



Simulink アドオンを用いたRapid Prototype 開発の事例紹介

2024 / 05 / 30

Kenichiroh Hara

In-Cabin Perception

アジェンダ

01 Woven by Toyota 会社紹介

02 Simulink® Desktop Real-Time™ (SLDRT) について

03 開発事例

01

Woven by Toyota
会社紹介



企業沿革

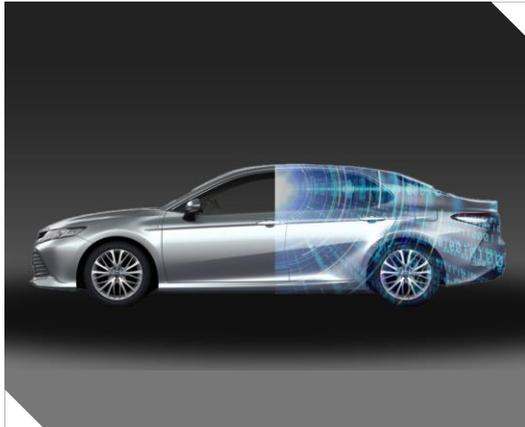


2018年3月
設立



2023年4月
社名変更

重点領域



Arene OS

- 最先端のソフトウェアプラットフォームと車両OS
- ソフトウェアを活用した新たな顧客価値の提供



Woven City

- モビリティのためのテストコースであり実証実験の街
- 様々なパートナーとともに「未来の当たり前」を発明



AD / ADAS

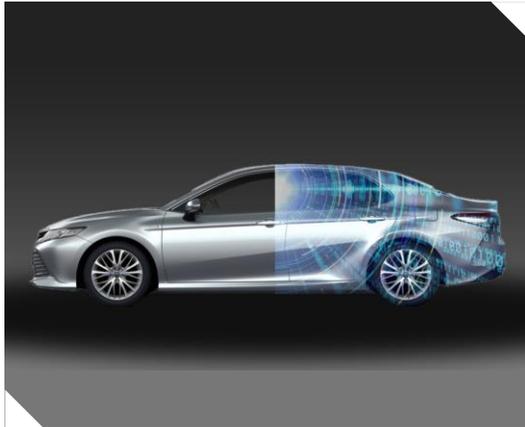
- 安心・安全な自動運転、先進運転システムの開発
- 地理空間インテリジェンス
- 人を中心としたモビリティ



Woven Capital

- グロースステージを対象としたベンチャーファンド
- モビリティの未来に向けたイノベーション創出のための投資を行う

重点領域



Arene OS

- 最先端のソフトウェアプラットフォームと車両OS
- ソフトウェアを活用した新たな顧客価値の提供



Woven City

- モビリティのためのテストコースであり実証実験の街
- 様々なパートナーとともに「未来の当たり前」を発明



AD / ADAS

- 安心・安全な自動運転、先進運転システムの開発
- 地理空間インテリジェンス
- 人を中心としたモビリティ



Woven Capital

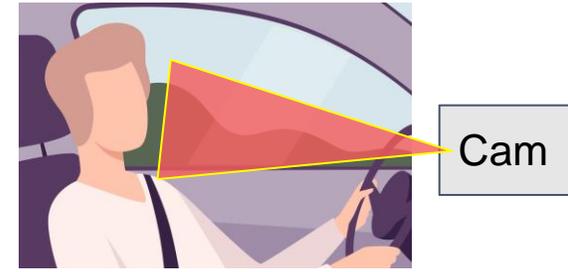
- グロースステージを対象としたベンチャーファンド
- モビリティの未来に向けたイノベーション創出のための投資を行う

ドライバーモニタリングシステムの紹介

ドライバーの姿勢異常を検出し、音や表示などでドライバーへ通知する



Simulink Desktop Real-Time 導入背景



・ 従来の問題点

- ・ USB ドングルによるライセンス管理等、シミュレータ開発の台数制限があった。
また、シミュレーション実行までの手順が多く取扱いにくい側面があった。
→ スキルの高いエンジニアに限定されがちで、台数制限もあり非効率
- ・ 社内準拠した設計ガイドラインに即したモデルによって、
従来のシミュレーション環境上で実行エラーが発生し、シミュレータの開発元から、
この設計モデルがサポート外という取扱いとされてしまった。
→ サポート外のまま開発が必要となった

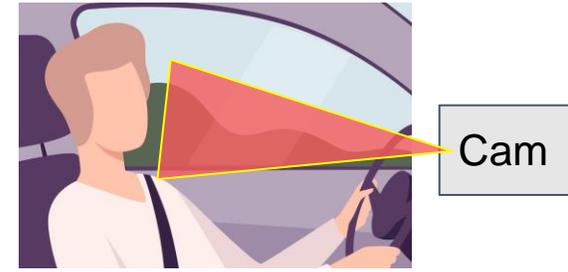
・ 目標

シンプルで取扱いやすいシミュレーション環境の構築をすること

・ アプローチ

「MATLAB® / Simulink® + Simulink Desktop Real-Time」 を活用

Simulink Desktop Real-Time を導入した結果



・ SLDRT を導入してよかった点

- ・ Matlab ライセンスを持つエンジニアすべてに使用可能となったため、SLDRTエンジニアが増え分業できるようになった。直感的で引継ぎやすかった。
- ・ PoC の準備時間を短縮できるようになった。(数週間→数日)
- ・ 車両への組付け/取り外しが容易になり複数車両展開も容易に分業できるようになった。

・ SLDRT の導入で困難だった点

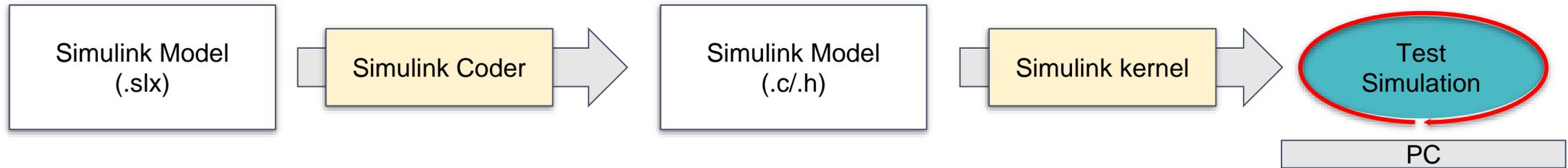
- ・ SLDRT に関する情報が Web 上にはほぼ皆無 (Mathworks のサポートの一本足打法)
- ・ リアルタイム性について理解を得られるのが難しい側面があった。

SLDRT で取り上げる内容

- Simulink Desktop Real-Time (SLDRT) アドオンって何？
- SLDRT ってどの程度リアルタイム？
- SLDRT って何がすごいのか？
- SLDRT を使って何の Prototype を作ったのか？
- Prototype を動かすのにどういう工夫が必要か？

・ Simulink Desktop Real-Time (SLDRT) アドオンって何？

パソコン上でリアルタイム環境を構築することができるアドオン



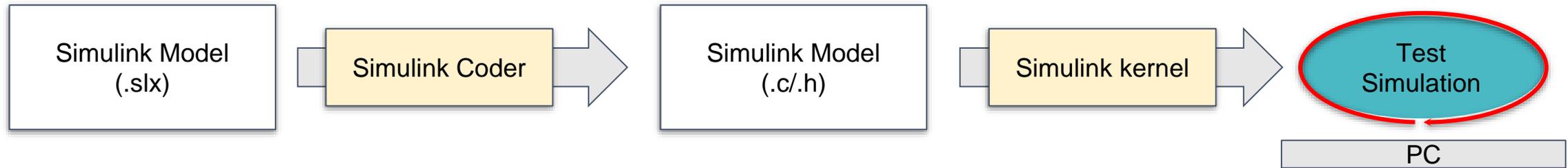
モデルシミュレーション実行としては同じ機能

	実行ボタン	実行方法	特徴
1	 実行 モデルをシミュレーション Ctrl+T	モデルのシミュレーション実行 (Simulink アプリ上で実行)	Simulink 標準のシミュレーション実行機能、追加アドオンなどは不要。
2	 Desktop Real-Time →  Run in Real Time	SLDRT 専用リアルタイムカーネルを使ったシミュレーション実行	追加アドオンで、SLDRT 専用カーネルを OS の外に配置し、Kernel モード実行では、OS 外で処理を行った結果を Simulink 画面上にログ表示。
3	 Simulink Real-Time →  Run on Target	PCと Ethernet 接続した HW 上でのモデルシミュレーション実行	別の専用の HW (Speedgoat 等) 上でシミュレーションを実行し、処理結果を Simulink 画面上にログ表示。

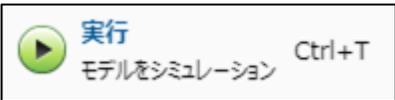
Matlab Software より画像引用

・ Simulink Desktop Real-Time (SLDRT) アドオンって何？

パソコン上でリアルタイム環境を構築することができるアドオン



モデルシミュレーション実行としては同じ機能

	実行ボタン	実行方法	特徴
1	 実行 モデルをシミュレーション Ctrl+T	モデルのシミュレーション実行 (Simulink アプリ上で実行)	Simulink 標準のシミュレーション実行機能、追加アドオンなどは不要。
2	 Desktop Real-Time →  Run in Real Time	SLDRT 専用リアルタイムカーネルを使ったシミュレーション実行	追加アドオンで、SLDRT 専用カーネルを OS の外に配置し、Kernel モード実行では、OS 外で処理を行った結果を Simulink 画面上にログ表示。
3	 Simulink Real-Time →  Run on Target	PCと Ethernet 接続した HW 上でのモデルシミュレーション実行	別の専用の HW (Speedgoat 等) 上でシミュレーションを実行し、処理結果を Simulink 画面上にログ表示。

Matlab Software より画像引用

SLDRT はどの程度 リアルタイムなのか (1/4)

SLDRTは PC 上で処理されるもの

- OS Kernel がプロセスの実行順序を決める並列処理が行われる
 - 優先順位を決められない。リソースの割込みもある。
 - リアルタイム性の保証ができない？

高性能な PC で処理すればリアルタイムになる？

- 常に処理時間内に間に合わせる
 - ベストエフォートによる実質リアルタイム処理を実現ということなのか？
たまたま割込みが入ったら保証できるのだろうか。



OS 上で動く論理のアーキではない

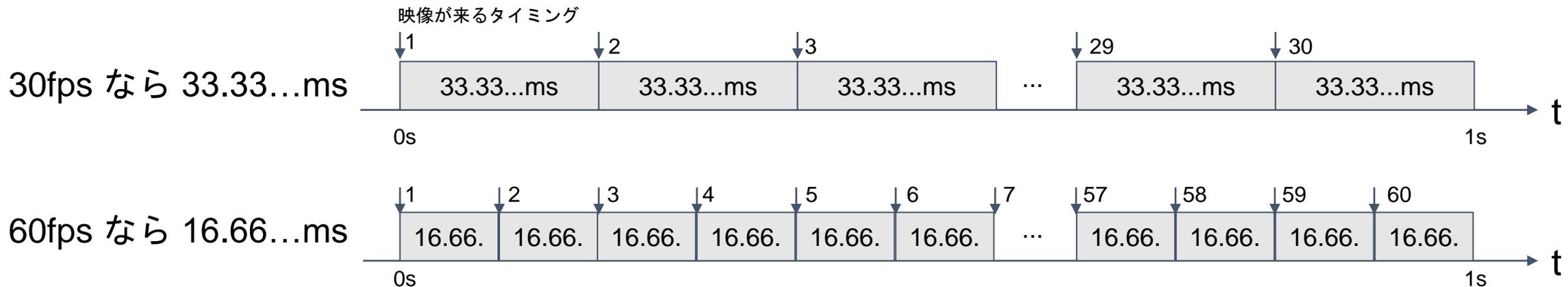
SLDRT はどの程度 リアルタイムなのか (2/4)

制御プログラミングにおけるリアルタイム処理とは

「決められた時間内に確実に計算を実施して、結果出力ができること」

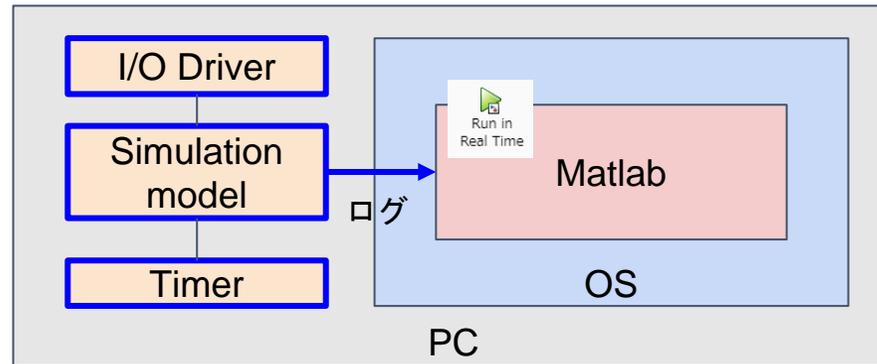


例) カメラの動画をフレームバイフレームで処理することが必要な場合、



SLDRT はどの程度 リアルタイムなのか (3/4)

SLDRT実行アーキテクチャ



■ Kernel Mode で動作

SLDRT の Kernel Mode 機能は、OS の Kernel-Mode といった特別な動作方法で実行させ、OS 外で Timer 管理を実施。また Simulation Model や I/O Driver も Timer 管理の中で処理を実施する。

- OS Kernel が決めるプロセスの並列処理とは別で動作
 - リアルタイムに動く。

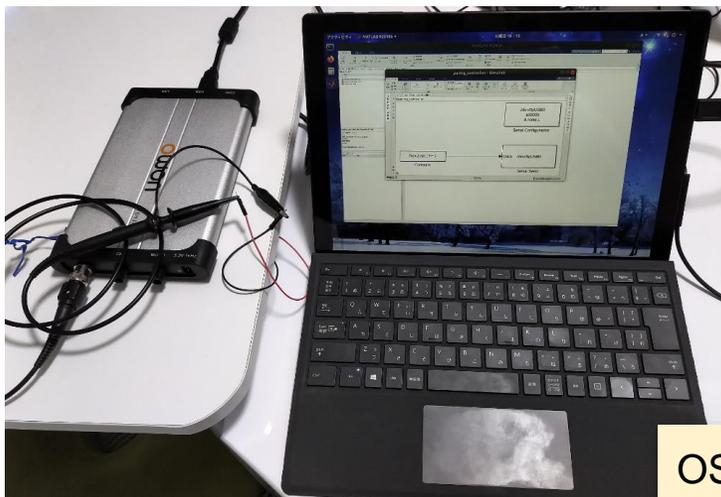
■ ログの出力結果

ログは、Simulink アプリ上に表示されるように通知される。

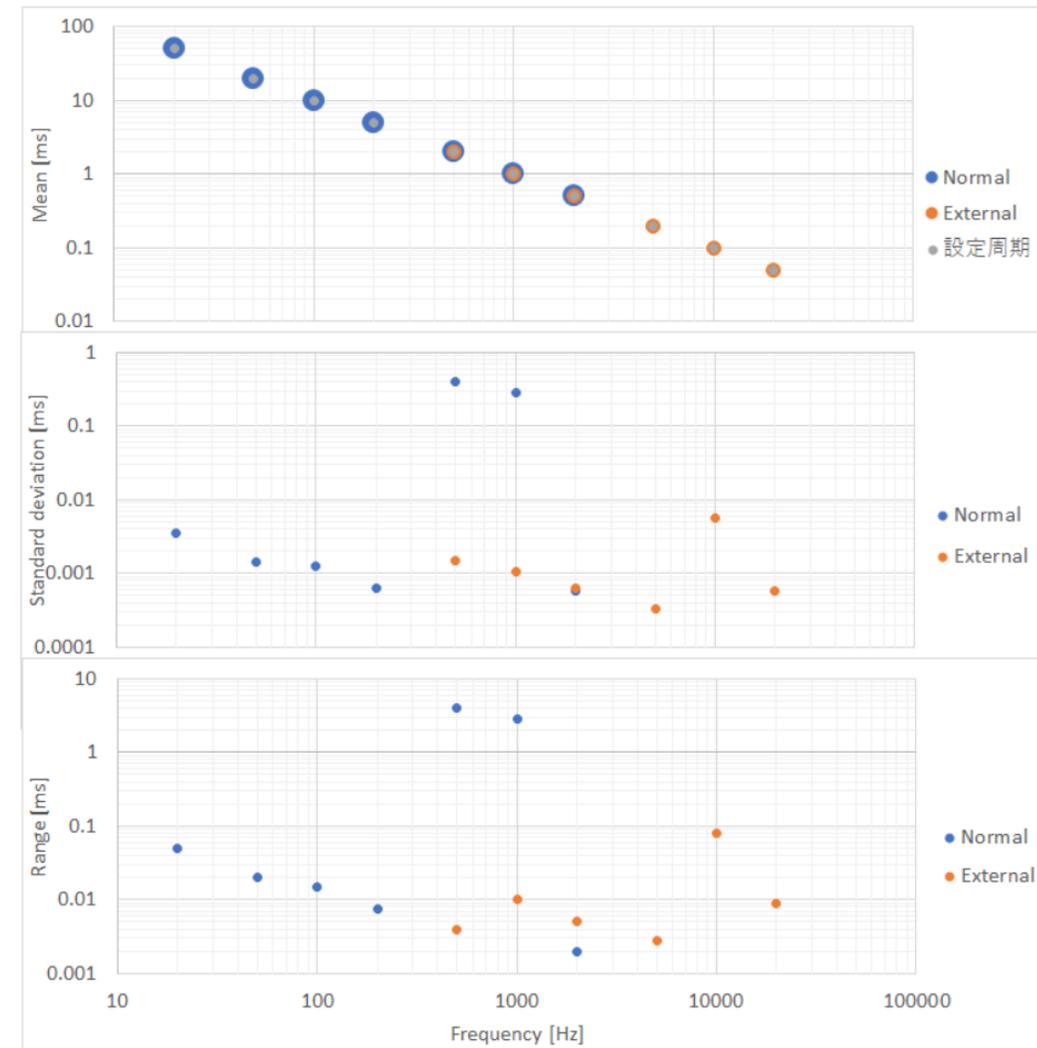
SLDRT はどの程度 リアルタイムなのか (4/4)

ThunderboltマルチファンクションボードMF644
を用いて Connected I/O Mode (旧Normal Mode) と
Kernel Mode (旧External Mode) の処理結果 →

※ 約10000Hz の処理時間 100 μ s オーダーになると標準偏差等から外れ値を確認できる。



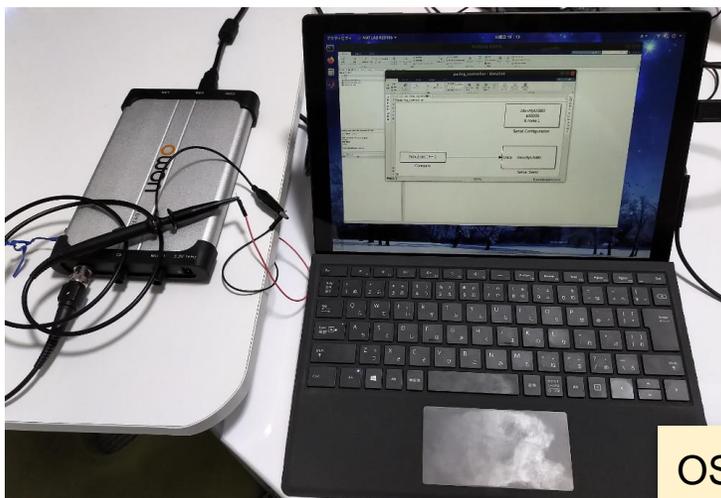
OS : Windows10 Enterprise 1903
CPU : Core i7-8650U
GPU : UHD Graphics 620
RAM : 16GB
Storage : SSD 512GB



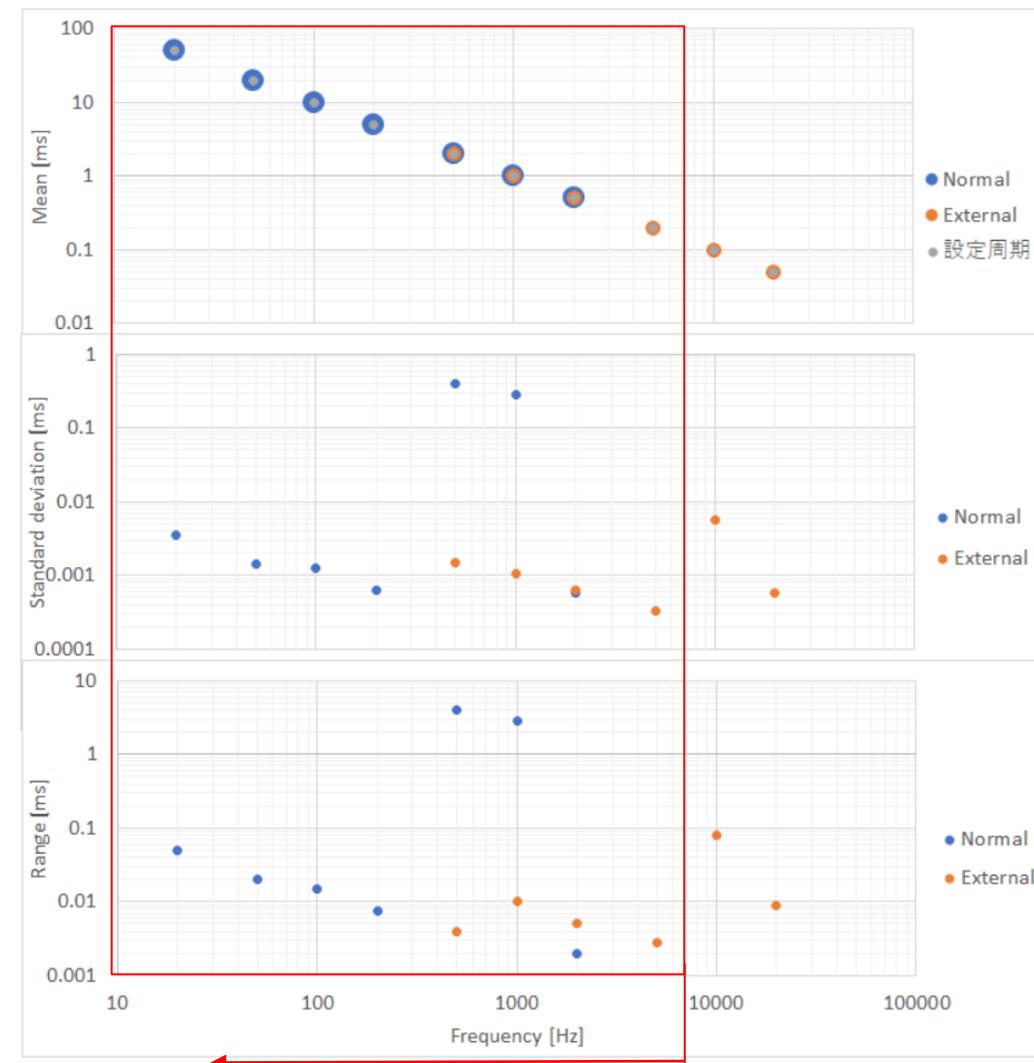
SLDRT はどの程度 リアルタイムなのか (4/4)

ThunderboltマルチファンクションボードMF644
を用いて Connected I/O Mode (旧Normal Mode) と
Kernel Mode (旧External Mode) の処理結果 →

※ 約10000Hz の処理時間 100 μ s オーダーになると標準偏
差等から外れ値を確認できる。

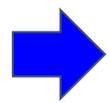


OS : Windows10 Enterprise 1903
CPU : Core i7-8650U
GPU : UHD Graphics 620
RAM : 16GB
Storage : SSD 512GB



SLDRT って何がすごいのか？

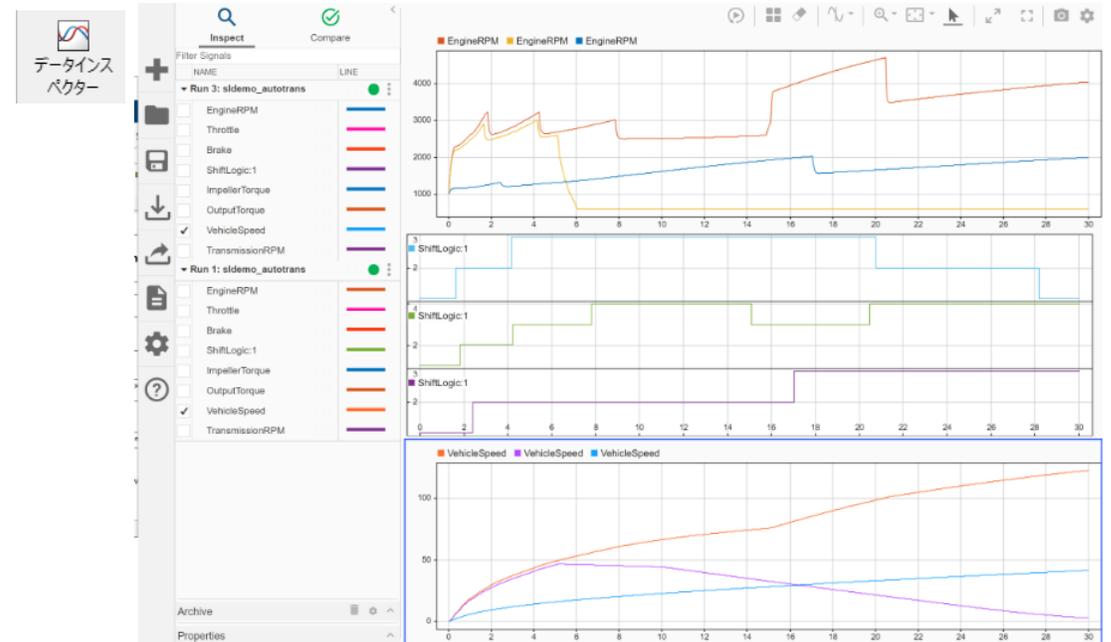
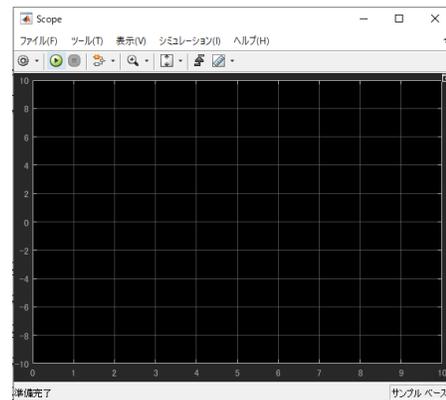
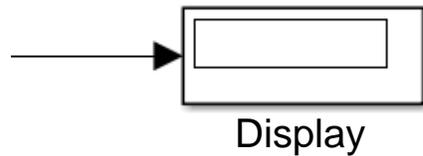
- ・ PC をリアルタイム制御デバイスのように扱える
- ・ PC と Matlab とアドオンライセンスのみ
※包括契約を結んでいれば追加料金が不要という
- ・ Simulinkモデルを修正してすぐに動かせる
- ・ 使い慣れた GUI のモデル上にログ表示できる
- ・ Borrow ライセンスがあれば StandAlone で動かせる



- ・ Rapid Prototype (ハード)の購入前の検討用としてお試しで動かせる
- ・ 開発コストを抑えられる可能性あり

SLDRT って何がすごいのか？

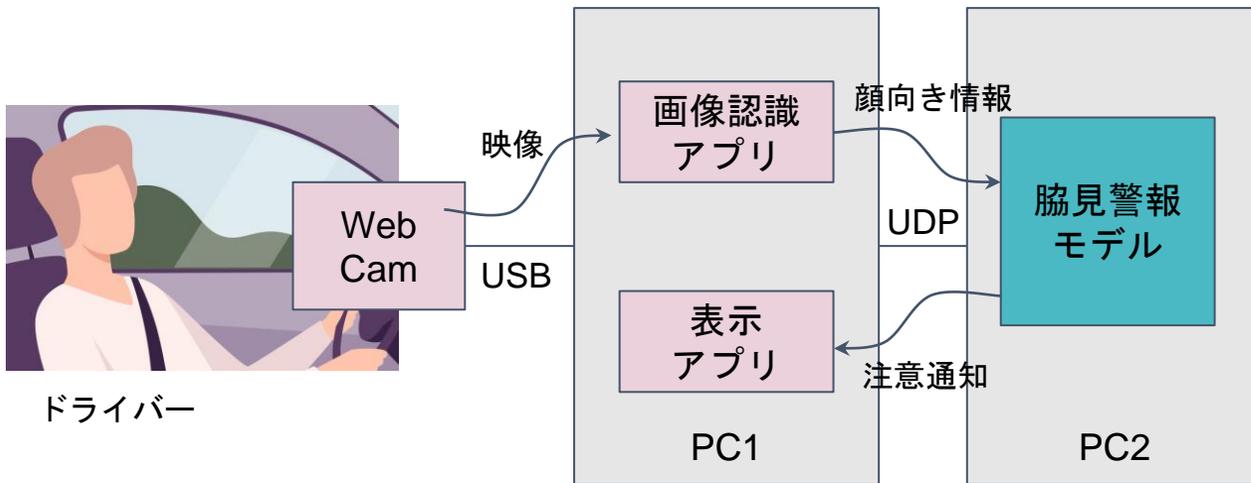
SLDRT を実行中のログ表示



使い慣れた GUI をそのまま活用することができる。
しかし、多用すると Windows のリソースが重くなる傾向があるので適度な数が好ましい。

SLDRT を使って何の Prototype を作ったのか？

ドライバーモニタリングシステムの PoC



- ・今回使用した構成
 - ・ Laptop PC x 2

マウスコンピュータ DAIV 6H
Windows 10 Pro (x64)
Core i9-12900H
64GB Memory
4TB SSD
NVIDIA GeForce RTX 3070 Ti Laptop GPU /

GDDR6 8GB

- ・ Matlab/Simulink 2021a
- ・ Simulink Desktop Real-Time
 - ・ Simulink Coder
- ・ USB Camera
- ・ 画像認識アプリ & 警報音再生アプリ

Demo movie

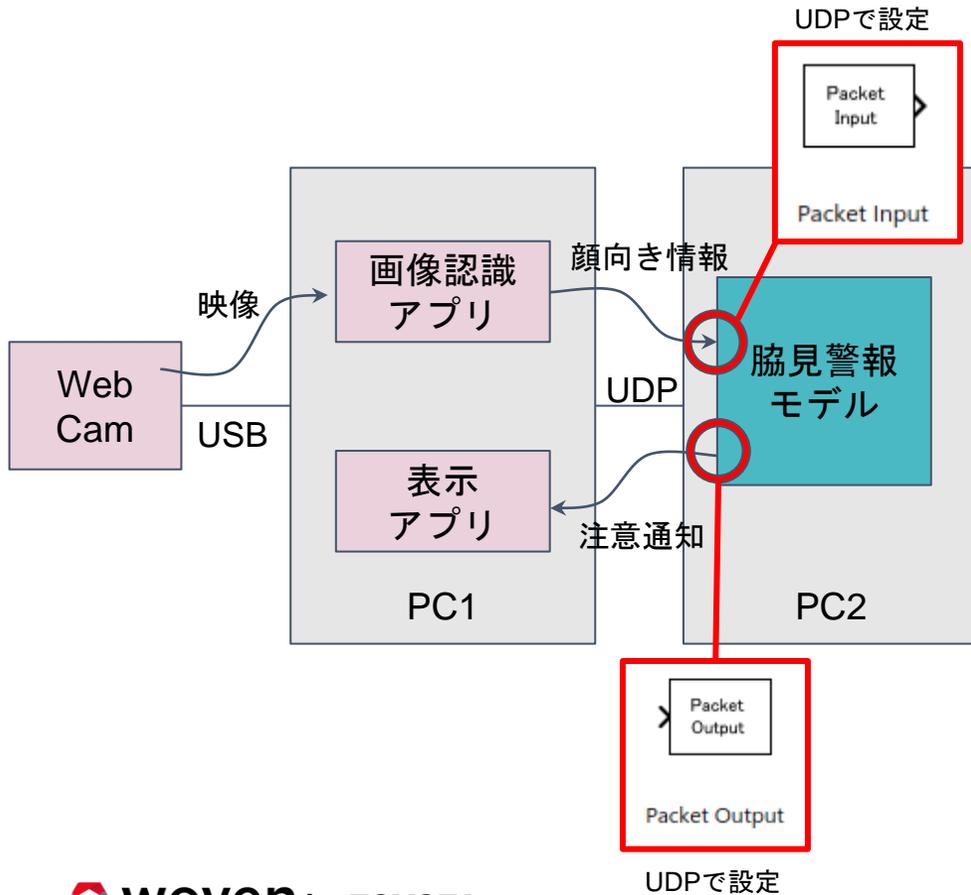
- ・ 脇見を検出する

ドライバーが右を向くと顔向きが検出され、ある閾値を一定時間超えると脇見として注意通知するデモを紹介いたします。

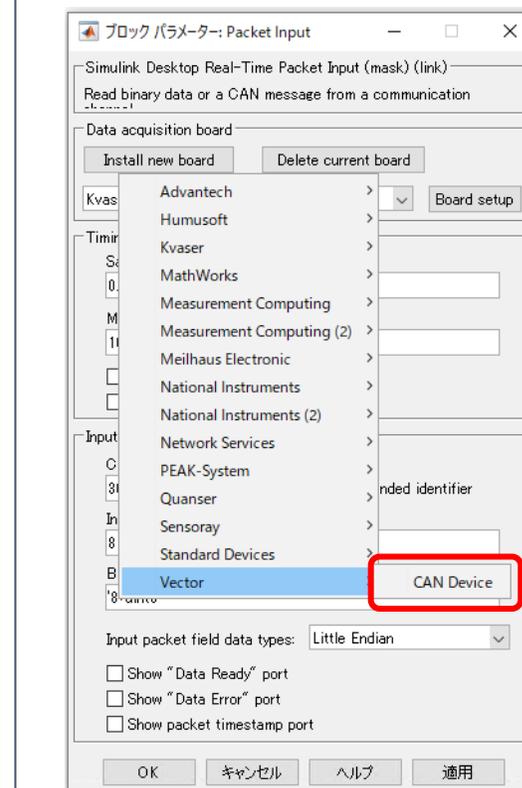


SLDRT を使って何の Prototype を作ったのか？

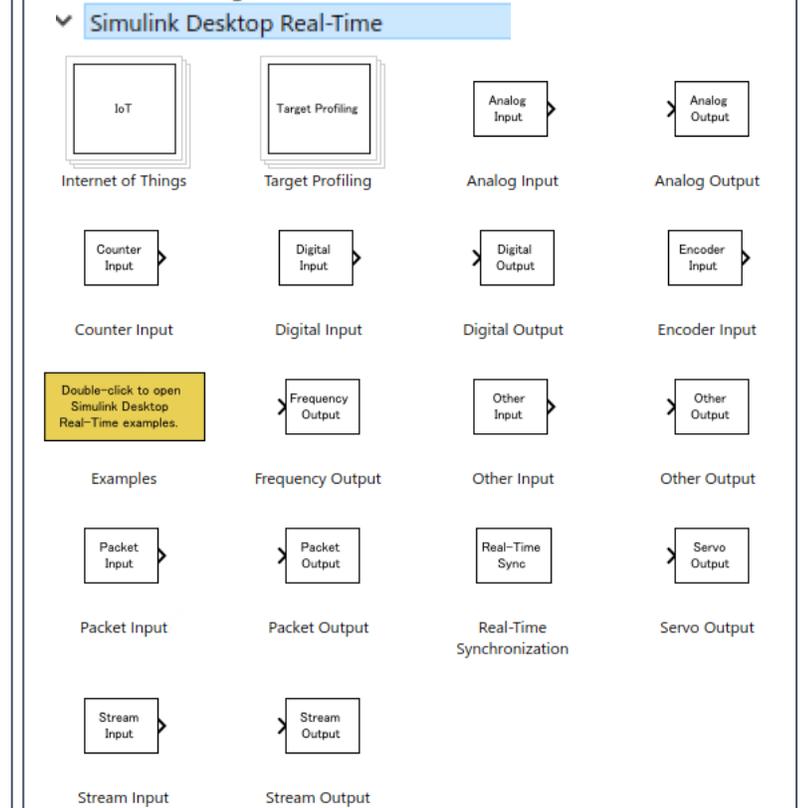
今回、Packet Input/Output ブロックで UDP 通信を実現。
Ethernet に限らず CAN、様々な I/O Port を持つため車両開発以外にも汎用性はそれなりにある。



UDP以外にも I/O 選定可能



他の SLDRT向け ライブラリ



Prototype を動かすのにどういう工夫が必要か？

Simulink DeskTop Real-Time で動かすための必要な準備・テクニック

製品設計モデルを動かすため、工夫・手当てしたポイント

項目	概要	備考
システムターゲットファイルの設定	sldrt.tlc に変更すること。	Kernel モードとして動かせるようになる。
IO のハーネス設計の追加	入力データの設計 出力データの設計	UDP で渡す場合、独自通信プロトコル設計が必要。
ライブラリブロックの汎化処理	製品モデルで使用しなければならない専用ライブラリは、Simulink Coder でコンパイルエラーになる場合、これらを汎用的な Simulink ライブラリに置き換える必要がある。 これを Matlab スクリプトで記述し、SLDRT 開始前に実行するのが望ましいです。	
メモリ配置の汎化処理	製品モデルで指定しなければならない専用メモリセクションがある場合、これらを汎用的なメモリセクションに置き換える必要がある。同様にスクリプトに記述が好ましい。	

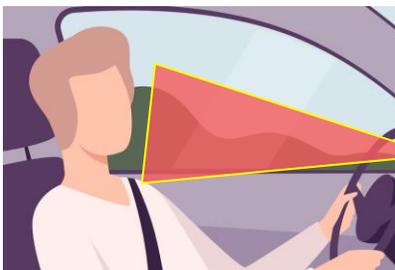
Prototype を動かすのにどういう工夫が必要か？

ビルドを成功させるためのもうひと工夫

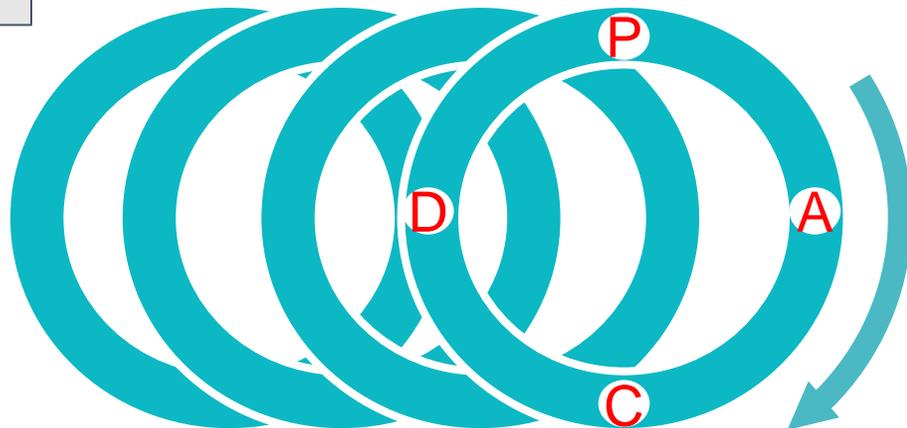
対処	概要	ヘルプ
SaveWithParameterizedLinksMsg を 'none' に設定する。	「ブロック線図はパラメーター付きのライブラリ リnkを含んでいます」という警告が出る場合は、 'none' にして無視します。	Link
ModelReferenceIOMismatchMessage は、'warning' にする。	親モデルと参照モデルのインターフェイスの不一致によるシミュレーションエラーを検出できるように設定する。	Link
SignalLabelMismatchMsg (信号ラベルの不一致) は、'warning' にする。	信号ラベルの不一致に対して警告する設定にする。	Link
UseDivisionForNetSlopeComputation (固定小数点の正味勾配計算に除算を使用)は、'off' にする。	整数乗算とそれに続くシフトを使って正味スケールリング計算を実行します。	Link
UseRowMajorAlgorithm (行優先の配列レイアウトに最適化されたアルゴリズムを使用)は、'off' に設定する。	既定設定がオフだが明示的なオフ設定をする。	Link
SharedChecksumLength の数字を確認。	チェックサムの長さを増やすことで競合の可能性を軽減させることができます。我々は 8 に設定。	Link
CustomInclude のカレントパスの設定	Custom Include ファイルを設定する場合のカレントパスの設定を明示的にしておく。	Link
CustomHeaderCode の作成	CustomHeaderCode に xxx.h ファイルを用意し、global な #define を定義する。	Link
ModelReferenceNumInstancesAllowed (最上位モデルごとに可能なインスタンスの総数)の設定をする	複数のモデルが参照構造にある場合は、Single にするべきか、Multi にするべきかを指定する必要があります。	Link



お使いの環境やロジックによっては、他にも対策が不要・必要なものがあると思いますが、少しでも助けになれば幸いです。



Cam



コンセプト/要求仕様検討

量産コード生成

この Prototype を使って毎月評価を実施。評価結果から得た対策・改善でソフトウェアのアップデートを繰り返し、ハードウェアが出来上がる頃には、一定品質のソフトウェアを織り込むことに成功した。

本講演で取り上げた内容まとめ

- Simulink Desktop Real-Time (SLDRT) アドオンって何？

→ PC上でリアルタイム環境を構築して動かせる Simulink アドオンソフト

- SLDRT ってどの程度リアルタイム？

→ 約1ms以上の周期なら RealTime な処理周期として取り扱えそうである。

- SLDRT って何がすごいのか？

→ Simulink を直接修正 Debug実行できる。Stand Alone 使用可。開発コストの節約に繋がられる。

- SLDRT を使って何の Prototype を作ったのか？

→ ドライバーモニタリングシステムの PoC (Driver in the loop) を作成

- Prototype を動かすのにどういう工夫が必要か？

→ 製品固有コードの汎化、Block 間参照関係の整理。Option 設定の見直し。
これらを m スクリプトにして、SLDRT 実行前に実施すると簡単な運用ができる。



Thank you

ありがとうございました

