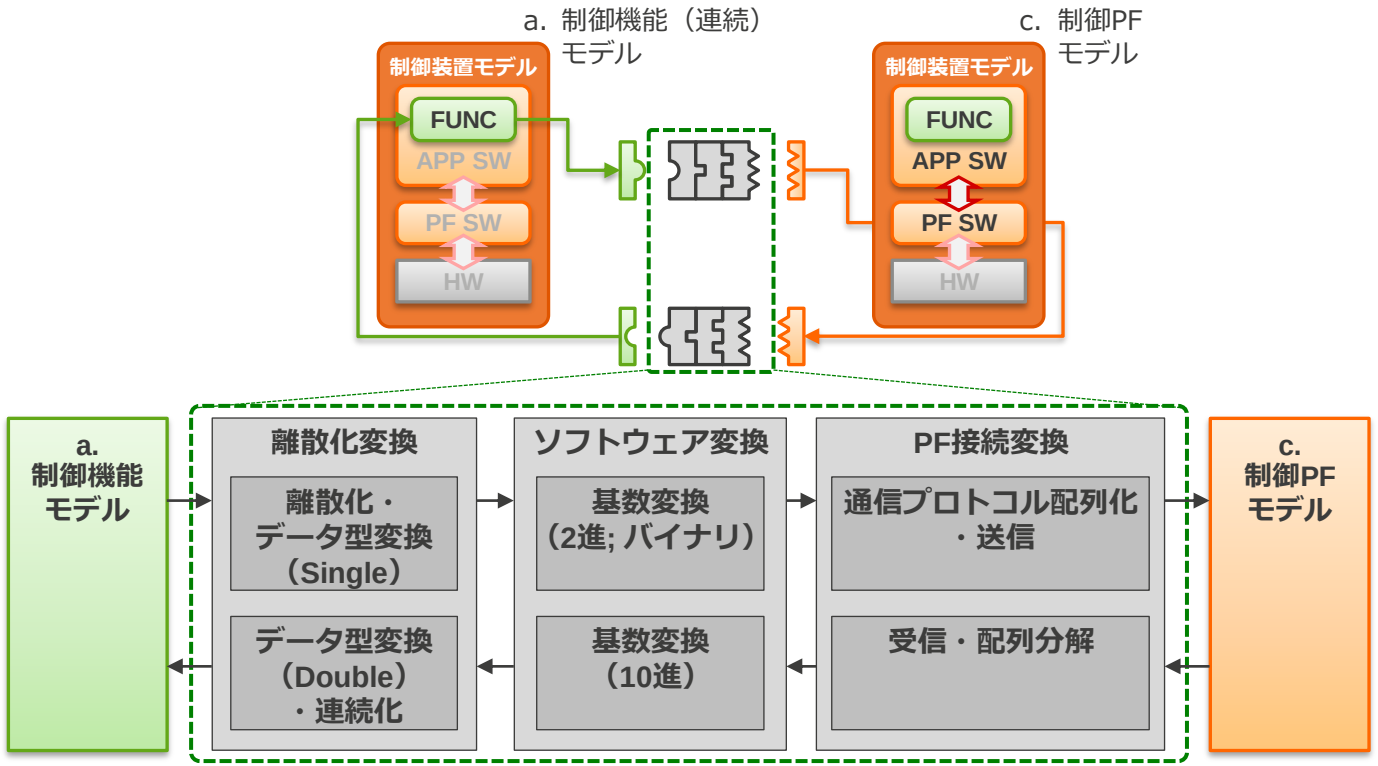
- 異なる抽象度のモデル間に設置する変換ブロックを事前に構築しておくことによって、データ変換アダプタを実現する。

–異なる抽象度のモデル接続に関する留意点（3/4）

モデル接続のための変換アダプタについて

- 入出力変換用のブロックを接続することで、抽象度の差分を解決する方法を、実現手段の一つとして提示する。
- 右図の例示では、前ページで選択したデータ変換、基数変換、通信プロトコル変換への対応のための変換アダプタを、ブロックの形式で実現し、挿入している。
- 各変換アダプタの入出力と内部機能を検討する際に、メタデータ項目を参照する。

例）制御機能（連続）モデルと制御PFモデル間の接続



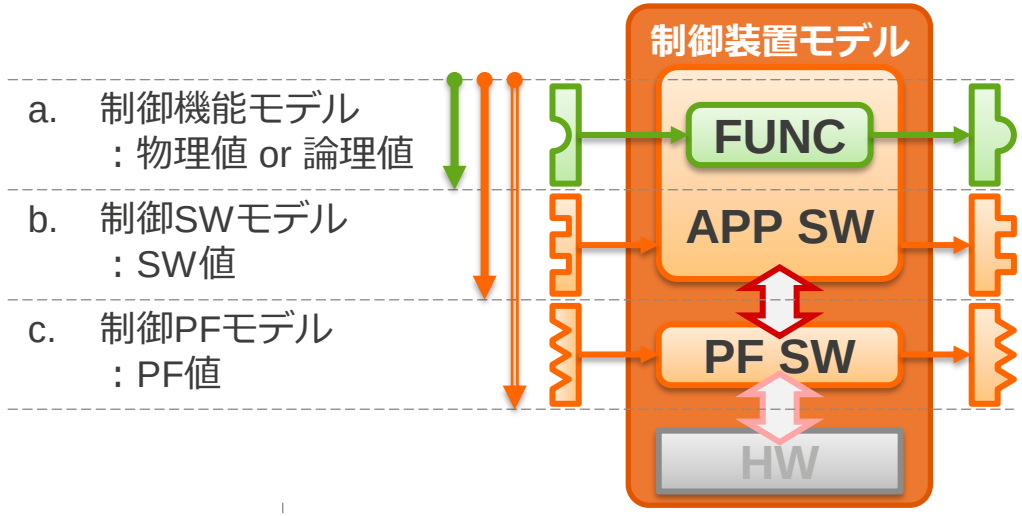
5. I/F項目の設定方法

- データ変換アダプタを準備する際には、それぞれの信号に含まれるデータの内容を参照して意図した入出力が接続されるようにI/Fを設定する。

– 異なる抽象度のモデル接続に関する留意点（4/4）

信号値に含まれるデータの情報（＝メタデータ）について

- 制御開発プロセスの異なる設計フェーズ間でモデルを接続する際は、入出力情報の内容についてあらかじめ設定しておくことで、モデル間の接続が容易となる。
- 下図の例示のように、設定されたデータ定義に基づいて、前述のデータ変換アダプタを事前に準備する。



例）

メタデータ項目 / データ仕様

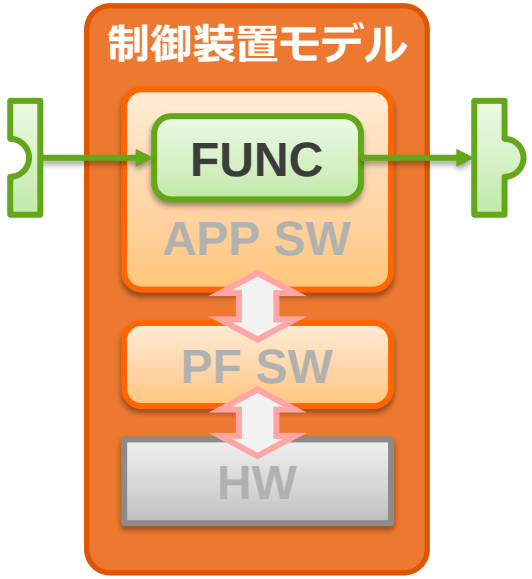
信号名称	信号内容	モデル抽象度	値範囲 最小値	値範囲 最大値	単位	入出力方向	データ型	初期値	LSB (分解能)	オフセット	バイトオーダー	通信周期	RAM値	選択 CAN 通信 フレーム名	ID (hex)	DLC (Byte)	データ位置 (Bit)	ビット長 (Bit)	送信元	送信先	通信速度	Ethernet 通信 Ethernet
trq_VCU_CNT_MG_Nm	モータトルク	a. 制御機能(連続)	0	5000	Nm	入力（Rx）	(Double)	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
		a. 制御機能(離散)	0	65536	–	入力（Rx）	Uint16	0	0.076	0	–	–	–	–	–	–	–	–	–	–	–	–
		b. 制御SW	0	0xFFFF	–	入力（Rx）	Binary	0	0.076	0	Little Endian	100ms	(RAM上のｱﾄﾞﾚｽ)	–	–	–	–	–	–	–	–	–
		c. 制御PF	0	0xFFFF	–	入力（Rx）	–	0	0.076	0	Little Endian	100ms	–	MG2Torque	0x60	8	8	16	MG2	HV	500kbps	–

6.　各抽象度モデルのI/F項目例【CAN 通信】

6. 各抽象度モデルのI/F項目例【CAN 通信】

－各抽象度モデルで使用するI/F項目例｜
－a. 制御機能モデル

a.　制御機能モデル



信号値

- 物理値または論理値
  （→ 対象データの意味を理解できる形式の数値）

メタデータ

- 信号名称
- 信号内容
- 値範囲（最小値｜最大値｜単位）
- 入出力方向

＝＝＝

- データ型
- 初期値
- LSB（分解能）
- オフセット

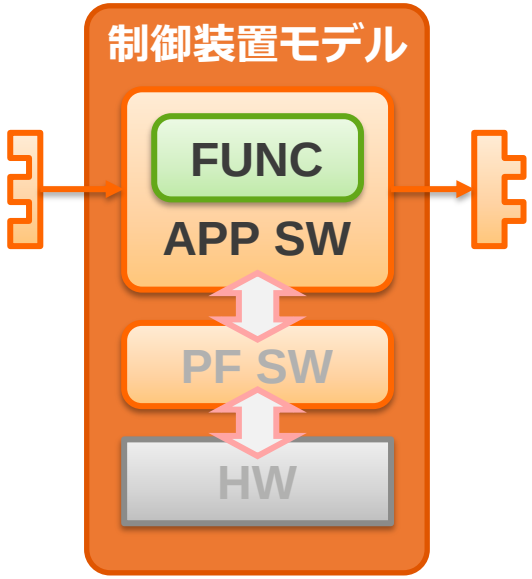
信号名称	信号内容	モデル抽象度	値範囲			入出力方向	データ型	初期値	LSB（分解能）	オフセット
			最小値	最大値	単位					
trq_VCU_CNT_MG_Nm	モータトルク	a. 制御機能(連続)	0	5000	Nm	入力（Rx）	(Double)	–	–	–
		a. 制御機能(離散)	0	65536	–	入力（Rx）	Uint16	0	0.076	0

6. 各抽象度モデルのI/F項目例【CAN 通信】

－各抽象度モデルで使用するI/F項目例｜

－ b. 制御SWモデル

b.　制御SWモデル



信号値

- SW値
  （→ 16進数 または 2進数に変換された数値）

メタデータ

- **メタデータ**（a.制御機能モデル）
  ＋
- 通信周期
- データ型
- 初期値
- LSB（分解能）
- オフセット
- バイトオーダー
- RAM値

信号名称	信号内容	モデル抽象度	値範囲			
			最小値	最大値	単位	入出力方向
trq_VCU_CNT_MG_Nm	モータトルク	b. 制御SW	0	0xFFFF	－	入力（Rx）

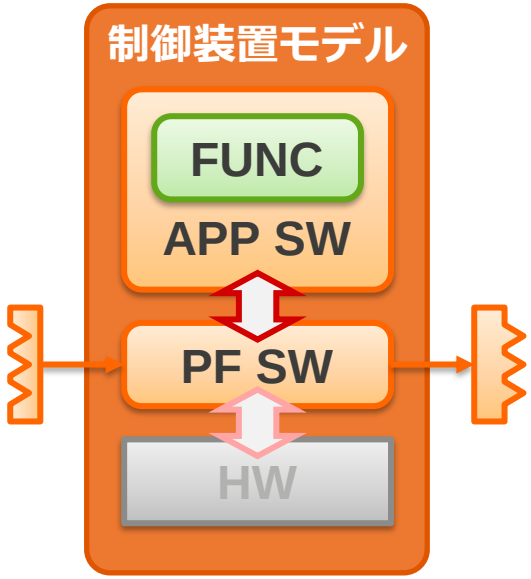
＋

データ型	初期値	LSB（分解能）	オフセット	バイトオーダー	通信周期	RAM値
Binary	0	0.076	0	Little Endian	100ms	(RAM上のｱﾄﾞﾚｽ)

6. 各抽象度モデルのI/F項目例【CAN 通信】

－各抽象度モデルで使用するI/F項目例｜
－ c. 制御PFモデル｜－CAN通信の場合

c.　制御PFモデル



信号値

- PF値
  （→ 対象マイコンでの通信形式に合わせた数値）

メタデータ

- **メタデータ**（a.制御機能モデル ＋ b.制御SWモデル）
  ＋
- CAN通信規格に合わせたメタデータ
  - フレーム名　・DLC　・ビット長　・送信先
  - ID　・データ位置　・送信元　・ボーレート

信号名称	信号内容	モデル抽象度	値範囲			入出力方向
			最小値	最大値	単位	
trq_VCU_CNT_MG_Nm	モータトルク	c. 制御PF	0	0xFFFF	－	入力（Rx）

＋

データ型	初期値	LSB（分解能）	オフセット	バイトオーダー	通信周期	RAM値
－	0	0.076	0	Little Endian	100ms	－

＋

CAN 通信							
フレーム名	ID (hex)	DLC (Byte)	データ位置 (Bit)	ビット長 (Bit)	送信元	送信先	ボーレート
MG2Torque	0x60	8	8	16	MG2	HV	500kbps

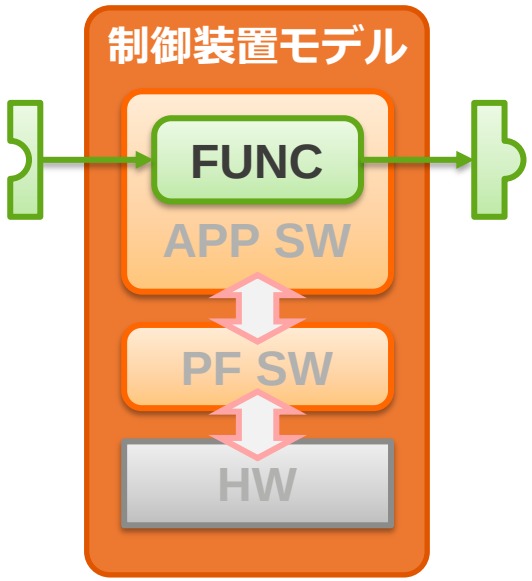
7.　各抽象度モデルのI/F項目例
【Ethernet 通信】

7. 各抽象度モデルのI/F項目例【Ethernet 通信】

－各抽象度モデルで使用するI/F項目例｜

－a. 制御機能モデル

a.　制御機能モデル



信号値

- 物理値または論理値
  （→ 対象データの意味を理解できる形式の数値）

メタデータ

- 信号名称
- 信号内容
- 値範囲（最小値｜最大値｜単位）
- 入出力方向

＝＝＝

- データ型
- 初期値
- LSB（分解能）
- オフセット

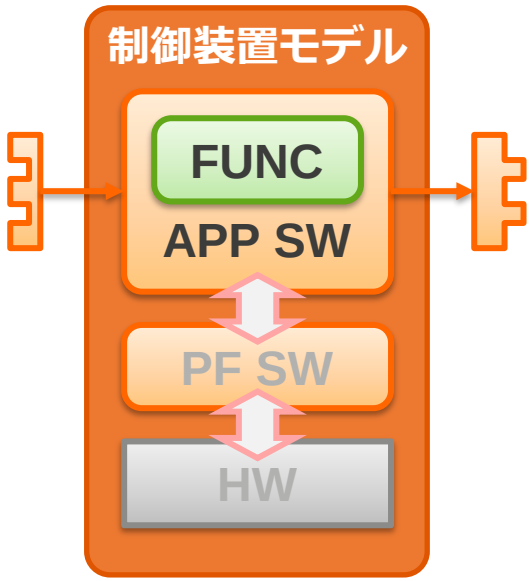
信号名称	信号内容	モデル抽象度	値範囲			入出力方向	データ型	初期値	LSB（分解能）	オフセット
			最小値	最大値	単位					
trq_VCU_CNT_MG_Nm	モータトルク	a. 制御機能(連続)	0	5000	Nm	入力（Rx）	(Double)	–	–	–
		a. 制御機能(離散)	0	65536	–	入力（Rx）	Uint16	0	0.076	0

7. 各抽象度モデルのI/F項目例【Ethernet 通信】

－各抽象度モデルで使用するI/F項目例｜

－ b. 制御SWモデル

b.　制御SWモデル



信号値

- SW値
  （→ 16進数 または 2進数に変換された数値）

メタデータ

- **メタデータ**（a.制御機能モデル）
  ＋
- 通信周期
- データ型
- 初期値
- LSB（分解能）
- オフセット
- バイトオーダー
- RAM値

信号名称	信号内容	モデル抽象度	値範囲			入出力方向
			最小値	最大値	単位	
trq_VCU_CNT_MG_Nm	モータトルク	b. 制御SW	0	0xFFFF	－	入力（Rx）

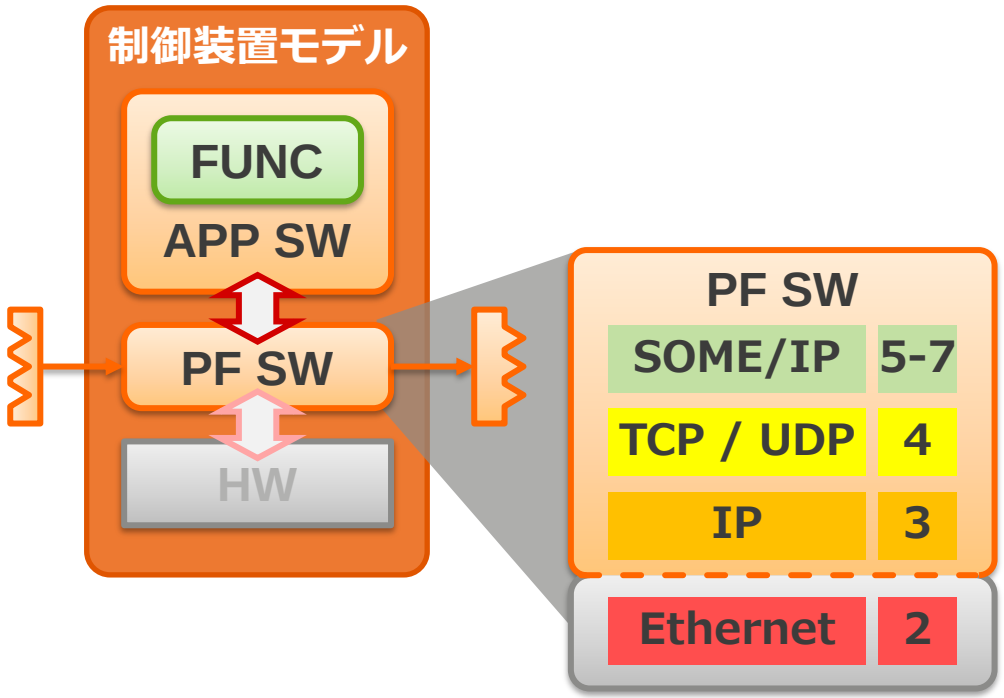
＋

データ型	初期値	LSB（分解能）	オフセット	バイトオーダー	通信周期	RAM値
Binary	0	0.076	0	Little Endian	100ms	(RAM上のｱﾄﾞﾚｽ)

7. 各抽象度モデルのI/F項目例【Ethernet 通信】

－各抽象度モデルで使用するI/F項目例｜
－ c. 制御PFモデル｜－Ethernet通信の場合

c.　制御PFモデル



信号値

- PF値
  （→ 対象マイコンでの通信形式に合わせた数値）

メタデータ

- **メタデータ**（a.制御機能モデル＋b.制御SWモデル）
  ＋
- 車載Ethernet通信規格に合わせたメタデータ
  - SOME/IP　（L7：アプリケーション層 ～ L5：セッション層）
  - TCP / UDP　（L4：トランスポート層）
  - IP　（L3：ネットワーク層）
  - イーサネット　（L2：データリンク層）

信号名称	信号内容	モデル抽象度	値範囲			
			最小値	最大値	単位	入出力方向
trq_VCU_CNT_MG_Nm	モータトルク	c. 制御PF	0	0xFFFF	－	入力（Rx）

＋

データ型	初期値	LSB（分解能）	オフセット	バイトオーダー	通信周期	RAM値
－	0	0.076	0	Little Endian	100ms	－

＋

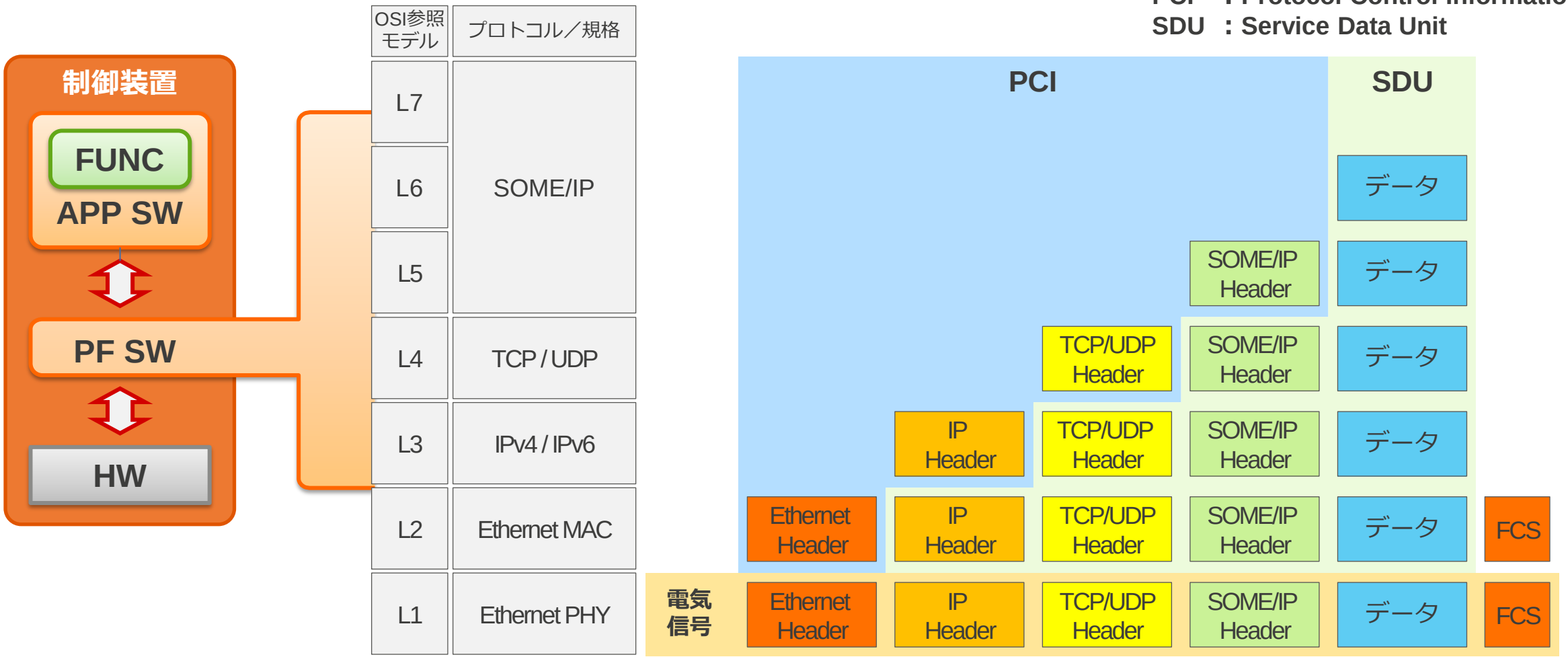
Ethernet 通信				
Ethernet	IP	選択		SOME/IP
		TCP	UDP	

Ethernet通信では、通信プロトコルがOSI参照モデルの各層で定義される。
本書では、アプリケーション層からデータリンク層までをPF値とした。

7. 各抽象度モデルのI/F項目例【Ethernet 通信】

– （補足）OSI参照モデルによる通信プロトコルの定義

PCI ：Protocol Control Information
SDU ：Service Data Unit

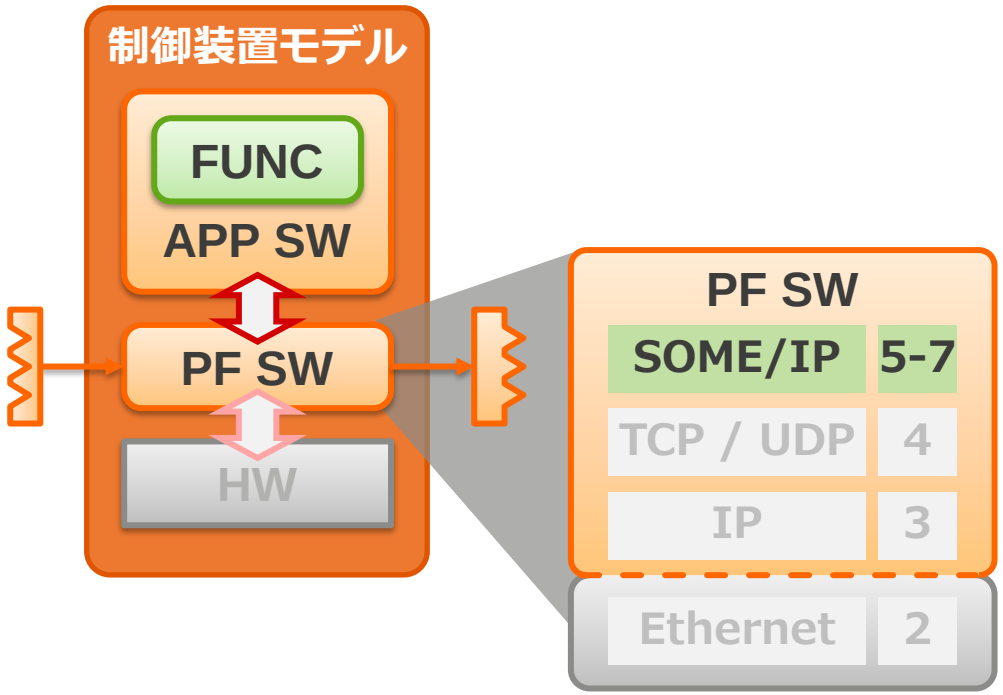


7. 各抽象度モデルのI/F項目例【Ethernet 通信】

－各抽象度モデルで使用するI/F項目例｜

－ c. 制御PFモデル｜－Ethernet通信の場合

c.　制御PFモデル



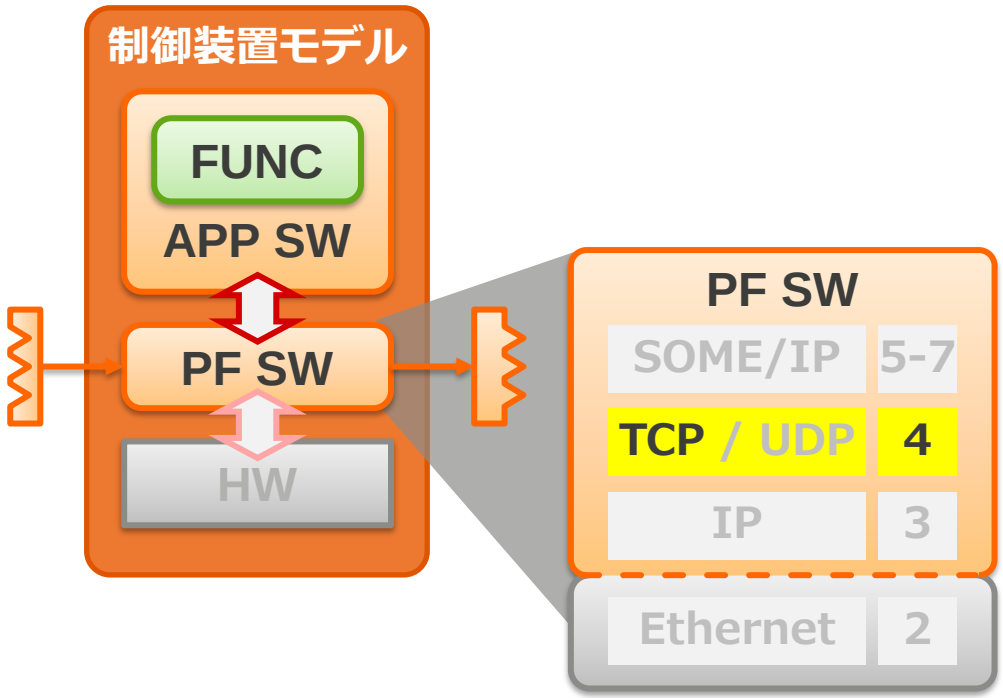
メタデータ（SOME/IP）	説明
Message ID	イベントを識別するための識別子
Length	メッセージの長さ
Request ID	クライアントIDとセッションID からなる一意の識別子
Protocol Version	使用される Some/IPヘッダー形式の識別
Interface Version	Some/IPサービスの 現在のバージョン説明
Message Type	メッセージのタイプ
Return Code	要求が正常に処理されたかを通知

7. 各抽象度モデルのI/F項目例【Ethernet 通信】

－各抽象度モデルで使用するI/F項目例｜

－ c. 制御PFモデル｜－Ethernet通信の場合

c.　制御PFモデル



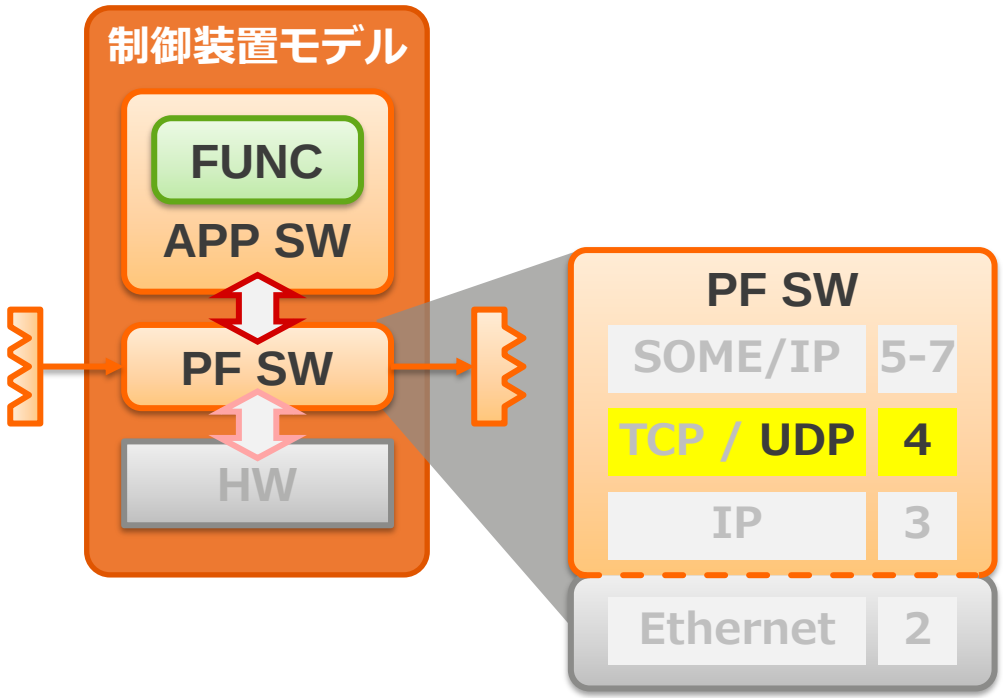
メタデータ（TCP）	説明
送信元ポート番号	送信元のポート番号の値
宛先ポート番号	宛先のポート番号の値
シーケンス番号	パケットに付けられる通し番号 受信した相手の確認応答番号を使用
確認応答番号	確認応答番号 受信した相手のｼｰｹﾝｽ番号＋ﾃﾞｰﾀｻｲｽﾞ
データオフセット	TCPヘッダの長さ
予約	全ビット「0」、将来の拡張用
コントロールフラグ	URG, ACK, PSH, RST, SYN, FIN の 6bit
ウィンドウサイズ	受信側が一度に受信可能なデータ量
チェックサム	TCPﾍｯﾀﾞとﾃﾞｰﾀ部のｴﾗｰﾁｪｯｸ用
緊急ポインタ	URGが「1」の場合に使用されるﾌｨｰﾙﾄﾞ
オプション	TCP通信の性能向上（拡張）用
パディング	「0」でTCPﾍｯﾀﾞ長を32bit整数に調整

7. 各抽象度モデルのI/F項目例【Ethernet 通信】

－各抽象度モデルで使用するI/F項目例｜

－ c. 制御PFモデル｜－Ethernet通信の場合

c.　制御PFモデル

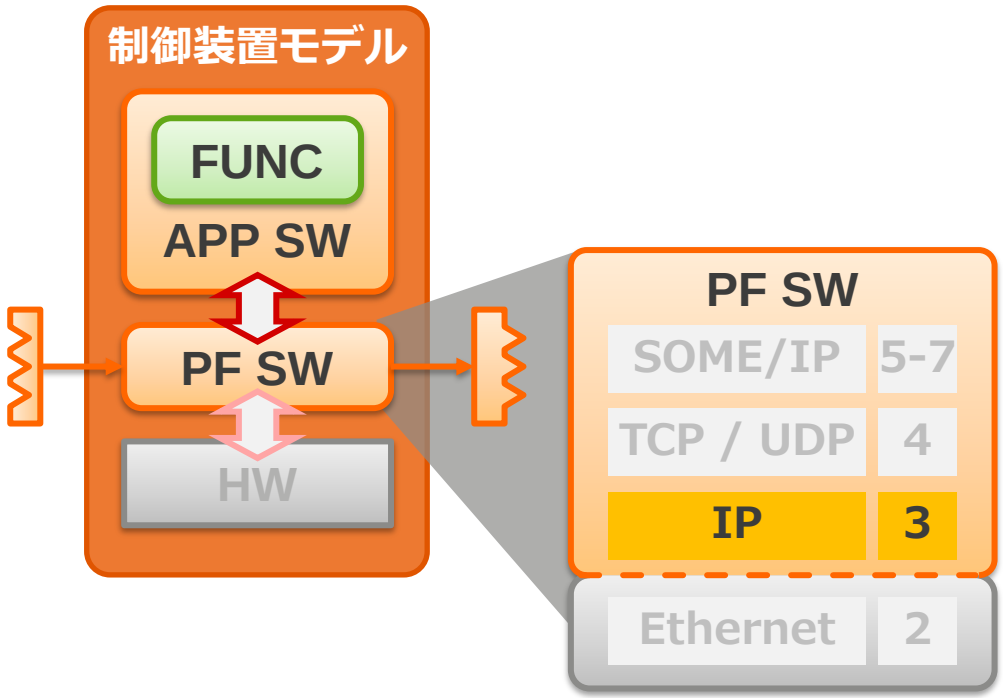


メタデータ（UDP）	説明
送信元ポート番号	送信元のポート番号の値
宛先ポート番号	宛先のポート番号の値
パケット長	UDPﾍｯﾀﾞとUDPﾃﾞｰﾀの長さを合計したサイズの値
チェックサム	UDPﾍｯﾀﾞとﾃﾞｰﾀ部のｴﾗｰﾁｪｯｸ用

7. 各抽象度モデルのI/F項目例【Ethernet 通信】

－各抽象度モデルで使用するI/F項目例｜
－ c. 制御PFモデル｜－Ethernet通信の場合

c.　制御PFモデル

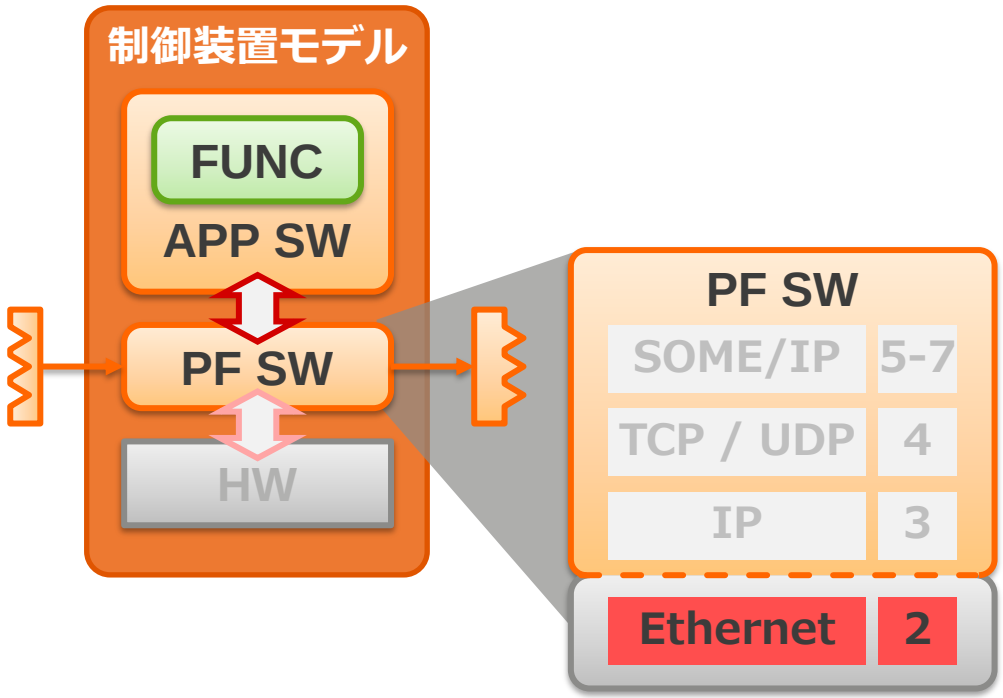


メタデータ（IP）	説明
バージョン	IPバージョン
ヘッダ長	IPヘッダの長さ
サービスタイプ	パケットの優先順位を指定する情報
パケット長	IPヘッダを含むパケット全体のサイズ
識別番号	分割ﾊﾟｹｯﾄを分割前ﾊﾟｹｯﾄに戻すための識別番号
フラグ	ﾊﾟｹｯﾄ分割（ﾌﾗｸﾞﾒﾝﾄ化）に関する制御情報
ﾌﾗｸﾞﾒﾝﾄｵﾌｾｯﾄ	分割ﾊﾟｹｯﾄの分割前ﾊﾟｹｯﾄでの位置情報
生存時間	ネットワークをルーティングできる数
プロトコル	上位のﾄﾗﾝｽﾎﾟｰﾄ層で使用しているプロトコル
ﾍｯﾀﾞﾁｪｯｸｻﾑ	IPヘッダの誤り検出のためのチェックサム情報
送信元IPアドレス	送信元のIPアドレスの情報
宛先IPアドレス	宛先のIPアドレスの情報
オプション	IPパケット通信の拡張情報
パディング	IPヘッダ長を32bit単位になるよう 0 埋め調整

7. 各抽象度モデルのI/F項目例【Ethernet 通信】

－各抽象度モデルで使用するI/F項目例｜
－ c. 制御PFモデル｜－Ethernet通信の場合

c.　制御PFモデル



メタデータ（Ethernet）	説明
プリアンブル	受信側と送信側で同期を行う際に使用される 先頭7(byte)：10101010 末尾1(byte)：10101011
宛先MACアドレス	宛先のMACアドレスの値
送信元MACアドレス	送信元のMACアドレスの値
長さ	データの長さの値
タイプ	上位層のプロトコル種類を示す値 例）IPv4：0x0800 など
FCS	フレーム全体のCRCチェック情報 エラー検出のため、送信、受信双方で同じチェックを実施

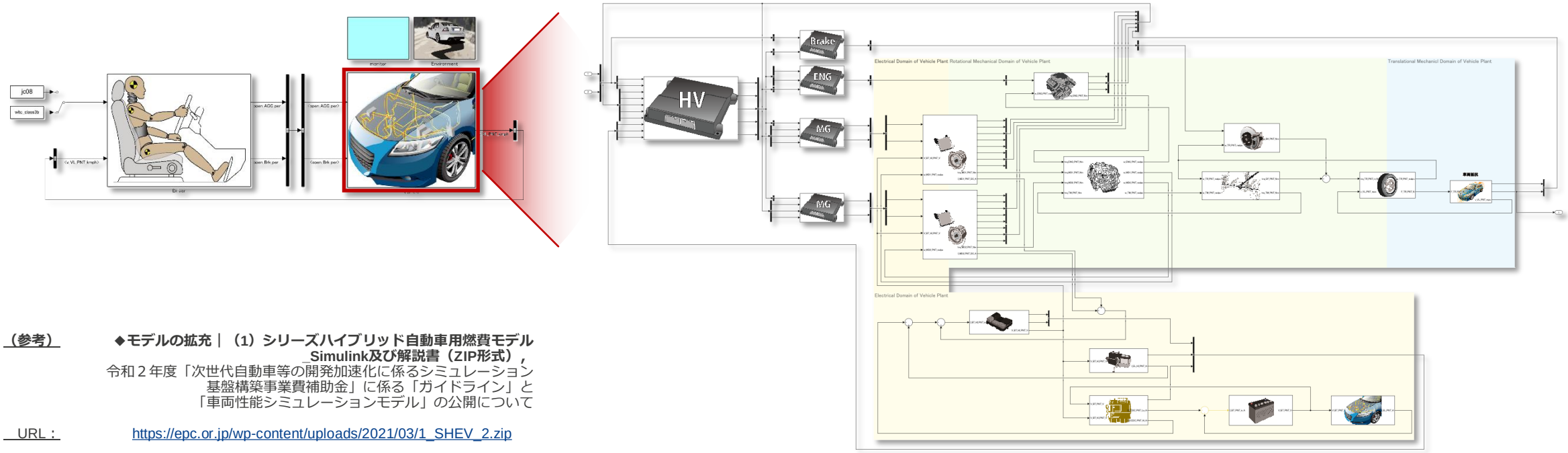
8.　モデル接続の具体事例

8. モデル接続の具体事例

- 本ガイドラインでは、経済産業省補助事業で提示された”シリーズハイブリッド自動車用燃費モデル”を対象に、CAN信号によるデータ授受を想定したモデル接続を解説する。

－CAN通信の I/Fの場合｜－事例の対象モデル（1/2）

本ガイドラインでは、「次世代自動車等の開発加速化に係るシミュレーション基盤構築事業費補助金」に係る「車両性能シミュレーションモデル」にて公開されている “シリーズハイブリッド自動車用燃費モデル”（以降、METI準拠モデル）を対象として、ガイドライン内で提示した接続方法を、一連のプロセスを通じて解説する。



（参考）　◆モデルの拡充｜（1）シリーズハイブリッド自動車用燃費モデル_Simulink及び解説書（ZIP形式），令和２年度「次世代自動車等の開発加速化に係るシミュレーション基盤構築事業費補助金」に係る「ガイドライン」と「車両性能シミュレーションモデル」の公開について

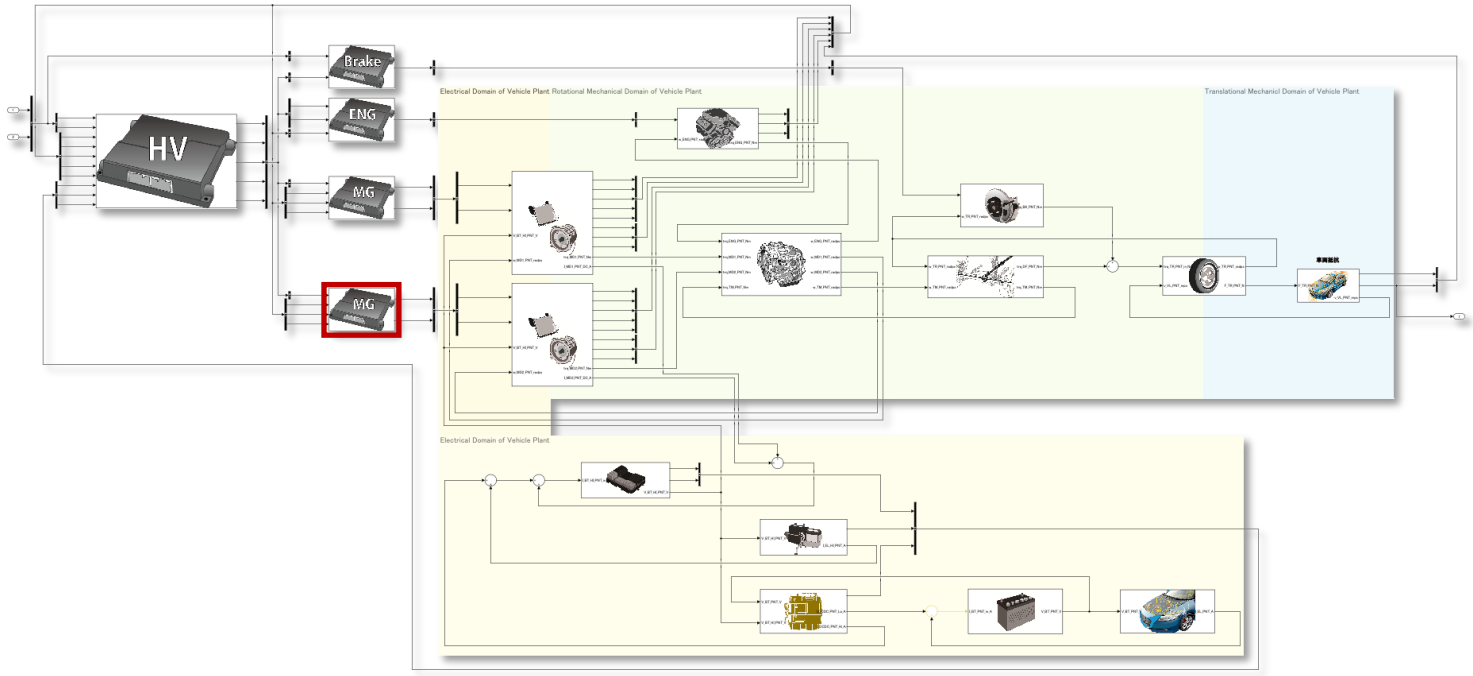
URL：　https://epc.or.jp/wp-content/uploads/2021/03/1_SHEV_2.zip

8. モデル接続の具体事例

- 本ガイドラインでは、経済産業省補助事業で提示された”シリーズハイブリッド自動車用燃費モデル”を対象に、CAN信号によるデータ授受を想定したモデル接続を解説する。

－CAN通信の I/Fの場合｜－事例の対象モデル（2/2）

抽象度の異なるシステムモデルを再現するため、本ガイドラインでは下図に示した走行用モータ制御装置の抽象度を a. 制御機能（連続）モデル から c. 制御PFモデル に変換して、以降の事例の解説を進める。



（参考）　◆モデルの拡充｜（1）シリーズハイブリッド自動車用燃費モデル_Simulink及び解説書（ZIP形式），令和２年度「次世代自動車等の開発加速化に係るシミュレーション基盤構築事業費補助金」に係る「ガイドライン」と「車両性能シミュレーションモデル」の公開について

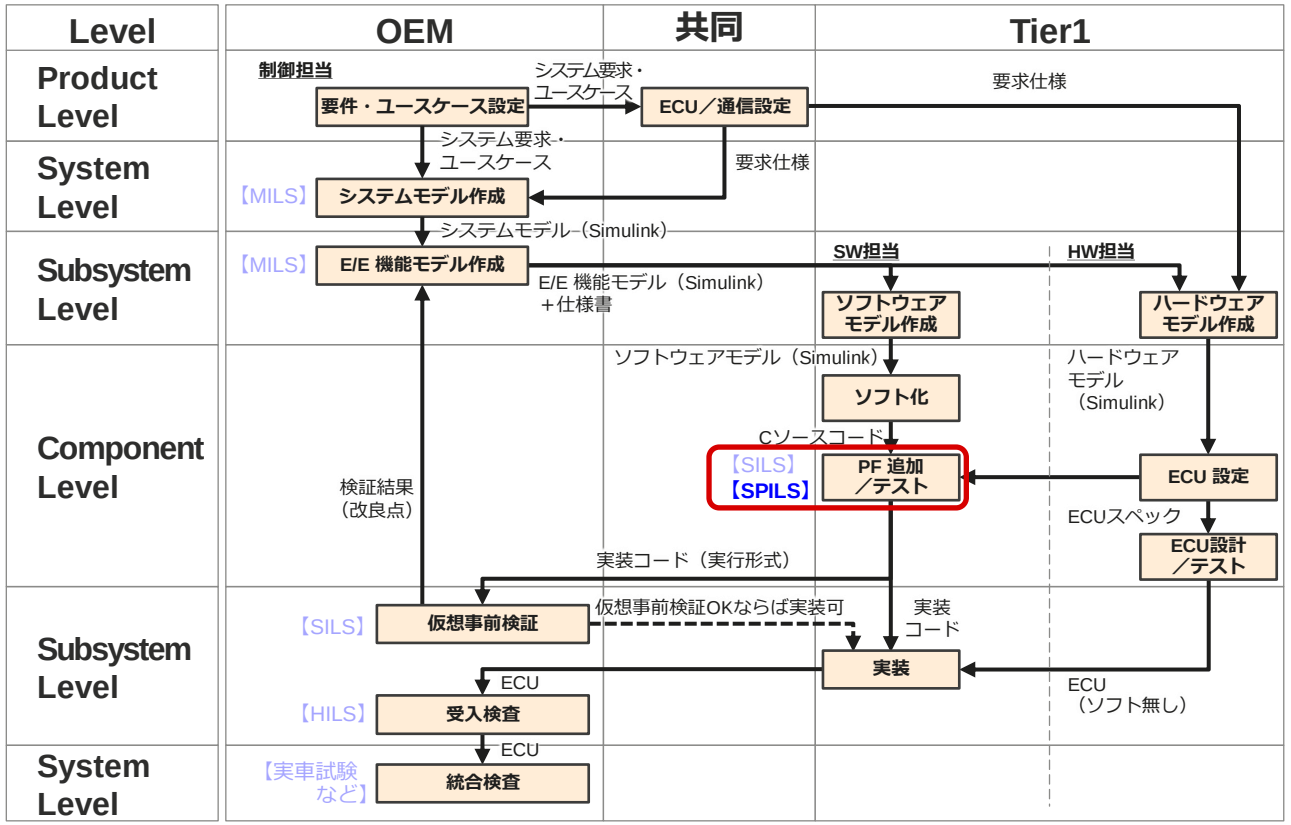
URL：　https://epc.or.jp/wp-content/uploads/2021/03/1_SHEV_2.zip

8. モデル接続の具体事例

- 以降の具体事例では、設計フェーズが実装前の状態まで進捗した時点を想定する。ここで、車両全体の制御機能モデル+プラントモデルを接続し、μCへの実行形式ファイルを実装する前に検証を実施する。

–CAN通信の I/Fの場合｜–具体事例の場面・状況設定（1/4）

- 本事例では、制御開発の設計フェーズが進捗し、実装先ECUのマイクロコントローラ（μC）仕様に対応したソフトウェアが、車両レベルで制御仕様を満足する動作を実現していることを確認する場面を想定する。
- 開発対象となる制御装置は、μC仕様に合わせたAPP SWと、MCALなどのPF SWまで統合された状態になっている。
- 車両レベルのモデルは、制御装置モデル、プラントモデルともに、抽象度の最も高いモデルとなっている。

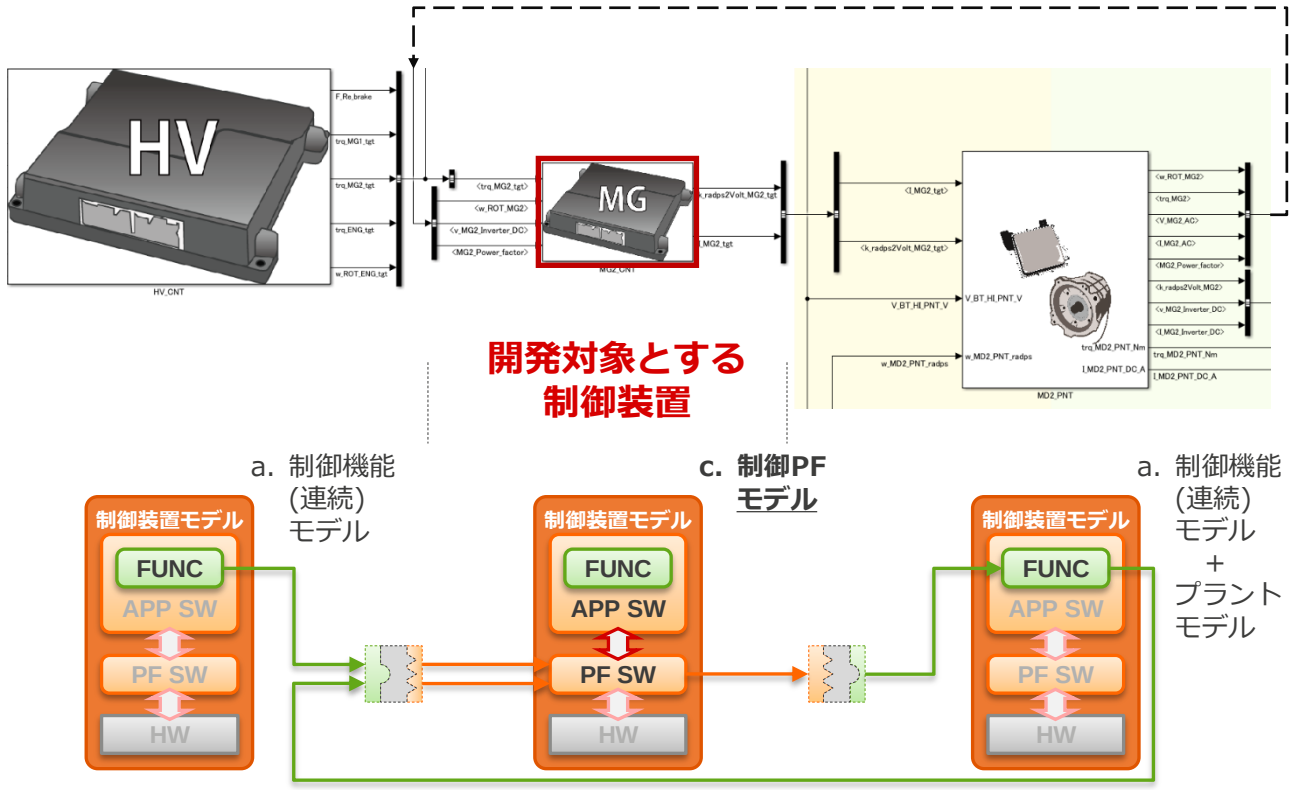


8. モデル接続の具体事例

- 具体事例に用いるモデルの状況に関する設定を下記に示す。

－CAN通信の I/Fの場合｜－具体事例の場面・状況設定（2/4）

- 開発対象である走行用モータ制御装置は、実装先ECUの仕様に合わせたソフトウェアの構築まで完了しており、本ガイドラインのPF SWまで準備されている。
- 上記のモータ制御装置を車両レベルで仮想検証するため、接続相手のモデルとして、各制御機能モデル、および同じ抽象度で再現されたプラントモデルを接続する。
- また、上記のモータ制御装置モデルは、Simulinkのような抽象度が高めのモデルを扱う制御シミュレータとは、別のツールで再現される。

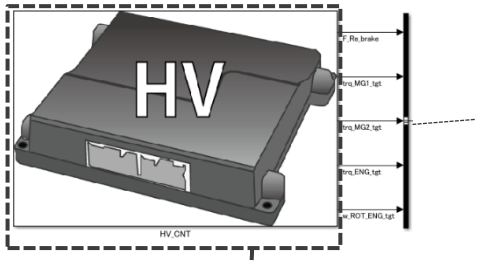


8. モデル接続の具体事例

- 具体事例に用いるモデルのうち、抽象度の高いモデル（制御機能モデル）については、METI準拠モデルをそのまま利用した。

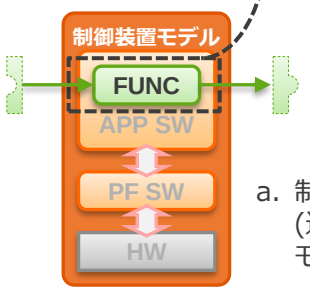
–CAN通信の I/Fの場合｜–具体事例の場面・状況設定（3/4）；補足

- METI準拠モデル内のハイブリッド制御装置、および走行用モータ装置と以降のプラントモデルは、最も抽象度の高い、a. 制御機能（連続）モデルである。



METI準拠モデル内の
ハイブリッド制御装置モデル

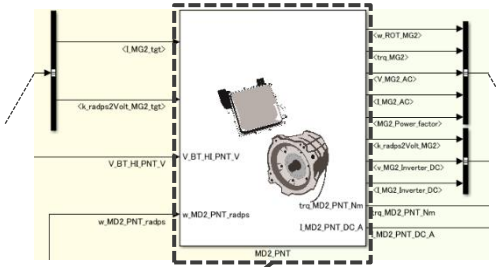
走行用モータ制御装置などの
後段の装置を制御するための
目標値は連続の物理量で計算
と出力が行われる



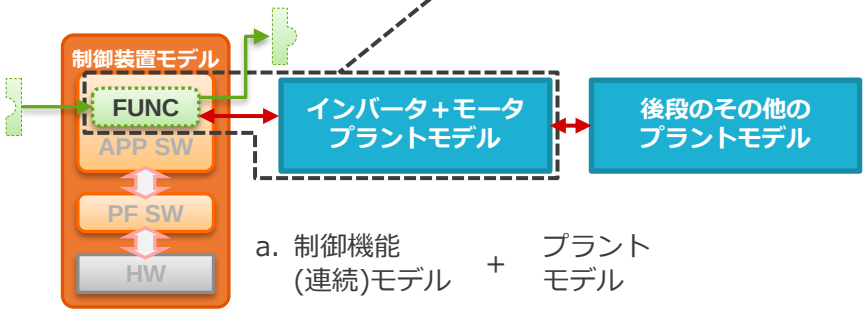
a. 制御機能
(連続)
モデル

METI準拠モデル内の
走行用モータプラントモデル

METI準拠モデルでは
目標値通りの挙動を想定しており、
入力の物理量は、サブシステム
ブロック内部で直接プラント
モデルに入力されている



左記と同様に、
出力の物理量は、直接
プラントモデルの物理量
が出力されている



ただし、本事例では
末端の制御機能が
備わっていることを
想定してモデルを
構築する

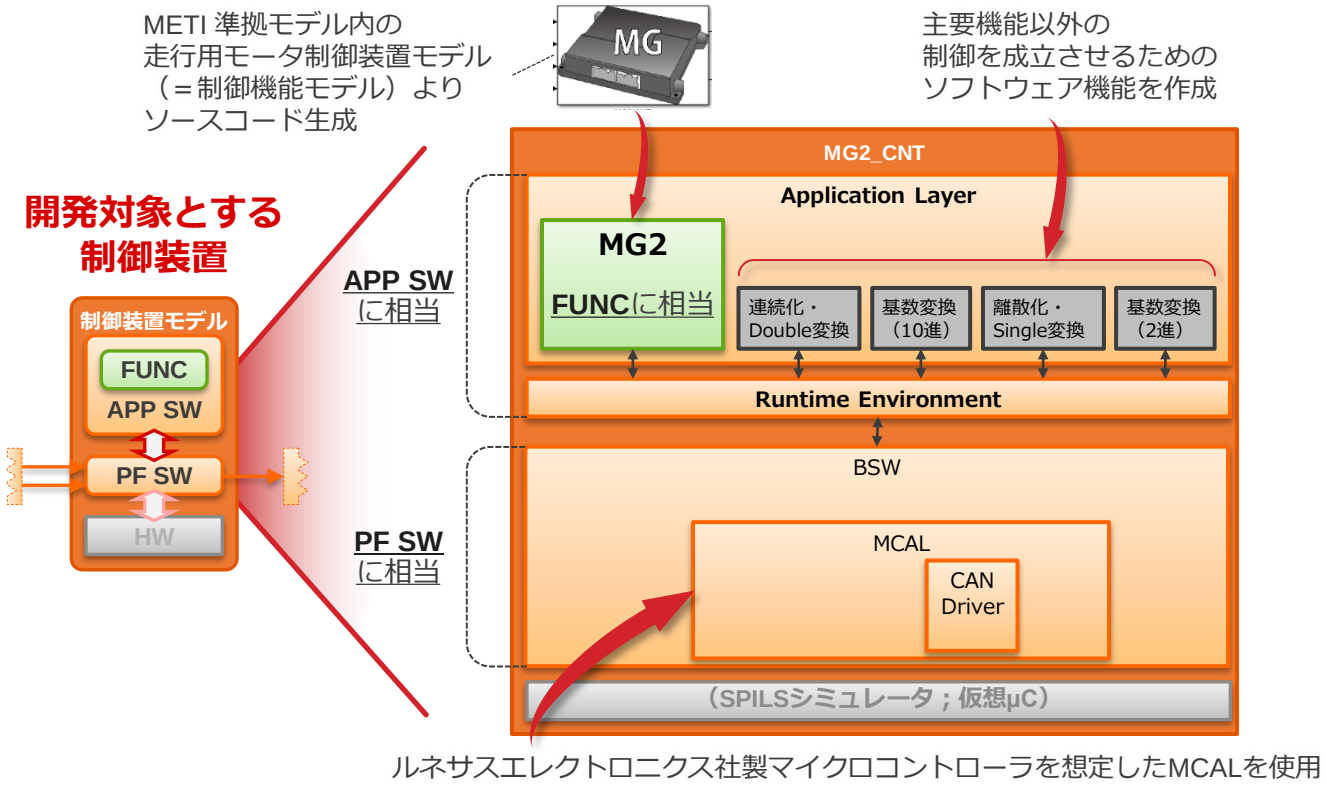
a. 制御機能
(連続)モデル
＋
プラント
モデル

8. モデル接続の具体事例

- 具体事例に用いるモデルのうち、抽象度の低いモデル（制御PFモデル）については、下図の変換によって準備を行った。

－CAN通信の I/Fの場合｜－具体事例の場面・状況設定（4/4）；補足

- 今回の事例では、METI準拠モデル内の走行用モータ制御装置の抽象度の変換を、右図の構成で実現している。
- 具体的には、METI準拠モデル（a. 制御機能モデル）から、今回の事例使用するモデル（c. 制御PFモデル）へ変換を行っている。
- FUNCや一部のAPPSWについては、生成後の修正が必要な場合もあるが、モデルからオートコードで生成する。
- RTEについては、AUTOSARであればXMLで記述された仕様から、ジェネレータツールでコードを生成可能である。
- BSW以下は、事前に準備されたソースコードが必要である。

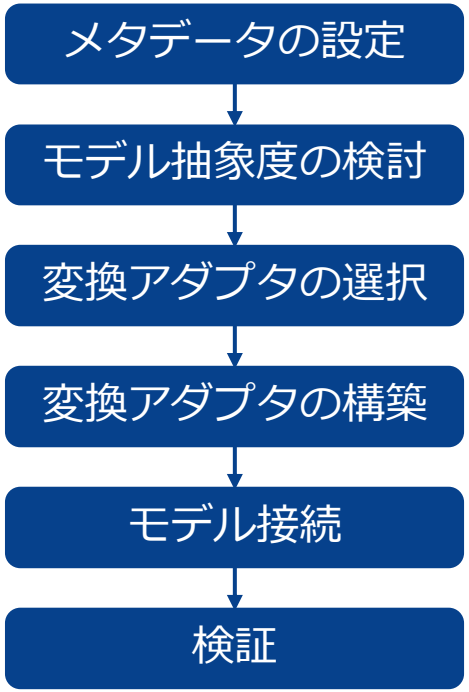


8. モデル接続の具体事例

- 入出力データのメタデータ設定、変換アダプタの選択と構築を実施した上で、異なる抽象度のモデルを接続し、対象の制御装置（モデル）の挙動を検証する。

－CAN通信の I/Fの場合｜－作業手順

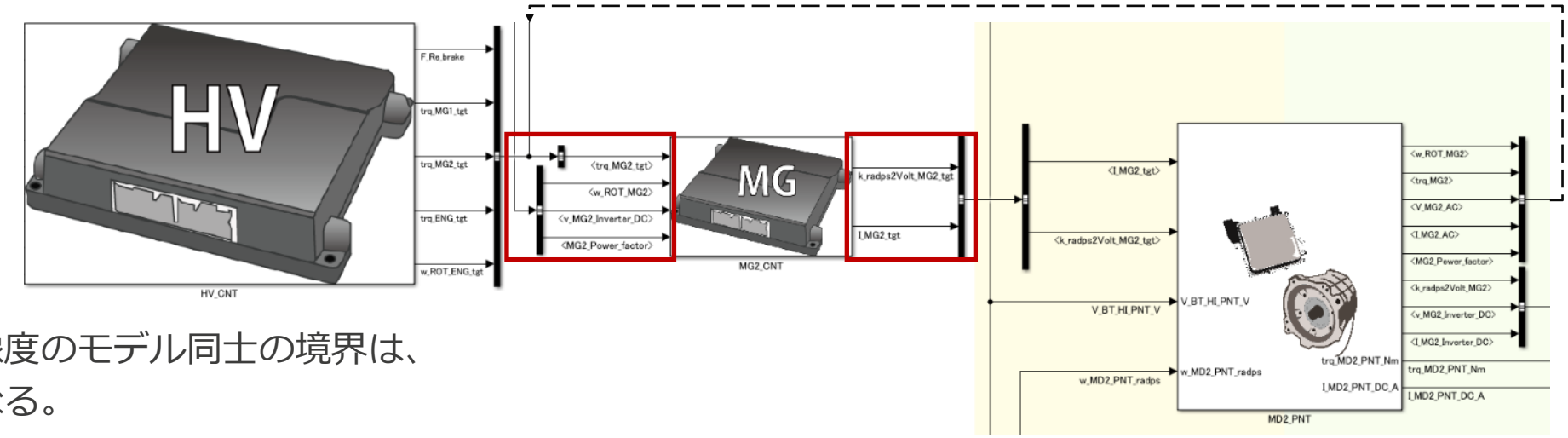
- モデル接続までの手順を右図に示した。
  まず、各抽象度のモデルが、どのような入出力を必要とするか、事前にメタデータを設定する。
  次に、接続されるモデル同士の抽象度を比較してそれぞれの入出力を接続するために必要な変換を検討しておく。検討内容に従って、データ変換アダプタを構築して、モデルを接続する。
  最後に、接続されたモデルを用いて検証を実施する。
- 実際の検証では、制御仕様書の内容を充足することを明らかにするが、本事例では検証までは含まない。



8. モデル接続の具体事例

- 異なる抽象度のモデル同士が接続される境界の入出力データに着目して、それぞれの抽象度のモデルが必要とするデータの形式などを明示するために、メタデータの設定を行う。

－CAN通信の I/Fの場合｜－メタデータの設定（1/2)



- 本事例では、異なる抽象度のモデル同士の境界は、上図の赤枠内の接続となる。
- 従って、上記の接続における入出力データに対して、メタデータの設定を実施する。
- なお、今回の事例では単純化のため、以降の走行用モータ制御装置への入力を、一つのバスとして扱う。

8. モデル接続の具体事例

- 今回の事例内容に合わせたメタデータの設定を下表のように実施した。

–CAN通信の I/Fの場合｜–メタデータの設定（2/2)

- 各入出力信号に対するメタデータ、およびパラメータは、今回の事例では右表のように設定した。
- 右表は、制御開発の進度に応じて定義される制御仕様や通信仕様などに合わせて、設定を進める。

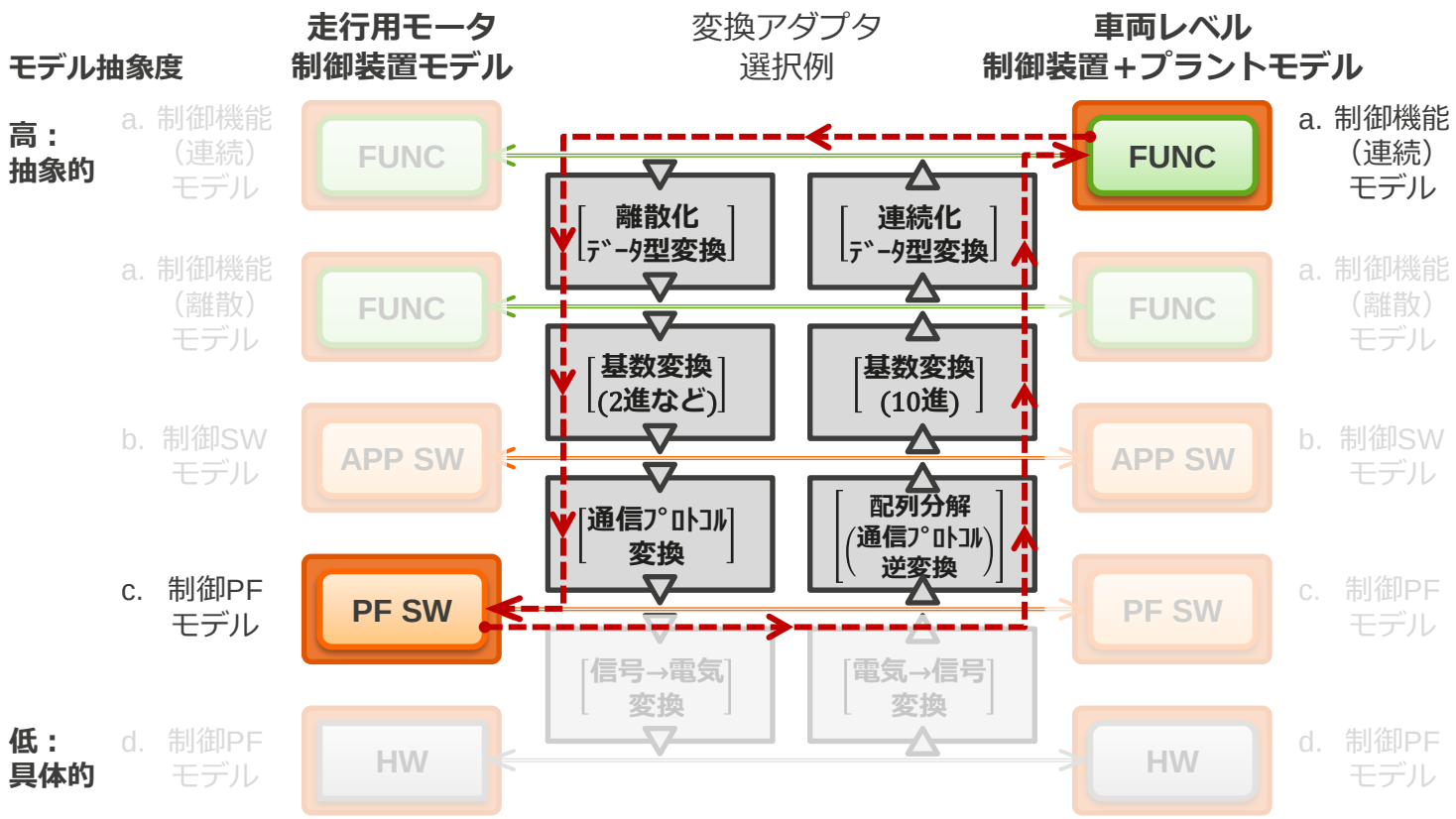
信号名称	信号内容	モデル抽象度	値範囲			入出力方向	通信周期	データ型	初期値	LSB（分解能）	オフセット	バイトオーダー	CAN 通信				
			最小値	最大値	単位								フレーム名	ID（hex）	DLC（Byte）	データ位置（Bit）	ビット長（Bit）
trq_MG2_tgt	モータトルク目標値	a. 制御機能(連続)	-200	200	Nm	入力（Rx）	–	Double	–	–	–	–	–	–	–	–	–
		a. 制御機能(離散)	0	65536	–	入力（Rx）	–	Single	0	0.0061036	200	–	–	–	–	–	–
		b. 制御SW	0	0xFFFF	–	入力（Rx）	10ms	Binary	0	0.0061036	200	Big Endian	–	–	–	–	–
		c. 制御PF	0	0xFFFF	–	入力（Rx）	10ms	–	0	0.0061036	200	Big Endian	IN-Data	0x1A	8	7	16
w_ROT_MG	モータ目標回転数	a. 制御機能(連続)	-2000	2000	rad/s	入力（Rx）	–	Double	–	–	–	–	–	–	–	–	–
		a. 制御機能(離散)	-1048576	1048575	–	入力（Rx）	–	Single	0	0.0038147	0	–	–	–	–	–	–
		b. 制御SW	0	0x1FFFFF	–	入力（Rx）	10ms	Binary	0	0.0038147	0	Little Endian	–	–	–	–	–
		c. 制御PF	0	0x1FFFFF	–	入力（Rx）	10ms	–	0	0.0038147	0	Little Endian	IN-Data	0x1A	8	16	21
v_MG_Inverter_DC	インバータ直流電圧値	a. 制御機能(連続)	0	400	V	入力（Rx）	–	Double	–	–	–	–	–	–	–	–	–
		a. 制御機能(離散)	0	65535	–	入力（Rx）	–	Single	0	0.0061036	0	–	–	–	–	–	–
		b. 制御SW	0	0xFFFF	–	入力（Rx）	10ms	Binary	0	0.0061036	0	Little Endian	–	–	–	–	–
		c. 制御PF	0	0xFFFF	–	入力（Rx）	10ms	–	0	0.0061036	0	Little Endian	IN-Data	0x1A	8	37	16
MG_Power_factor	モータ力率	a. 制御機能(連続)	0	1	–	入力（Rx）	–	Double	–	–	–	–	–	–	–	–	–
		a. 制御機能(離散)	0	255	–	入力（Rx）	–	Single	0	0.0039216	0	–	–	–	–	–	–
		b. 制御SW	0	0xFF	–	入力（Rx）	10ms	Binary	0	0.0039216	0	Big Endian	–	–	–	–	–
		c. 制御PF	0	0xFF	–	入力（Rx）	10ms	–	0	0.0039216	0	Big Endian	IN-Data	0x1A	8	63	8
k_radps2Volt_MG_tgt	逆起電力係数目標値	a. 制御機能(連続)	0	2	–	出力（Tx）	–	Double	–	–	–	–	–	–	–	–	–
		a. 制御機能(離散)	0	255	–	出力（Tx）	–	Single	0	0.0078431	0	–	–	–	–	–	–
		b. 制御SW	0	0xFF	–	出力（Tx）	10ms	Binary	0	0.0078431	0	Big Endian	–	–	–	–	–
		c. 制御PF	0	0xFF	–	出力（Tx）	10ms	–	0	0.0078431	0	Big Endian	OUT-Data	0x1B	4	7	8
I_MG_tgt	電流目標値	a. 制御機能(連続)	-500	500	A	出力（Tx）	–	Double	–	–	–	–	–	–	–	–	–
		a. 制御機能(離散)	-262144	262143	–	出力（Tx）	–	Single	0	0.0038147	0	–	–	–	–	–	–
		b. 制御SW	0	0x7FFFF	–	出力（Tx）	10ms	Binary	0	0.0038147	0	Big Endian	–	–	–	–	–
		c. 制御PF	0	0x7FFFF	–	出力（Tx）	10ms	–	0	0.0038147	0	Big Endian	OUT-Data	0x1B	4	15	19

8. モデル接続の具体事例

- 今回の事例で想定するモデルの抽象度を横並びで比較して検討し、入出力の接続を成立させるためのデータ変換アダプタを選択する。

－CAN通信の I/Fの場合｜－モデル抽象度の検討と変換アダプタの選択

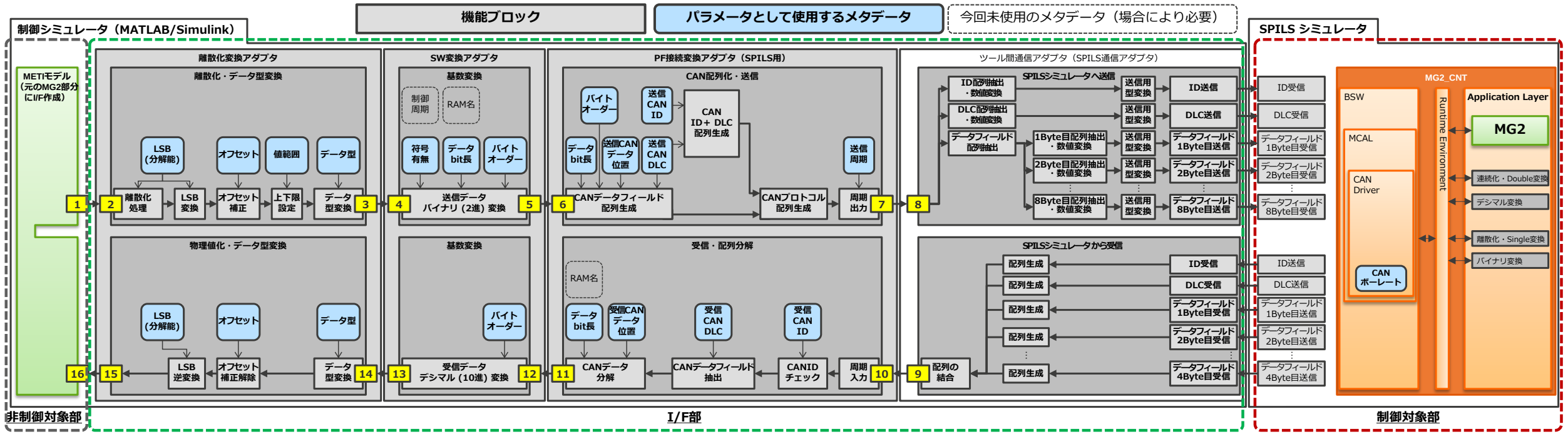
- 場面・状況設定にて説明した通り、
  今回は c. 制御PFモデルと、
  a. 制御機能（連続）モデルを接続する。
- モデル同士の抽象度の差分を解消するために、右図の赤点線上の変換アダプタを選択する。



8. モデル接続の具体事例

- 事前に設定したメタデータの情報に基づき、各抽象度モデルが必要とする入出力データの形式の差を解消するような変換アダプタを実際に構築し、異なる抽象度のモデルを接続する。

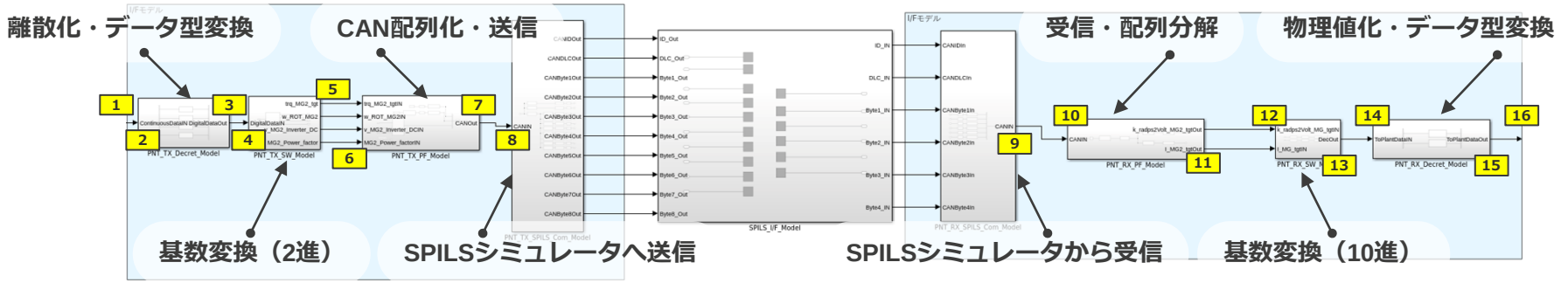
－CAN通信の I/Fの場合｜－モデル接続（1/7）

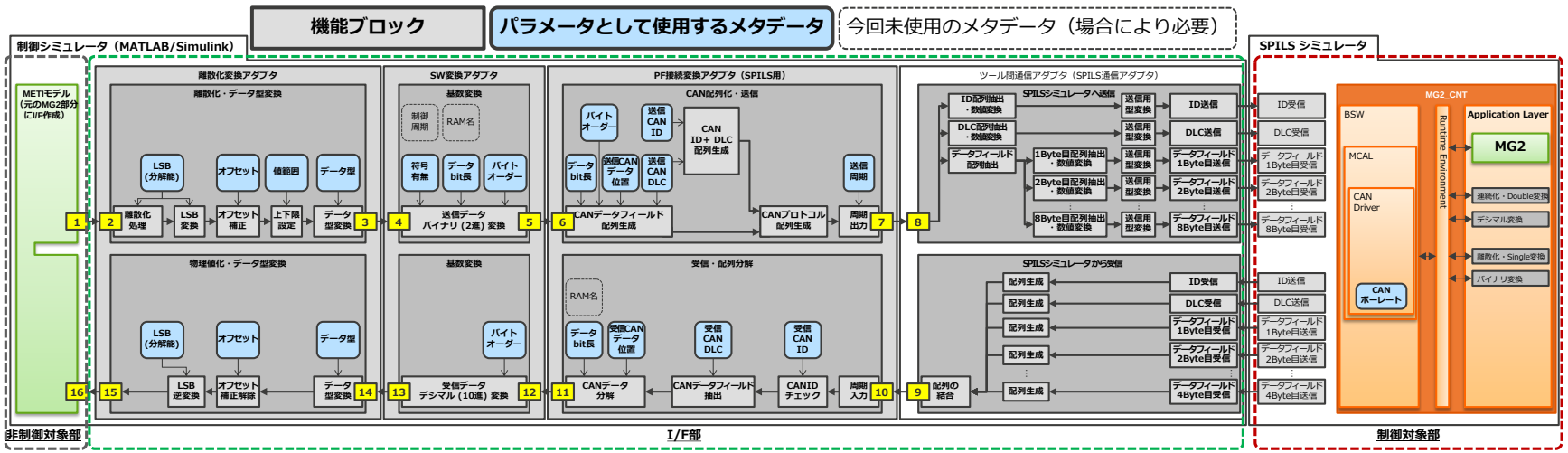


8. モデル接続の具体事例

- 下図に、検討したアーキテクチャと実際に準備したモデルとの対比を示す。

－CAN通信の I/Fの場合｜－モデル接続（2/7）



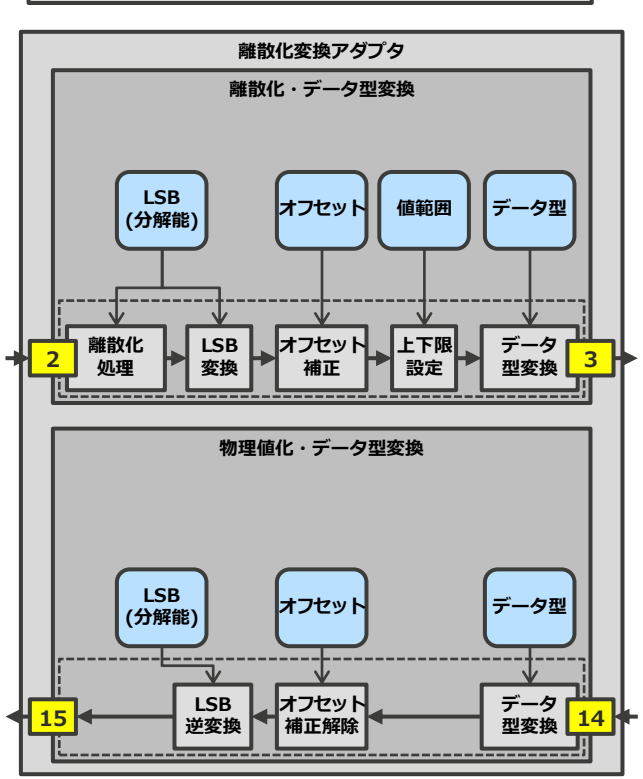


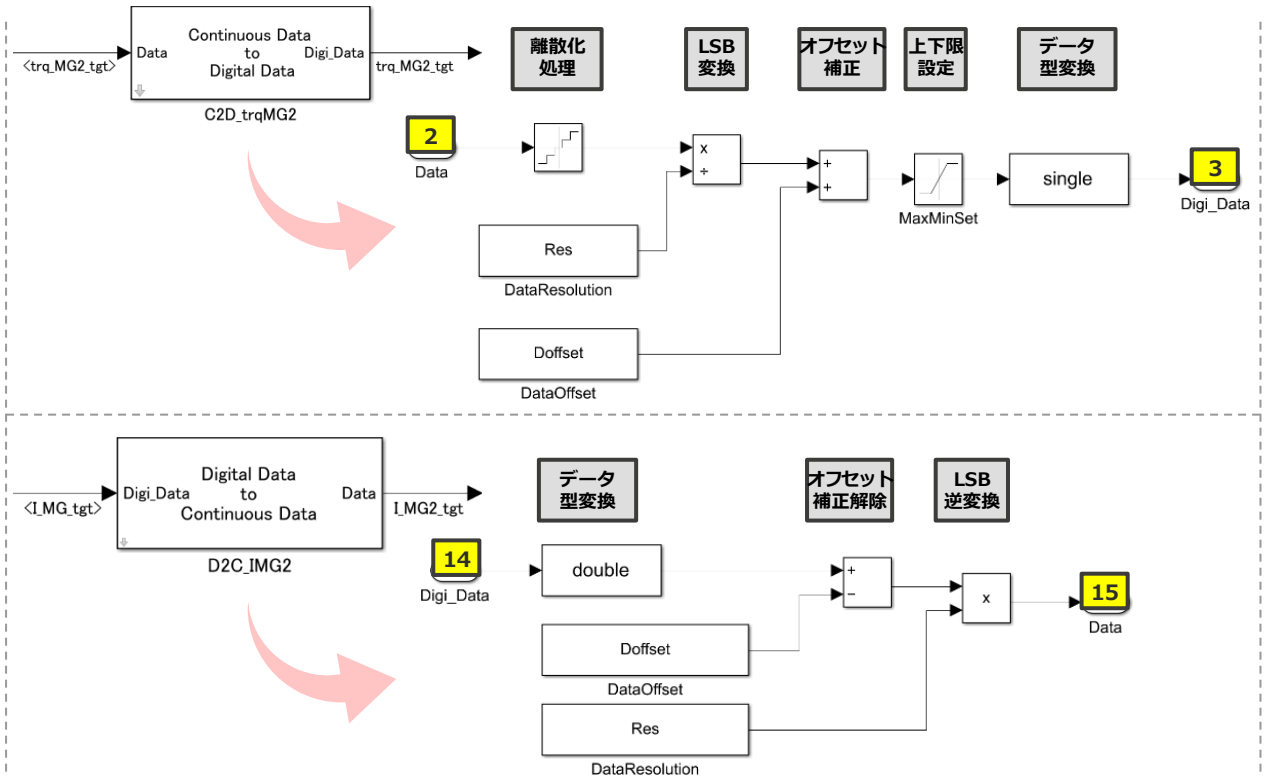
8. モデル接続の具体事例

- 離散化変換アダプタは、現実を想定した物理量（連続量）で検討される制御機能、コンピュータで扱われる離散量で検討される制御機能とを接続するための変換アダプタである。

－CAN通信の I/Fの場合｜－モデル接続（3/7）

離散化変換アダプタ





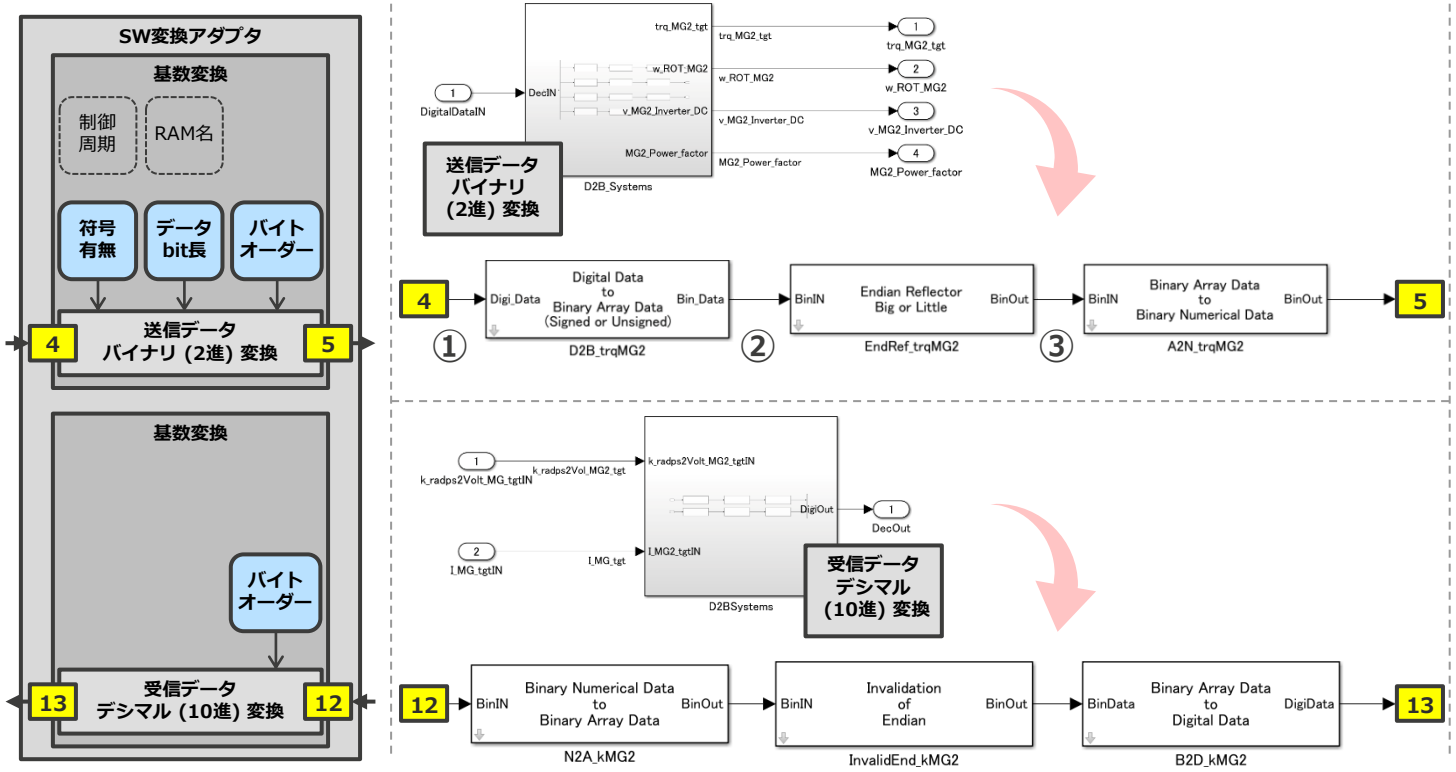
- 左図は、一つの入出力データに着目した際の離散化および連続化の変換アダプタを表す。
- 離散化変換は、データのオフセットや上下限制限、丸めなどが発生する変換の構成としている。
- 連続化変換は、本事例では物理値化変換に簡易化し、補間を省略した。
  また、離散化変換と対比して逆変換の構成とした。

8. モデル接続の具体事例

- SW変換アダプタは、コンピュータの記憶域やデジタル通信を想定し、離散化したデータを2進数（16進数）へ基数変換する、または逆に10進数へ戻すための変換アダプタである。

－CAN通信の I/Fの場合｜－モデル接続（4/7）

SW変換アダプタ



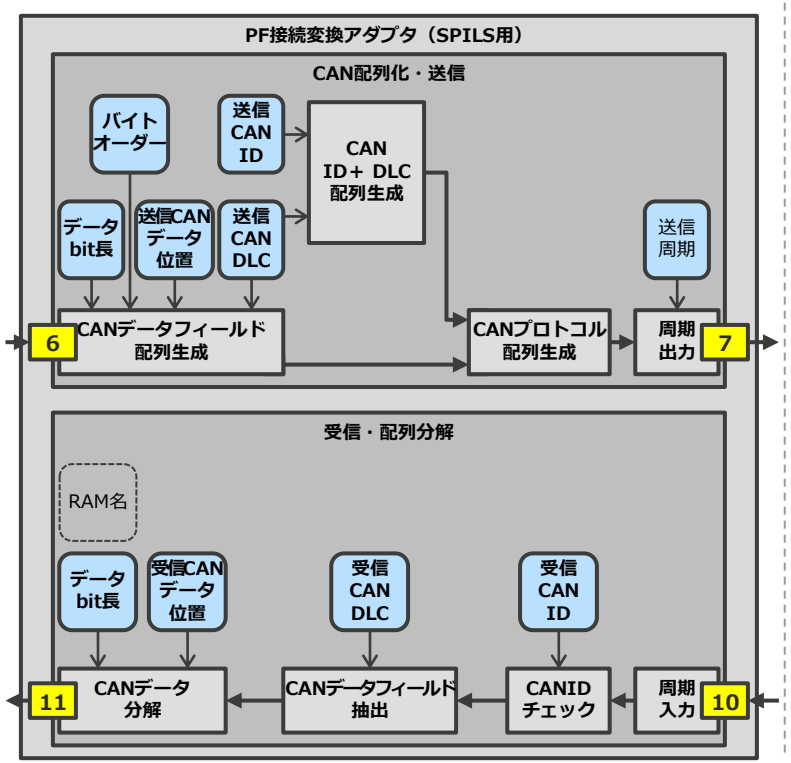
- 今回は、入出力データの基数変換のみを実行している。もし必要があれば、制御周期やデータ格納のためのRAM名の設定などを行ってもよい。
- それぞれのデータ変換について着目すると、バイナリ変換は、離散化された10進数のデータを、①2進数の配列データに変換し、②バイトオーダーに従って配列の順序変換を実行している。加えて、後のデータの扱いを考慮し、③2進数の配列をまとめて一つの数値データへ変換している。
- 本事例でのデシマル変換は、バイナリ変換の逆変換である。

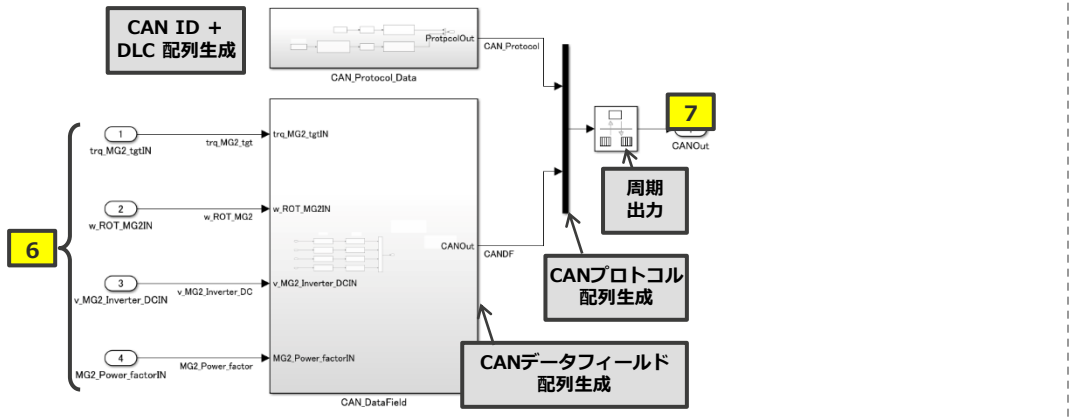
8. モデル接続の具体事例

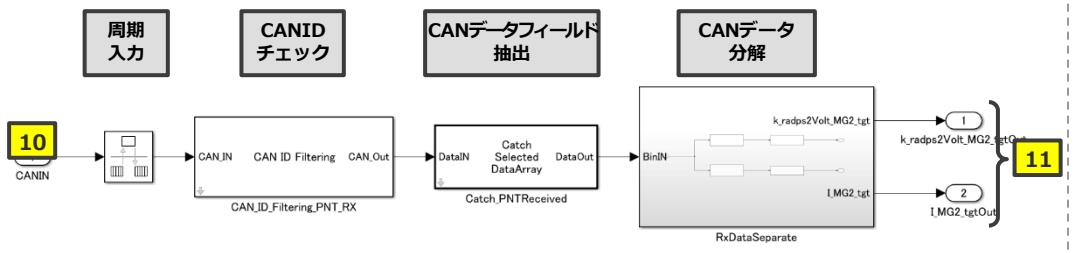
- CAN通信を想定したPF接続変換アダプタは、対象マイクロコントローラ（μC）のCANインターフェースに合わせたデータの授受を可能とするための変換アダプタである。

－CAN通信の I/Fの場合｜－モデル接続（5/7）

PF接続変換アダプタ（1/3）







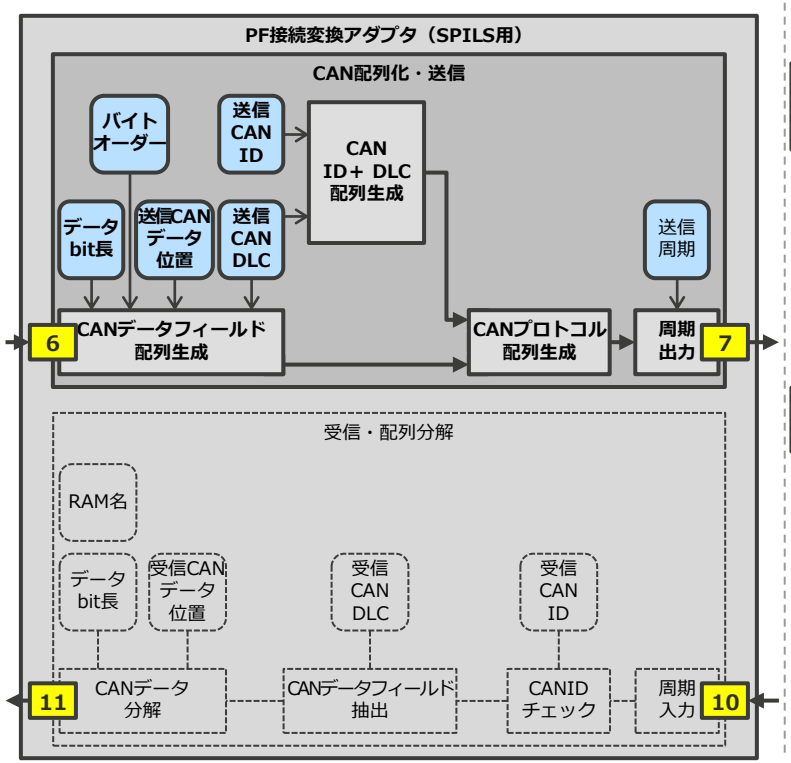
- 本事例では、CAN配列化変換をヘッダーの一部とデータに分割してそれぞれ生成し、合わせて配列としている。これは、ヘッダーやフッターの追加、通信プロトコルの変更を想定した構造としたためである。
- 配列分解は、受信対象のCANフレームであるかを判別した後、データフィールドから個別のデータを抽出し、取り出している。

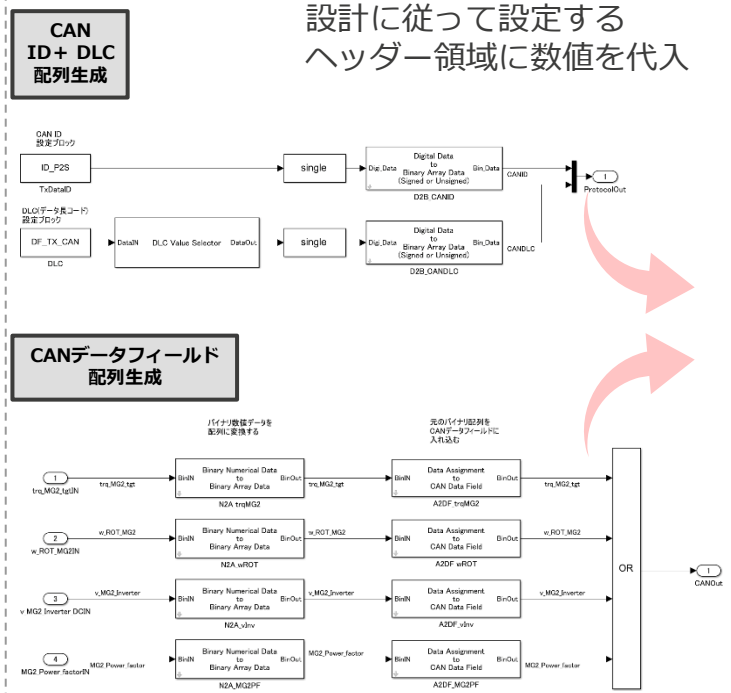
8. モデル接続の具体事例

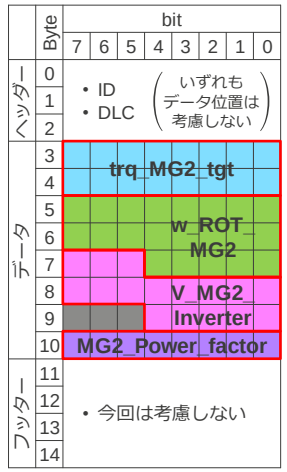
- CAN通信を想定したPF接続変換アダプタは、対象マイクロコントローラ（μC）のCANインターフェースに合わせたデータの授受を可能とするための変換アダプタである。

－CAN通信の I/Fの場合｜－モデル接続（6/7）

PF接続変換アダプタ（2/3）







DLCで設定したデータフィールドのデータ長に合わせてデータの配列を生成

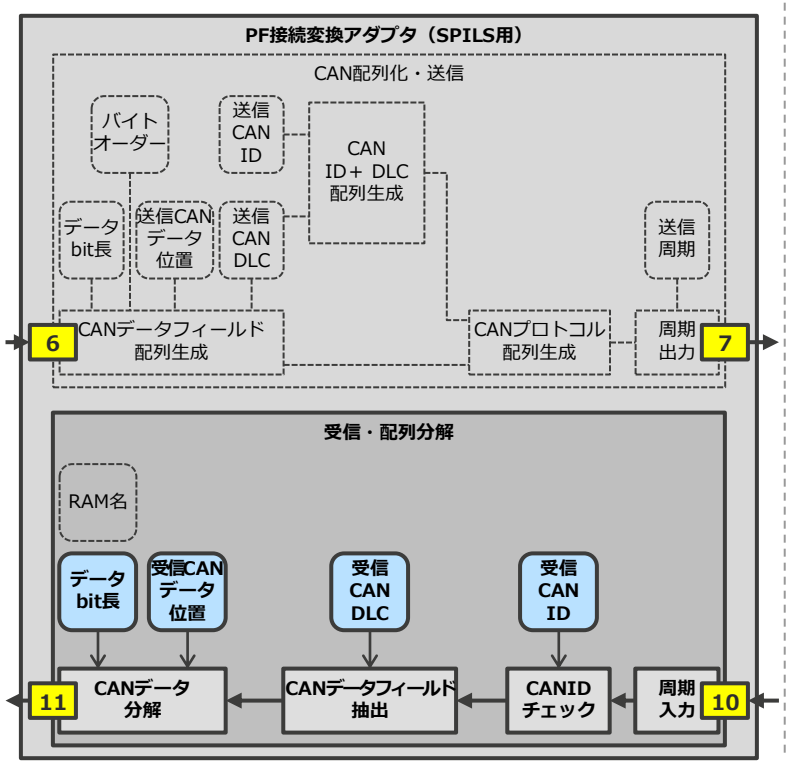
- 本事例では、CAN配列化変換をヘッダーの一部とデータに分割してそれぞれ生成し、合わせて配列としている。これは、ヘッダーやフッターの追加、通信プロトコルの変更を想定した構造としたためである。
- 配列分解は、受信対象のCANフレームであるかを判別した後、データフィールドから個別のデータを抽出し、取り出している。

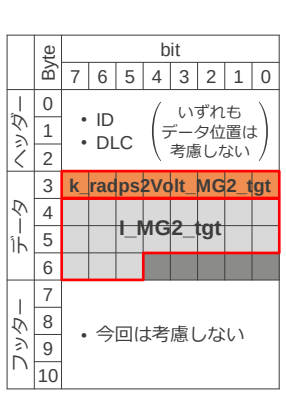
8. モデル接続の具体事例

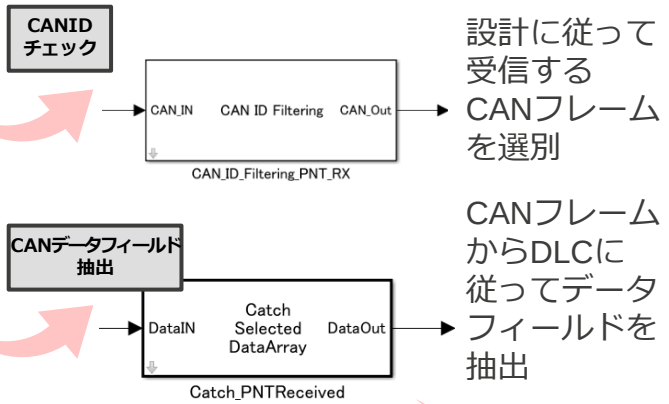
- CAN通信を想定したPF接続変換アダプタは、対象マイクロコントローラ（μC）のCANインターフェースに合わせたデータの授受を可能とするための変換アダプタである。

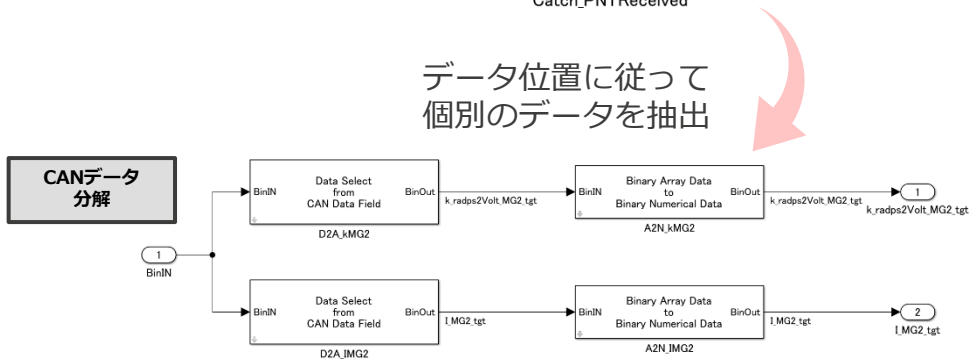
－CAN通信の I/Fの場合｜－モデル接続（7/7）

PF接続変換アダプタ（3/3）









- 本事例では、CAN配列化変換をヘッダーの一部とデータに分割してそれぞれ生成し、合わせて配列としている。これは、ヘッダーやフッターの追加、通信プロトコルの変更を想定した構造としたためである。
- 配列分解は、受信対象のCANフレームであるかを判別した後、データフィールドから個別のデータを抽出し、取り出している。

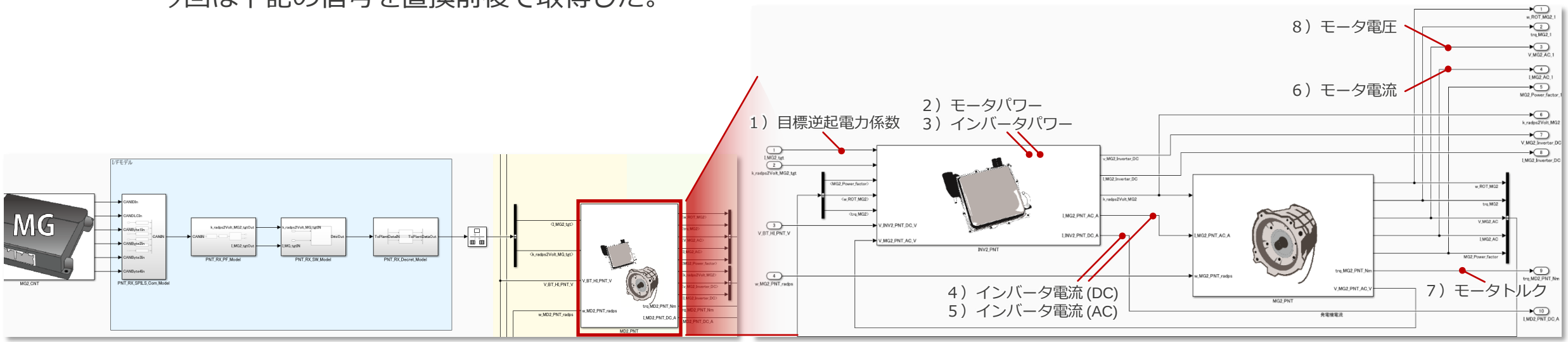
8. モデル接続の具体事例

- METI準拠モデルと今回の事例にて接続したモデルについて、それぞれシミュレーションを実行し、動作確認を実施した。

－CAN通信の I/Fの場合｜－検証（モデル動作確認）

目的 ：a. 制御機能（連続）モデル（METI準拠モデル）の一部を
c. 制御PFモデル（制御APP＆MCAL）に置換したモデル（＝SPILSモデル）について、
元の置換前の制御機能（連続）モデル（＝MILSモデル）との値の傾向を比較する。

方法 ：制御対象であるモータドライブシステムより、同一の信号を置換前後でそれぞれ取得し、値を並べて比較する。
今回は下記の信号を置換前後で取得した。



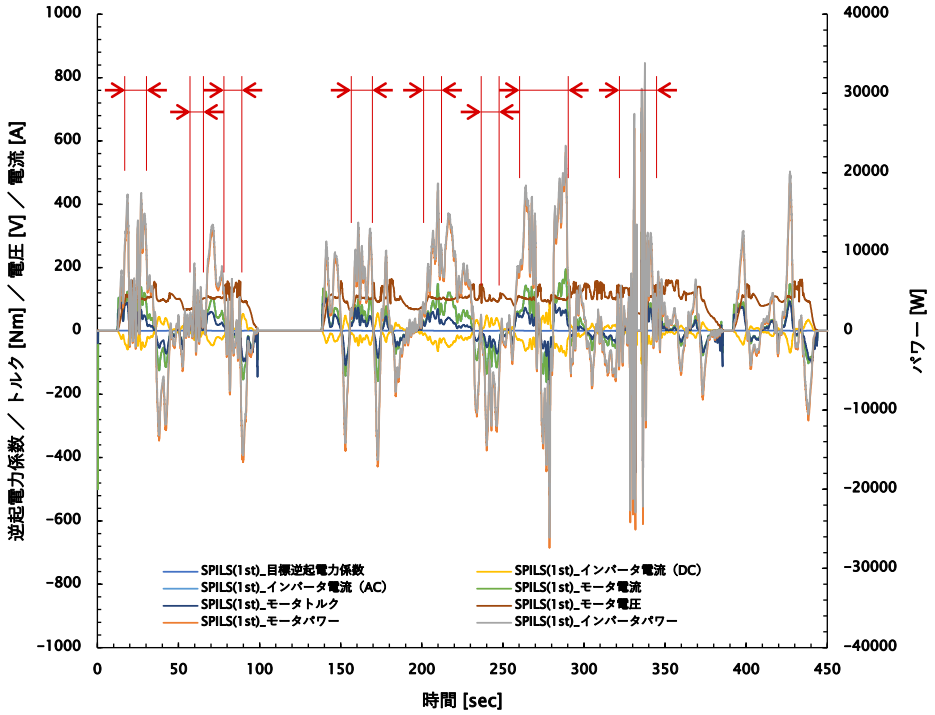
8. モデル接続の具体事例

- 今回の事例にて接続したモデルとMETI準拠モデルについて、それぞれシミュレーションを実行し、動作確認を実施した。

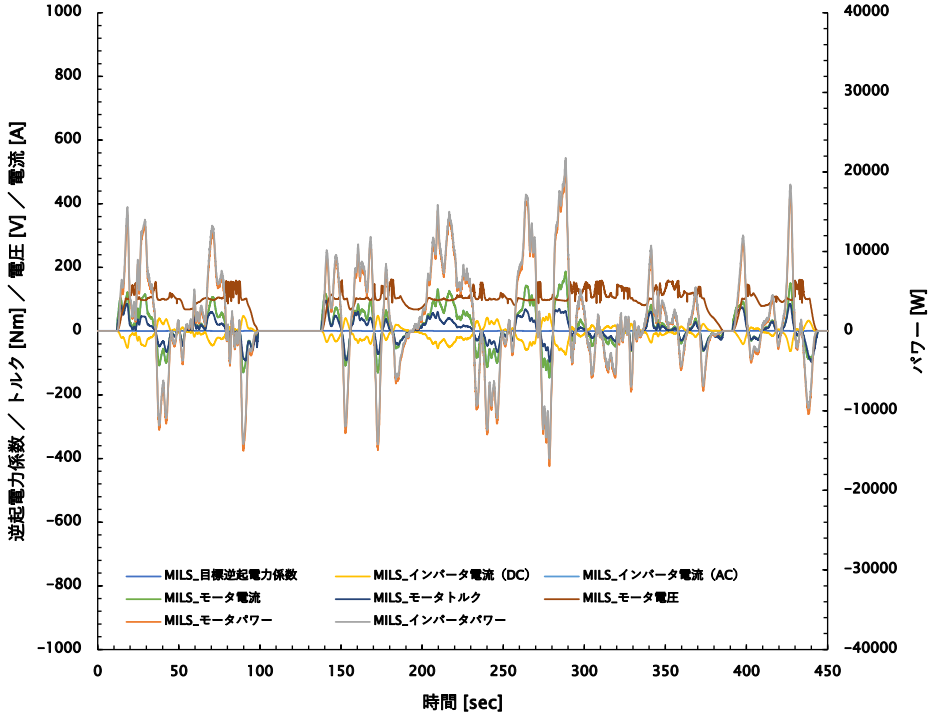
－CAN通信の I/Fの場合｜－検証（モデル動作確認）

結果 ：

SPILS（CAN通信周期 ≒ 190 ms）



MILS



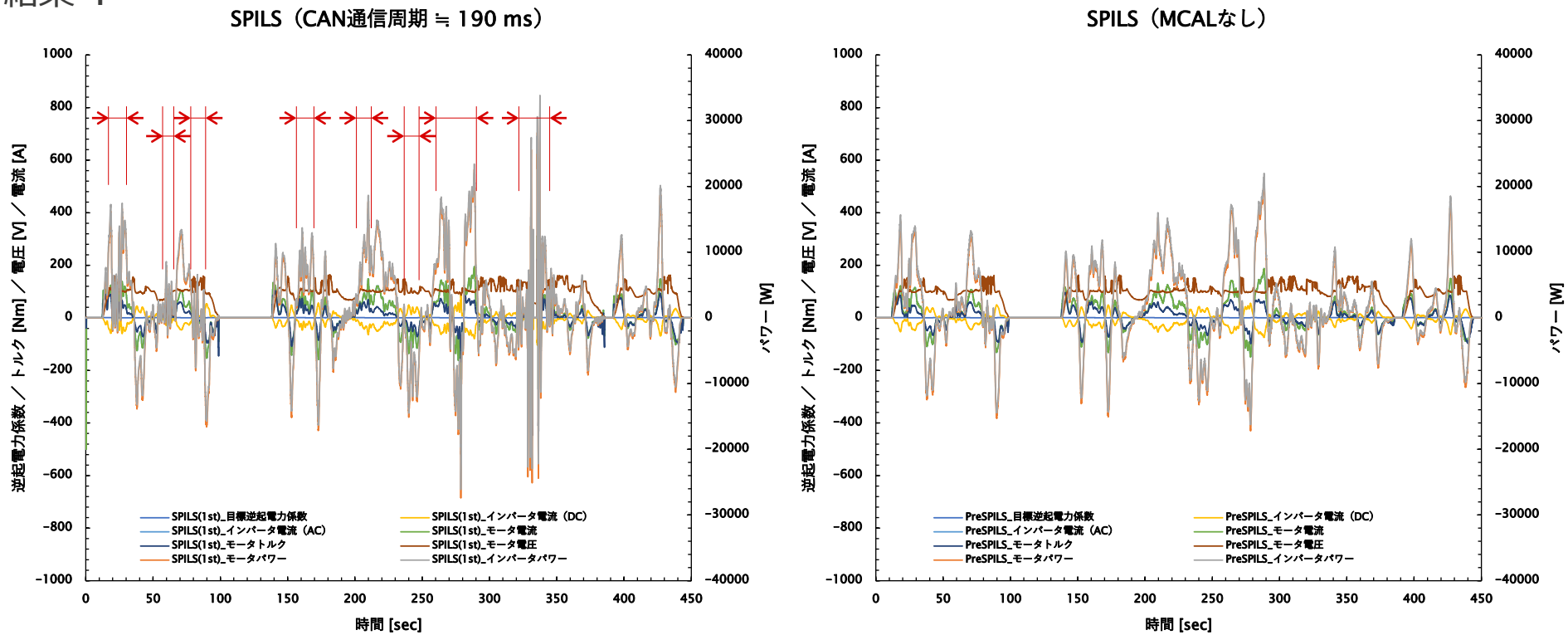
- 左図がSPILSモデルの結果を示している。また、比較のため、右図にモデル変換前のMILSの結果を示す。
- 傾向は一致しており、データ変換アダプタによる接続でシミュレーションを実行可能であることを確認できた。
- **値の変化が著しい箇所で振動が確認**された。

8. モデル接続の具体事例

- 振動の原因を確認するため、前ページと同一の環境、かつMCALを使用しないシミュレーションを実行して、結果を比較した。

－CAN通信の I/Fの場合｜－検証（モデル動作確認）

結果 ：



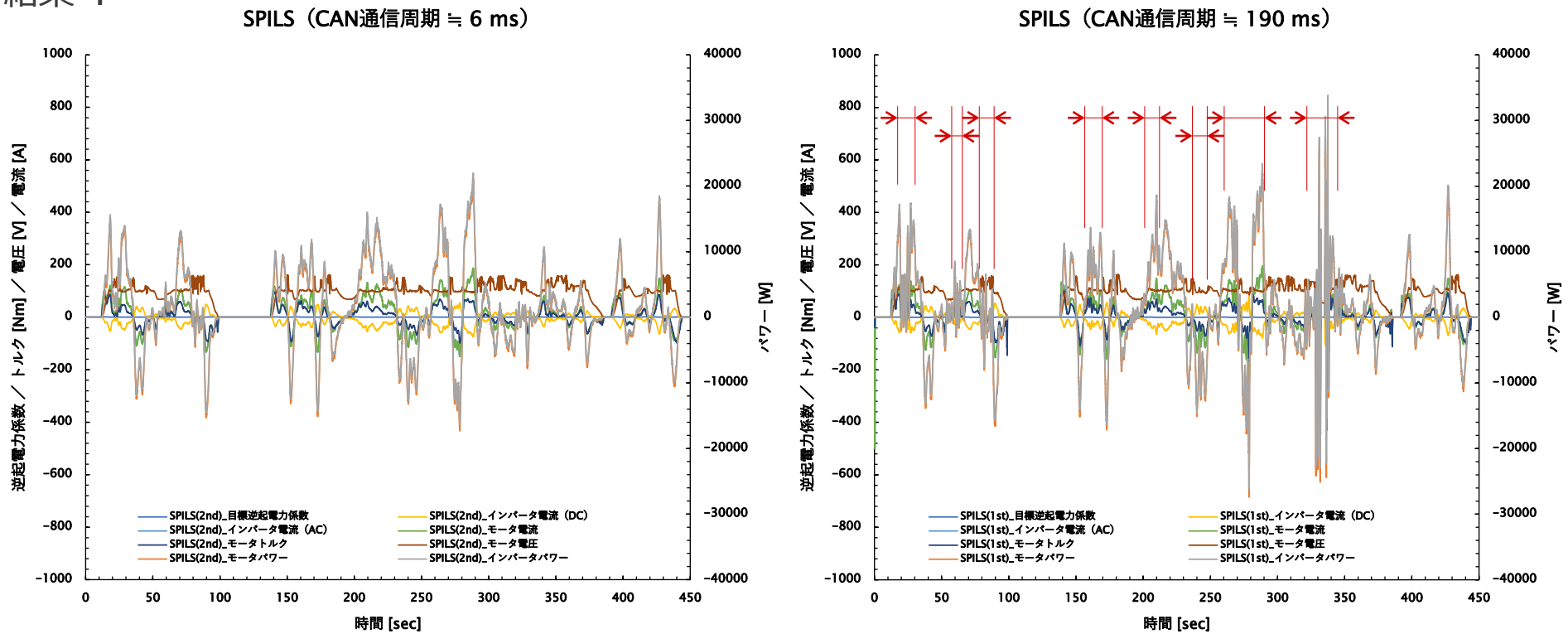
- MCALの影響を確認するため、MCALなしのシミュレーションを実施した。
- 左図はMCALを実装、右図はMCALなしのシミュレーションである。
- 右図に振動がないことからMCAL周りに振動の原因があることが示唆される。
- 振動の原因が、 MCAL側のCAN通信周期の大きさにあると推定し、通信周期の再調整を実施。

8. モデル接続の具体事例

- 振動の原因と推定したSPILSモデル側のCAN通信周期を修正して、再度シミュレーションを実行した。

－CAN通信の I/Fの場合｜－検証（モデル動作確認）

結果 ：



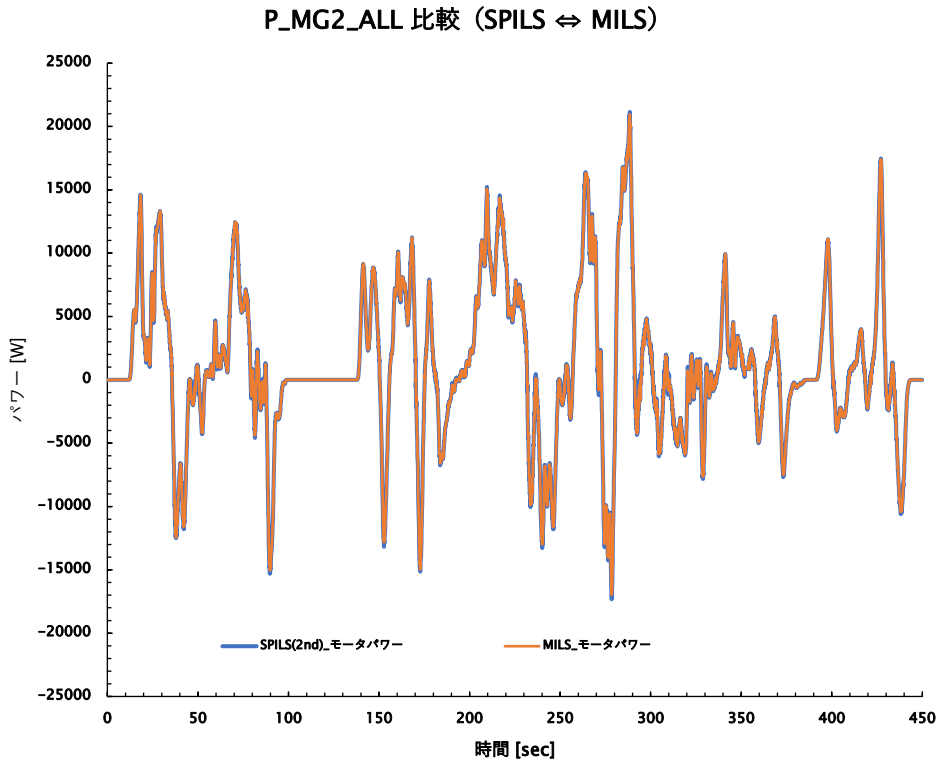
- 左図がCAN通信周期の修正後、右図が修正前の結果をそれぞれ示している。
- 修正の結果、振動が軽減した。
- 修正前の結果が、モデル変換前のMETI準拠モデルの結果に一致するかの確認を実施した。

8. モデル接続の具体事例

- 振動の原因と推定したSPILSモデル側のCAN通信周期を修正して、再度シミュレーションを実行した。
  ⇒ 再実行時の結果を、モデル変換前のMETI準拠モデル実行結果と比較した。

－CAN通信の I/Fの場合｜－検証（モデル動作確認）

結果 ：



- 代表例として、計測したデータのうち
  モータパワーの比較結果を図に示す。
- 結果が一致して重なっていることが
  グラフから読み取れる。
  また、他の結果も同様に一致している。

8. モデル接続の具体事例

－CAN通信の I/Fの場合｜－小まとめ（モデル動作確認）

- 提案したメタデータ、およびメタデータを適用して接続を具体化するためのデータ変換アダプタを使用して、抽象度違いのモデルを接続し、シミュレーションを実行することが出来た。
- また、 1回目の試行にて、SPILSモデル側（制御装置モデル側）の設定に不具合があったことで結果に意図しない振動が発生したが、設定の修正によってモデルの変換前後での結果の一致性も確認することが出来た。
- 上記より、システムレベルの検討に使用するような抽象度の高いモデルに対し、設計が進んだ抽象度の低い制御装置のモデルを接続してシミュレーションを実行することで、制御装置の不具合が反映された全体の挙動を結果として得られることが分かった。
- 今回定義した方法が、実装前の仮想検証に対して有効な方法の一つであることが示唆された。

8. モデル接続の具体事例

- 今回のモデル接続事例を遂行するにあたり、実際に注意を要した内容をとりまとめ、下記に記載した。

－CAN通信の I/Fの場合｜－TIPS（実際にモデル接続を実施してみての注意点）

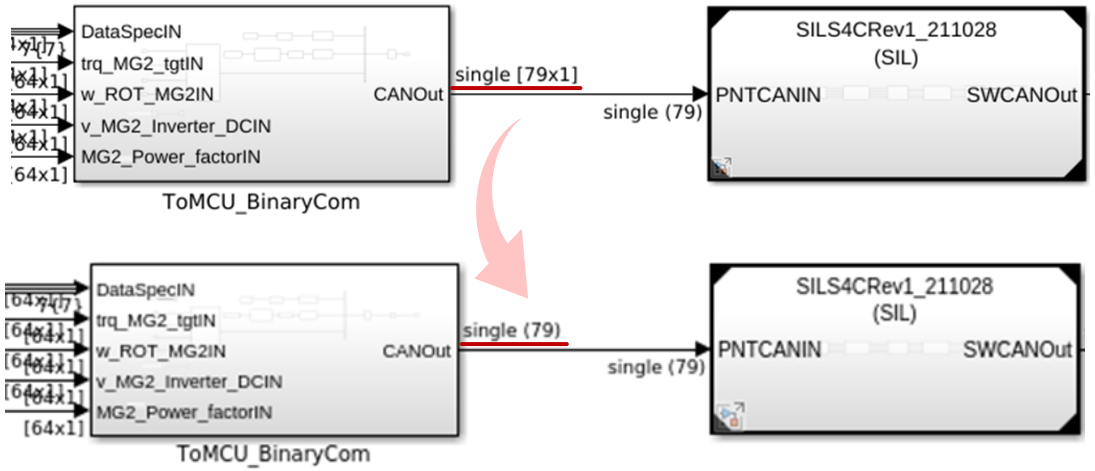
#	発生したトラブル	発生状況			発生箇所	対応	備考
		いつ	どのような方法で	何をしようとした			
	接続 I/F に起因したトラブル						
1	データ型 不一致による 実行エラー	既存のMILSモデルとSILSモデル を接続してシミュレーションを 実行しようとしたとき	特に設定を気にせず そのまま使用して	Simulinkでの シミュレーション実行	MILSモデル ⇒ SILSモデル への入力部	MILSモデル側の出力 データをベクトル型 から配列へ変更した	今回は配列を使用したが、 扱うデータサイズによって、 他の方法にも可能性あり
	その他のトラブル（ガイドラインのスコープ外であるが課題となる可能性の大きい内容）						
2	設定不一致 エラー	既存のMILSモデルとSILSモデル を接続してシミュレーションを 実行しようとしたとき	特に設定を気にせず そのまま使用して	Simulinkでの シミュレーション実行	コンフィギュレーション パラメーター設定	コード生成用モデルの 設定をMILSモデルに 合致させた	マニュアルの作成を推奨 ⇒トラブルの性質上、 ガイドラインには不向き
3	参照先不明 エラー	既存のMILSモデルとSILSモデル を接続してシミュレーションを 実行しようとしたとき	特に設定を気にせず そのまま使用して	Simulinkでの シミュレーション実行	MILSモデルの モニタサブシステム内	モデル内で対象となる 'From' ブロックを 削除した	同上
4	プロセッサ 不一致エラー	SPILSからSILSに立ち戻って シミュレーションをしようと したとき	SPILS用に用意した 制御装置モデルを そのまま使用して	コード生成	コード生成時の 選択項目	RELマイコン⇒ Intel（正しい指定） に設定変更した	I/F起因ではないが、関係 者間をまたぐときに発生 し得るトラブル
5	ビルドエラー	SPILS用の実行ファイルを 生成しようとしていたとき	特に設定を気にせずに 生成されたコードなどを そのまま使用して	開発対象マイコン用の 実行形式ファイルの ビルド	生成コードの データ型とビット長の 設定	開発対象マイコンの定義に 合わせてモデル設定を修正 し、再コード生成した	抽象度違いモデルの接続に、 入出力データのメタデータ 以外情報の必要性も示唆

8. モデル接続の具体事例

- 今回のモデル接続事例を遂行するにあたり、実際に注意を要した内容をとりまとめ、下記に記載した。

－CAN通信の I/Fの場合｜－TIPS（実際にモデル接続を実施してみての注意点）

#	発生したトラブル	発生状況			発生箇所	対応	備考
		いつ	どのような方法で	何をしようとした			
	接続 I/F に起因したトラブル						
1	データ型 不一致による 実行エラー	既存のMILSモデルとSILSモデルを接続してシミュレーションを実行しようとしたとき	特に設定を気にせずにそのまま使用して	Simulinkでのシミュレーション実行	MILSモデル ⇒ SILSモデル への入力部	MILSモデル側の出力データをベクトル型から配列へ変更した	今回は配列を使用したが、扱うデータサイズによって、他の方法にも可能性あり



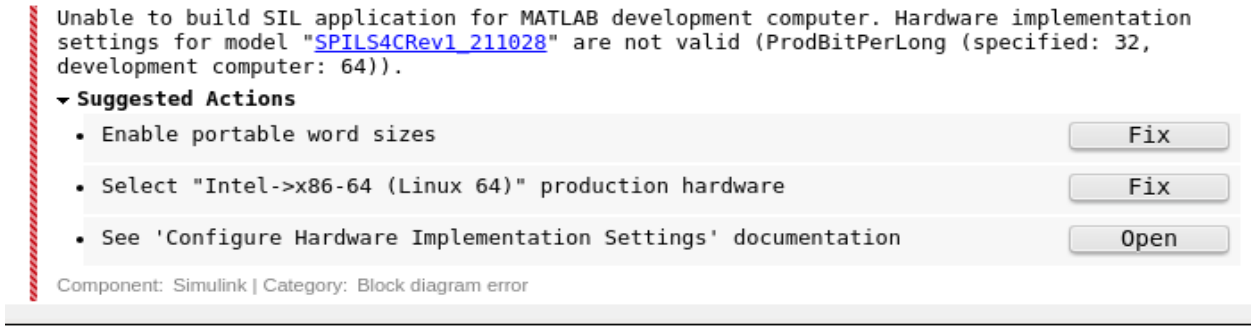
- 接続 I/F に起因のする内容のため、本ガイドラインでの引き続きの検討を要する。
- 2進数データを整数に再変換して送信する方法も考えられるが、今回は桁落ちが発生したため配列を利用した。データの送り方は複数の方法が考えられるため、検討を継続する。

8. モデル接続の具体事例

- 今回のモデル接続事例を遂行するにあたり、実際に注意を要した内容をとりまとめ、下記に記載した。

－CAN通信の I/Fの場合｜－TIPS（実際にモデル接続を実施してみての注意点）

#	発生したトラブル	発生状況			発生箇所	対応	備考
		いつ	どのような方法で	何をしようとした			
	その他のトラブル（ガイドラインのスコープ外であるが課題となる可能性の大きい内容）						
2	設定不一致 エラー	既存のMILSモデルとSILSモデルを接続してシミュレーションを実行しようとしたとき	特に設定を気にせずにそのまま使用して	Simulinkでのシミュレーション実行	コンフィギュレーションパラメーター設定	コード生成用モデルの設定をMILSモデルに合致させた	マニュアルの作成を推奨 ⇒トラブルの性質上、ガイドラインには不向き



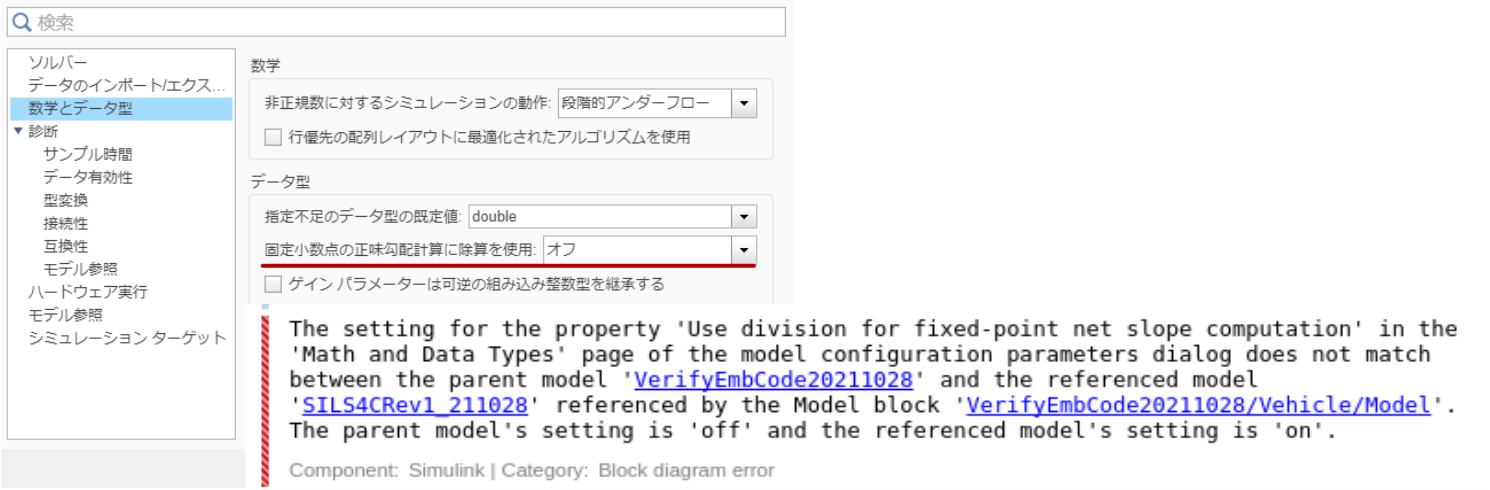
- 生成したソースコードが、どのハードウェア（開発対象マイコン／汎用CPU／…etc.）に向けたものか、関係者間で確認されていない場合に、同様のトラブルが発生し得る。

8. モデル接続の具体事例

- 今回のモデル接続事例を遂行するにあたり、実際に注意を要した内容をとりまとめ、下記に記載した。

－CAN通信の I/Fの場合｜－TIPS（実際にモデル接続を実施してみての注意点）

#	発生したトラブル	発生状況			発生箇所	対応	備考
		いつ	どのような方法で	何をしようとした			
	その他のトラブル（ガイドラインのスコープ外であるが課題となる可能性の大きい内容）						
3	参照先不明 エラー	既存のMILSモデルとSILSモデルを接続してシミュレーションを実行しようとしたとき	特に設定を気にせずにそのまま使用して	Simulinkでのシミュレーション実行	MILSモデルのモニタサブシステム内	モデル内で対象となる‘From’ブロックを削除した	マニュアルの作成を推奨 ⇒トラブルの性質上、ガイドラインには不向き



- 抽象度違いのモデルを接続する際に注意する点であるため、TIPSに立項して注記した。
- 必要であれば、マニュアル作成を推奨する。

8. モデル接続の具体事例

- 今回のモデル接続事例を遂行するにあたり、実際に注意を要した内容をとりまとめ、下記に記載した。

－CAN通信の I/Fの場合｜－TIPS（実際にモデル接続を実施してみての注意点）

#	発生したトラブル	発生状況			発生箇所	対応	備考
		いつ	どのような方法で	何をしようとした			
	その他のトラブル（ガイドラインのスコープ外であるが課題となる可能性の大きい内容）						
4	プロセッサ 不一致エラー	SPILSからSILSに立ち戻って シミュレーションをしようと したとき	SPILS用に用意した 制御装置モデルを そのまま使用して	コード生成	コード生成時の 選択項目	RELマイコン⇒ Intel（正しい指定） に設定変更した	I/F起因ではないが、関係 者間をまたぐときに発生 し得るトラブル

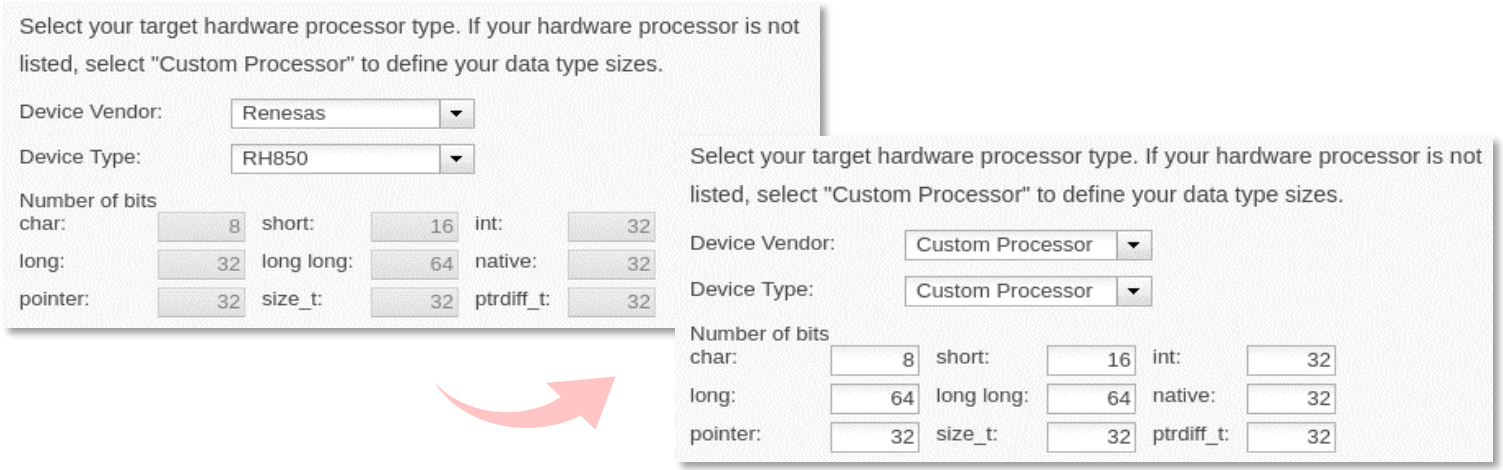
- 観測するデータのモニタリング方法に関する内容である。
- データのモニタリングについて、抽象度違いのモデル間での信号授受が発生する場合について、必要であればマニュアル作成を推奨する。

8. モデル接続の具体事例

- 今回のモデル接続事例を遂行するにあたり、実際に注意を要した内容をとりまとめ、下記に記載した。

－CAN通信の I/Fの場合｜－TIPS（実際にモデル接続を実施してみての注意点）

#	発生したトラブル	発生状況			発生箇所	対応	備考
		いつ	どのような方法で	何をしようとした			
	その他のトラブル（ガイドラインのスコープ外であるが課題となる可能性の大きい内容）						
5	ビルドエラー	SPILS用の実行ファイルを生成しようとしていたとき	特に設定を気にせずに生成されたコードなどをそのまま使用して	開発対象マイコン用の実行形式ファイルのビルド	生成コードのデータ型とビット長の設定	開発対象マイコンの定義に合わせてモデル設定を修正し、再コード生成した	抽象度違いモデルの接続に、入出力データのメタデータ以外情報の必要性も示唆



- 関係者間で本項目のように、モデルやソースコードの内部情報の確認プロセスが必要な場合を想定して立項した。
- 入出力データのメタデータ以外にも考慮が必要な項目が存在することが示唆される。