技術レポート　Technical Report

CAN バスによるマイコン間通信の構築と検証

Construction and Verification of Inter-Microcontroller Communication over a CAN Bus

レポート番号	MTIC-TR-2026-001
団体	NPO 法人松戸技術革新センター M-TIC　松戸ロボット研究会
著者	大城 公平
発行日	2026 年 10 月 25 日（A4 版・改訂）　初版（A3 ポスター）：2026 年 8 月 21 日
対象	機体内ネットワーク検証系（Raspberry Pi Pico 2 W ×2）
分類	技術レポート／通信・制御
機体状態の基準日	2026 年 8 月 21 日

本レポートの数値と機体状態について　本検証系は研究開発段階にあり随時変更されるため、記載の仕様は基準日時点のものであり、現在の実機と一致することを保証しない。数値には種別を付す：**公称**＝カタログ・設計値、**実測**＝実際に計測した値（記載桁が測定分解能）、**概算**＝目測・推定値、**計算値**＝実測・公称値からの計算。

概要

ロボットの機能が増えると、1 台のマイコンにすべてを担わせる構成は配線・処理の両面で限界を迎える。本検証では、複数のマイコンを少ない配線で結ぶ機体内ネットワークとして、自動車で実績のある CAN（Controller Area Network）バスを Raspberry Pi Pico 2 W 2 台の間に構築した。MCP2515 を直接制御する最小ドライバを MicroPython で実装し、500 kbps で送信ノードから 0.5 秒ごとにカウンタと速度指令を送った結果、受信ノードでカウンタの欠番なく受信できた。通電前に H-L 間の終端抵抗（60 Ω、実測）を確認して物理層を先に保証する手順は、不具合の切り分けを単純にするうえで有効と考えられる。送るデータはモーターの速度指令値（Hz）を想定した形式とし、受信側でその値を取り出して表示できることを確認した。さらに専用の受信プログラムで欠番を自動計数する 1 時間の連続受信試験を行い、7147 フレームを欠番 0・バスエラーなしで受信した。試験は机上の短い配線（130 mm）で、モーターを近くで回さずに行った。配線を長くした場合やモーターの近くを通した場合、多ノード化、実機での ID 設計は次の段階で確かめる。

1. 背景と目的

1.1 開発背景

ロボットの機能が増えるほど、1 台のマイコンにすべてを担わせる構成は配線・処理の両面で限界を迎える。複数のマイコンで役割を分担し、それらを少ない配線で確実に結ぶ「機体内ネットワーク」が必要になる。CAN バスは差動 2 線式で電位差により信号を伝えるためノイズに強く、モーターやインバータが近くにあるロボット内部に適する。同じ 2 本の線に複数のノードをぶら下げられ、自動車・産業機器で採用実績があり、エラー検出機構を備える。

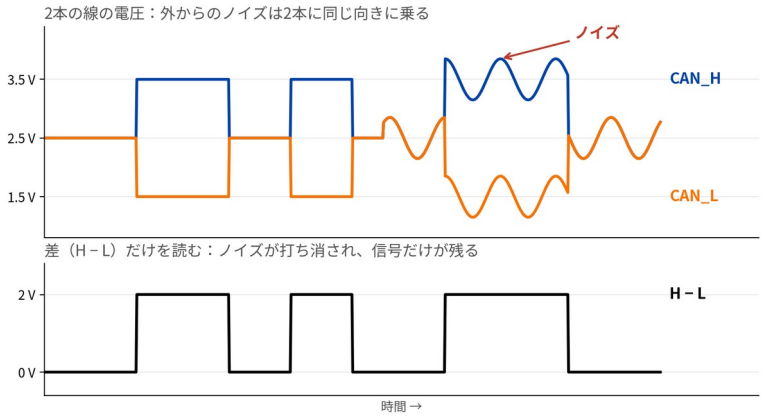


図 1　差動 2 線式がノイズに強い理由（概念図）

CAN バスは CAN_H と CAN_L の 2 本の線の電圧の差で 0 と 1 を伝える（図 1）。外からのノイズは 2 本の線に同じ向きに乗るため、差を取ると打ち消されて信号だけが残る。これがモーターやインバータの近くでも通信が乱れにくい理由である。

1.2 目的と目標

Raspberry Pi Pico 2 W 間に CAN バスを構築し、通信が成立することを確認する。送るデータはモーターの速度指令値（Hz）を想定した形式とし、受信側でその値を取り出せることを確認する。実際にモーターを駆動する構成は次の段階とする。

目標：500 kbps で 2 ノード間の通信が成立し、連続受信でフレームの取りこぼしがないこと（図 2）。

2. システム構成

2.1 ハードウェア

区分	仕様
マイコン	Raspberry Pi Pico 2 W ×2（MicroPython v1.28.0）
CAN トランシーバ基板	Waveshare Pico-CAN-B ×2（MCP2515 CAN コントローラ搭載。Pico に直接載せる拡張基板）
接続	SPI 0 番系統：SCK=GP6／MOSI=GP7／MISO=GP4／CS=GP5（基板内配線済み）
ビットレート	500 kbps（設定値）
終端抵抗	120 Ω × 両端（基板上ジャンパー ON、公称）
ノード間配線	H-H・L-L・G-G の 3 本、長さ 130 mm（実測・5 mm 単位）
送信データ	8 バイト中 3 バイトを使用：1 バイト目＝カウンタ（0〜255 で循環）、2〜3 バイト目＝速度指令［Hz］（例：0x0190＝400）。実受信データで確認

Copyright © 2026 Matsudo Technology Innovation Center. All rights reserved.

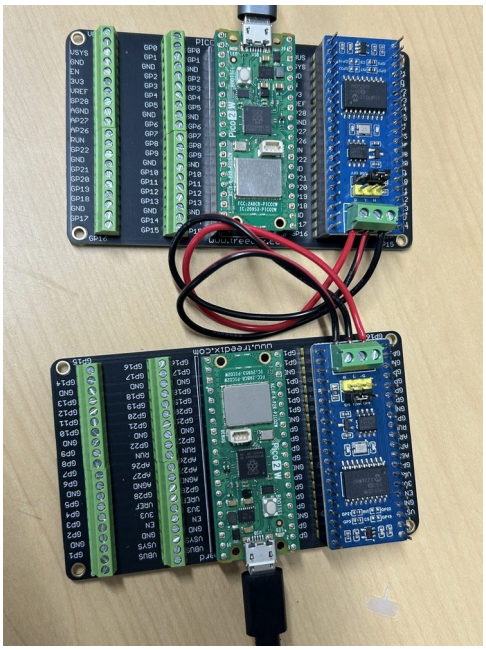


図 2　構築した 2 ノード CAN バス

2.2 配線と物理層の確認手順

ノード間の配線は H（CAN High）・L（CAN Low）・G（GND）の 3 本のみである。通電前にテスターで H-L 間の抵抗を測定し、120 Ω 終端の 2 並列に相当する 60 Ω（実測、テスター表示値）を確認してから電源を投入した。物理層を先に保証することで、以降の不具合の切り分けをソフトウェアに限定できる。配線・組立手順を図 3 に、終端ジャンパーの実装を図 4 に示す。

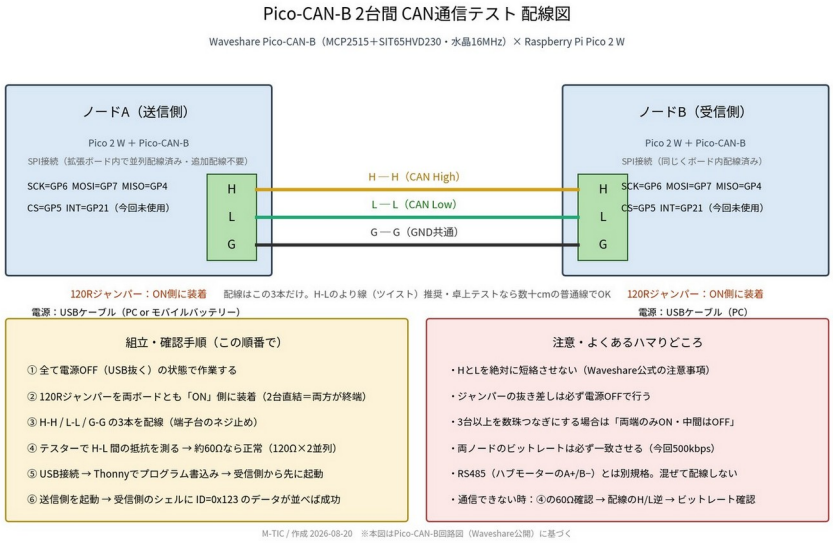


図 3　配線図・組立手順・注意点（筆者作成）

Copyright © 2026 Matsudo Technology Innovation Center. All rights reserved.

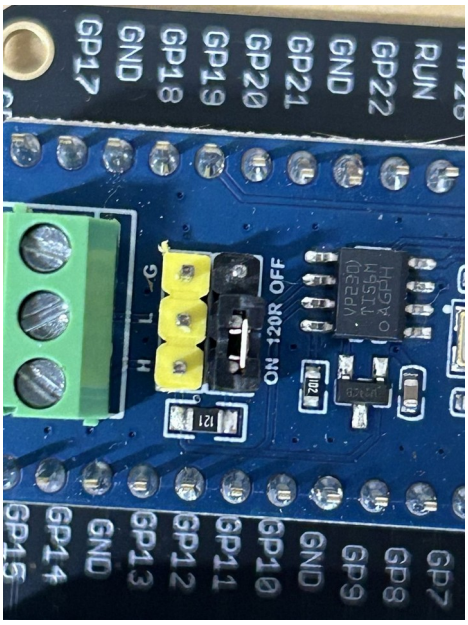


図 4　バス配線と終端ジャンパー（Pico-CAN-B 基板上）

2.3 ソフトウェア

- ドライバ：MCP2515 を直接制御する最小ドライバを MicroPython で実装（外部ライブラリ不使用）
- 送信ノード：0.5 秒ごとに ID=0x123 でカウンタ値と速度指令値［Hz］を送信。起動時に自動実行されるよう保存し、給電のみで動作
- 受信ノード：PC に接続し、受信フレームと取り出した速度指令値をシェルに表示（モーターは接続していない）

3. 検証方法と結果

3.1 検証方法

送信ノードを給電のみで起動し、受信ノードを PC に接続してシェル上で受信ログを取得した。カウンタ値の連番性で取りこぼしを判定した。屋内、ノード間配線長 130 mm（実測・5 mm 単位）。

長時間の安定性は、受信側に専用の試験プログラムを載せて評価した。試験プログラムは受信フレームのカウンタ（1 バイト目、0〜255 で循環）を追跡して循環を補正し、番号の飛びを欠番として自動で数える。あわせて MCP2515 のエラーフラグ（EFLG レジスタ）の最大値を記録する。計測時間は 3600 秒（1 時間）とした。試験のようすと結果の画面を図 5・図 6 に示す。

3.2 結果

```
ノード B 起動：受信待ち（500K bps）
受信 #44  ID=0x123  speed=400Hz
受信 #45  ID=0x123  speed=400Hz
受信 #46  ID=0x123  speed=400Hz
（以降連続受信・欠番なし）
```

図 5　受信ノードのログ（実測）

項目	結果
通信成立（500 kbps）	成功
フレームの取りこぼし	ゼロ（カウンタ欠番なし、実測）
送信ノードの自走	給電のみで動作

Copyright © 2026 Matsudo Technology Innovation Center. All rights reserved.

項目	結果
H-L 間の終端抵抗	60 Ω（実測、テスター表示値）
1 時間連続受信（3600 s）	受信 7147 フレーム／送信 7147 フレーム（カウンタ番号 0→7146 の差から）、欠番 0（実測）
エラーフラグ最大値	0x00（バスエラーなし、実測）
ID 違いのフレーム	0（実測）
平均送信周期	0.504 s（3600 s ÷ 7146 間隔、計算値）。設定 0.5 s より送信側の処理時間ぶん長いと推定
バス負荷率	0.03 %（計算値：1 フレーム約 75 ビット÷500 kbps＝150 µs を 0.504 s ごと）

```
[結果まとめ] 経過 3600 秒（60.0 分）
 受信フレーム数  : 7147
 最初/最後の番号 : 0 / 7146
 送信された数    : 7147（番号の差から計算）
 欠番（取りこぼし）: 0
 ID 違いのフレーム : 0
 エラーフラグ最大 : 0x00
```

図 6　1 時間連続受信試験の結果（受信ノードの出力、実測）

4. 課題と考察

- **多ノード化：**2 ノードでは問題が出なかったが、ノードを増やした場合の ID 競合・バス負荷・アービトレーションの挙動はまだ確かめていない。ID 設計（優先度の割り当て）を事前に決める必要がある。
- **実機環境：**本検証は机上の短距離配線で行った。モーター駆動系の近傍に配線した場合のノイズ耐性と、配線長を伸ばした場合の終端条件はまだ確かめていない。
- **エラー処理：**最小ドライバのため、エラー発生時の復帰処理は実装していない。1 時間の試験ではエラーが一度も発生しなかったが、ノイズの多い実機環境では復帰処理が必要になる可能性がある。

Copyright © 2026 Matsudo Technology Innovation Center. All rights reserved.

5. 今後の展望

5.1 検証計画

計測項目	方法	判定基準
受信した指令でモーターを動かす	受信ノードにモータードライバを接続し、受信した速度指令値でモーターを駆動する	指令値どおりの速度で回ること
3 ノード以上での通信	駆動・アーム・センサーのノードを追加し、ID 設計を決めて連続送受信	全ノードで欠番なし
配線長と安定性	同じ 1 時間受信テストを、配線長 130 mm → 1 m → 実機の配線長で繰り返す	各条件で欠番ゼロ・エラーフラグ 0x00
モーター近傍のノイズ耐性	CTR やインホイールのモーター駆動中に、配線をモーターの近くに通して同じテスト	欠番・エラーフラグを記録し、配線ルールを決める
より長時間の運転	同じ試験プログラムで 8 時間（1 日の稼働を想定）の連続受信	欠番ゼロ・エラーフラグ 0x00

5.2 応用の可能性

開発中の移動ロボットでは、駆動・アーム・センサーを別マイコンに分担させ、CAN バスで統合する構成を検討する。信号線はノード数によらず CAN_H・CAN_L の 2 本で済む（これに GND を加える）。このあとは、ノード数・配線長・モーターとの距離・運転時間を変えて同じ試験を繰り返し、どこまで安定に通信できるかを数字で押さえていく。

6. まとめ

Raspberry Pi Pico 2 W 間に CAN バスを構築し、500 kbps で取りこぼしのない通信を確認した。1 時間の連続受信試験では 7147 フレームを欠番 0・バスエラーなしで受信し、機体内ネットワークとして基本的な安定性があることを確かめた。終端抵抗の事前確認（H-L 間 60 Ω）により物理層を先に保証する手順は、切り分けを単純にするうえで有効と考えられる。速度指令値を想定したデータを送り、受信側で値を取り出せることも確認した。今後は多ノード化・モーター近傍でのノイズ耐性・配線長の影響を同じ試験方法で検証し、実機への搭載に進む。

参考資料

Waveshare, “Pico-CAN-B” 製品ドキュメント

Microchip, “MCP2515 Stand-Alone CAN Controller with SPI Interface” データシート

改訂履歴　初版（2026-08-21）A3 ポスターとして発行。　A4 版（2026-09-30）差動 2 線式の説明図、配線長の実測、送信データの形式、1 時間連続受信試験（7147 フレーム・欠番 0）の結果と検証計画を追加。　2026-10-25　文章を見直し、誤記を訂正。

Copyright © 2026 Matsudo Technology Innovation Center. All rights reserved.