Confidential
※複写および第三者への開示をお断りいたします。

# ラズパイを使った車載 CAN 情報収集とクラウド利用の実験をしてみた!

2022年2月21日

株式会社サニー技研 ビジネス開発事業部
テクノベート課　濱邊 真也

© 2022 Sunny Giken Inc. All rights reserved.

# 1.あらまし

Confidential
※複写および第三者への開示をお断りいたします。

- 近年、自動運転・コネクテッド・MaaS・・など、CAN 通信情報を取得して、様々なサービスや分析に活用することが盛んに行われています。身近に利用できるエッジコンピュータで同様のことをやってみようということでラズパイを使った CAN 情報収集の実験を行いました。
  ラズパイを柱に CAN インターフェース接続、データ加工、無線によるクラウド連携の技術、そして実験で得られた結果や気づきなどを紹介します。

- 登壇者略歴
  - 2014年 ～ モータ制御やRTOS等の研究開発に従事
  - 2016年 ～ 車載ネットワークのプラットフォーム開発に従事
  - 2020年 ～ 車載ネットワーク向けセキュリティやIoTを組み合わせた製品の研究開発に従事

© 2022 Sunny Giken Inc. All rights reserved.

## 2. 実験動機[1/2]

- 製品開発業務の要望

多種多様なPoCを作り、製品の幅を広げる手がかりを得たい

しかし

- 携わる製品の多くが時間制約の厳しいCANを含んだシステム
  - 低級言語による開発, OSSが少ない, 専用ドライバが必要

などなど。。。

開発コストが大きく, PoCつくりもたいへん

リアルタイム性を担保しつつ、開発コストを小さくしたい

© 2022 Sunny Giken Inc. All rights reserved.

# 3. 実験動機[2/2]

- 手軽に使えるRaspberry Piの利用を検討
  - Raspberry Piが手軽なのは、汎用OSが利用できるから
    - 課題：
      - 汎用OSでは、時間制約を守ることが難しい

      - リアルタイムOSでは、OSSの利用が難しい（開発コスト増）

時間制約の保証は、専用ハードウェアに任せ、
他の処理はRaspberry Piで実現できないか？

**実験してみた**

© 2022 Sunny Giken Inc. All rights reserved.

# 4. 専用ハードウェアとRaspberry Piの協調

Confidential
※複写および第三者への開示をお断りいたします。

- Raspberry Piと専用ハードウェアの得意分野をいいとこ取り

## Raspberry Piの得意分野

- クラウドとの接続
  - 世の中に開発実績が豊富
  - 充実したオープンソースソフトウェアが存在
- 設定UI
  - Webブラウザを介したUIが容易に実装できる

開発期間短縮

## 専用ハードウェア得意分野

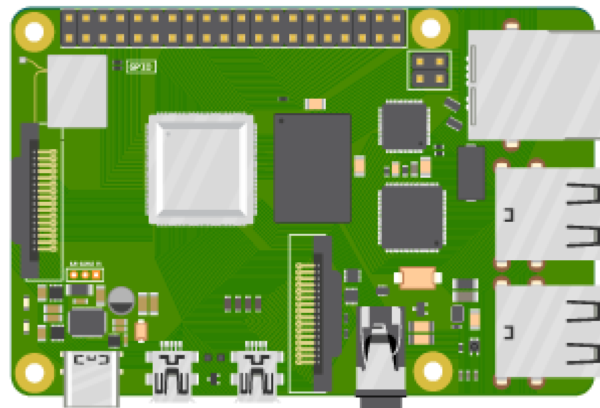
- CAN通信のログ生成
  - usecオーダーで正確なタイムスタンプが取得可能
- CANメッセージ送信制御
  - 厳密時間で定期送信ができる

時間制約の保証

© 2022 Sunny Giken Inc. All rights reserved.

# 5. 実験で利用するハードウェア

- RaspberryPi4
  - CPU
    - ARM Cortex-A72
  - RAM
    - 4GB
  - OS
    - Ubuntu 20.04
  - 記憶メディア
    - SD/SSD



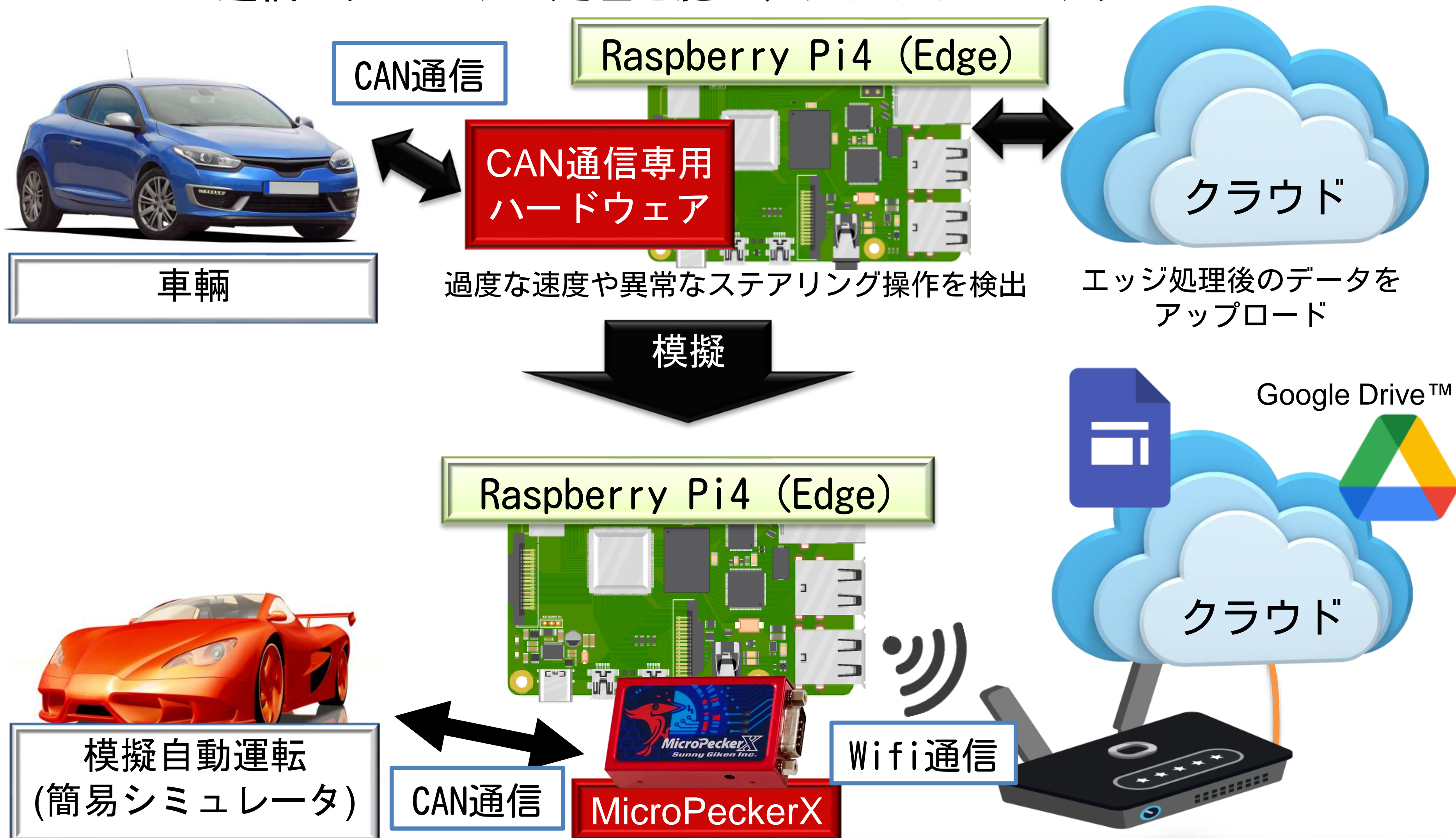
- 専用ハードウェア
  - MicroPeckerX
    - 既製品
    - CAN/CAN-FDに対応
    - Linux対応
    - 実車にも使われるトランシーバ/マイコンを搭載

© 2022 Sunny Giken Inc. All rights reserved.

# 6. 実験のハードウェア構成

Confidential
※複写および第三者への開示をお断りいたします。

自動運転の精度向上や不具合解析に役立てるシステムを想定
CAN通信ログにエッジ処理を施し、クラウドにアップロード

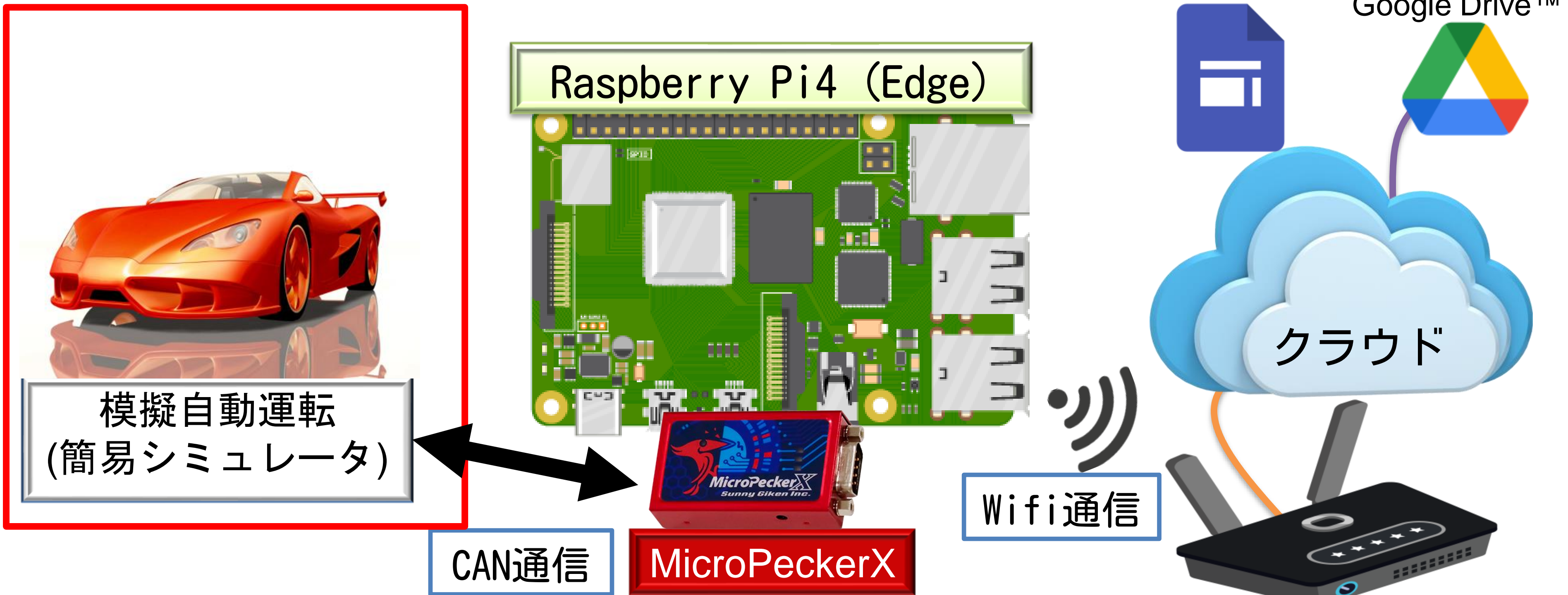



© 2022 Sunny Giken Inc. All rights reserved.

# 8. 実験シナリオと構成(模擬自動運転)

Confidential
※複写および第三者への開示をお断りいたします。

- 運転走行中のCAN通信を模擬することを担当する
  - Ede処理が必要な情報を生成する
  - 生成する情報は「速度情報」と「ステアリング舵角」

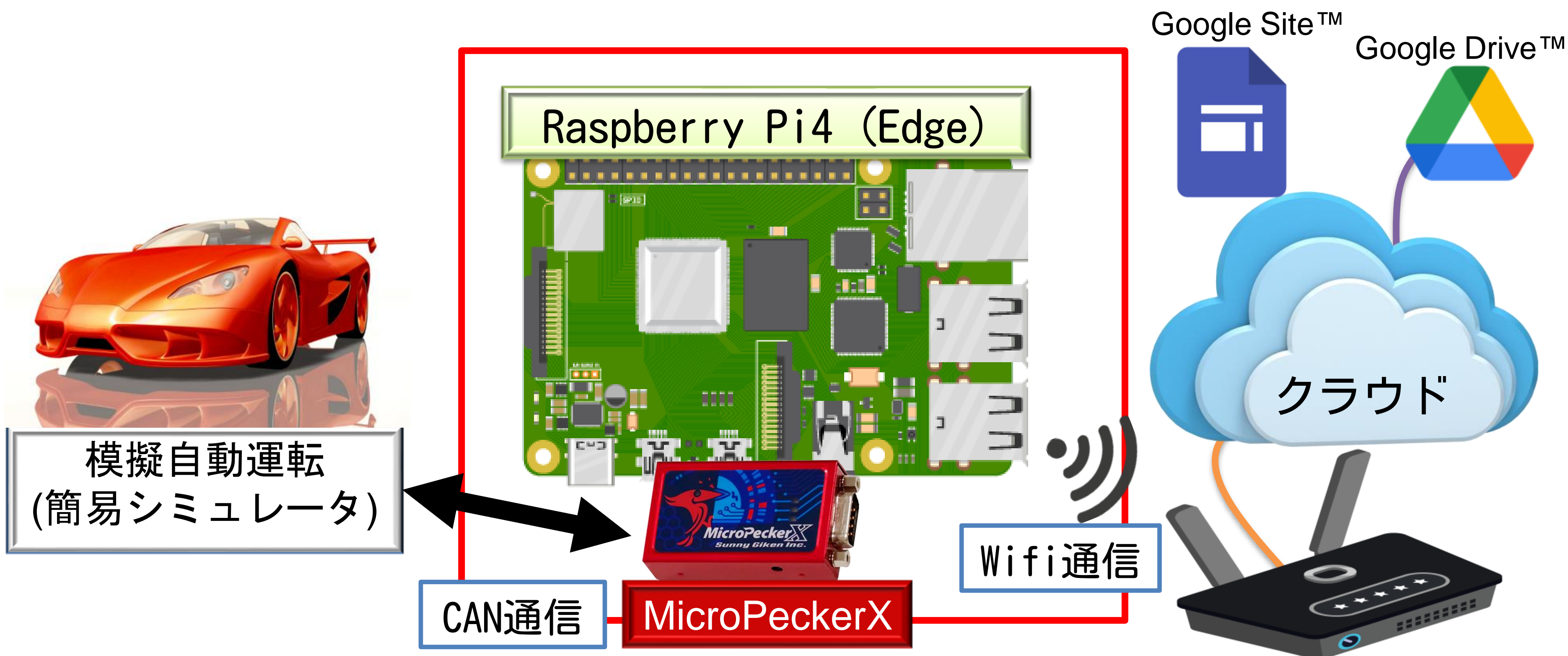



© 2022 Sunny Giken Inc. All rights reserved.

# 9. 実験シナリオと構成(Edge)

Confidential
※複写および第三者への開示をお断りいたします。

- リアルタイムデータのエッジ処理を担当する
  - 車両情報を取得及び処理を行う
  - クラウドサーバに処理済みデータをアップロードする

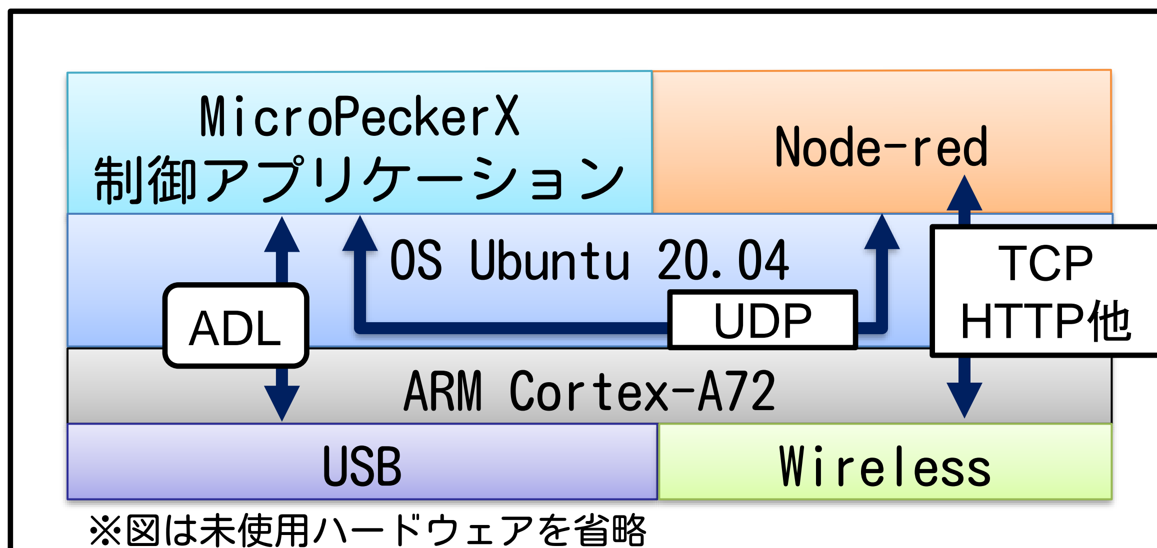


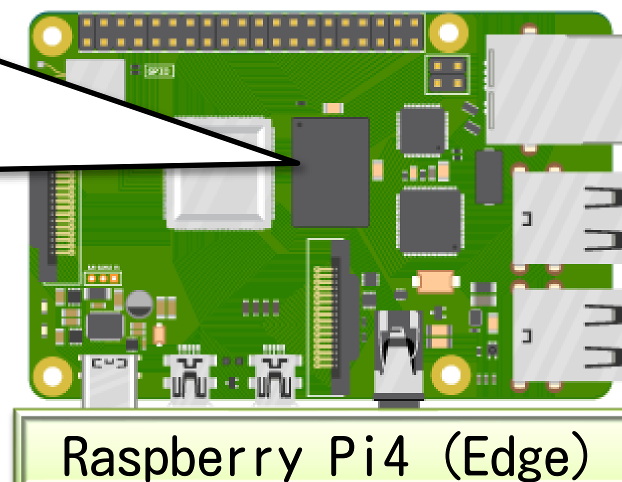

© 2022 Sunny Giken Inc. All rights reserved.

# 10. 実験シナリオと構成(Edge:詳細[1/4])

Confidential
※複写および第三者への開示をお断りいたします。

- Raspberry Pi4のソフトウェア構成概要
  - MicroPeckerX制御アプリケーション
    - CAN/CAN-FDデータの取得、シグナルデータの抽出
    - 開発言語：C++(20)
  - Node-REDアプリケーション
    - MicroPeckerX制御アプリケーションの設定用UI
    - データ整理、クラウド通信
    - 開発言語：ブロックダイアグラム、一部JavaScript



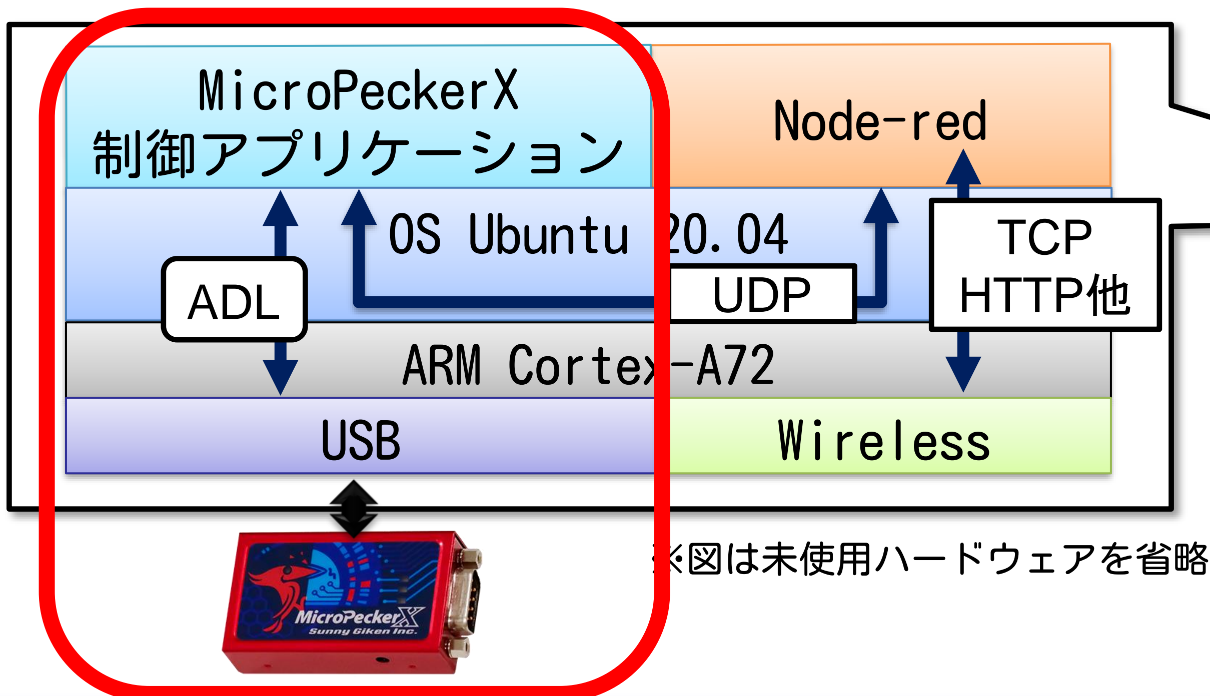



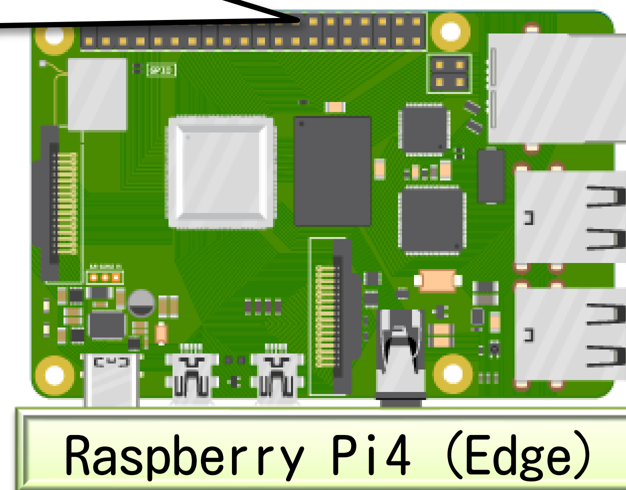

© 2022 Sunny Giken Inc. All rights reserved.

Sunny Giken Inc.

# 11. 実験シナリオと構成(Edge:詳細[2/4])

- MicroPeckerX制御アプリケーション(開発言語の選択)
  - リアルタイム性を重視してC++言語で開発
    - CANシグナルデータの取得はビット切り出しを要し、メモリを直接扱う言語でなければパフォーマンスが厳しい
      - 一般的な言語では、C又はC++のみ候補
    - それでも開発の容易性と保守性を堅持したい
      - STLが豊富でラムダ式記述も可能なC++20を選択



Raspberry Pi4 (Edge)

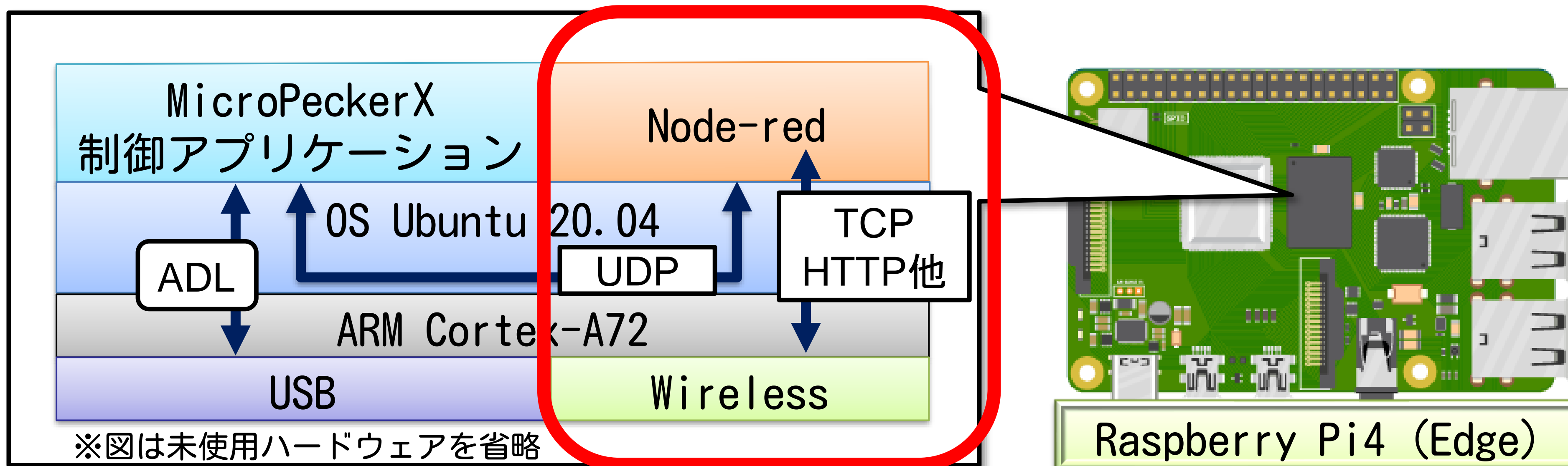
※図は未使用ハードウェアを省略


© 2022 Sunny Giken Inc. All rights reserved.
Sunny Giken Inc.

# 12. 実験シナリオと構成(Edge:詳細[3/4])

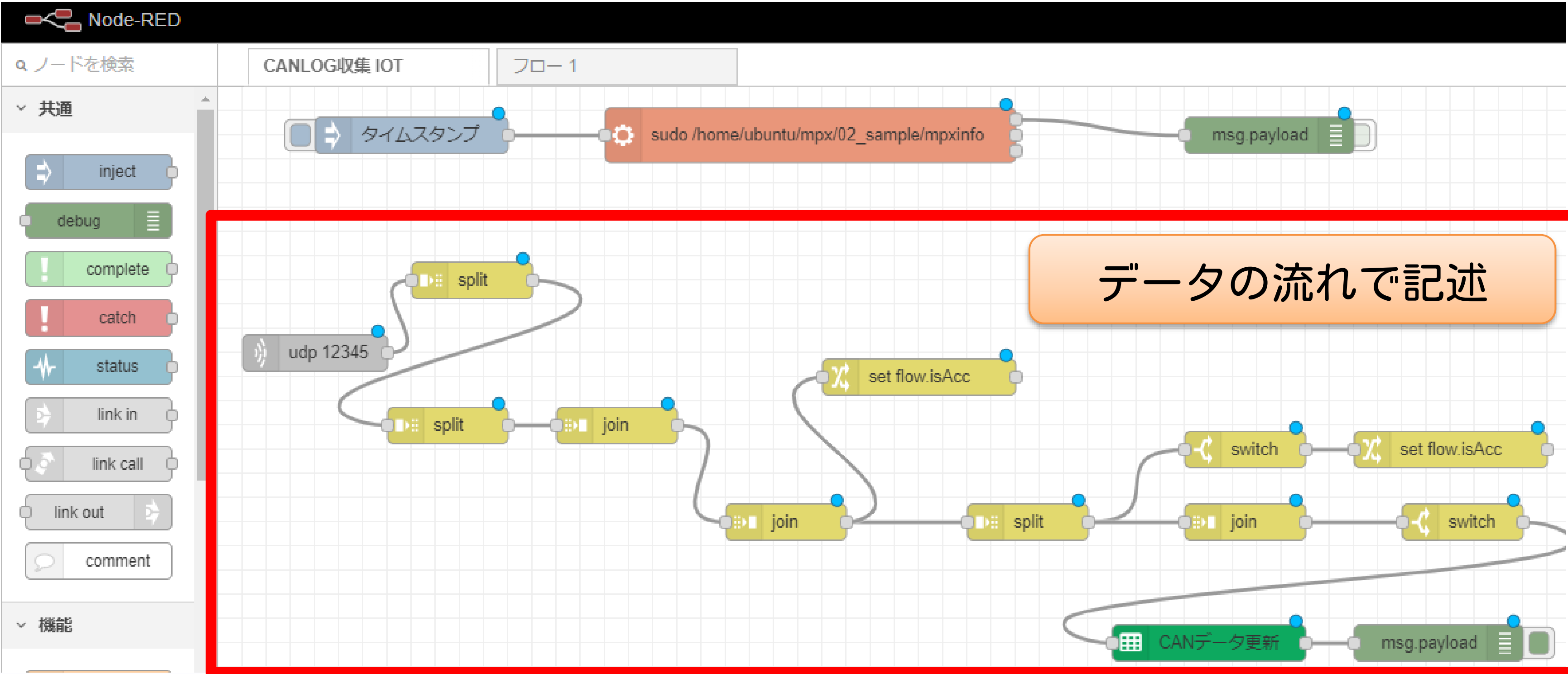
Confidential
※複写および第三者への開示をお断りいたします。

- Node-REDアプリケーションの役割
  - MicroPeckerX制御アプリケーションの設定用UI
    - Raspberry Pi単体ではHMIが無いため
  - データ整理、クラウド通信
    - MicroPeckerXからCAN通信ログとシグナル情報を受け取り、必要な情報のみ、Google Drive内のGoogle スプレッドシートに書き込む



# 13. 実験シナリオと構成(Edge:詳細[4/4])

- Node-RED
  - ローコード/ノーコードで開発できる
    - プログラマでなくてもわかりやすい
    - 場合によっては実装、保守も任せることも可能？
  - データフローで記述する
    - CANのログデータはストリームで受け取るため、親和性が高い



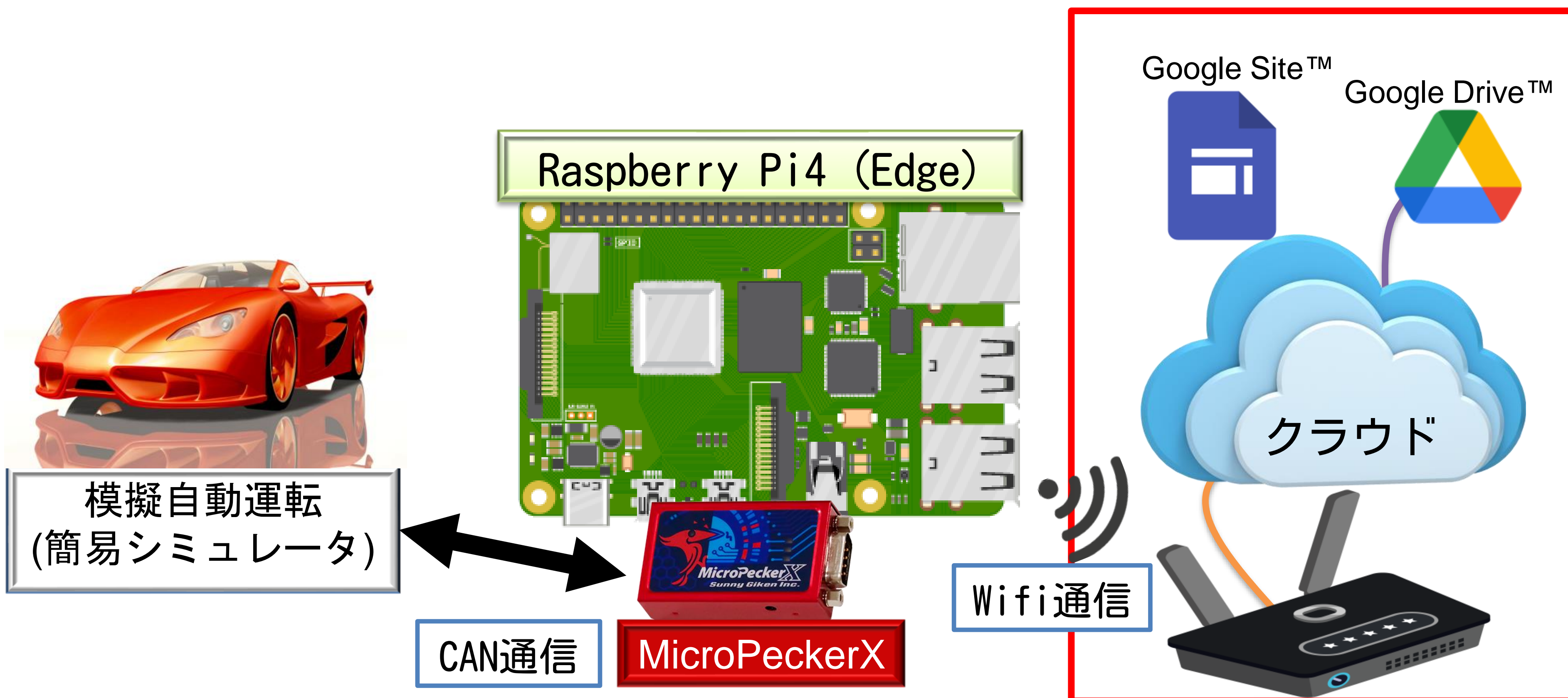

© 2022 Sunny Giken Inc. All rights reserved.

Sunny Giken Inc.

# 14. 実験シナリオと構成(クラウド)

Confidential
※複写および第三者への開示をお断りいたします。

- 異常なログの保持とクラウド側のユーザインタフェースを担当する
  - クラウドはGoogle Drive と Google サイトで構成する
  - クラウドに対する接続は、WiFiを介して行う

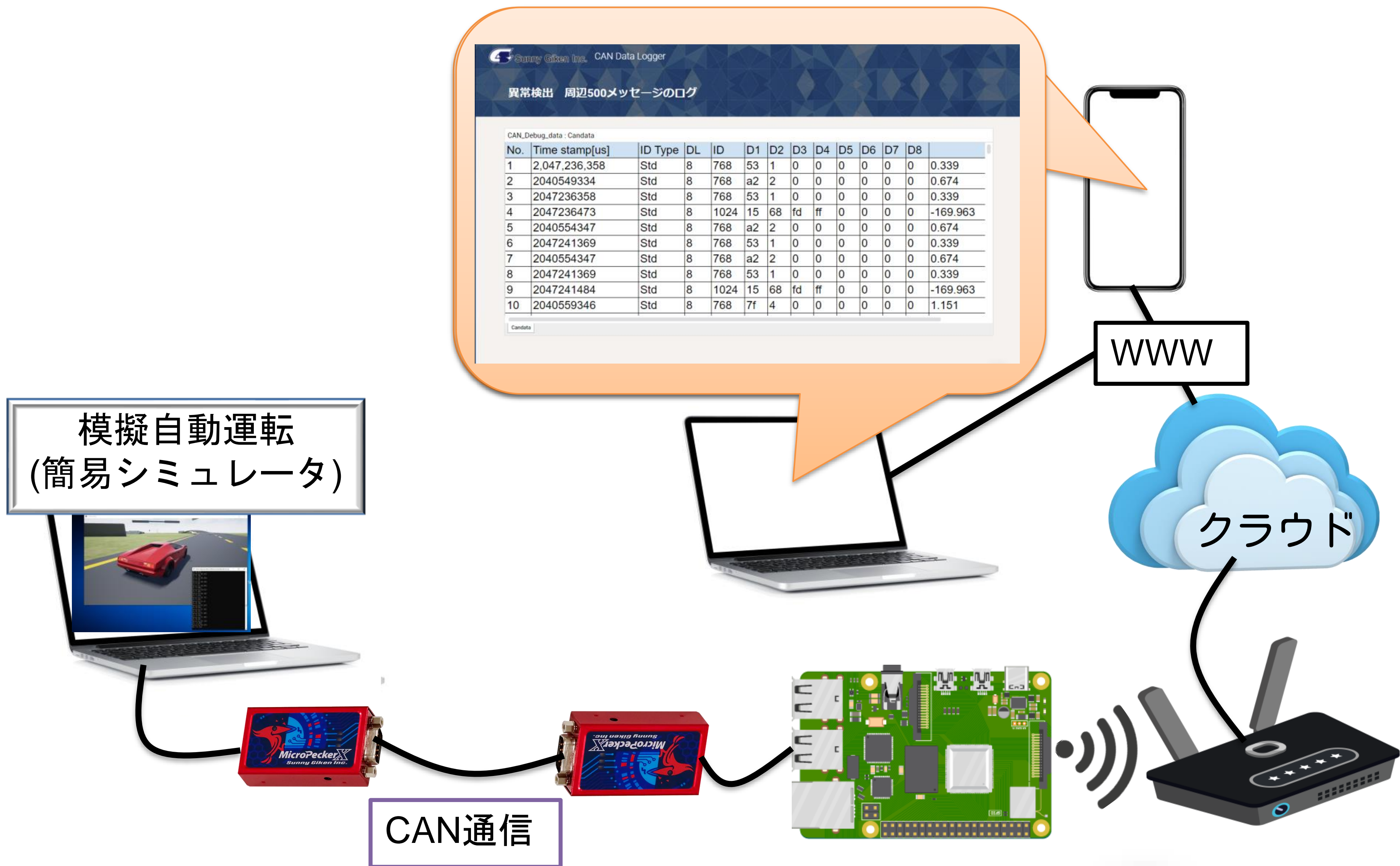



© 2022 Sunny Giken Inc. All rights reserved.

# 15. 実験デモ

Confidential
※複写および第三者への開示をお断りいたします。

- 今回の実験についてデモンストレーションを行います



Sunny Giken Inc. CAN Data Logger

異常検出　周辺500メッセージのログ

CAN_Debug_data : Candata

| No. | Time stamp[us] | ID Type | DL | ID | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2,047,236,358 | Std | 8 | 768 | 53 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0.339 |
| 2 | 2040549334 | Std | 8 | 768 | a2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0.674 |
| 3 | 2047236358 | Std | 8 | 768 | 53 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0.339 |
| 4 | 2047236473 | Std | 8 | 1024 | 15 | 68 | fd | ff | 0 | 0 | 0 | 0 | -169.963 |
| 5 | 2040554347 | Std | 8 | 768 | a2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0.674 |
| 6 | 2047241369 | Std | 8 | 768 | 53 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0.339 |
| 7 | 2040554347 | Std | 8 | 768 | a2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0.674 |
| 8 | 2047241369 | Std | 8 | 768 | 53 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0.339 |
| 9 | 2047241484 | Std | 8 | 1024 | 15 | 68 | fd | ff | 0 | 0 | 0 | 0 | -169.963 |
| 10 | 2040559346 | Std | 8 | 768 | 7f | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 1.151 |

Candata



© 2022 Sunny Giken Inc. All rights reserved.

Confidential
※複写および第三者への開示をお断りいたします。

- 時間制約を守ることができた
  - CANバスに流れたメッセージの個数および時間間隔と等しいタイムスタンプのログを記録している。
- 専用ハードウェアに時間制約に関わる処理を任せ、その他の処理のみRaspberry Piで実現できた
  - CANメッセージの受信タイミングは、タイムスタンプに含まれる。送信側もMicroPeckerXに任せることで正確な時間間隔を実現。
- リアルタイムシステムを構築するのと比べ短期間で実現できた
  - 慣れていない状況で1週間程度。慣れるとさらに効率が上がる見込み。

Raspberry Piと専用ハードウェアの組み合わせでPoC開発に利用できる性能が発揮できた。

© 2022 Sunny Giken Inc. All rights reserved.

# 17. 気づき(一覧)

Confidential
※複写および第三者への開示をお断りいたします。

- 気付き
  - ストレージの品質
  - アライン制約
  - リアルタイム系とオープン系でデータ型の共用はしない
  - 処理は自分で作らない
  - OSSライセンスの精査は十分に
  - 必要情報の精査を確実に
  - などなど、、、

© 2022 Sunny Giken Inc. All rights reserved.

# 18. 気づき(Raspberry Pi利用時の注意点)

Confidential
※複写および第三者への開示をお断りいたします。

- ストレージの品質
  - SDカードは、保証された書き込み可能な回数が少ない。
    - 不用意な選択を行うと、数か月という短期間で破損した。
  - 対策
    - 産業グレードを選ぶこと。監視カメラ向けなどで、高耐久を謳うものもあり、これらを選ぶのも手。
    - 必ず、SDを丸ごとバックアップする。
    - コスト、大きさが許す場合は、SSDを使う方法もある。現行のRaspberry Pi4では、ファームウェアが対応済み。

© 2022 Sunny Giken Inc. All rights reserved.

# 19. 気づき(リアルタイム系とオープン系の接続)

Confidential
※複写および第三者への開示をお断りいたします。

- アライン
  - 組み込み系は、メモリ制約に対応するため、アライン制約無し(1byteアライン)が多い。Raspberry Piは、ARMプロセッサのため、厳密な4byteアライン。
  - 対策
    - アライン済みの領域を確保し、組み込みデータを全てコピーして処理する。
    - 全ての変数を4byte以上の型で記述する。
  - 注意
    - 構造体アクセスしてはいけない。パッキングして差し示す場所が正しかったとしても、例えば4の倍数でないアドレスにint型(32bit)でアクセスして例外が発生する。

© 2022 Sunny Giken Inc. All rights reserved.

Sunny Giken Inc.

# 20. 気づき(全体)

Confidential

※複写および第三者への開示をお断りいたします。

- リアルタイム系とオープン系でデータ型の共用はしない
  - リアルタイム系から渡されたデータをそのまま使用していないか？<br>リアルタイム系に渡すデータを構造体で渡していないか？
- 処理は自分で作らない
  - 大抵の処理は、OSSで存在する。特にメモリ制御系は、CPUの特性に合わせて効率化されている。基本的にGCCの最適化を信じてよい。
- OSSライセンスの精査は十分に
  - 思わぬところでソース開示制約に引っかかる可能性あり。
- 必要情報の精査を確実に
  - クラウドにアップロードする情報を精査すること。<br>思わぬコスト増の可能性がある。

© 2022 Sunny Giken Inc. All rights reserved.

# 21. 今から始める方へのおすすめ[1/2]

Confidential
※複写および第三者への開示をお断りいたします。

- Linux PCを用意すること
  - 同一ディストリビューションが一番良いが、無ければDebian系統のディストリビューションを用意するのが賢明(例：Ubuntu)
    - Raspberry Piのシステムは、一般的にDebian系のシステム。万が一起動しない場合も、Linux PCがあれば、SDカードを直接開き設定の見直しができる。
    - SDカードのパーティションをグラフィカルに直接扱うことができる。
      - SDカードをバックアップする際、Linux PCを用いてパーティション容量を削減してからバックアップするのがおすすめ。表示された容量の中でもばらつきがあり、削減せずにバックアップしてしまうと、同一容量の別SDカードに書き込めないことがある。

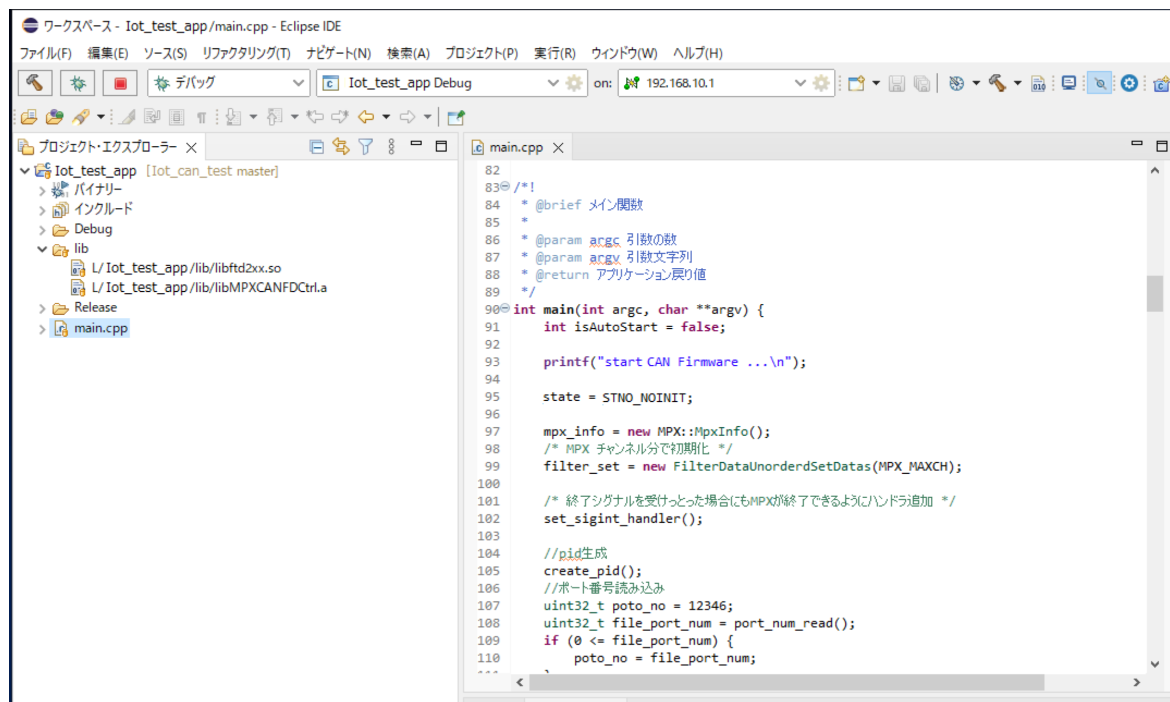
© 2022 Sunny Giken Inc. All rights reserved.

Sunny Giken Inc.

# 22. 今から始める方へのおすすめ[2/2]

Confidential
※複写および第三者への開示をお断りいたします。

- クロスコンパイラを用意すること(Linux以外のOSを常用の方)
  - 常用PCと比べるとさすがに非力。
  - GUI環境や強力な支援機能を持つ開発環境は、やはりWindowsが多い。
  - PoC開発では、知見が乏しい分野に踏み込みがち。新機能に対応するコーディングは覚えるのではなく、支援機能に任せる。



Confidential
※複写および第三者への開示をお断りいたします。

# ご清聴ありがとうございました

© 2022 Sunny Giken Inc. All rights reserved.