

MBDツールによるAUTOSAR Adaptive Platform向け アプリケーション開発事例

2023/5/31

パーソルクロステクノロジー株式会社：高橋 良輔

名古屋大学 A2Pコンソーシアム：高田 光隆, 松原 豊

目次

1. AUTOSARとは
2. AUTOSAR Adaptive Platformに関する
コンソーシアム型共同研究の活動紹介
3. MATLAB[®]/Simulink[®]を使用した開発
4. 今後の展望

1. AUTOSARとは

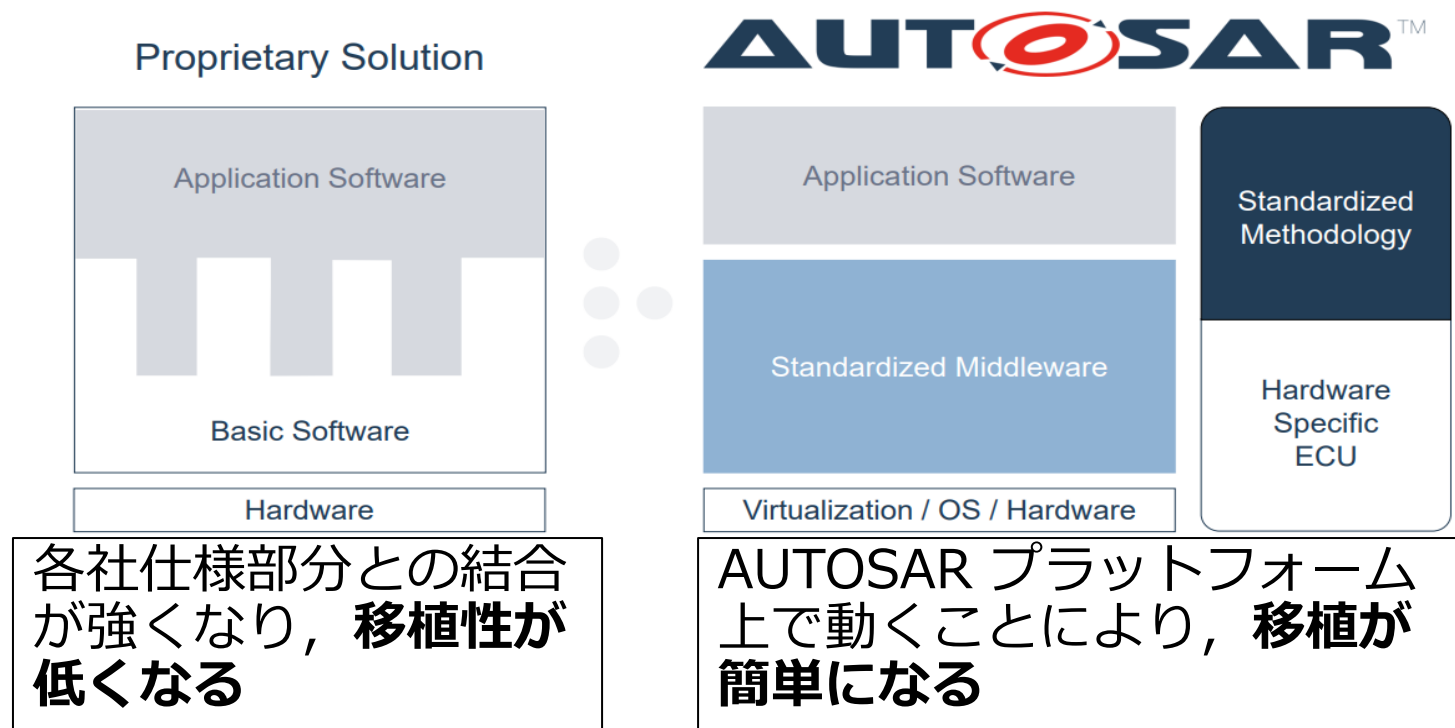
AUTOSARとは

- **AUT**omotive **O**pen **S**ystem **AR**chitecture の略
- 2003年7月、ドイツの自動車メーカー(OEM)とTier1サプライヤを中心に設立された、**車載ソフトウェアプラットフォームの仕様の標準化**を主な目的とした、自動車業界のグローバルパートナーシップを指す
- 2008年頃から広く量産車に適用（現在のClassic Platform）され始め、欧州での新規開発では適用される機会が増加している
- 現在は、7つの異なる参加形態(**パートナーシップ**)に分かれ、世界各国のOEM、Tier1サプライヤ、半導体メーカー、ソフトウェアベンダ、ツールベンダ等、**350以上の企業、団体が参加している**
- **従来の制御系を処理する電子制御ユニット（ECU）**を主な対象にする「**Classic Platform（CP）**」、**自動運転や先進運転支援システム（ADAS）**などを対象とする「**Adaptive Platform（AP）**」の2種類が存在する
 - Classic Platform 2003年～
 - Adaptive Platform 2016年～



AUTOSARのモチベーション

- 機能拡大に伴う、**車載ソフト/ハードの複雑さ**を管理する
- **製品の変更、更新、改良を柔軟にサポート**する
- 製品ライン内/間のソリューションのスケールビリティ
- 機能の統合/乗換の**拡張性、柔軟性**を高める
- 車載ソフト/ハードの**品質と信頼性**を改善する

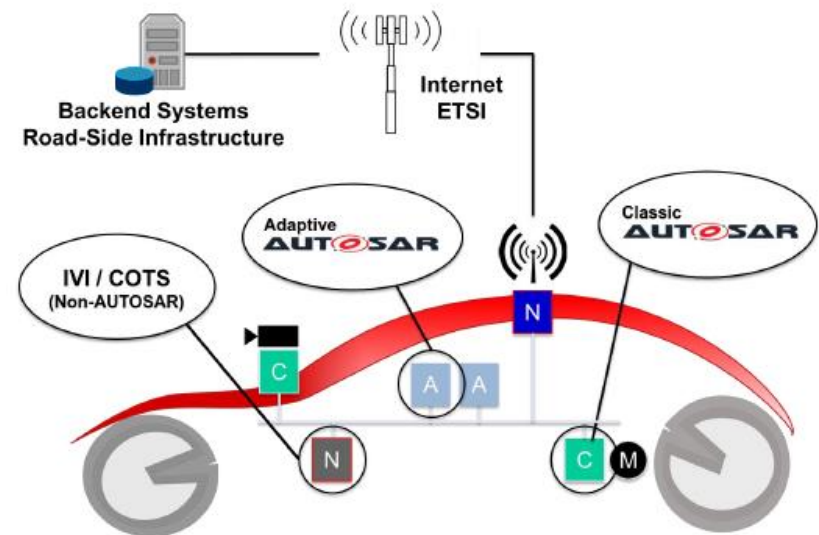


https://www.autosar.org/fileadmin/user_upload/AUTOSAR_EXP_Introduction_Part1.pdf

Adaptive Platform 開発の背景

- 先進運転支援システム（ADAS）や自動運転など、高機能・高性能で、かつ高い安全性が求められる **次世代ECU向け**として、CPよりも、**機能が豊富で柔軟性の高い**ソフトウェアプラットフォームに対する需要が高まってきた
- 上記に対応したソフトウェアプラットフォームを **Adaptive Platform**と位置づけ、**仕様とプロトタイプ実装**を進めている

CP, AP, それ以外のソフトウェアプラットフォームが混在する状況を想定



AUTOSAR_EXP_PlatformDesign.pdf

2. AUTOSAR Adaptive Platform に関するコンソーシアム型共同研究 の活動紹介

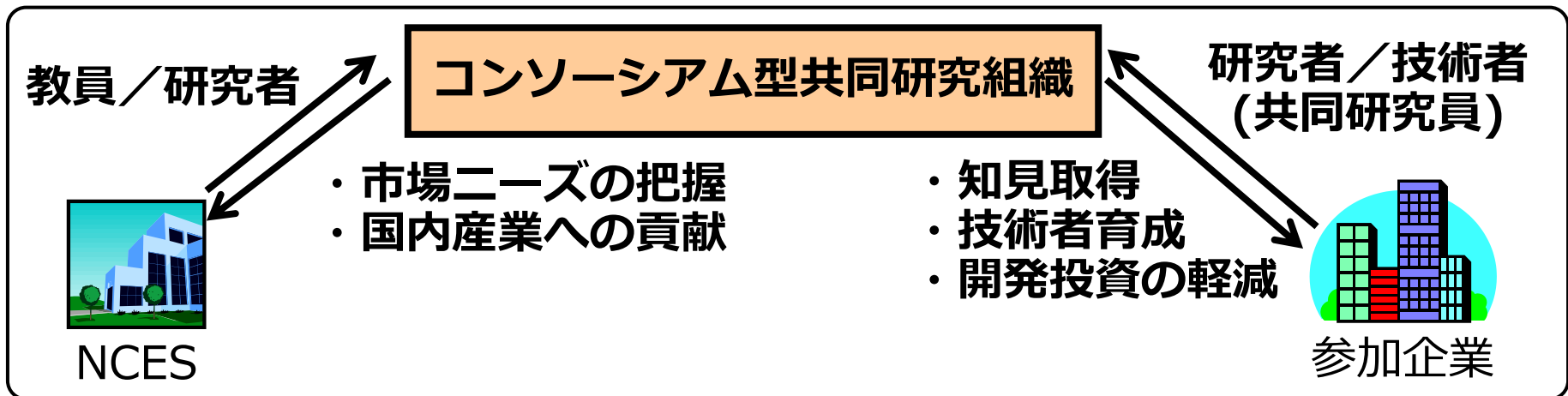
AUTOSARに対するNCESの取組み

NCESとは

- 名古屋大学 大学院情報学研究科 附属組込みシステム研究センター
- 大学が持つ技術シーズを用いて産業界が必要とする技術課題を解決する産学連携を基本とした組織

日本におけるAUTOSARの課題解決のため、NCESと複数企業（のべ40社以上）により**コンソーシアム型共同研究組織**を設立

2011年度～2013年度	： ATK2コンソーシアム
2014年度～2016年度	： APコンソーシアム
2017年度～	： A2Pコンソーシアム
2018年度～2021年度	： APToolコンソーシアム



A2Pコンソーシアム

• AUTOSAR Adaptive Platformに関する コンソーシアム型共同研究

- AUTOSAR Adaptive Platform (A2P) を対象とし、過去にNCESで実施したAUTOSARのコンソと同様のコンソーシアム型共同研究

• 研究開発の目標

- AUTOSAR Adaptive Platform(AP)プロトタイプ開発に参画し, APのノウハウを蓄積
- 参加企業におけるAPを使用したシステム開発への技術的支援
- 日本のAUTOSARに対する貢献をアピール

AUTOSAR AP導入初期における課題

1 : 仕様書などの文書数が膨大

- 仕様書数が膨大（FO: 約40本, AP: 約60本）, かつ各仕様書もページ枚数が多い（Specification of Communication Managementは500ページ!）
- どれから読み始めるのが良いのか分からない

2 : デモンストレータ（APD）を「とりあえず動かしてみる」敷居が高い

- 同梱されているサンプルのビルド&実行エラー（最近のリリースでは改善されたが…）
- サンプルの構成把握が難しく, 改造が面倒

3 : アプリケーション開発環境が提供されていない

- 設定ファイル（ARXML(AUTOSAR XML)）を手作業で編集
 - ARXMLの仕様が膨大ですべてを把握するのは困難
- 本研究活動で作業が軽減

現状の課題：AP開発の立ち上げコストが非常に高い！

仕様翻訳書, 解説書の作成

APの仕様書

仕様書の入手先

- https://www.autosar.org/nc/document-search/?tx_sysgsearch_pi1%5Bquery%5D=&tx_sysgsearch_pi1%5Bcategory%5D%5B146%5D=146

R20-11リリース場所

仕様書の構成

フォルダ名	内容
AdaptiveFoundation	APの基本的な機能について
AdaptiveService	APの標準サービスについて
General	全体概要/概念について
MethodologyAndManifest	AP実現の方法論と方針
ReleaseDocumentation	リリースドキュメント

それぞれのフォルダには、AUTOSAR_XXで始まるファイルがあり、XXの部分で、内容の種類を判別できる
例: AUTOSAR_EXP_ARACoMAPI.pdf

文書の種類

AUTOSAR_XX	内容
RS	Requirement Specification ソフトウェア仕様要件
EXP	Explanation 他ドキュメントにも記載された説明資料
SWS	Software Specification ソフトウェア仕様
MMOD	MetaModel
TPS	Template Specification テンプレート仕様
TR	Technical Report テクニカルレポート

CM概要 -サービス指向通信- (3/3)

- サービスディスカバリのインターフェース：
通信相手との間(ECU内/ECU間)の通信経路の動的な設定を行う

<サービスディスカバリによる通信経路確立>

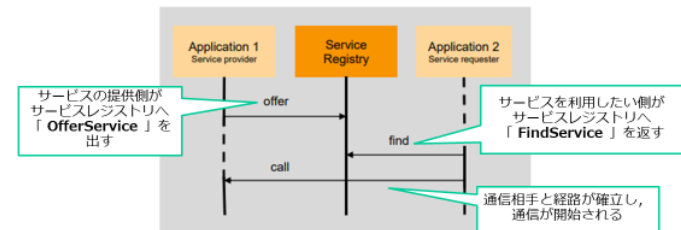


Figure 7.2: Service-Oriented Communication

AUTOSAR_SWS_CommunicationManagement.pdf

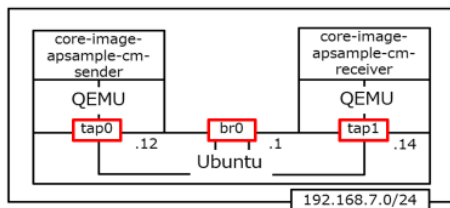
QEMU 実行方法 (1/5)

- ビルド環境の構築, ビルドの実施まではRaspberryPiと同様の手順を行う

- QEMUで実施する場合は下記のコマンドを実施し, ツールをインストールする

```
$ sudo apt-get install qemu bridge-utils uml-utilities
```

- ECU間の通信を行う場合は, 下記図の通り, tapとbridgeを使用し接続を行う



演習4 CM (受信側) (1/8)

- 演習4で使用するファイル

```
WS/sample-applications/apsamples/  
APSampleCmReceiver/  
• src/apsample_cm_receiver.cpp  
• src/apsample_cm_receiver_activity.cpp  
• src/main_apsample_cm_receiver.cpp  
• include/apsample_cm_receiver.hpp  
• include/apsample_cm_receiver_activity.hpp
```

- 演習4-1 ヘッダのインクルード

ARXMLファイルから自動生成されたヘッダは, 以下のディレクトリに生成される

下記ヘッダファイルをインクルードする

```
nces/apsample/cm/apsamplecm_proxy.h
```

(解答ファイル: apsample_cm_receiver_activity.hpp)

Adaptive Platform Demonstrator

- **AUTOSAR AP**では標準化活動として、仕様開発の他に、リファレンスコードとして**APD(Adaptive Platform Demonstrator)**を開発している

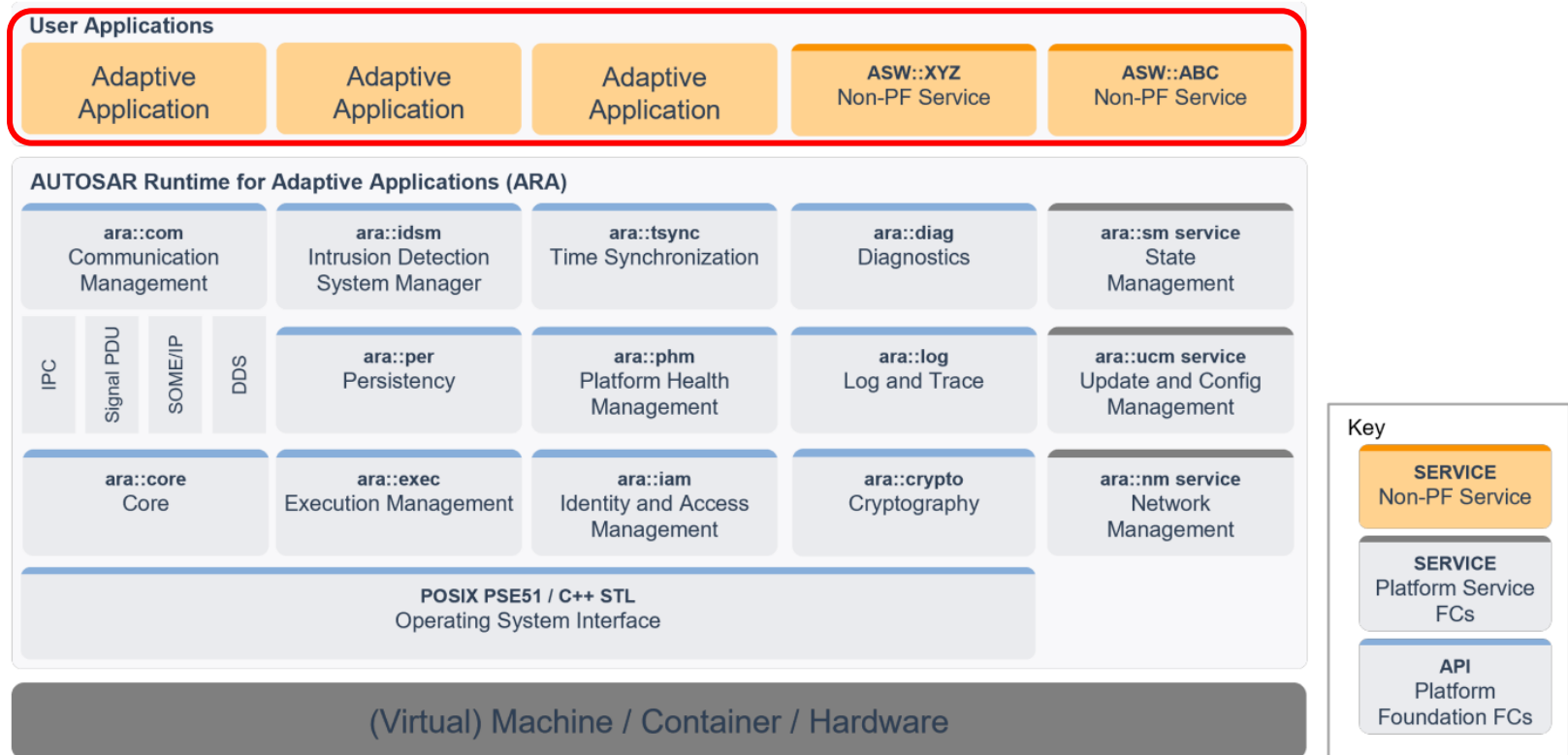


Figure 4.1: AP architecture logical view

AUTOSAR_EXP_PlatformDesign.pdf

サンプルプログラムの開発 : APSample

APDを基に，基本的な機能理解用のサンプルプログラムの開発，APD用環境構築手順書を作成

演習1 EM (2/6)

・演習1-1 ヘッダをインクルード

EMのAPIは下記のヘッダファイルで定義されている
下記ヘッダファイルをインクルードする

```
<ara/exec/execution_client.h>
```

(解答ファイル : `apsample_em.hpp`)

・演習1-2 ExecutionClientクラスをインスタンス化する

EMにアプリケーションステートを報告するAPIは，下記のクラスで定義されている

```
namespace ara {  
  namespace exec {  
    class ExecutionClient {  
      // ara::exec::ExecutionClient  
      //クラスのインスタンス化は<クラス名> <インスタンス名>;のように記述する  
    }  
  }  
}
```

このクラスを任意のインスタンス名でインスタンス化する

※コンストラクタに処理は無いため，そのままインスタンス化が可能

(解答ファイル : `apsample_em.hpp`)

APSample一覧

CONFIDENTIAL
関係者以外

- ・現在用意しているAPSampleは以下の通り
- ・通信対象も決められているため，実施時は下記を参照しそれぞれデプロイする
(デプロイ方法，実行方法は後述)

ECU1 (192.168.7.12)	ECU2 (192.168.7.14)
APSampleEM	※1
APSampleEm2	※1
APSampleCmSender	APSampleCmSender
APSampleCmProvider	APSampleCmProvider
APSampleCmFieldSender	APSampleCmFieldSender
APSampleDiagCounter	※2



※1 EM, EM2はECU2通信しないため，ECU2側に対象アプリが存在しない (詳細はAPSampleEm詳細.pptxを参照)

※2 DiagはPCと通信するため，ECU2側に対象アプリが存在しない (詳細はAPSampleDiag詳細.pptxを参照)

NCES 名古屋大学大学院情報科学研究科
自動車システム研究センター

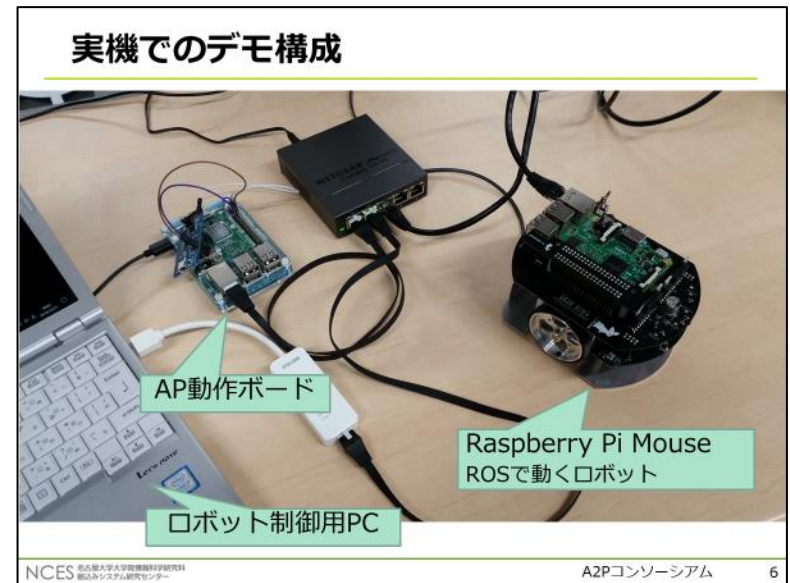
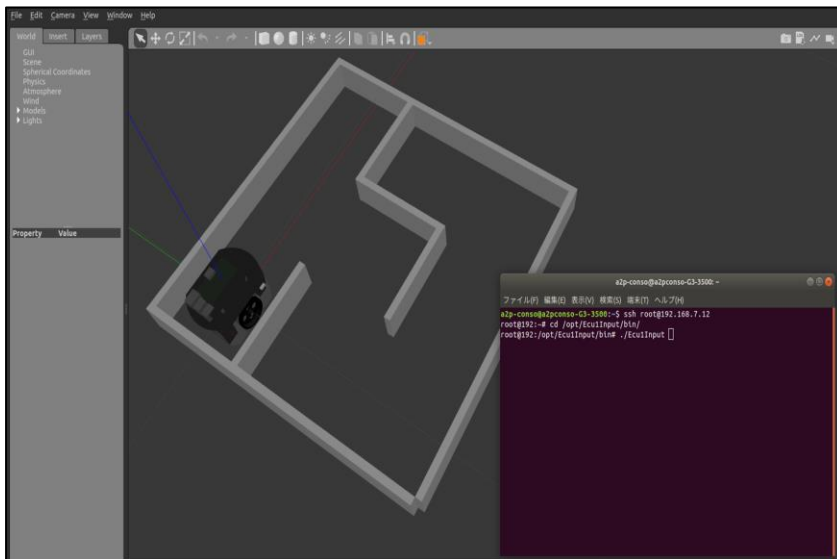
A2Pコンソーシアム

25

同サンプルプログラムを演習に使用した社会人向け講座 (NEP AUTOSAR AP入門) を開催

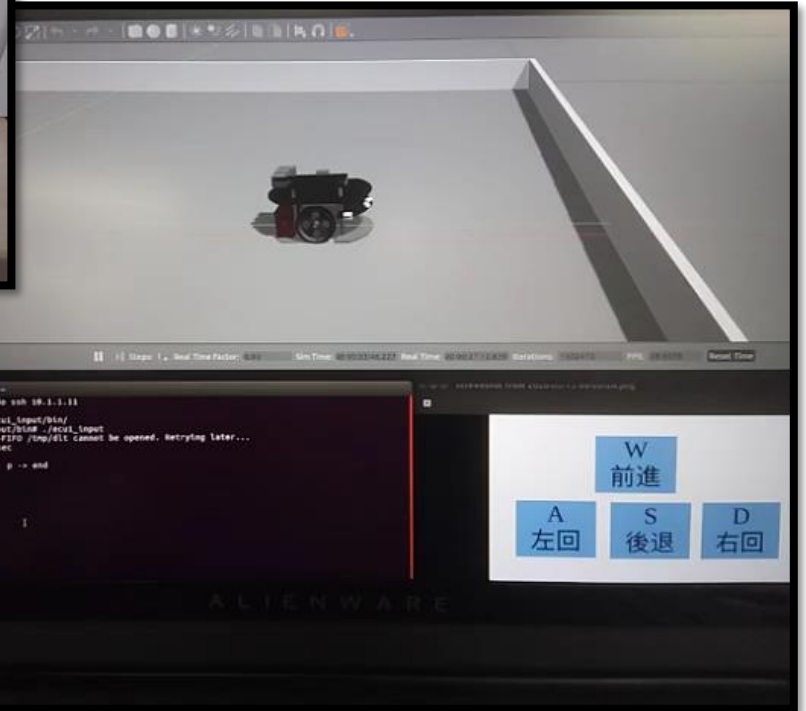
RaspberryPiMouse用デモプログラム

RaspberryPiMouse実機を使用した場合と，ROSシミュレータ(Gazebo)を使用した場合の2パターンに対応するラズパイマウスデモプログラムを作成



AUTOSAR(AP)の基礎知識を基に，アプリケーションを開発する能力を獲得することを目的とし，ラズパイマウスデモプログラムを使用した学習教材を作成

RaspberryPiMouse用デモプログラム



3. MATLAB/Simulink を使用した開発

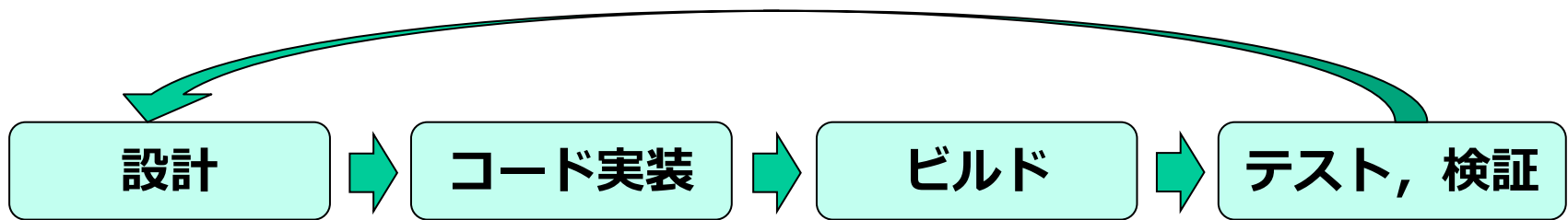
パーソルクロステクノロジー紹介

社名	パーソルクロステクノロジー株式会社 (PERSOL CROSS TECHNOLOGY CO., LTD.)
代表者	代表取締役社長 正木 慎二
本社	東京都新宿区西新宿 2-1-1 新宿三井ビルディング 5 1 階
設立	1979年10月11日
資本金	4億9,500万円
従業員数	9,706名 (2022年12月時点) ※当社所属の全雇用形態の従業員数
事業内容	テクノロジーソリューション事業 自動車・航空・家電・ロボットなどの設計・開発・実験における請負・派遣サービス ITシステムやアプリケーションのシステム開発・インフラ設計・運用における派遣・準委任・フリーランスサービス AIやDXを活用したIoT、モビリティサービスの導入支援
ホームページ	https://persol-xtech.co.jp/
企業資格	労働者派遣事業許可番号「派13-316579」 有料職業紹介事業許可番号「13-ユ-315141」 ISO9001*、プライバシーマーク
親会社	パーソルホールディングス株式会社



*ISO9001:2015 取得拠点 名古屋オフィス、上尾オフィス/上尾R&Dセンター、横浜オフィス/横浜R&Dセンター、刈谷R&Dセンター、名古屋R&Dセンター、栃木さくら事業所、刈谷テストセンター、宇都宮オフィス/宇都宮R&Dセンター、大阪オフィス/大阪R&Dセンター、神戸オフィス

AP用アプリ開発の課題解決に向けて



これら一般的なプロセスに対してAUTOSAR導入初期では
アプリ開発のハードルが高い

A2Pコンソ活動として改善に向けた成果物を作成

APDを使用した成果物作成時の課題として

- 1回の検証を行うのに時間がかかる
- 複数のファイルを作成しているため、実際の動作イメージが実装コードからつきにくい

MATLAB/Simulinkを使用したモデルベース開発

MATLAB/Simulink対応の期待

近年の自動車開発に多く利用されているMBD開発をAUTOSAR APでも行うことができる事例を提示



さらなるAUTOSAR APへの関心の向上, 普及の促進

- AUTOSAR業務従事者だけでなく, MBD開発を行っている組込み技術者へAUTOSAR普及をアピール

早期検証, コードの自動生成等のMBD開発の利点を活用したアプリ開発



開発効率の向上

- ブロック単位での検証が可能になる
- AUTOSAR CPコード等からの置き換えが容易になる

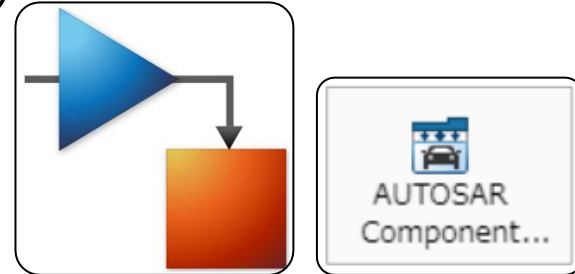
MATLAB/Simulink対応概要

①



デモアプリ(C++コード)

②



Simulink®モデル化

③



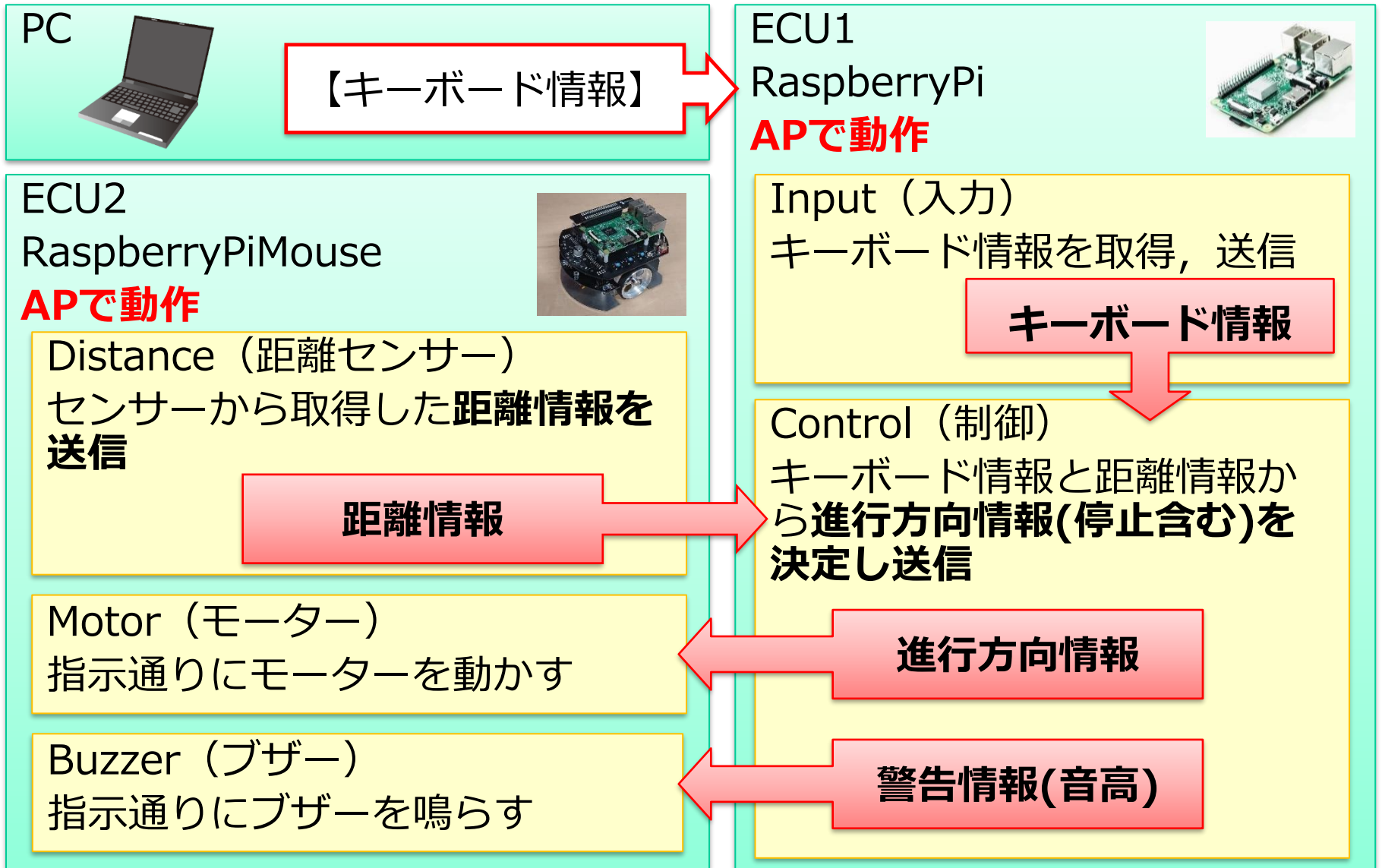
コード生成

④

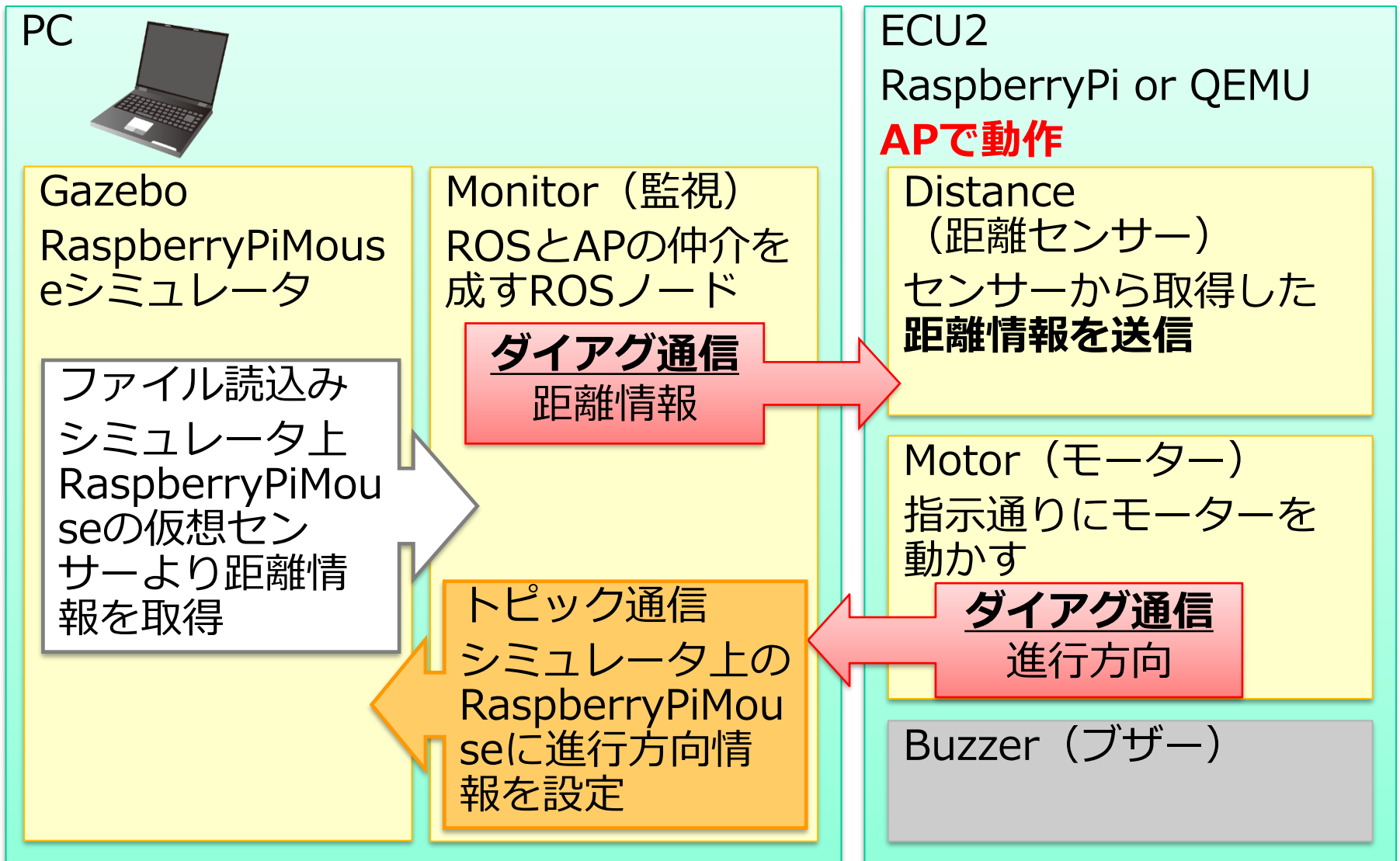


生成コードで確認

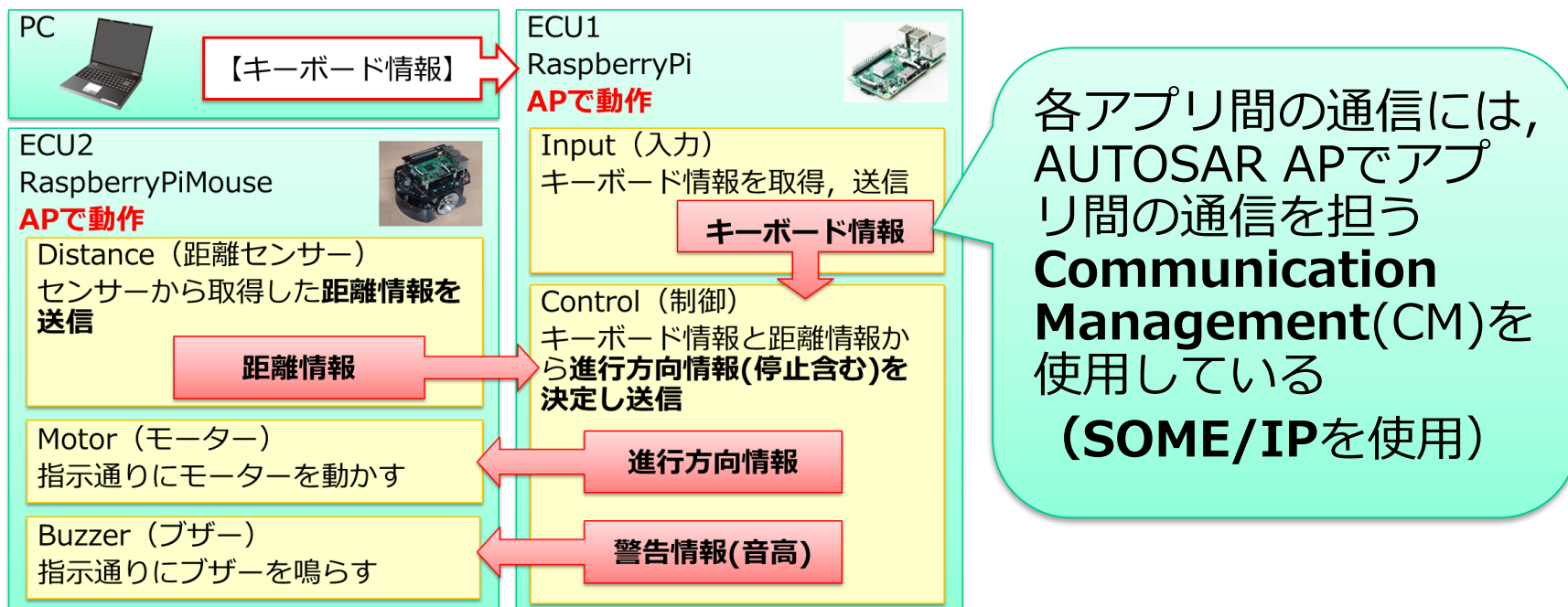
① RaspberryPiMouseデモ概要



① RaspberryPiMouseデモ概要



① Service Interface



- アプリ間のインターフェイスは**Service Interface**と呼ばれ、大きく下記3つで構成されている

Event	データの送受信 (Pub/Sub)
Method	サービスの呼び出し (Client/Server)
Field	遠隔からフィールド値を取得, 更新

② AUTOSAR Blockset

AUTOSAR Blockset

- AUTOSAR アプリケーションの**モデリング&シミュレーション**
- **モデルからのAUTOSARコード&ARXMLファイルの自動生成**

R2018b以前

Embedded Coder Support Package for AUTOSAR Standard

- Embedded Coder®ライセンスがあれば無料で利用可能
- **Classic Platformのみに対応**
- **モデルの編集のみでもEmbedded Coderが必要**

R2019b以後

AUTOSAR Blockset

- Simulinkのアドオン製品
- **Classic Platformに加えてAdaptive Platformに対応**
- **モデルの編集はSimulinkのみで可能**
- コード生成やSIL/PIL検証にはEmbedded Coderが必要

② AUTOSAR Blockset

Classic Platform

MATLAB Release	AUTOSAR Release
R2023a	4.0, 4.1, 4.2 (Rev 4.2.1, 4.2.2), 4.3 (Rev 4.3.1), 4.4, R19-11, R20-11, R21-11
R2022b	4.0, 4.1, 4.2 (Rev 4.2.1, 4.2.2), 4.3 (Rev 4.3.1), 4.4, R19-11, R20-11
R2022a	4.0, 4.1, 4.2 (Rev 4.2.1, 4.2.2), 4.3 (Rev 4.3.1), 4.4, R19-11
R2020b, R2021a/b	4.0, 4.1, 4.2 (Rev 4.2.1, 4.2.2), 4.3 (Rev 4.3.1), 4.4
R2019a/b, R2020a	2.1, 3.0, 3.1, 3.2 (Rev 3.2.2), 4.0, 4.1, 4.2 (Rev 4.2.1, 4.2.2), 4.3 (Rev 4.3.1)

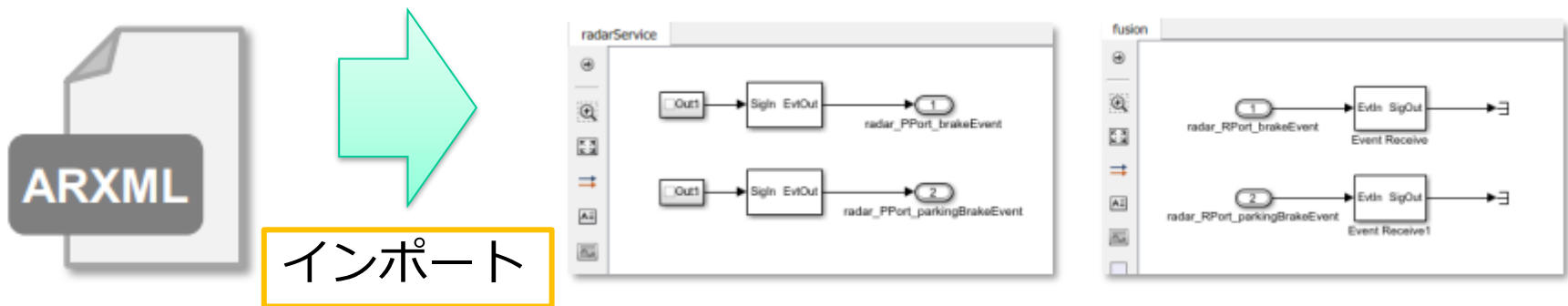
Adaptive Platform

MATLAB Release	AUTOSAR Release
R2023a	4.0, 4.1, 4.2 (Rev 4.2.1, 4.2.2), 4.3 (Rev 4.3.1), 4.4, R19-11, R20-11, R21-11
R2022b	R18-10, R19-03, R19-11, R20-11
R2021a/b, R2022a	R18-10, R19-03, R19-11
R2020a/b	R18-10, R19-03
R2019a	R18-10

② Simulinkモデルの作成

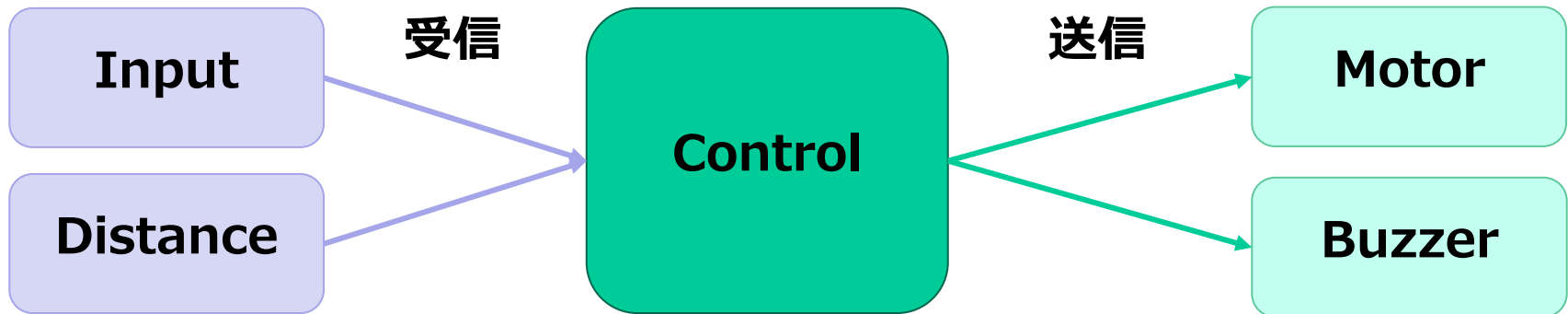
Simulinkモデルを設計し，ソフトウェアコンポーネントにマッピング

- **ARXMLファイル**からソフトウェアコンポーネントとコンポジションの記述をインポートし，AUTOSAR AP向けのSimulinkモデルを生成
(トップダウン)
- ARXMLファイルを入力情報として使用するほかに，SimulinkモデルからARXMLファイルを出力することも可能
(ボトムアップ)

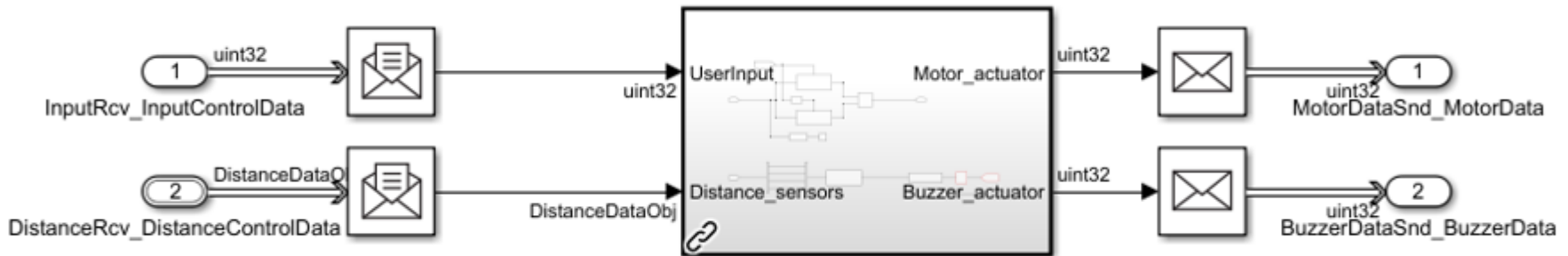


② AP用コードのSimulinkモデル化

Control



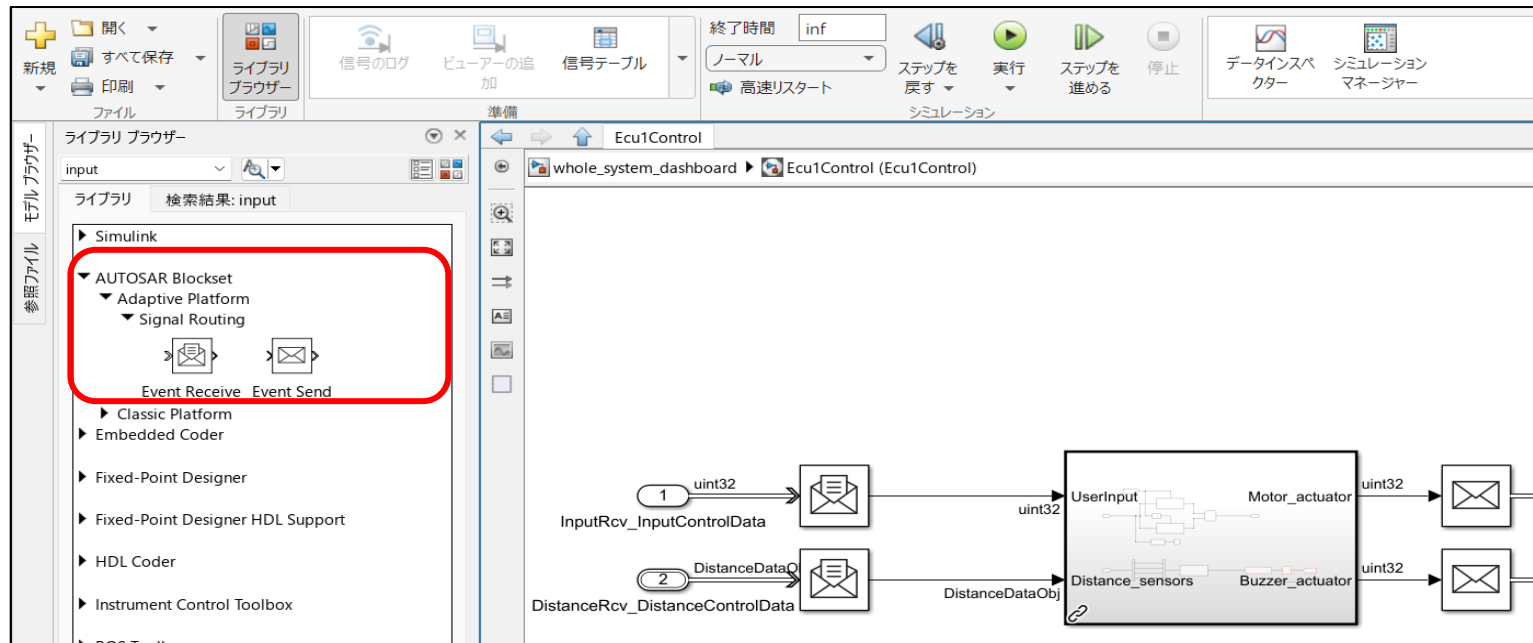
Simulinkモデル化



Event Receive

Event Send

② ライブラリブラウザ



Event Receiveブロック

信号の値とデータ型を保持したまま，入カイベントが信号に変換



Event Sendブロック

信号の値とデータ型を保持したまま、入力信号がイベントに変換

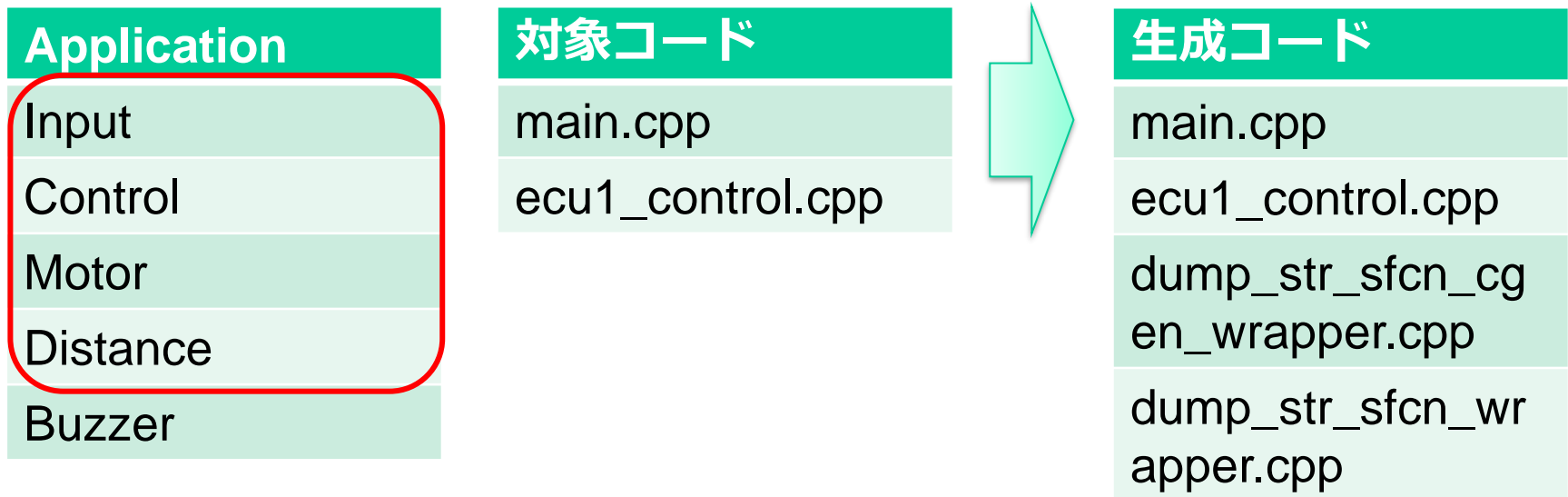
③ コード生成

The screenshot displays the AUTOSAR Component Designer interface. The top menu bar includes 'シミュレーション', 'デバッグ', 'モデル化', '書式設定', 'アプリ', and 'AUTOSAR x'. The 'コード生成' (Code Generation) button is highlighted with a red box. Below the menu, the '次のコード' (Next Code) section shows 'Ecu1Control'. The main workspace contains a Simulink model diagram with the following components and connections:

- InputRcv_InputControlData (uint32) connects to an input port of the Ecu1Control block.
- DistanceRcv_DistanceControlData (DistanceDataObj) connects to an input port of the Ecu1Control block.
- The Ecu1Control block contains sub-blocks: UserInput, Motor_actuator, Distance_sensors, and Buzzer_actuator.
- UserInput outputs uint32 to Motor_actuator.
- Distance_sensors outputs uint32 to Buzzer_actuator.
- Motor_actuator outputs uint32 to an output port.
- Buzzer_actuator outputs uint32 to an output port.
- The output ports connect to MotorDataSnd_MotorData (uint32) and BuzzerDataSnd_BuzzerData (uint32).

- AUTOSAR Component Designerを使用しコード生成が可能
- 作成したSimulinkモデルからコードを生成

③ コード生成



- 4つのアプリを生成コードへ置き換える
- 暫定処置として2か所生成コードを修正
 1. **Logging機能用ヘッダーファイルをAPDに合わせ変更**
APDは会員向けのデモンストレーターのため、対象リリースバージョン仕様と合致していない箇所がある
 2. InstanceIdentifierの先頭へ "SOME/IP:" を追加
APDに合わせ" SOME/IP:20"のように文字列として使用する

③ 生成コード

- 通信の確立（サービスの提供登録）

```
192 // Model initialize function↓
193 void Ecu1Control::initialize()↓
194 {↓
195     // Initialize service provider instance - BuzzerDataSnd↓
196     BuzzerDataSnd = std::make_shared< nces::mouse::portinterfaces::skeleton::↓
197         ControlBuzzerSkeleton >(ara::com::InstanceIdentifier("SOME/IP:17"), ara::com::↓
198         MethodCallProcessingMode::kEventSingleThread);↓
199     BuzzerDataSnd->OfferService();↓
```

- データの送信（Event Send）

```
182 ↓
183     // Send: '<Root>/Event Send'↓
184     // Send event↓
185     MotorDataSnd->MotorData.Send(rtb_Motor_actuator);↓
186 ↓
187     // Send: '<Root>/Event Send1'↓
188     // Send event↓
189     BuzzerDataSnd->BuzzerData.Send(0U);↓
190 }↓
```

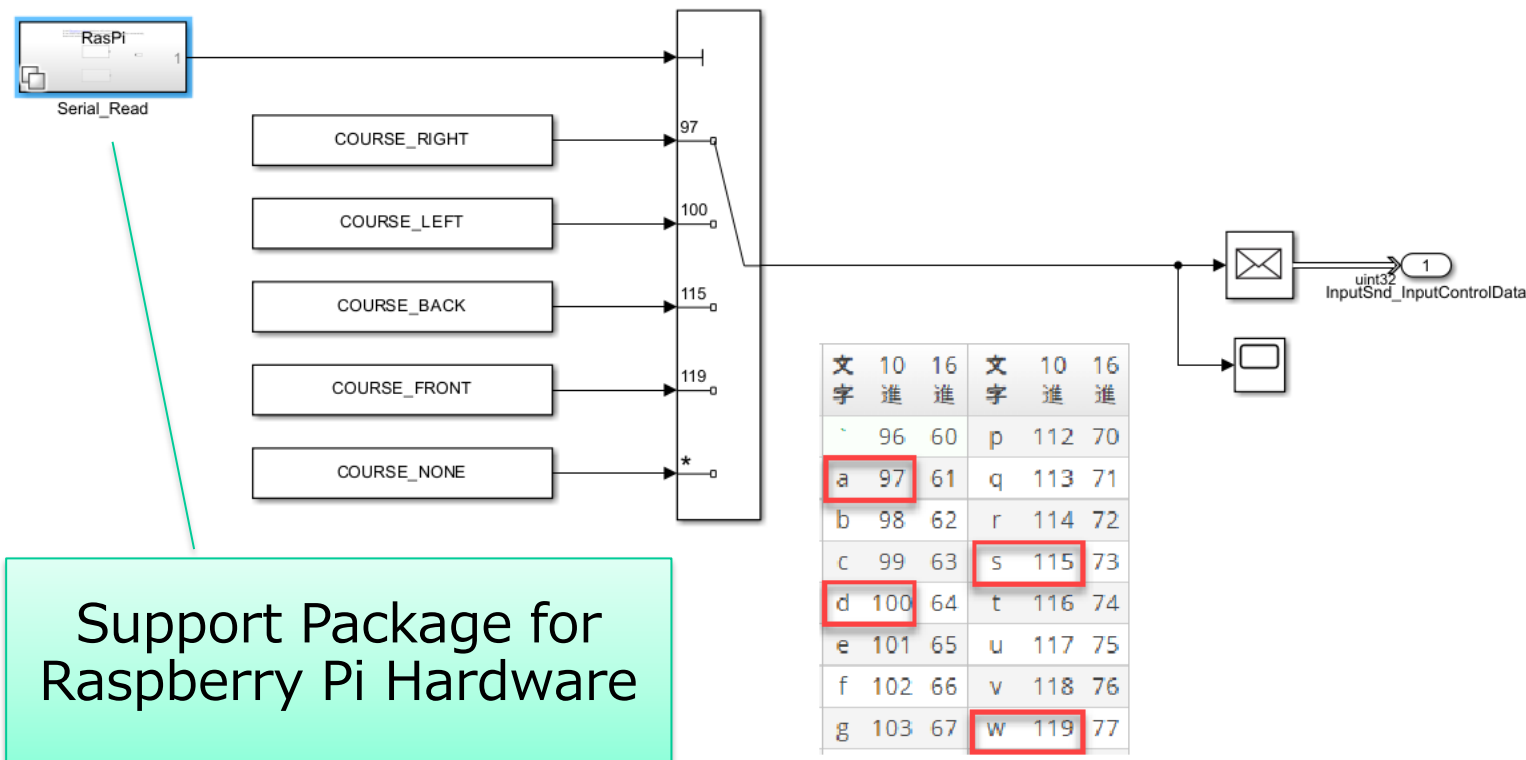
- データの受信（Event Receive）

```
112 // Model step function↓
113 void Ecu1Control::step()↓
114 {↓
115     std::shared_ptr<ara::core::Result<size_t>> resultPtr;↓
116     uint32_T rtb_Motor_actuator;↓
117     if (InputRcv) {↓
118         resultPtr = std::make_shared< ara::core::Result<size_t> >↓
119             (InputRcv->InputControlData.GetNewSamples(std::move(std::bind(&Ecu1Control::↓
120                 InputRcvInputControlDataReceive, this, std::placeholders::_1)), 1U));↓
121         resultPtr->ValueOrThrow();↓
122     }↓
```

④ 動作確認 - RaspberryPiMouse実機 -

1. RaspberryPiMouse実機を使用する

- キーボード入力の代わりにシリアル通信を使用する



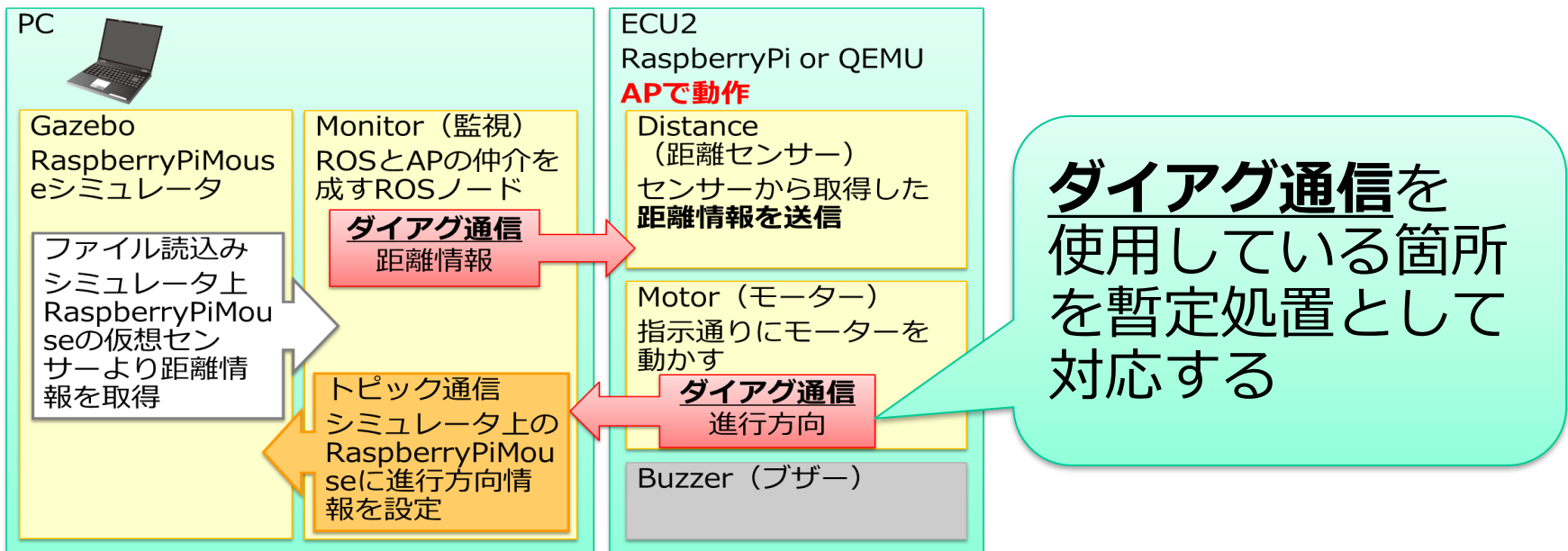
- Raspberry Piハードウェアサポートパッケージを使用し、シリアル通信にて読み取った値をinputにて処理をする

④ 動作確認 -シミュレータ-

2. Gazeboシミュレータを使用する

シミュレータを使用した検証には2パターン行った

生成コードに置き換えイメージを作成, 動作確認



- **ダイアグ通信の機能がR2022bのAUTOSAR Blocksetでは対応していないため, S-Function Builderブロックを使ってダイアグ通信APIの呼び出しを自動生成コードに挿入し対応した (S-Functionは外部C/C++コードを呼び出すためのブロック)**

④ 動作確認 - シミュレータ -

GazeboのraspimouseモデルとSimulinkを直接接続したコシミュレーション動作画面

The image displays a Simulink simulation environment on the left and a Gazebo 3D simulation environment on the right. The Simulink window shows a control system with blocks for 'Ecu1Control', 'Ecu2Distance', 'MotorDataRcv', and 'BuzzerDataRcv'. A 'Forward' button is highlighted in a blue box, with a green callout box pointing to it containing the text '画面以上に進行方向ボタンを作成' (Create a forward direction button on the screen). The Gazebo window shows a 3D view of the 'RaspberryPiMouseV2' robot in a virtual environment. A green callout box at the bottom of the Gazebo window contains the text 'Simulinkからの指示に従い動作' (Operate according to instructions from Simulink). The Gazebo interface includes a 'Property' table for the selected 'wall_clone_0' object:

Property	Value
name	wall_clone_0
is_static	<input checked="" type="checkbox"/> True
self_collide	<input type="checkbox"/> False
enable_wi...	<input type="checkbox"/> False
pose	
link	

At the bottom of the Gazebo window, the status bar shows 'Real Time Factor: 0.70' and 'Sim Time: 00 00:12:51.943'.

MATLAB Simulink対応結果のまとめ

- 一部を除き, 既存のAUTOSAR APアプリをMATLAB Simulinkへ置き換え, APDを使用し動作を確認した
 - ARXMLファイルからSimulinkモデルの作成
 - 作成したSimulinkモデルで動作検証, コード生成
 - 生成コードを使用したAPD動作イメージの作成
- AUTOSAR APに関する新たな開発事例を提示することができた



MBDの利点がAUTOSAR AP開発でも活用できる可能性を認識し, AUTOSAR APの開発手法の1つとして有効

AUTOSAR AP開発の可能性の拡大

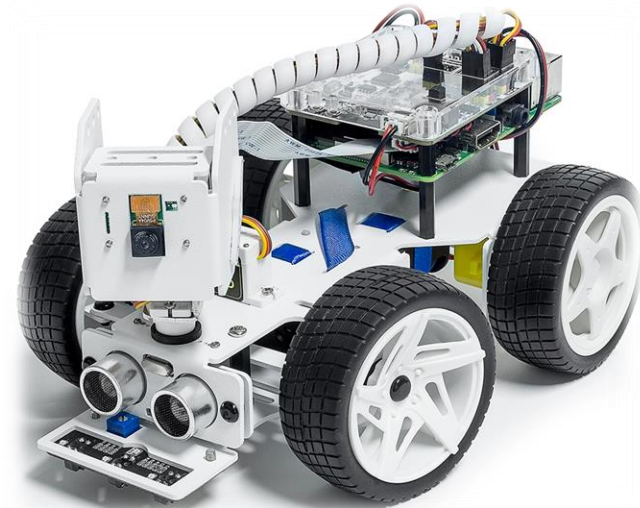
今後の展望

- 新しいバージョンへの追従

- Adaptive Platform Demonstrator
: R22-11-1, R23-11
- AUTOSAR Blockset : R2302a

- 新しいユースケースの増加

- アプリケーション
- パッケージ
- ボード



新たなAUTOSAR AP開発事例を提示

A2Pコンソ参加について

研究参加

- 出向者にAUTOSAR APの情報とノウハウが蓄積
- (出向者帰社後) 蓄積した情報と知見を活用して, AUTOSAR APを用いたシステムの立上げが可能
- AUTOSAR AP公開物に対するコンソーシアム活動の知見 (内部構成, 機能や評価結果) や成果物を無償で利用
- AUTOSAR活動を通じた, 出向者の英語力の向上

オブザーバ参加

- AUTOSAR AP公開物に対するコンソーシアム活動の知見 (内部構成, 機能や評価結果) をまとめた調査/評価/研究レポートを無償で利用できる

(個別共同プロジェクト)

- 技術項目が限定されている場合は, 大学との個別共同研究も可能

NCES お問い合わせ先

ご意見・ご質問のある方はお問い合わせください



名古屋大学大学院情報学研究科

附属組込みシステム研究センター

Tel : 052-789-4228 Fax: 052-789-4273

URL: <https://www.nces.i.nagoya-u.ac.jp/a2p-conso/>

email: a2p-staff@nces.i.nagoya-u.ac.jp