

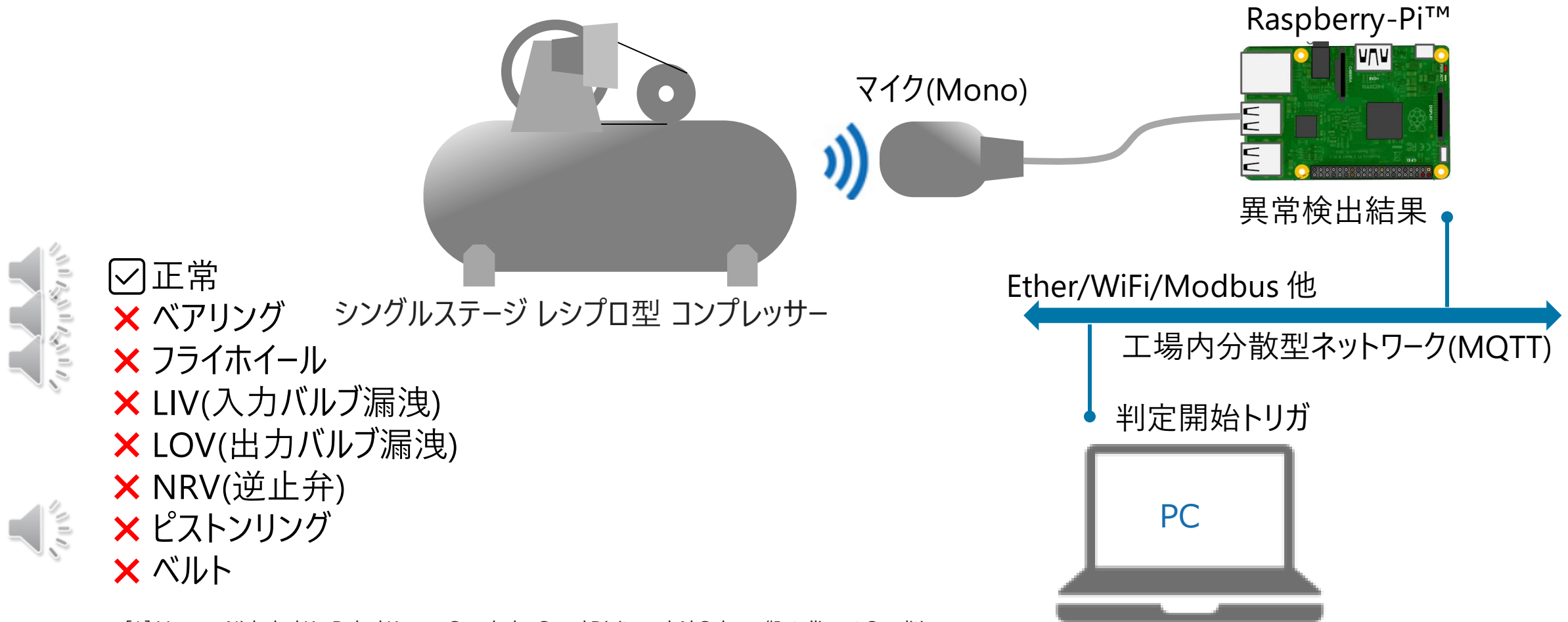


## Session2 : MATLABでカンタンRaspberry Pi 実装

---

MathWorks Japan  
アプリケーションエンジニアリング部

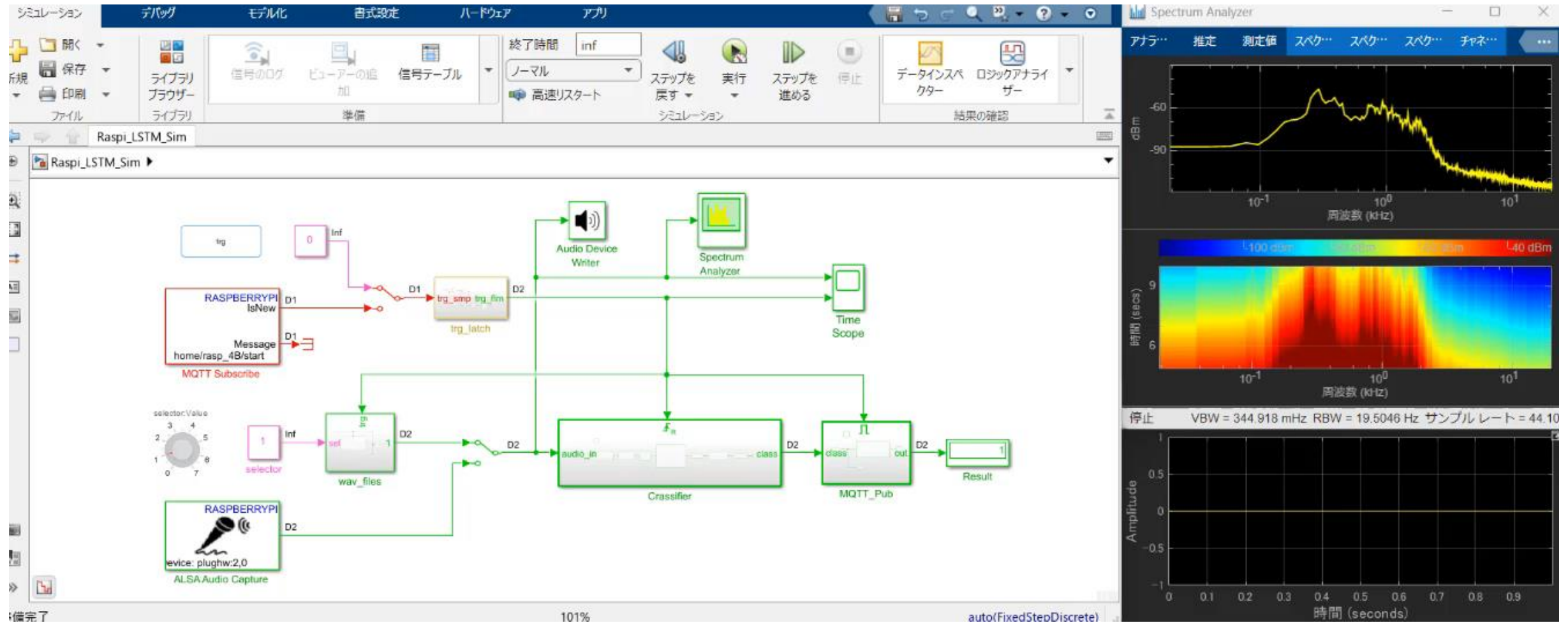
# 工程での異常検出・予知保全をエッジデバイスで



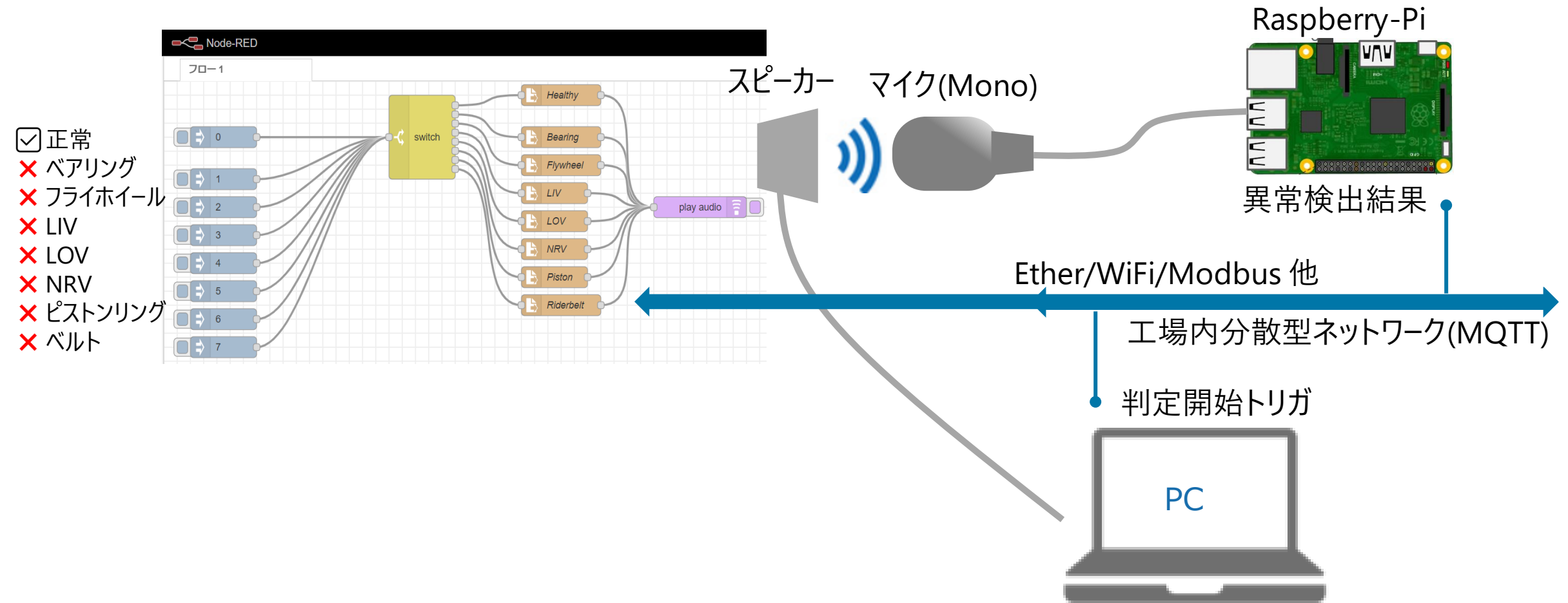
[1] Verma, Nishchal K., Rahul Kumar Sevakula, Sonal Dixit, and Al Salour. "Intelligent Condition Based Monitoring Using Acoustic Signals for Air Compressors." *IEEE Transactions on Reliability* 65, no. 1 (March 2016): 291–309. <https://doi.org/10.1109/TR.2015.2459684>.

# Demo シミュレーション

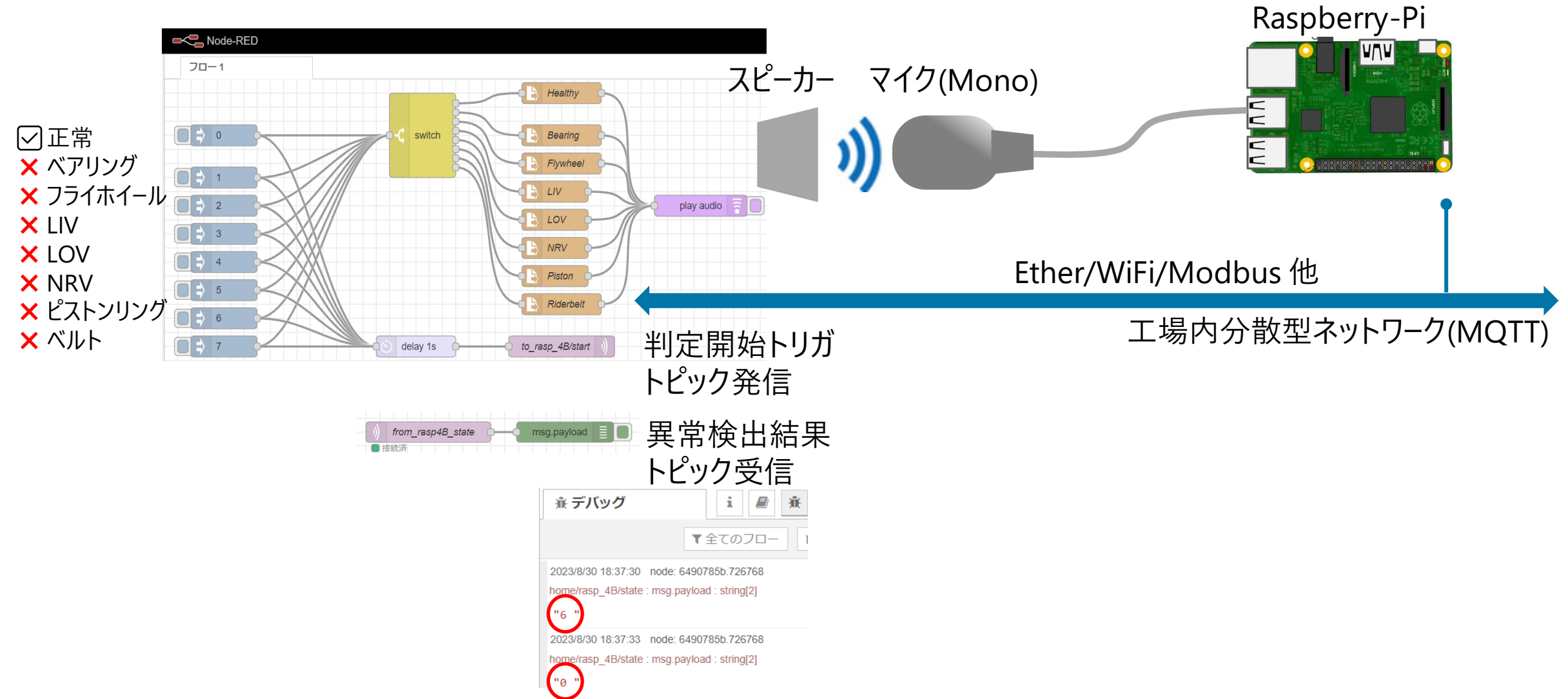
保存された音源で可視化しながらシミュレーション



## Demo 実機での構成 (音源はスピーカーで代替)



# Demo 実機での構成 (通信)



## Demo 実機での構成 (システム)

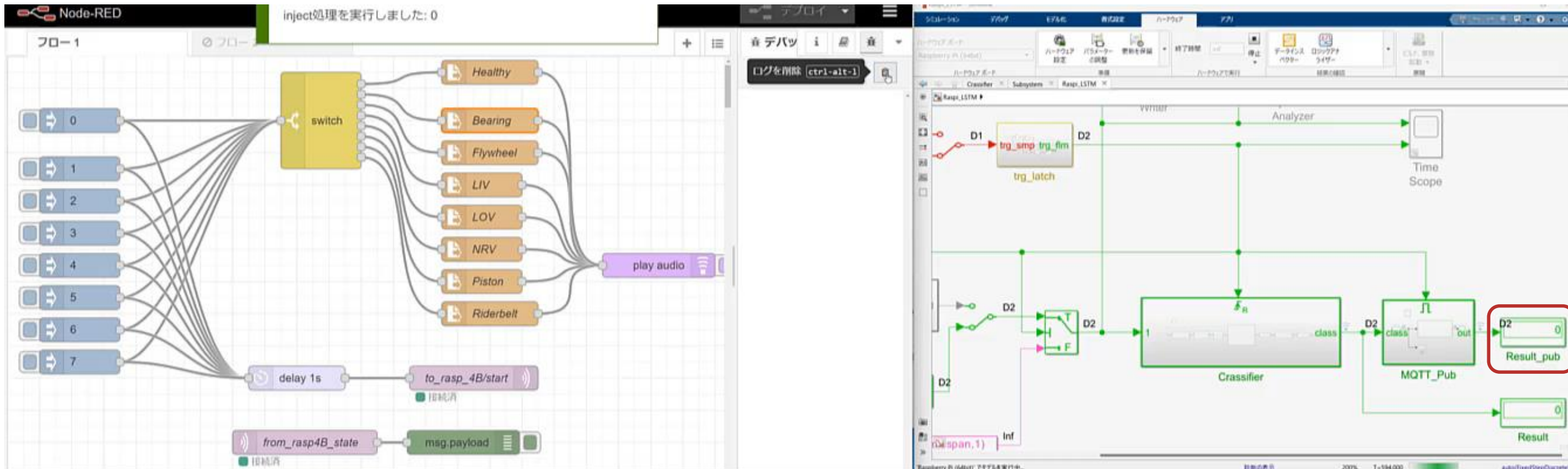
Raspberry-Piのマイク入力から音声を拾い  
エッジAIとしてRaspberry-Pi側で故障診断をして  
MQTT経由で診断結果を通知





# Demo 実機動作

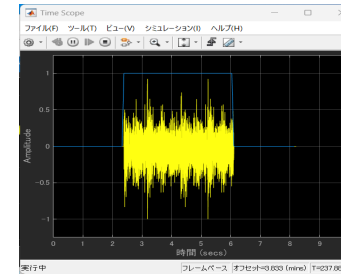
Raspberry-Piのマイク入力から音声を拾い (ハードウェア入力)  
Raspberry-Pi側で故障診断をして (エッジAI アルゴリズム)  
MQTT経由で診断結果を通知 (通信プロトコル対応)



# エッジデバイスに実装するメリット

データーをクラウドに転送して処理する構成との比較

- 通信形態の選択肢  
処理後の結果のみ/事象が起きたときのみ送信  
データ量を圧縮でき、常時接続不要  
->フィールドでの運用 (LoRa/ LPWA /LTE)
- セキュリティ  
通信を傍受してデータそのものにアクセスされる危険がない
- 応答性  
フィードバックループや出力がある場合、通信ラグなく応答性がよい
- ポータビリティ  
実証実験・コンセプト確認用に便利



音声データ  
(フレームデータ)



“Bearing”

認識結果  
(数文字)

デメリット：ハードウェア制約からくる処理性能の限界->所要リソース削減の工夫や調整が重要





やってみたいと思うけど...

ハードウェア周りの設定がめんどくさそう...

アルゴリズムは？ 専門知識がないと難しいのでは？ エッジデバイスで処理できるの？

ネットワークや外部通信デバイスとの接続は？ 実運用できるの？

-> MATLAB/ Simulinkを使えば簡単です。エッジデバイスの開発例を基に説明します

# エッジデバイス開発に必要な要素



開発環境  
可視化/デバッグ

ハードウェア周りの設定はサポートパッケージで簡単に



エッジデバイス向け  
アルゴリズム開発

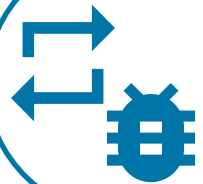
MATLABでアルゴリズム開発、実装のためのチューニング



通信プロトコル  
ミドルウェア対応

よく使用される通信プロトコルにも簡単に接続(MQTT/CAN他)

# エッジデバイス開発に必要な要素



開発環境  
可視化/デバッグ



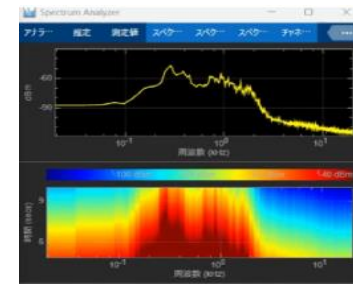
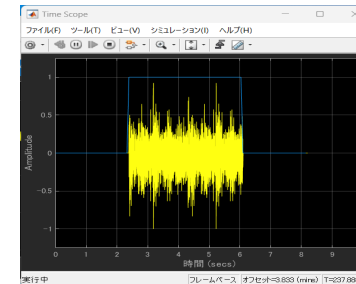
エッジデバイス向け  
アルゴリズム開発



通信プロトコル  
ミドルウェア対応

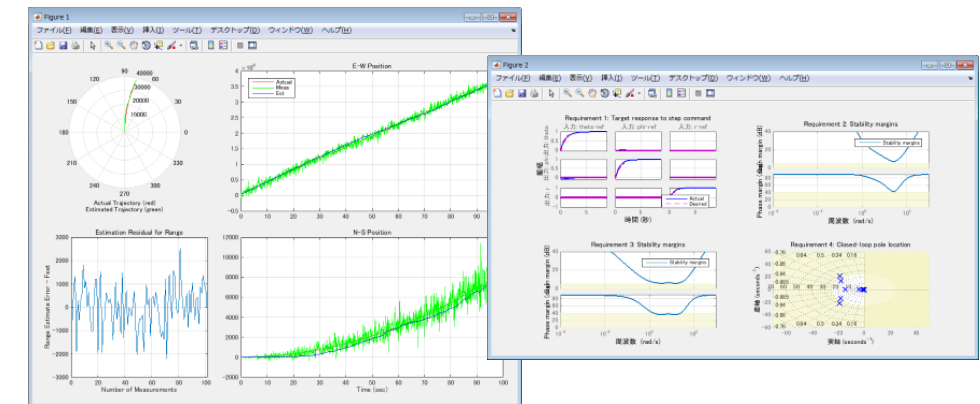
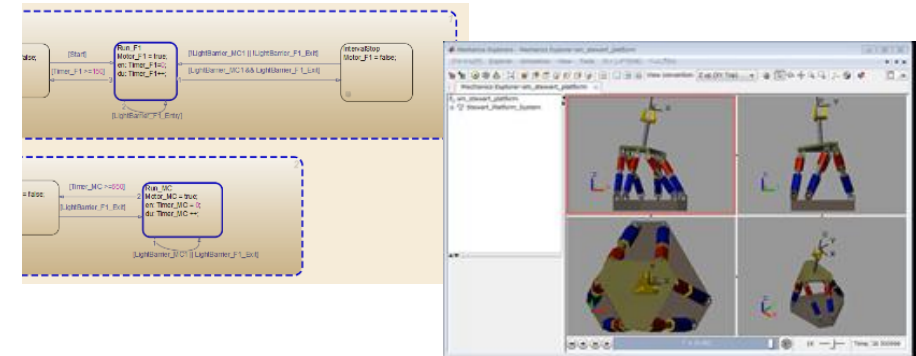
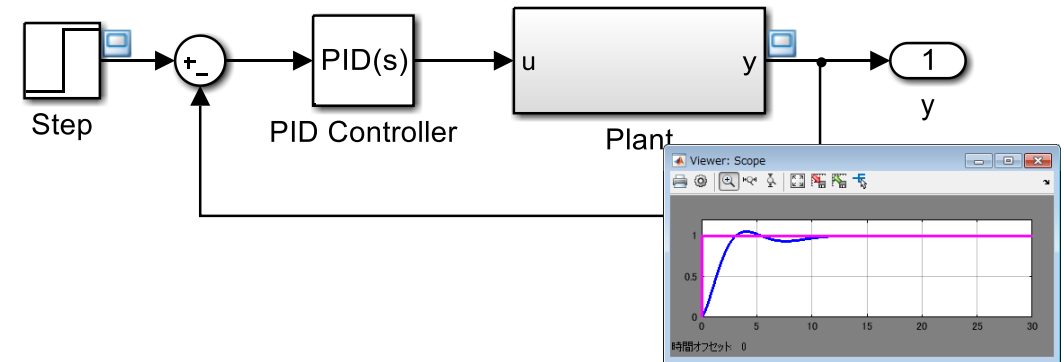
ハードウェア周りの設定はサポートパッケージで簡単に

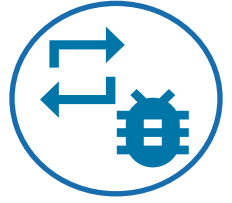
第三者にもわかりやすく、直感的にシステムを構築できるSimulinkで、各部の信号やパラメータを可視化しながらデバック



# Simulinkとは

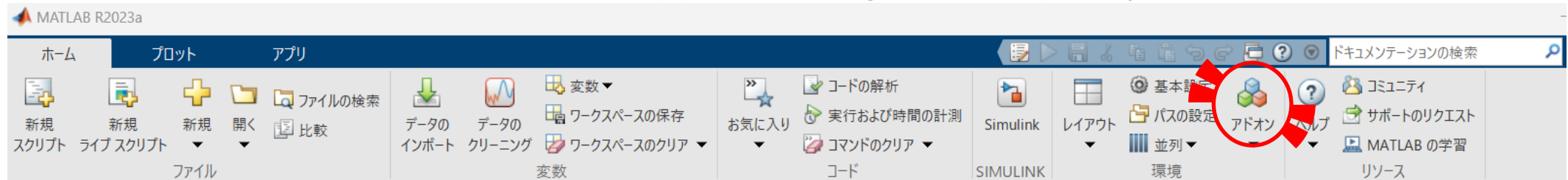
- 見やすい・作りやすいブロック線図環境
  - データや処理の流れが明瞭
  - アイディアを即座に具現化・共有
- 幅広いアプリケーション領域をカバー
  - 動的システム、イベントドリブン
  - 制御、信号処理、画像処理、AI
  - 様々なデバイスへの実装
- MATLAB環境との完全な統合
  - スクリプトによるバッチシミュレーション
  - 豊富な設計・解析・技術計算ライブラリへのアクセス





# Raspberry Pi との接続 (サポートパッケージ)

- OSが管理するハードウェアやビルドシステムとMATLAB/Simulinkを接続するためのパッケージ
- “アドオン”から簡単ダウンロード “Simulink Support Package for Raspberry Pi Hardware”



- ハードウェア入出力を簡単に実現するSimulinkブロック

WebCam(USB)



CAN



I2C



MQTT



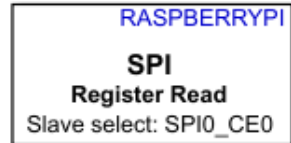
マイク



シリアル  
(USB/仮想ポートも)

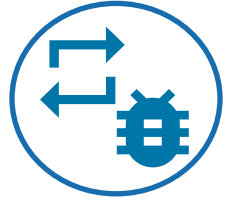


SPI

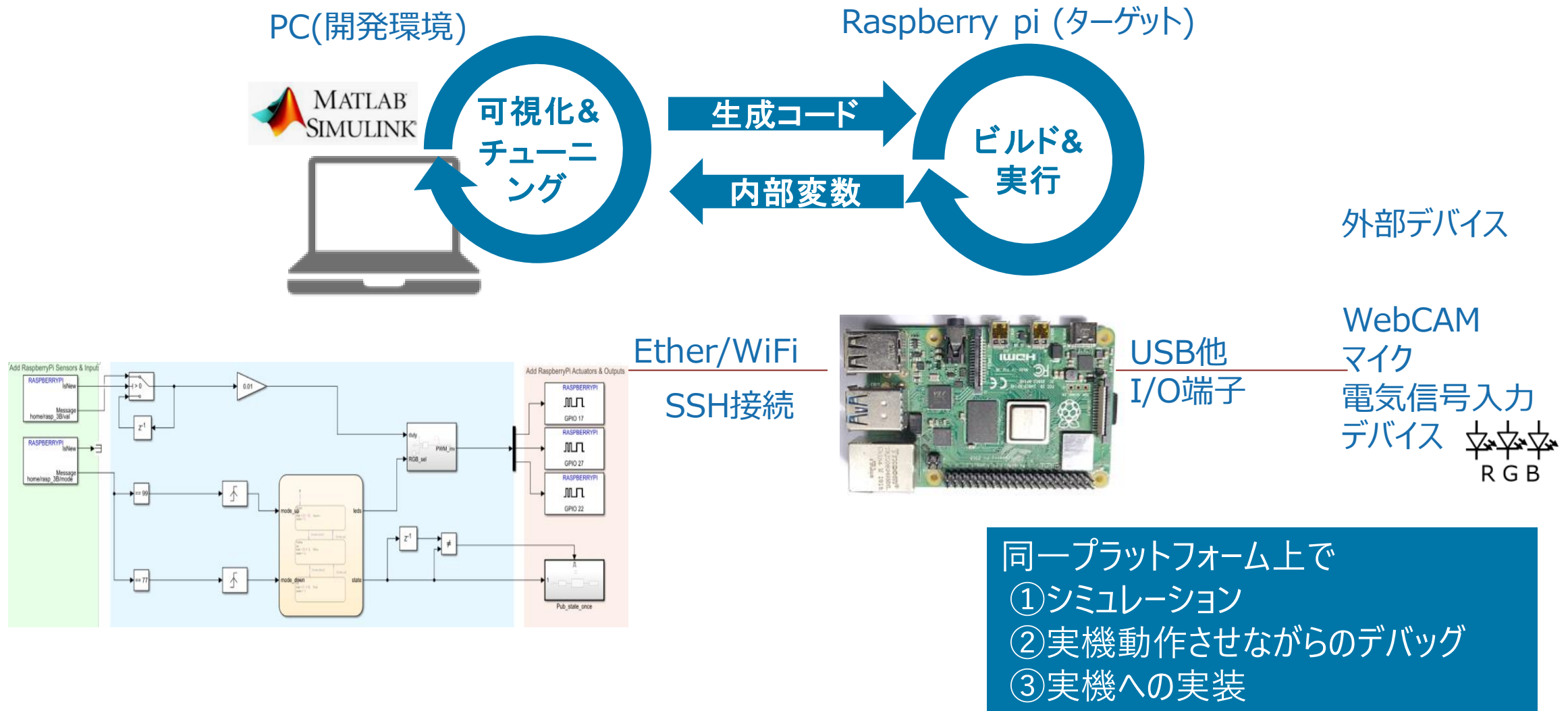


ROS

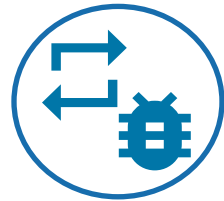




# MATLAB/Simulink 開発プラットフォームの構成



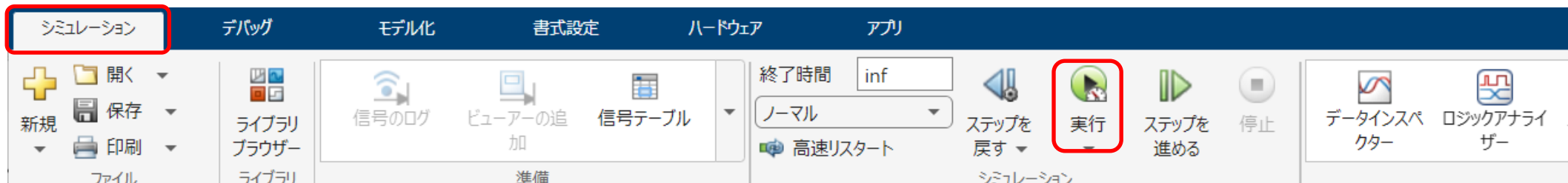




# シミュレーションと実機実行の切り替え

## ①シミュレーション

仮想入出力で動作検証



## ②実機動作させながらのデバッグ

実際の実機データを扱うには、"ハードウェア"タブの"監視と調整"からエクスターナルモードで実施

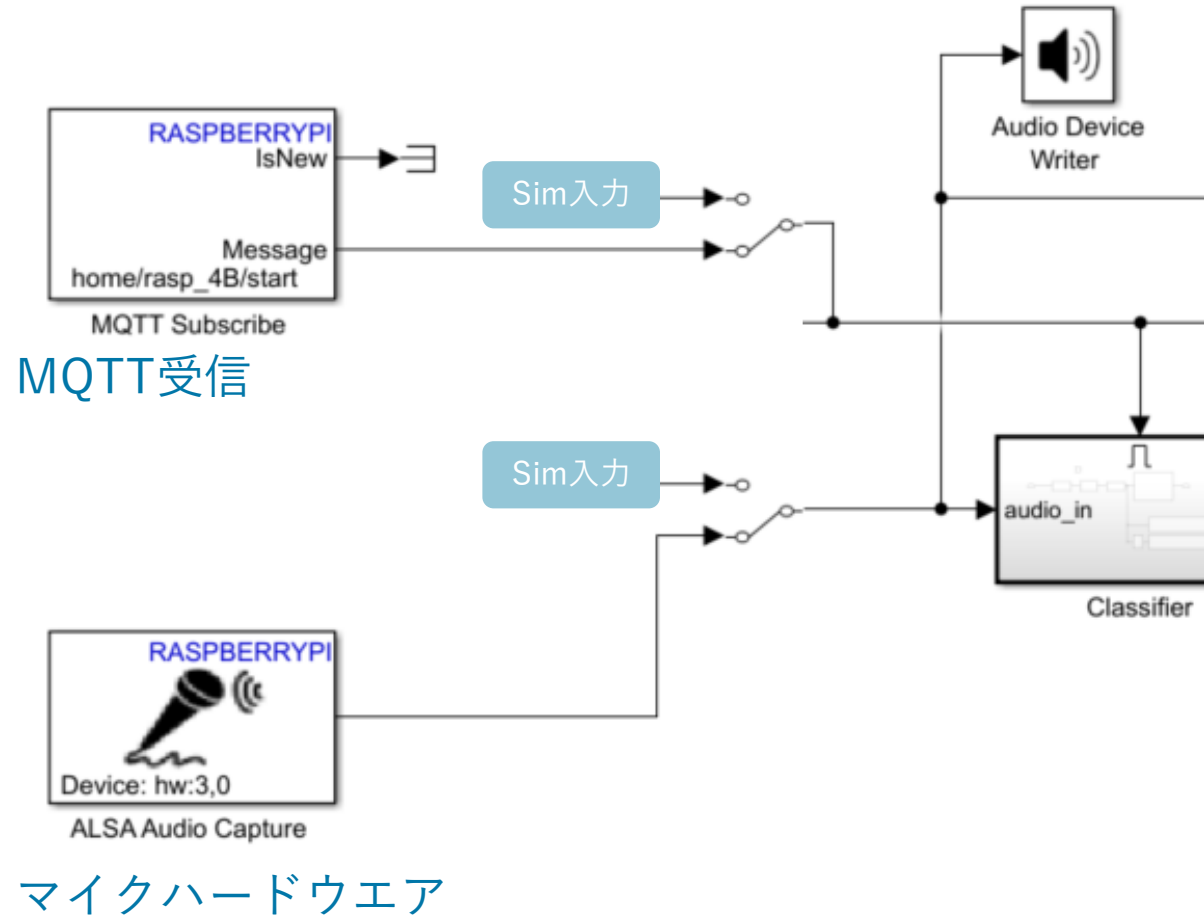
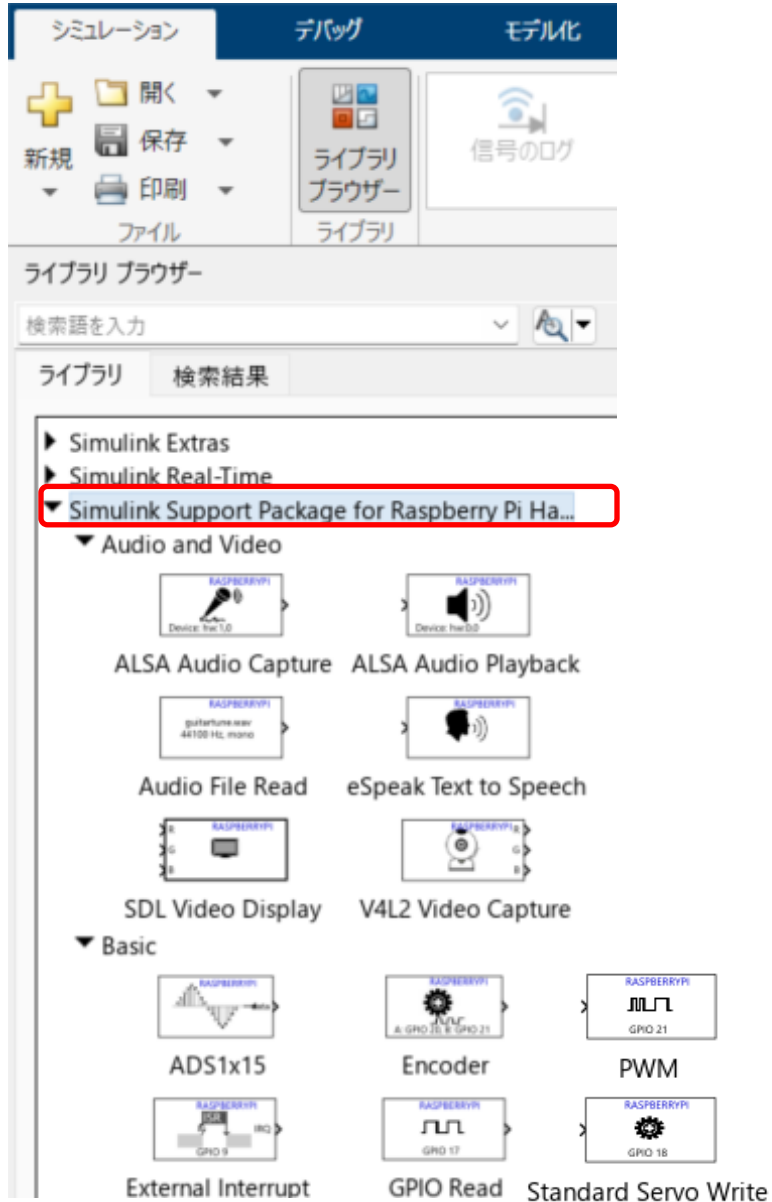
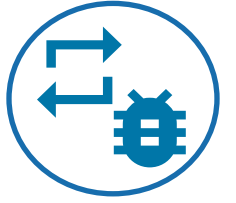


## ③実機への実装

ボタン一つでコード生成、ターゲットへの転送、ターゲットでのビルドを自動実行

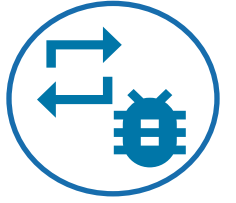


# ハードウェア入出力



MQTT受信

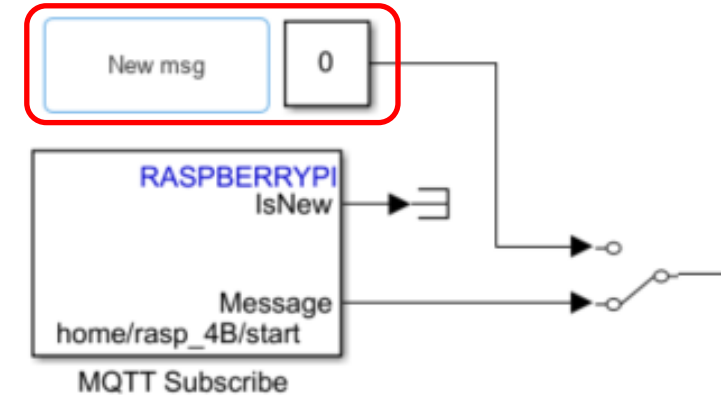
マイクハードウェア



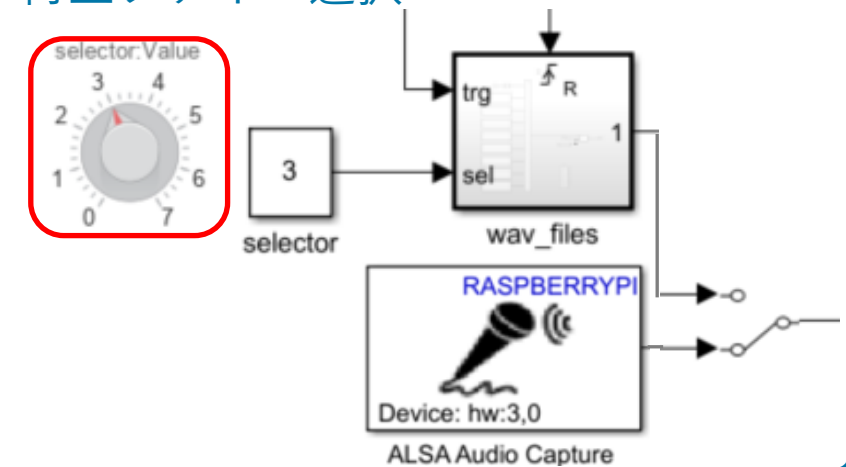
# シミュレーション入出力

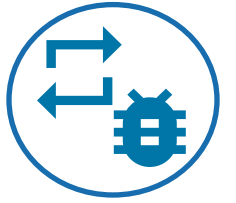


## MQTT受信代替

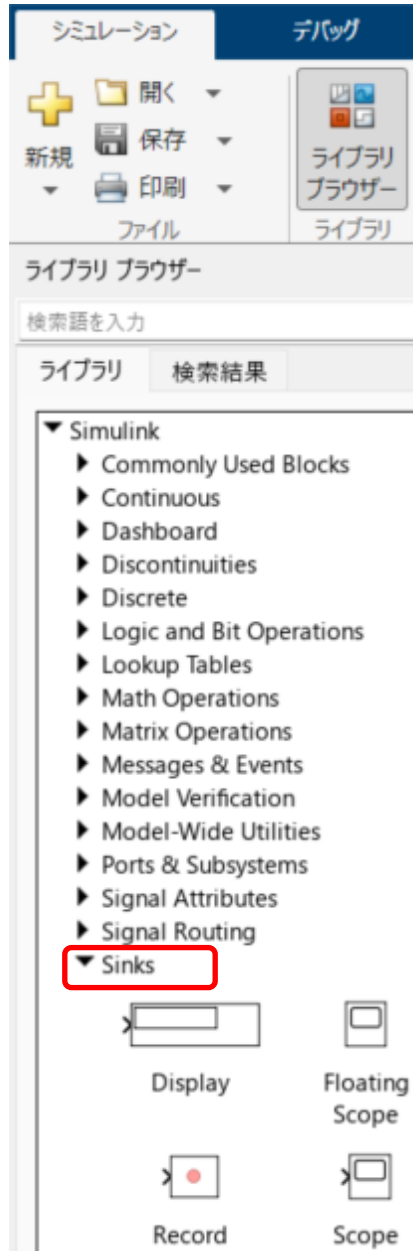


## 再生ファイル選択

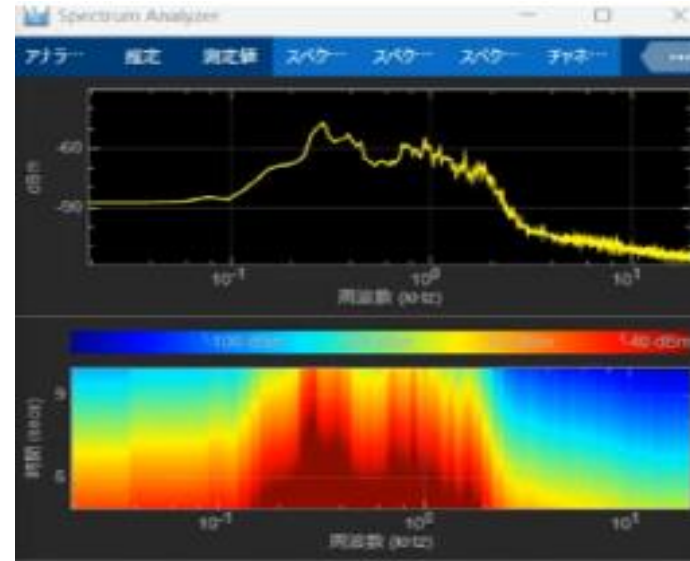




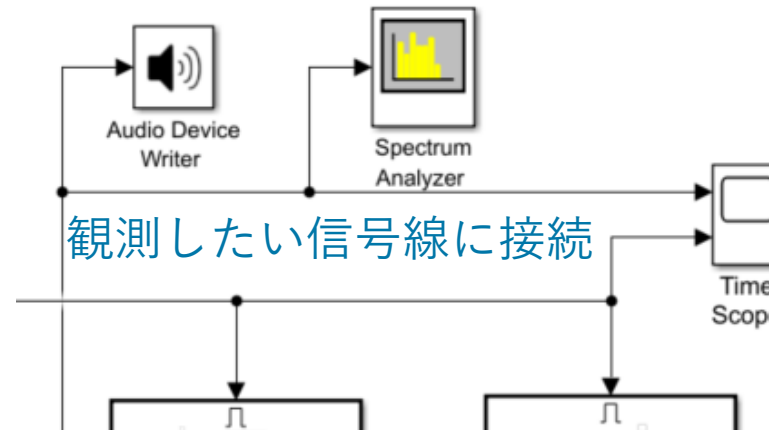
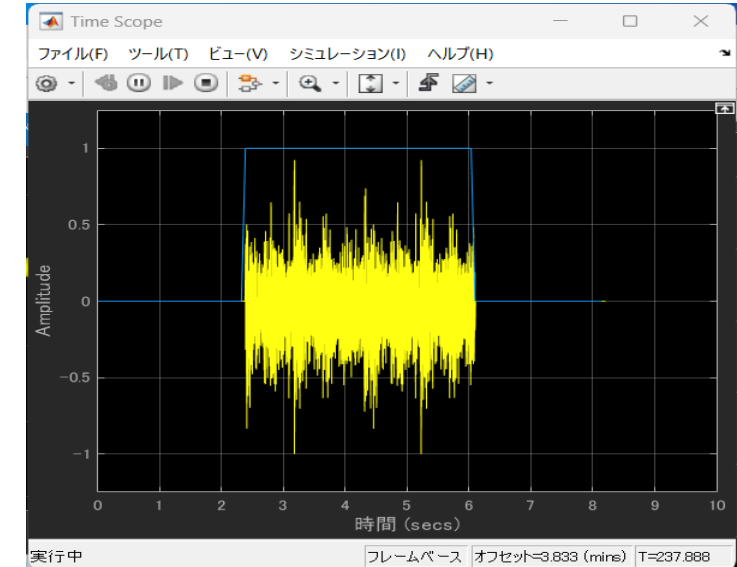
# 可視化/検証ブロック



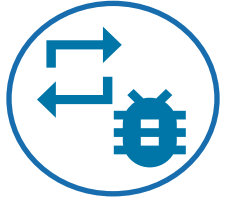
スペクトラムアナライザ



Time Scope



# まずはLチカ! シミュレーション入出力とハードウェア入出力



- シミュレーション
- 実機動作させながらのデバッグ
- 実機への実装

## <シミュレーション>

仮の入力、クリティカルなシナリオ



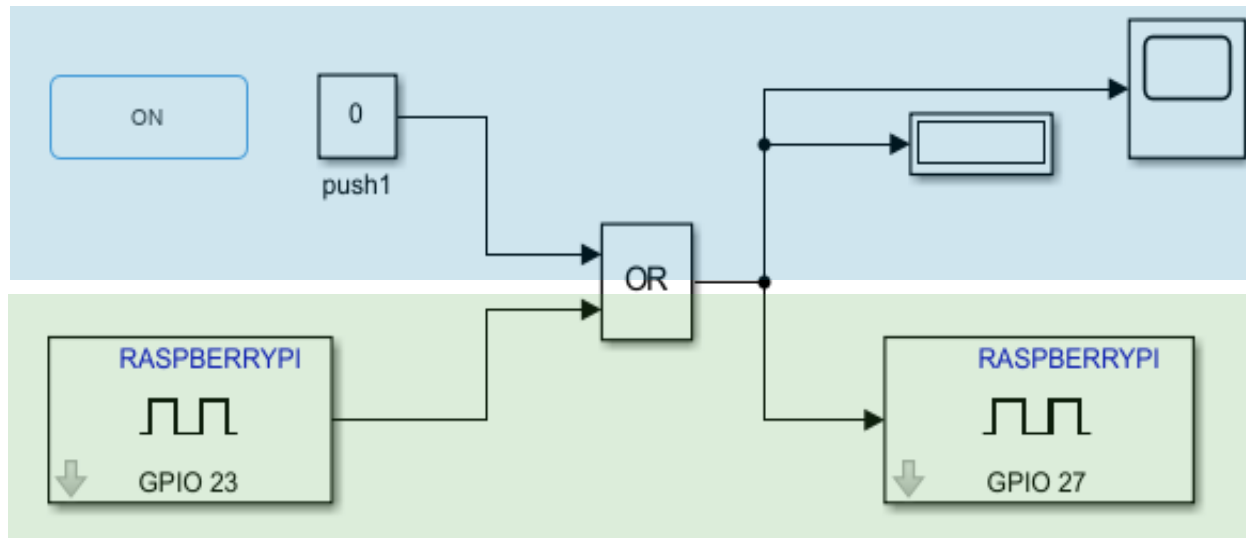
## <デバッグ>

任意の信号を可視化



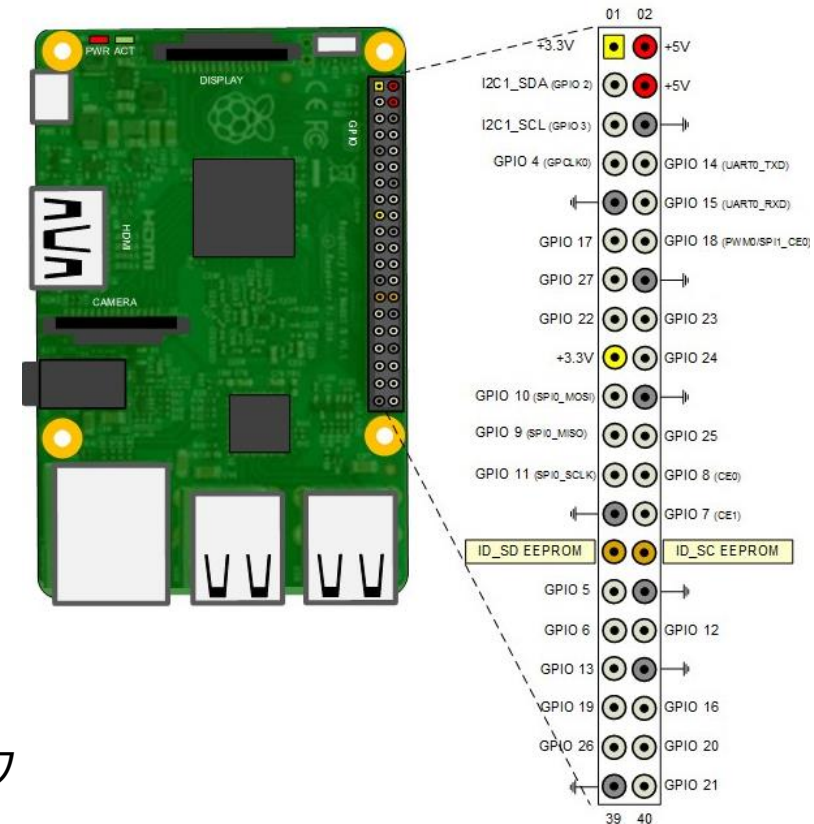
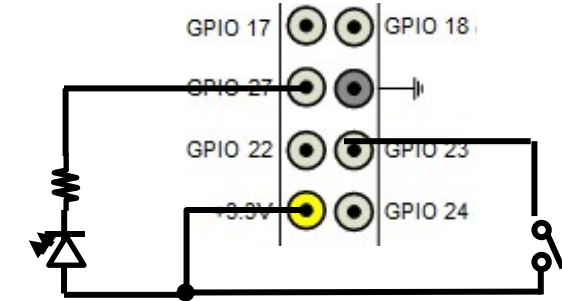
## <実機入力>

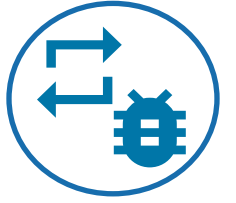
ハードウェア入力ブロック



## <実機出力>

ハードウェア出力ブロック



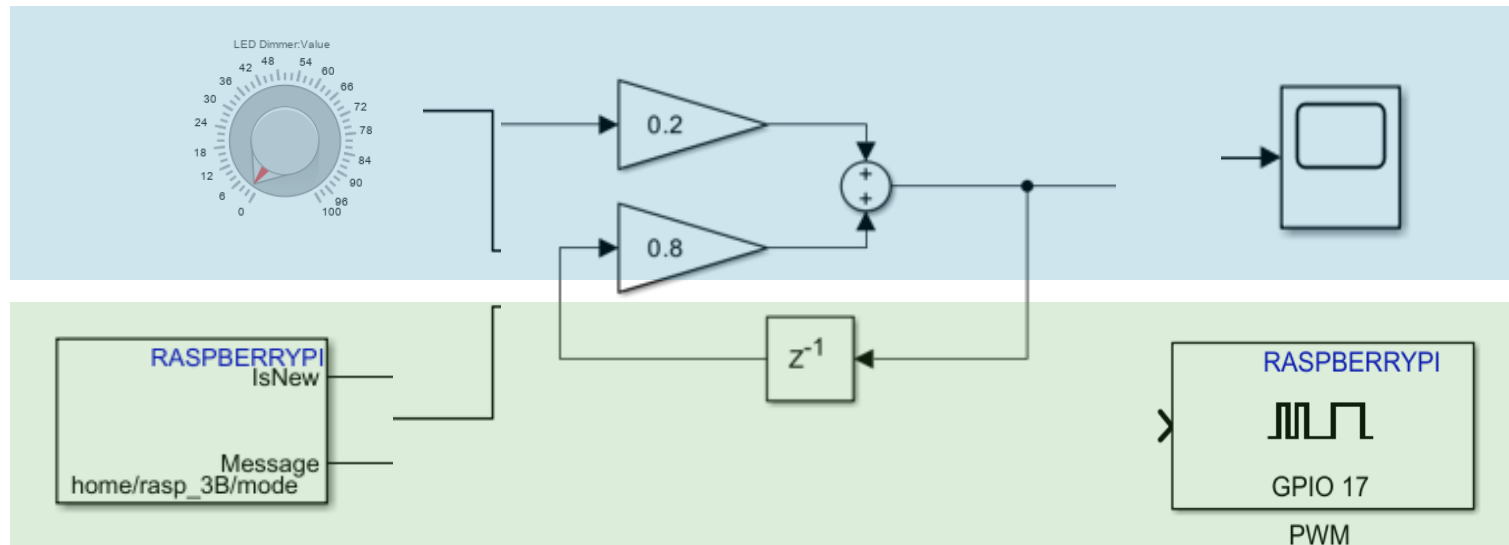


# Lチカを少し発展

電流値を入力

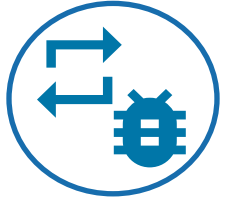
電流値がゆっくり変わるように

電流値をモニタ

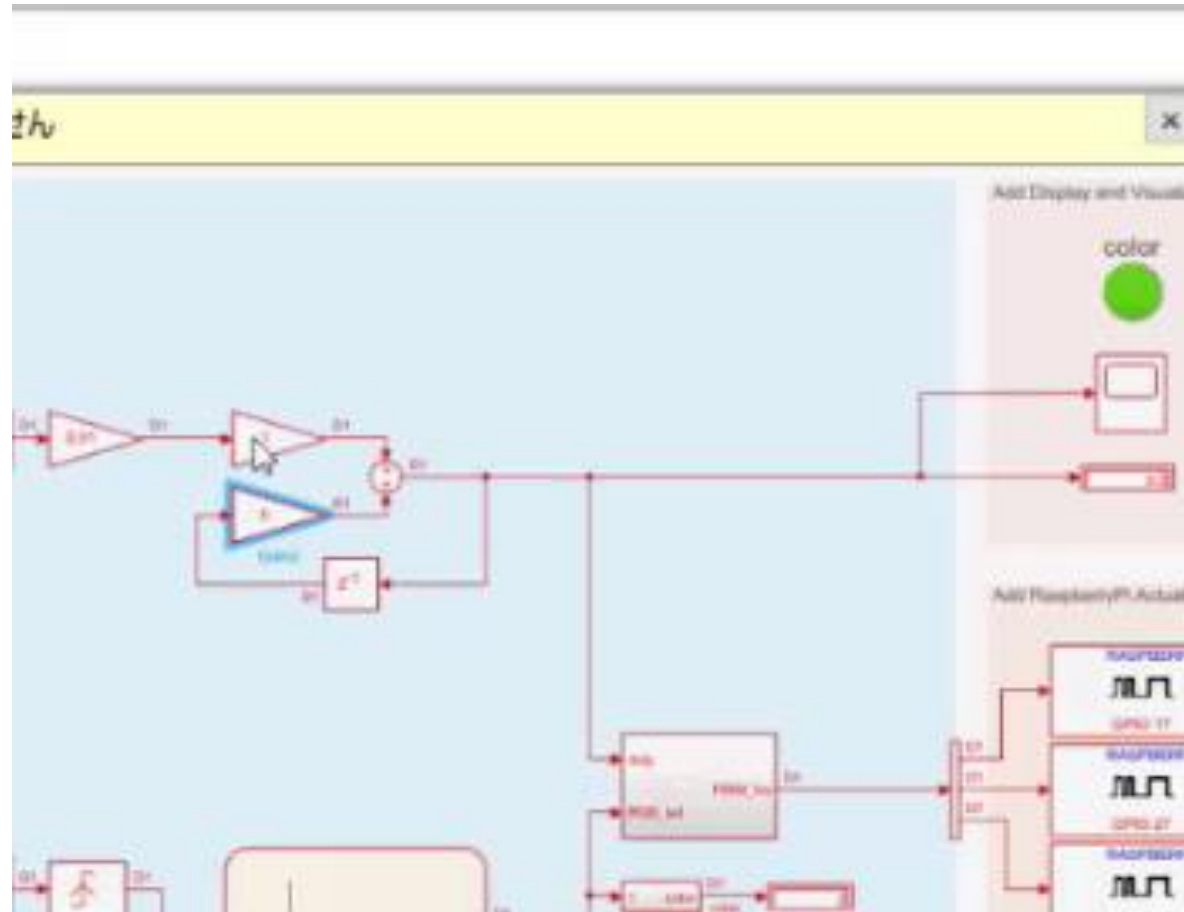


GPIO->PWMに  
DUTYを入力して明るさ調整

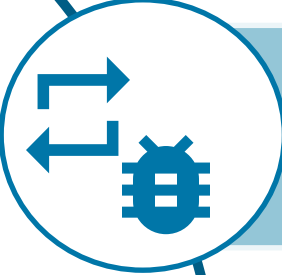




## Lチ力を少し発展




# エッジデバイス開発に必要な要素



開発環境  
可視化/デバッグ



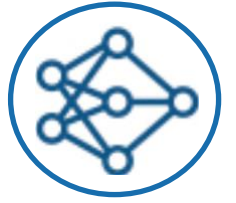
エッジデバイス向け  
アルゴリズム開発



通信プロトコル  
ミドルウェア対応

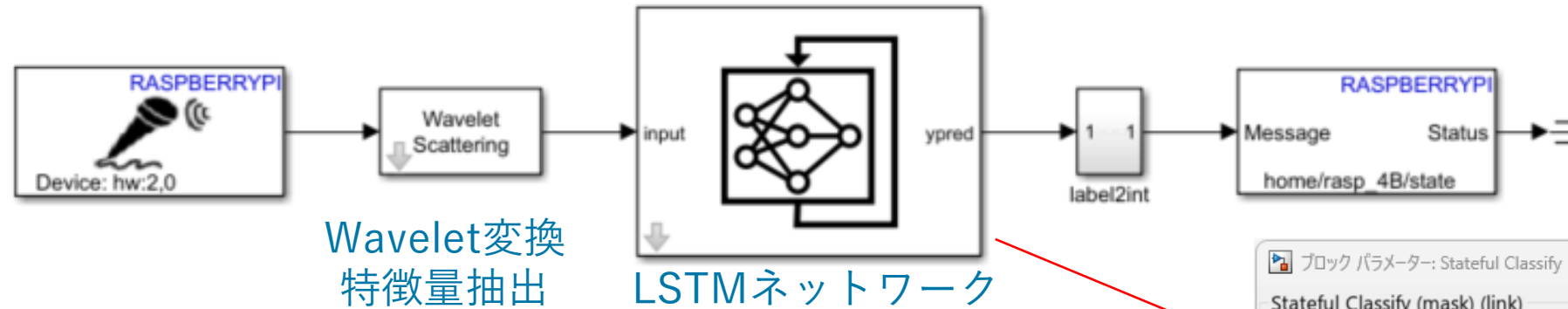
MATLABでアルゴリズム開発、実装のためのチューニング

Simulink環境で、AI アルゴリズムもブロックを配置して実現  
MATLAB環境で、AIモデルのチューニング (学習/精度向上/省メモリ)



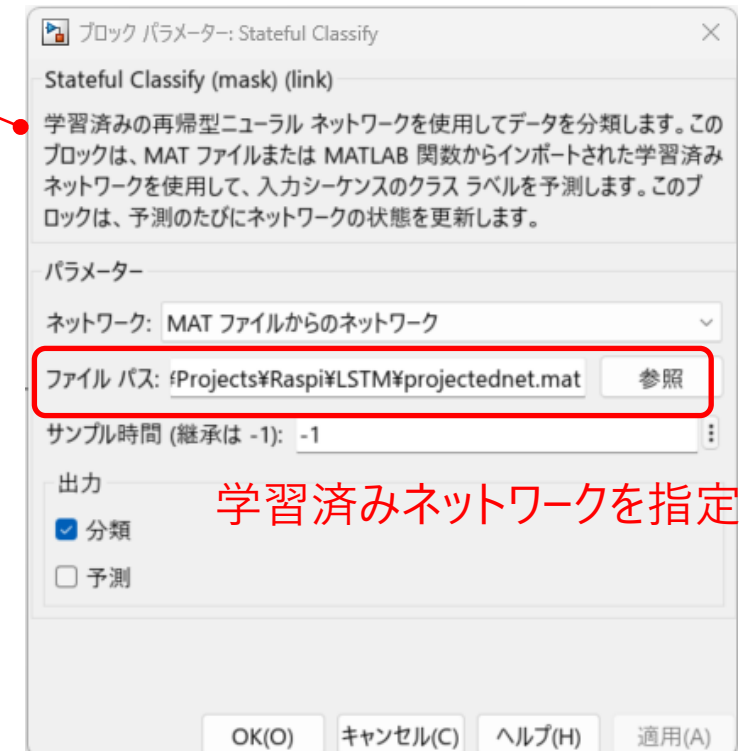
# アルゴリズム例 異常検出

ハードウェアも含んだシステム全体の記述はSimulinkで



Simulinkから呼び出すネットワークの学習や調整はMATLABで

関連リンク：[予知保全・異常検知のためのデータセントリックAI  
王道! 時系列データで学ぶ6種の特徴抽出と異常検知](#)

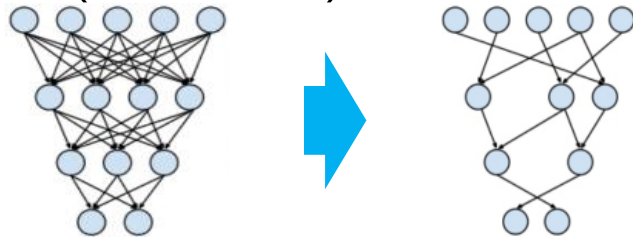


学習済みネットワークを指定

# ネットワークサイズの削減

メモリ節約/実行速度向上のため、  
MATLABで精度と削減効果を可視化しながら、ネットワークに必要なメモリを削減

## - 枝刈り (プルーニング)

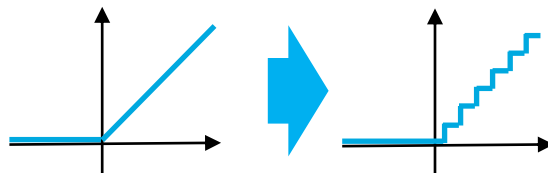


## - 投影 (プロジェクション)

$$Wx \rightarrow WQQ^t x$$

Wの代わりにQを保存

## - 量子化



### ▼ 枝刈り

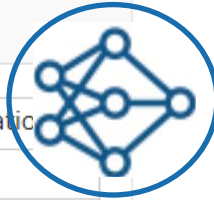
<code>taylorPrunableNetwork</code>	Network that can be pruned by using first-order Taylor approximation
<code>forward</code>	学習用の深層学習ネットワーク出力の計算
<code>predict</code>	推論用の深層学習ネットワーク出力の計算
<code>updatePrunables</code>	Remove filters from prunable layers based on importance scores
<code>updateScore</code>	Compute and accumulate Taylor-based importance scores for pruning
<code>dlnetwork</code>	カスタム学習ループ向けの深層学習ネットワーク

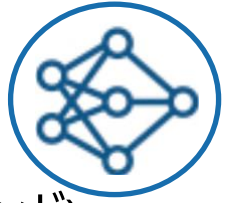
### ▼ 投影

<code>compressNetworkUsingProjection</code>	Compress neural network using projection
<code>neuronPCA</code>	Principal component analysis of neuron activations

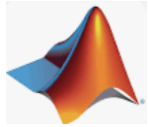
### ▼ 量子化

<code>dlquantizer</code>	Quantize a deep neural network to 8-bit scaled integer data types
<code>dlquantizationOptions</code>	Options for quantizing a trained deep neural network
<code>calibrate</code>	深層ニューラル ネットワークのシミュレーションと範囲の収集
<code>quantize</code>	Quantize deep neural network
<code>validate</code>	Quantize and validate a deep neural network
<code>quantizationDetails</code>	Display quantization details for a neural network
<code>estimateNetworkMetrics</code>	Estimate network metrics for specific layers of a neural network
<code>equalizeLayers</code>	Equalize layer parameters of deep neural network





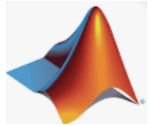
# 投影(プロジェクション)でネットワークサイズを削減する例



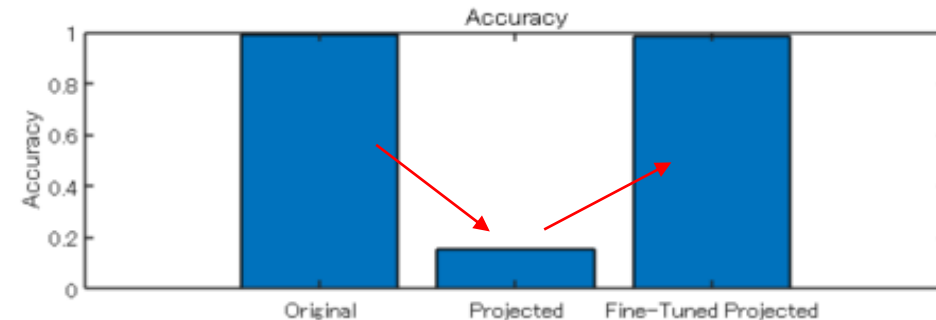
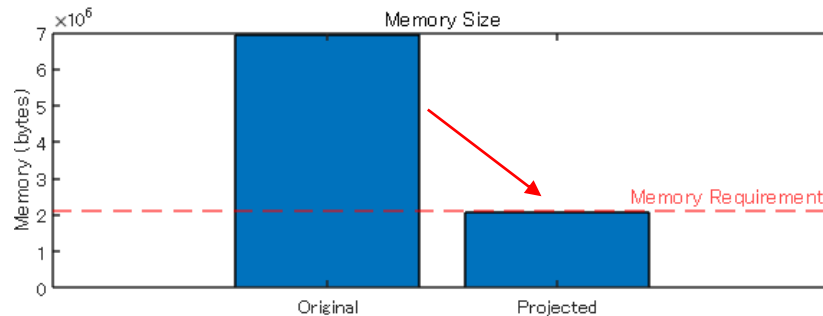
対象のネットワークに対し、投影したネットワークを生成 (compressNetworkUsingProjectionコマンド)  
学習パラメータの削減率を指定できる (LearnablesReductionGoalオプション)

```
netProjected = compressNetworkUsingProjection(dlnet,npca,LearnablesReductionGoal=0.95);
```

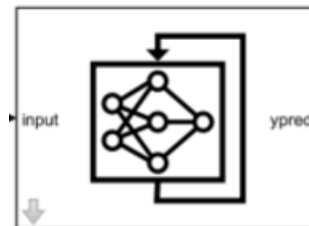
Compressed network has 87.1% fewer learnable parameters.  
Projected layers explain on average 95.0% of layer activation variance.



生成されたネットワークを再学習してファインチューニング – 分類精度はほぼ回復



ファインチューニング後、ネットワークを.mat形式で保存し、Simulinkブロックのメニューから読み込む

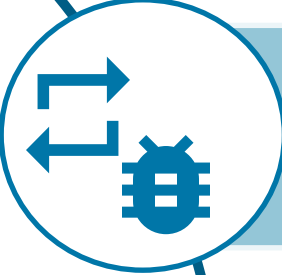


パラメーター

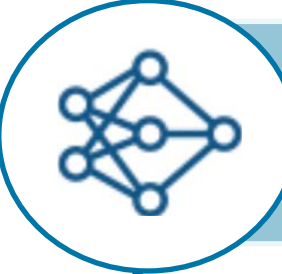
ネットワーク: MAT ファイルからのネットワーク

ファイル パス: Projects\Raspi\LSTM\projectednet.mat 参照

# エッジデバイス開発に必要な要素



開発環境  
可視化/デバッグ



エッジデバイス向け  
アルゴリズム開発



通信プロトコル  
ミドルウェア対応

よく使用される通信プロトコルにも簡単に接続(MQTT/CAN他)  
受信パケットからヘッダ等を削除して必要なデータのみ抜き出す処理も  
ブロックを配置して必要な項目をGUIで記入するだけ！

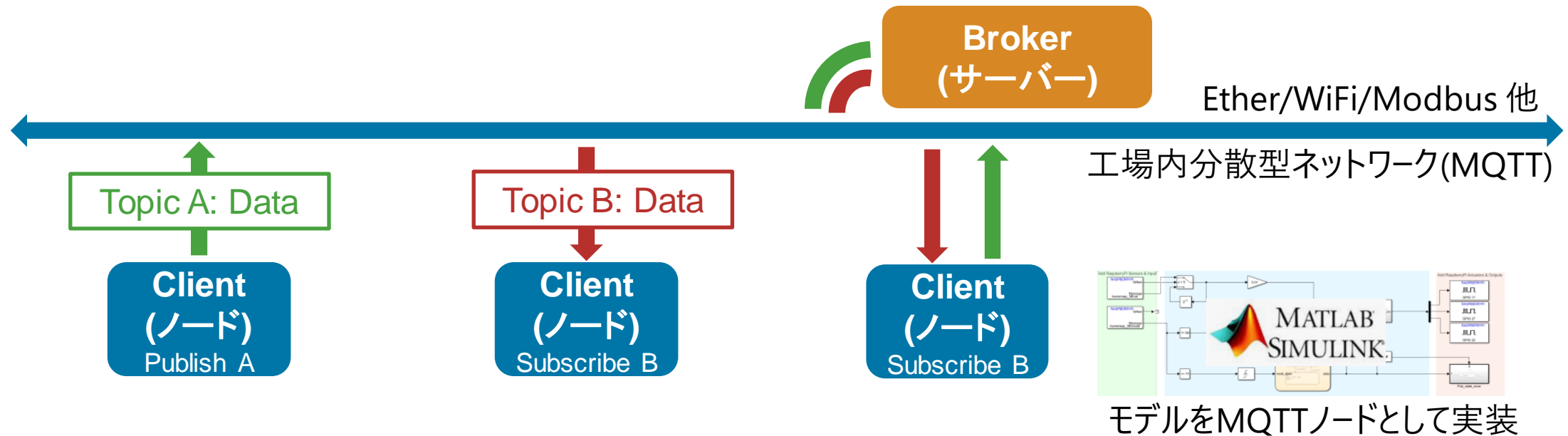




## ミドルウェア 分散型システム

処理はネットワーク上に分散したノード単位で実行

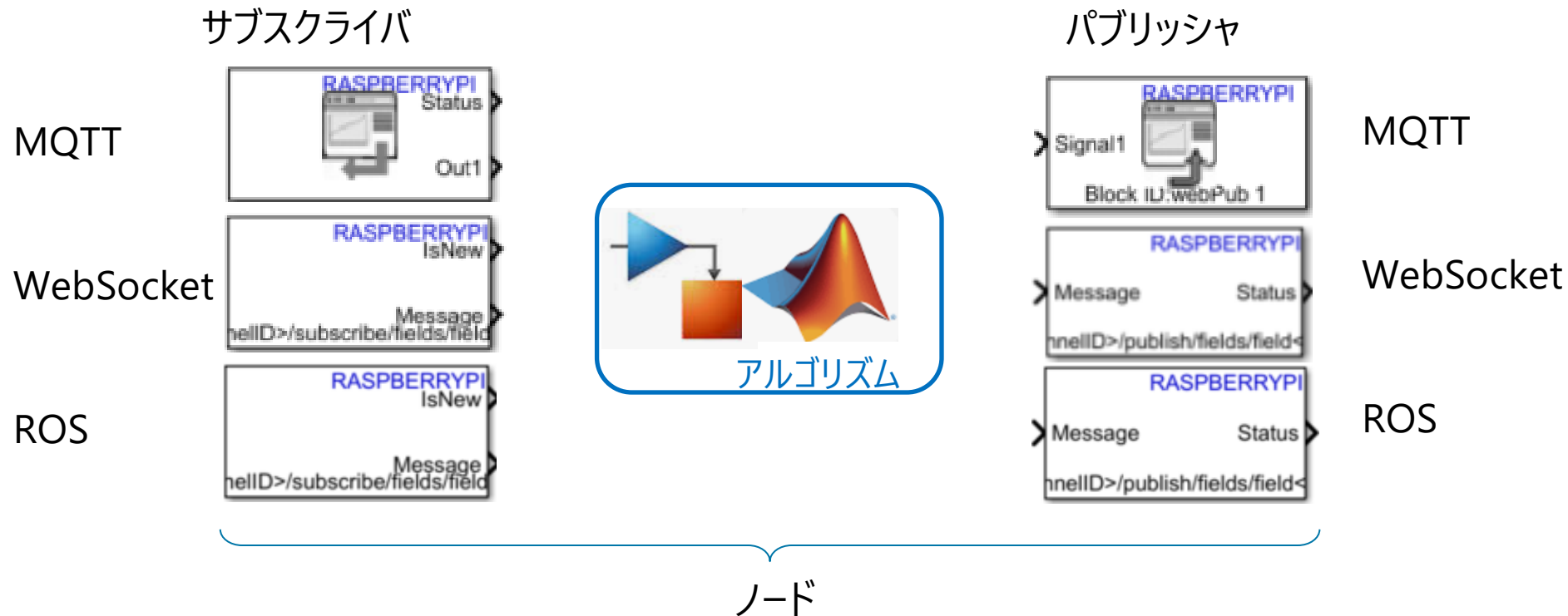
- ・ クライアント(ノード)はトピックでデータを送受信
- ・ ブローカー(サーバー)はトピックを仲介
- ・ QoSを設定できる (データの優先順位や頻度)





# 対応ミドルウェア MQTT / WebSocket / ROS

- MQTT: 軽量プロトコル、IoT/M2M向き
- WebSocket: 軽量プロトコル、IoT/M2M向き
- ROS: 基本的には常時接続して利用



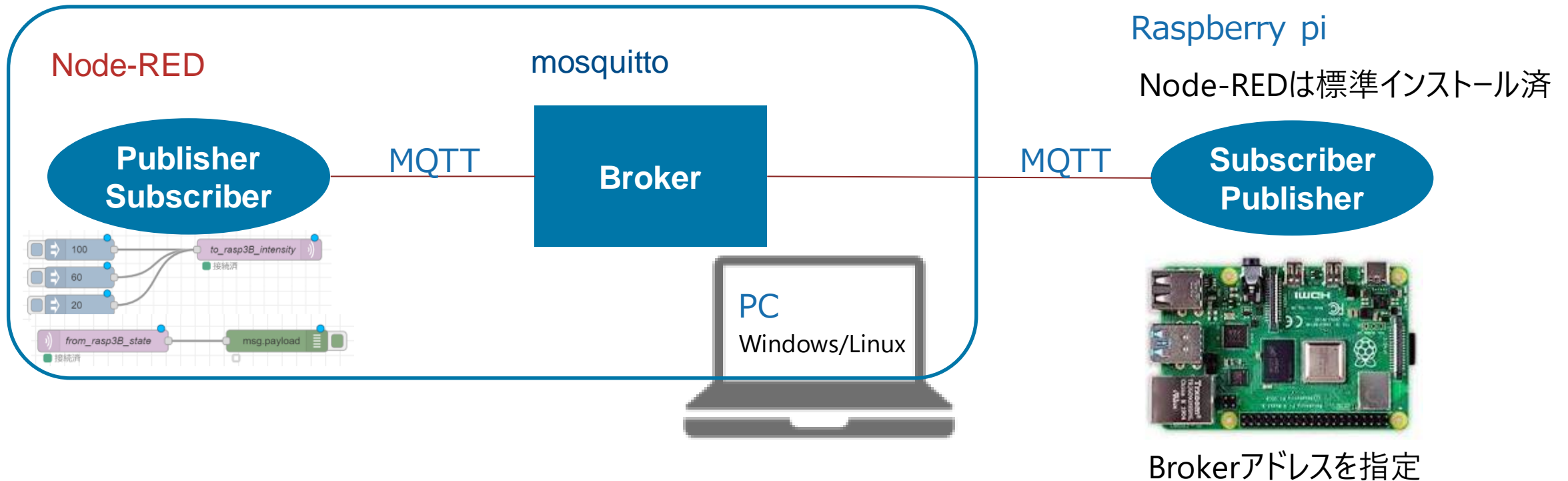
Topic:	home/rasp_4B/start
QoS:	0
Message length (...)	1
<input type="checkbox"/> Output the last received message	

トピックや  
QoSを設定



# MQTTネットワークの構築例

MATLAB開発PC上でも構築できる。ネットワークに関してMATLAB特有の設定は不要  
Ether/WiFi/Modbusなどの基盤ネットワーク上に、Broker(サーバー)を立てる







# CAN通信

ネットワークの物理層を含むプロトコルはOSが管理する(Ether/WiFi/RoLa/LTEなど)  
CANの場合はCAN-Utills(ドライバ)に準拠したCAN-USBデバイスを用いて接続



## CAN Receiveブロック

Parameters	
CAN interface:	can0
<input type="checkbox"/> Set up CAN interface	
Data to be output...	Raw data
Identifier type:	Standard (11-bit identifier)
Message ID:	100
Message length:	8
Sample time:	-1

## CAN Unpackブロック 必要な信号のみ取得

パラメーター	
Data to output as:	raw data
CANdb file:	raw data
Message list:	(none)
Message	
Name:	CAN Msg
Identifier type:	Standard (11-bit identifier)
CAN Identifier:	0
Length (bytes):	8

## CAN Transmitブロック

Parameters	
CAN interface:	can0
<input type="checkbox"/> Set up CAN interface	
Data is input as:	Raw data
<input type="checkbox"/> Wait until data sent	
<input type="checkbox"/> Output status	
Message	
Identifier type:	Standard (11-bit identifier)
Message ID:	100
Message length:	8
<input type="checkbox"/> Request remote frame	

## まとめ



開発環境  
可視化/デバッグ

ハードウェア入出力はブロックを配置して実現  
シミュレーション/ハードウェア実行を可視化してデバッグ



エッジデバイス向け  
アルゴリズム開発

MATLABのソリューションを展開  
実装に適した、省メモリのためのチューニング



通信プロトコル  
ミドルウェア対応

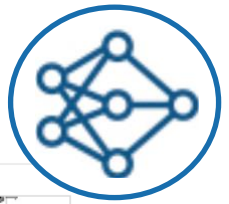
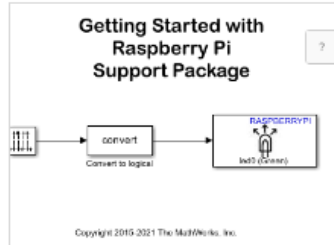
CANやMQTT・ROSなどのプロトコルから  
必要な情報のみ入出力するブロック



# 豊富な実施例

## Examples

### Modeling — Examples

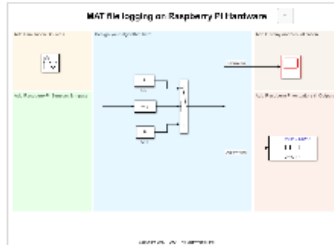



**Getting Started with Raspberry Pi Support Package**

Copyright 2015-2021 The MathWorks, Inc.

**Getting Started with Simulink Support Package for Raspberry Pi...**

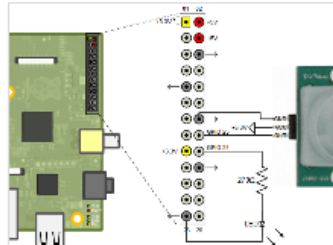
Use Simulink® Support Package for Raspberry Pi® Hardware to run a Simulink model on Raspberry Pi hardware.



**MAT file logging on Raspberry Pi hardware**


**Log Signals in MAT-File Format Using Raspberry Pi**

Log signals from a Simulink® model on Raspberry Pi® hardware in the MAT-file format.



**Detect Motion Using Passive Infra-Red Sensor Interfaced with...**

Use Raspberry Pi® hardware to interface to a motion sensor and control an external LED.



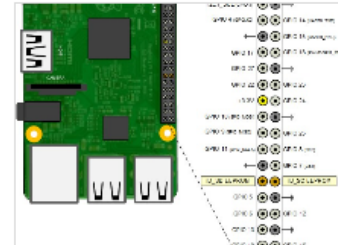
**Get Started with Deploying Web-Based Customizable...**

Use the Simulink® Support Package for Raspberry Pi® Hardware to create an interactive dashboard display



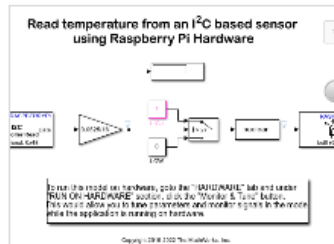
**Get Started with Audio Signal Processing Using Raspberry Pi**

Use the Simulink® Support Package for Raspberry Pi® Hardware to get started with audio signal processing



**Communicate with EEPROM Using Raspberry Pi**

Use Simulink® Support Package for Raspberry Pi® Hardware to read from and write to an SPI EEPROM.

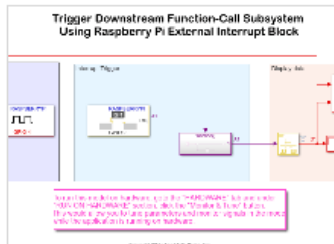


**Read temperature from an I2C based sensor using Raspberry Pi Hardware**

Copyright © 2018-2020 The MathWorks, Inc.

**Read Temperature from TMP102 Sensor**

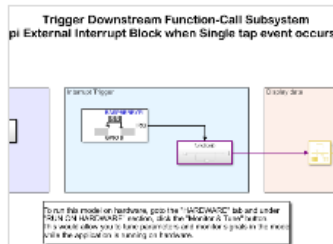
Illustrates how to use Simulink® Support Package for Raspberry Pi® Hardware to configure and read temperature from a



**Trigger Downstream Function-Call Subsystem Using Raspberry Pi External Interrupt Block**

**Trigger Downstream Function-Call Subsystem Using...**

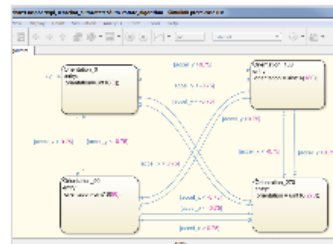
Use the Simulink® Support Package for Raspberry Pi® Hardware to trigger a downstream function-call in the



**Trigger Downstream Function-Call Subsystem Using Raspberry Pi External Interrupt Block when Single tap event occurs**

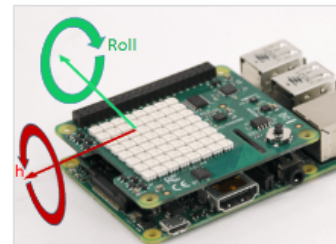
**Trigger Downstream Function-Call Subsystem Using...**

Use the Simulink® Support Package for Raspberry Pi® Hardware to trigger a downstream function-call in



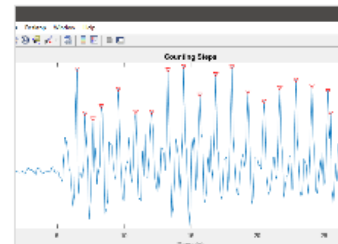
**Auto-Rotate Image Displayed on Raspberry Pi Sense HAT LED Matrix**

Develop a Simulink® model to implement an algorithm to read the Accelerometer On-board Sense HAT and control the



**Control LEGO MINDSTORMS EV3 Robot Using Raspberry...**

Read the Accelerometer on Raspberry Pi® Sense HAT and use this data to control the motion of a LEGO®



**Count Steps Using Raspberry Pi Sense HAT**

Demonstrates an application that counts the number of steps a person walked while holding a Raspberry Pi® Sense HAT.

# その他のハードウェアサポートパッケージ

代表的なデバイスにはハードウェア サポートパッケージをご用意

## NVIDIA Jetson / Drive



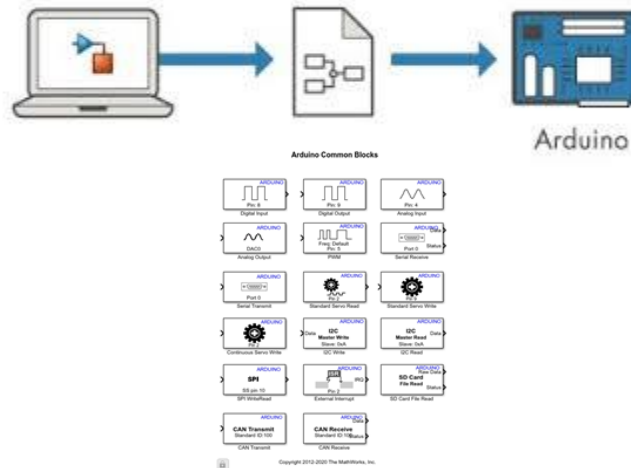
[MATLAB Coder Support Package for NVIDIA Jetson and NVIDIA DRIVE Platforms](#)

MATLAB Coder™

GPU Coder™

R2021a

## Arduino



[Getting Started with Arduino® Hardware](#)

Simulink Coder™

## その他マイコンボード

Texas Instruments

[Get Started with C2000 Microcontroller Blockset](#)

STマイクロエレクトロニクス

[Simulink Coder Support Package for STMicroelectronics Nucleo Boards](#)

NXP K64F

[Simulink Coder Support Package for NXP FRDM-K64F Board](#)

ご清聴ありがとうございました

## 必要なライセンス

<Wavelet/LSTM>  
Deep Learning Toolbox™  
Signal Processing Toolbox

<CAN Unpack>  
Embedded Coder  
Vehicle Network Toolbox  
Simulink Real-Time

<ROS>  
ROS Toolbox