

RaspberryPi アダプタユニット(S1UNT0001) ユーザーガイド

・概要

Raspberry Pi アダプタユニット (S1UNT0001) は、Raspberry Pi をマウントし、プラグインユニットとして使用するユニットです。



S1UNT0001 に Raspberry Pi を実装した様子

・仕様

- ・適合モデル
 - Raspberry Pi 2 Model B
 - Raspberry Pi 3 Model B
 - Raspberry Pi 3 Model B+
- ・CAN 通信ライン：1
- ・RS232C 通信ライン：1
- ・RTC バックアップ電池：CR1220
- ・電源入力：DC12V ~ 24V
- ・PCB 外形寸法：100mm × 160mm (基板厚：1.6mm)

・免責事項について

本書に記載されている使用方法と異なる使い方をした場合、または本書に記載されていない方法で使用した場合、その結果で生じる直接的、間接的な損害に対して、当社はいかなる責任も負いません。また利益の損失、物理的な損失、その他いっさいの費用について責任を負いません。

・製品の用途について

本製品は、電子回路システムの試作、実験、検証を想定して設計、製造されています。下記に示す人命、重大事故に関わる機器、装置には絶対に使用しないでください。

- ・医療用機器
- ・航空、宇宙関連装置
- ・有毒な液体、気体を扱う装置
- ・原子力設備などの放射能を取り扱う装置

・注意事項



本製品を取り扱う際には、十分に静電気対策を行ってください。



電源を入れた状態でのコネクタの抜き差しは、絶対に行わないでください。



電源を OFF する際には、Raspberry Pi のシステムがシャットダウンしていることを確認してください。
(電源断対策を行っている場合は除く)

1 回路構成の詳細

本製品を使用する前に、接続図、部品表で回路の詳細をご確認ください。接続図、部品表は製品情報ページからダウンロードできます。

2 機能の詳細

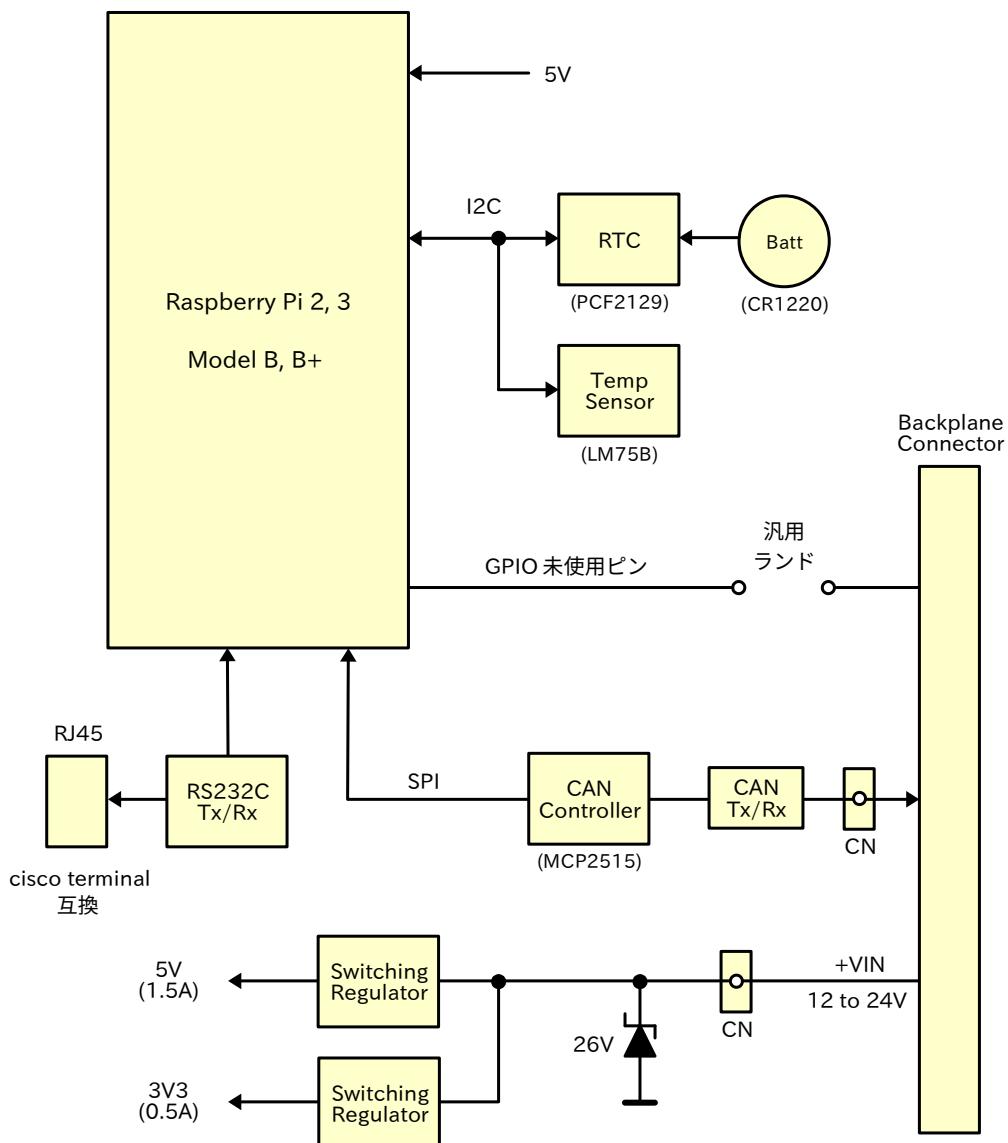


図 2.1 S1UNT0001 ブロック図

2.1 電源

電源入力 +VIN には DC12~24V を入力します。 通常はバックプレーンコネクタを介して入力しますが、バックプレーンを使用せず単体で動作させたい場合は、基板上の CN11 から入力することもできます。

入力した電源 +VIN から Raspberry Pi 用に +5V、周辺回路用に +3.3V を生成します。

電源ラインの途中には、R7, R8 (0Ω) を実装しています。この抵抗をシャント抵抗に交換することで簡易的に消費電流を測定できます。

Raspberry Pi 用の 5V レギュレータ(U8)の最大電流は 1.5A です。消費電流がこの範囲を超えないように注意してください。特に大きな電流を必要とする USB 機器は接続しないことを推奨します。

2.2 CAN コントローラ

CAN コントローラ (MCP2515) は SPI バスに接続されています。ユーザアプリケーションから使用するためには、まずデバイスとして認識させる必要があります。

2.3 温度センサ

温度センサ (LM75B) は、I2C バスに接続しています。LM75B は、I2C アドレスの下位 3 ビットをハード的に設定できるようになっています。本ユニットでは 3 ピンとも GND に接続していますので、アドレスは 0x48 になります。ユーザアプリケーションでアクセスする際には、このアドレスを指定してください。

2.4 リアルタイムクロック

リアルタイムクロック (PCF2129AT) は、I2C バスに接続しています。 (I2C アドレスは 0x51)
「5 Raspberry Pi の設定例」にはシステムデバイスとして認識する方法の例を記載しています。システムデバイスとして認識されると標準のコマンドで時刻の読み出し、設定などの操作ができるようになります。

BT1 電池ホルダにコイン電池 CR1220 を取り付けると、入力電源が OFF になっても、リアルタイムクロックの内部時計が動作を継続して時刻がバックアップされます。

この時刻は、次回起動時に読み出されてシステムの時刻としてセットされます。

2.5 シリアルポート

シリアル通信の送受信信号を RS232C レベルに変換して RJ45 コネクタに接続しています。シリアルポートは、ユーザアプリケーションで使用したり、シリアルログインのポートとして使用することができます。設定方法の例を「5 Raspberry Pi の設定例」に記載していますのでご覧ください。

フロントパネルからアクセスできる RJ45 コネクタのピンアサインは、cisco シリアルコンソール用の RJ45 ピンアサインと互換性を持たせていますので、市販の RS232C-RJ45 変換ケーブル、USB-RJ45 変換ケーブルが使用できます。

また、サブラック内でシリアル通信を行う機器と接続するために、基板上にコネクタを用意しています。RS232C 通信を行う機器と接続する場合は、CN8 を使用します。（この時 CN2 RJ45 は使用不可）3.3V LVTTL の信号でシリアル通信を行う場合は CN9 を使用します。（R5, R6 の 0Ω 抵抗を外す）

2.6 未使用的信号

未使用的 GPIO 信号は、ランド、コネクタなどに接続しています。詳しくは関連資料の接続図を参照願います。

3 ボード単体での動作

バックプレーンを使用せず、基板単体で動作させることができます。通常はバックプレーンコネクタを介して電源を入力しますが、基板上には電源入力用のコネクタ(CN11)が取り付けられていますので、外部から直接 DC 電源を供給することができます。

CAN 通信ラインのコネクタも取り付けられていますので、CAN 通信などを行うことができます。その他にも、コネクタ接続用のランドを設けていますので必要に応じてご利用ください。

4 RaspberryPi の取り付け、取り外し

4.1 取り付け

- (1) 取り付け作業の前に、フロントパネルを取り外します。
- (2) 付属のネジ、スペーサを Raspberry Pi に取り付けます。

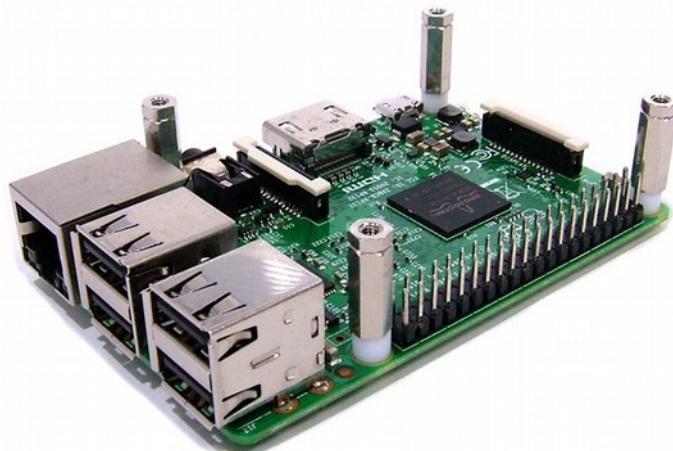
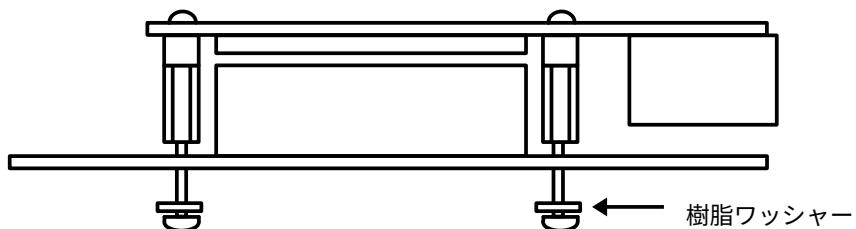


図 4.1 ネジ、スペーサの取り付け

- (3) 取付時に Raspberry Pi がたわまないように、露出しているコネクタピンをゴムシート、紙片などでカバーした固い材質のブロック材を当てながらゆっくりと嵌合させていきます。
まっすぐ挿入せず、左右に少しづつ入れていくと力を入れ過ぎないようにできます。

4.2 取り外し

- (1) 取り外し作業の前に、フロントパネルを取り外します。
- (2) Raspberry Pi を取り付けているネジ(M2.5, L5, バインドネジ)を取り外し、付属のネジ(M2.5, L10, バインドネジ)を樹脂ワッシャと共に Raspberry Pi に 3mm 程のスペースを空けて取り付けます。



- (3) 一度に押し上げようとせず、左右のネジの高さを調整しながら左右を少しづつ（3mm 程度）下に押しつけていきます。



無理に力をかけると、コネクタが破損する可能性があります。

5 Raspberry Pi の設定例

5.1 設定環境

下記の環境を例に説明します。

- ・使用モジュール：Raspberry Pi Model 3B+
- ・OS：Rasbian Buster Lite
- ・初期設定はシリアル接続のコンソールから行う。
- ・RTC バックアップ用の電池 CR1220 を取り付ける。

電源、シリアルコンソールケーブル、マイクロ SD カードなどは、必要に応じて準備してください。

5.2 マイクロ SD の準備

以下の手順でマイクロ SD を準備します。

- (1) OS イメージをダウンロードして、マイクロ SD カードに書き込む。
- (2) マイクロ SD を Windows や Linux などの計算機にマウントする。
- (3) /boot/config.txt の末尾に以下の行を追加する。

```
dtoverlay=mcp2515-can0,oscillator=16000000,interrupt=25
dtoverlay=spi-bcm2835

dtoverlay=i2c-rtc,pcf2129

dtoverlay=pi3-disable-bt
```

- ・MCP2515 には 16MHz のクリスタルが接続されています。
- ・MCP2515 の割り込みラインは、GPIO25 に接続されています。
- ・RTC IC には PCF2129AT が接続されています。

- (4) 設定したマイクロ SD カードを Raspberry Pi に取り付ける。

5.3 システムの基本設定

以下の手順でシステムを起動します。

- (1) シリアルコンソールケーブル (115200bps, 8bit, パリティ無し) 及び LAN ケーブルを接続する。
- (2) 電源を投入する。
- (3) ログインプロンプトが表示されたら、ID:pi, Password:raspberry でログインする。
(ログイン直後にパスワードを変更することをお勧めします。)
- (4) raspi-config でシステムの設定を行う。

```
$ sudo raspi-config
```

SPI, I2C インターフェースを有効にする。

- Interfacing Options : SPI を on
- Interfacing Options : I2C を on

以下の項目は必要に応じて設定する。

- Interfacing Options : SSH サーバー機能
- Localisation Options : Locale (例 : ja_JP.UTF-8)
- Localisation Options : Timezone (例 : Asia-Tokyo)
- Advanced Options : Expand Filesystem (最大サイズにする)

- (5) 必要に応じて、インストールパッケージの更新を行う。操作の前にインターネットにアクセスできる環境であることを確認する。

```
$ sudo apt update --allow-releaseinfo-change      (注)  
$ sudo apt upgrade
```

注：2019-10 時点では、リモートとローカルで package suite が違うために上記のオプションが必要になっています。

5.4 RTC の設定

LAN に接続しない状態で起動した場合に、システム時刻が RTC で保持している時刻に設定されるようにシステムを設定します。

- (1) 以下のコマンドで fake-hwclock サービスを停止する。

```
$ sudo update-rc.d -f fake-hwclock remove  
$ sudo systemctl disable fake-hwclock
```

- (2) root 権限で /etc/default/hwclock を開き編集する。

変更前 10 行目 #HWCLOCKACCESS=yes
変更後 10 行目 HWCLOCKACCESS=yes

変更前 18 行目 #HCTOSYS_DEVICE=rtc0
変更後 18 行目 HCTOSYS_DEVICE=rtc0

- (3) root 権限で /lib/udev/hwclock-set を開き 7~9 行目をコメントアウトする。

変更前
7 if [-e /run/systemd/system] ; then
8 exit 0
9 fi

変更後
7 # if [-e /run/systemd/system] ; then
8 # exit 0
9 # fi

5.5 簡易チェック

5.5.1 リアルタイムクロック

以下の手順で確認します。

- (1) システムをシャットダウンし、LAN ケーブルを外す。
- (2) 5 ~ 10 分後に電源を投入する。
- (3) システム起動後、時刻を確認し現在時刻と数秒の誤差で一致していることを確認する。

5.5.2 CAN コントローラ

以下のコマンドでエラーが発生しないことを確認します。

```
$ sudo ip link set can0 up type can bitrate 1000000
```

5.5.3 温度センサ

製品情報ページから温度センサにアクセスするサンプル python ソフトウェアをダウンロードできます。
別紙の「SUG-0009 温度センサ サンプルソフトウェア ユーザーガイド」を参照して動作の確認を行ってください。

改訂履歴

日付 バージョン	変更内容
2020-01-27 ver 1.0	初版発行