

PicoJig, PicoJig-WL マニュアル

目次

1 必ずお読み下さい.....	3
2 概要.....	4
2.1 PicoJig-WL の概要.....	4
2.2 PicoJig の概要.....	4
3 内容物.....	5
3.1 ファームウェア(FW).....	5
3.2 PC アプリ.....	5
4 セットアップ.....	6
4.1 Pi Pico または Pi Pico W に FW を書き込む.....	6
4.2 PC 側のセットアップ.....	6
5 LED.....	7
5.1 PicoJig の FW が書き込まれた Pi Pico の LED 点灯内容.....	7
5.2 PicoJig-WL が書き込まれた Pi Pico W の LED 点灯内容.....	7
6 メイン画面と起動.....	8
6.1 メイン画面.....	8
6.2 USB モードでの起動.....	9
6.3 Wi-Fi モードでの起動.....	10
6.4 FW エラーの確認.....	11
6.5 Flash メモリ内の設定データの消去.....	11
7 無線 LAN 設定.....	12
7.1 無線 LAN 設定画面.....	12
8 GPIO.....	14
8.1 GPIO で使用しているピン.....	14
8.2 GPIO 画面.....	15
9 ADC.....	16
9.1 ADC で使用しているピン.....	16
9.2 ADC 画面.....	16
10 UART.....	17
10.1 UART で使用しているピン.....	17
10.2 UART 画面.....	17
11 SPI.....	19
11.1 SPI で使用しているピン.....	19
11.2 SPI の注意事項.....	19
11.3 SPI 画面.....	21
12 I2C.....	23
12.1 I2C で使用しているピン.....	23
12.2 I2C の注意事項.....	23
12.3 I2C 画面.....	23
13 PWM.....	25
13.1 PWM で使用しているピン.....	25

13.2 PWM 画面.....	25
14 PicoJigApp のプロジェクトとライブラリ.....	26

1 必ずお読み下さい

※PicoJig や PicoJig-WL を使用する場合、必ず塩町ソフトウェアのウェブサイトの利用規約を確認して下さい。

＜利用規約の URL＞

<https://sites.google.com/view/shiomachisoft/%E5%88%A9%E7%94%A8%E8%A6%8F%E7%B4%84>

なお、PicoJig/PicoJigWL を使用したり本書の内容を行ったりして発生したいかなるトラブル・損失・損害についても塩町ソフトウェア(PicoJig/PicoJig-WL の作成者)は一切責任を負いません。

2 概要

本書は、PicoJigとPicoJig-WLのマニュアルです。
PicoJigとPicoJig-WLの概要は以下の通りです。

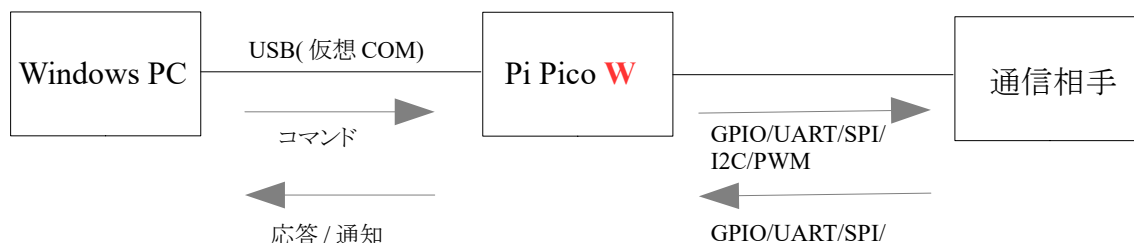
2.1 PicoJig-WLの概要

PicoJig-WLは、USB(仮想COM)またはWi-Fi(TCPソケット通信)経由でRaspberry Pi Pico **W**のGPIO/UART/SPI/I2C/ADC/PWMを制御するファームウェアとPCアプリです。
USBモードとWi-Fiモードの2種類があります。

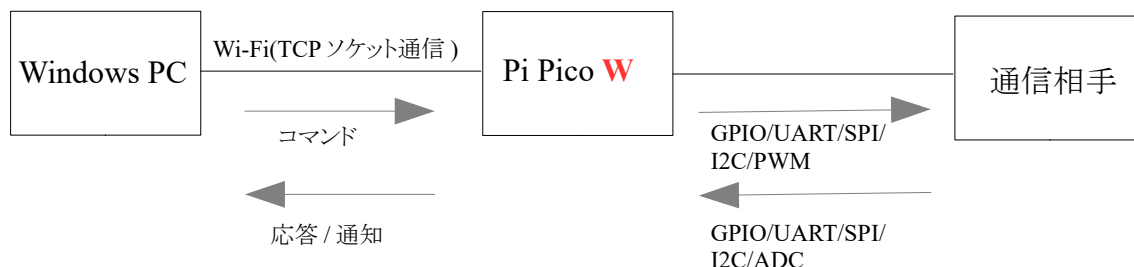
- ・マイコン基板はRaspberry Pi Pico **W**を使用します。
- ・Wi-Fiモードの場合、Pi Pico WはTCPサーバーになります。PCはTCPクライアントになります。
- ・Wi-Fiモードでは、2.4GHz帯を使用するWi-Fi規格「IEEE 802.11b/g/n」をサポートする無線LANルーターが必要です。
- ・Pi Pico WのSPI, I2Cはマスタです。

<システム構成>

■USBモード



■Wi-Fiモード



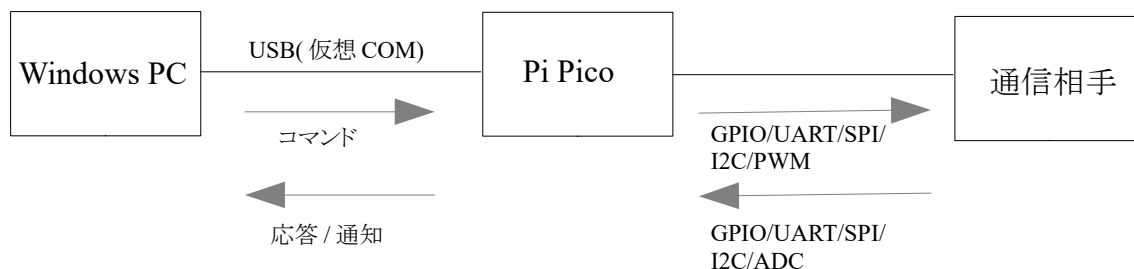
2.2 PicoJigの概要

PicoJigは、USB(仮想COM)経由でRaspberry Pi PicoのGPIO/UART/SPI/I2C/ADC/PWMを制御するファームウェアとPCアプリです。

- ・マイコン基板はRaspberry Pi Picoを使用します。
- ・Pi PicoのSPI, I2Cはマスタです。

<システム構成>

■USBモード



3 内容物

3.1 ファームウェア(FW)

(1) PicoJig_XXXXXXXXX.uf2

※XXXXXXXXXはバージョン日付になります。
PicoJig 用の FW であり、Pi Pico に書き込みます。

(2) PicoJig_WL_XXXXXXXXX.uf2

※XXXXXXXXXはバージョン日付になります。
PicoJig-WL 用の FW であり、Pi Pico W に書き込みます。

3.2 PC アプリ

(1) PicoJigApp_XXXXXフォルダ

※XXXXはバージョンになります。
このフォルダには、PicoJigApp(Windows PC 上で実行するアプリ)のバイナリが含まれます。
PicoJigApp は、PicoJig/PicoJig-WL 兼用です。
PicoJigApp_XXXXXフォルダは、PC の適当な場所(デスクトップなど)にフォルダごとコピーして下さい。

Windows について、.NET Framework 4.6.2 以上の.NET Framework 4.x.x が有効になっている必要があります。
.NET 5 以上とは互換性はありません。
※.NET Framework の有効化は自己責任です。

(2) JigApp.zip

PicoJigApp のソース一式です。
PicoJigApp は Visual Studio2022 で作成されています。

PicoJig/PicoJig-WL の FW が書き込まれた Pi Pico/Pi Pico W とコマンドをやりとりするライブラリの cs ファイルは、以下です。

- JigCmd_Common.cs
- JigCmd_Recv.cs
- JigCmd_Send.cs
- JigCmd_Virtual.cs
- JigSerial.cs
- JigTcpClient.cs

※ライブラリの説明はまだ記載できていません。

4 セットアップ

4.1 Pi Pico または Pi Pico W に FW を書き込む

以下は、Pi Pico または Pi Pico W に FW を書き込む手順です。

<注意>

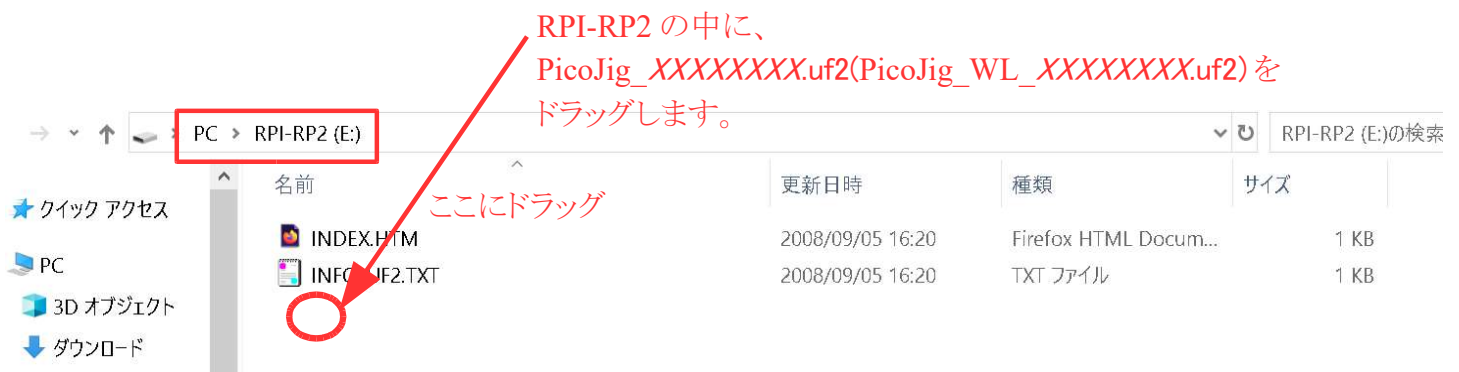
※PicoJigを使用する場合は、Pi Pico に PicoJig_XXXXXXXX.uf2 を書き込みます。

※PicoJig-WLを使用する場合は、Pi Pico W に PicoJig_WL_XXXXXXXX.uf2 を書き込みます。

- (1) Pi Pico(Pi Pico W)の白いボタンを押しながら PC と Pi Pico(Pi Pico W)を USB ケーブルで接続します。
すると、RPI-RP2 のドライブが認識されます。



- (2) RPI-RP2 の中に PicoJig_XXXXXXXX.uf2(PicoJig_WL_XXXXXXXX.uf2)をドラッグします。



以上で、FW の書き込みは終了です。

なお、Pi Pico(Pi Pico W)の電源が ON したタイミングで FW は起動します。

4.2 PC 側のセットアップ

- (1) PicoJigApp_XXXXXフォルダ PC の適当な場所(デスクトップなど)にフォルダごとコピーして下さい。
なお、PicoJigApp は、PicoJig/PicoJig-WL 兼用です。

Windows について、.NET Framework 4.6.2 以上の .NET Framework 4.x.x が有効になっている必要があります。
.NET 5 以上とは互換性はありません。

※.NET Framework の有効化は自己責任です。

5 LED

5.1 PicoJig のFWが書き込まれた Pi Pico のLED 点灯内容

- FW がエラーを検出していない場合、LED は 500ms 間隔で点滅します。
- FW がエラーを検出している場合、LED は 100ms 間隔で点滅します。

5.2 PicoJig-WL が書き込まれた Pi Pico W のLED 点灯内容

- FW がエラーを検出しておらず、かつ、無線 LAN ルーターと接続できていない場合、LED は 500ms 間隔で点滅します。
- FW がエラーを検出おらず、かつ、無線 LAN ルーターと接続できている場合、LED は点灯します。
- FW がエラーを検出している場合、LED は 100ms 間隔で点滅します。

6 メイン画面と起動

6.1 メイン画面

PicoJigApp - Monitor stopped:Not connected.

connect

[1] ☒ USB Mode

COM Port:
COM8 [2] v

[3] ☐ Wi-Fi Mode
(PicoW Only)

IP address of the destination server:
192.168.10.100 [4]

[6] disconnected [5] connect

APP/FW Information

APP Name: PicoJigApp [7]
APP Version: 2.0.0.0
FW Name: ---
FW Version: ---
Unique Board ID: ---

FW Error:

[10]

[8] NW Config

[9] Erase setting data in flash memory

[11] clear

6.2 USB モードでの起動

※USB モードは、PicoJig と PicoJig-WL の両方で使用できます。

- (1) Pi Pico を USB ケーブルで接続してから 10 秒程度待った後に PicoJigApp_XXXXX フォルダの中の PicoJigApp.exe をダブルクリックします。

※10 秒程度待つのは、Windows が Pi Pico の仮想 COM を認識するのに時間がかかるためです。

PicoJigApp.exe をダブルクリックすると<メイン画面>の章のメイン画面が表示されます。

- (2) <メイン画面>の[1]を ON にして USB モードを選択します。

- (3) <メイン画面>の[2]で Pi Pico の COM 番号を選択した後に、[5]のボタンを押します。

<メイン画面>の[6]の表示が”connected”に変わっていれば USB モードで Pi Pico と接続できています。

エラーのメッセージボックスが出る場合は、以下の事を試して下さい。

- ・[2]のリストに COM 番号が複数ある場合、[2]の COM 番号の選択を変更してから[5]を押す。
- ・Pi Pico の USB ケーブル接続を確認し、10 秒待ってから、PicoJigApp.exe を再起動する。

<メイン画面>の[6]の表示が”connected”に変わると、<メイン画面>の[8][9][11]のボタンが有効になります。
また、[7]に FW 情報が表示されます。

6.3 Wi-Fi モードでの起動

※Wi-Fi モードは、PicoJig-WL のみ使用できます。

(1) 最初に、＜USB モードでの起動＞の章の手順します。

※Pi Pico W の Flash メモリに無線 LAN 設定を保存するために、最初は USB モードで起動する必要があります。

(2) ＜メイン画面＞の[8]の中の[NW Config]ボタンを押して＜NW Config 画面＞を表示して無線 LAN 設定を行います。

※無線 LAN 設定は、Pi Pico W の Flash メモリに保存されるため毎回行う必要はありません。

※無線 LAN 設定の操作方法については、＜NW Config 画面＞の章を参照して下さい。

(3) Pi Pico W の LED が点滅ではなく点灯になっていることを確認します。

(Pi Pico W が無線 LAN ルーターと接続できていることを確認します。)

無線 LAN 設定が終わっていると、Pi Pico W は無線 LAN ルーターへの接続を試みます。

無線 LAN ルーターと接続できた場合、LED が点滅ではなく点灯になります。

※LED が点滅のままで点灯にならない場合、以下を行って下さい。

- ・電波干渉が起きるような機器が Pi Pico W の近くにないか確認して下さい。
- ・無線 LAN 設定に間違いがないか確認して下さい。

(4) Pi Pico W の仮想 COM をクローズします。

↓

＜メイン画面＞の[5]のボタンの表示が disconnect であることを確認し、[5]のボタンを押します。

そして、＜メイン画面＞の[6]の表示が”disconnected”になるのを確認します。

以下の(5)～(8)で Pi Pico W と TCP 接続します。

(5) Pi Pico W の LED が点滅ではなく点灯(=無線 LAN ルーターと接続済み)であることを確認します。

(6) ＜メイン画面＞の[3]を押して Wi-Fi モードを選択します。

(7) ＜メイン画面＞の[4]に TCP 接続したい Pi Pico W の IP アドレスを指定します。

※なお、PC と Pi Pico W の IP アドレスのネットワーク部は同じである必要があります。

(8) Pi Pico W と TCP 接続するために、[5]のボタンの表示が connect であることを確認してから[5]のボタンを押します。

[6]の表示が connected になれば、Pi Pico W と TCP 接続は成功しています。

(Pi Pico W と Wi-Fi モードで接続できています。)

＜メイン画面＞の[6]の表示が”connected”に変わると、＜メイン画面＞の[8][9][11]のボタンが有効になります。

また、[7]に FW 情報が表示されます。

6.4 FW エラーの確認

FW が認識したエラーは<メイン画面>の[10]に表示されます。

FW が認識したエラーをクリアしたい場合は、<メイン画面>の[11]を押します。

FW が認識するエラーは例として以下のようなものがあります。

<例>

- WDT タイムアウトでマイコンがリセットした。
- UART:Framing error
- UART:Parity error
- UART:Break error
- UART:Overrun error
- I2C:address not acknowledged, or, no device present.
- I2C通信でタイムアウト
- バッファに空きがないので要求データを破棄した(USB送信)
- バッファに空きがないので要求データを破棄した(UART送信)
- バッファに空きがないので要求データを破棄した(UART受信)
- バッファに空きがないので要求データを破棄した(I2C送信/受信)

6.5 Flash メモリ内の設定データの消去

以下の設定データは、Pi Pico(Pi Pico W)の Flash メモリ内の後方に保存されます。

- 無線 LAN 設定
- GPIO 設定
- UART 設定
- SPI 設定
- I2C 設定

※PicoJig(PicoJig-WL)をもう使用しない場合は、<メイン画面>の[9]のボタンで Flash メモリ内の後方に保存されている設定データを消去することをお勧めします。

7 無線 LAN 設定

7.1 無線 LAN 設定画面

無線 LAN 設定画面は、＜メイン画面＞の[8]の中の[NW Config]ボタンを押すと表示されます。

(1) [1]のボックスに国コードを入力します。

＜例＞

日本:JP

アメリカ:US

(2) [2]のボックスに Pi Pico W の IP アドレスを幾つにしたいかを入力します。

＜例＞

Pi Pico W の IP アドレスを 192.168.10.100 にしたい場合:

192.168.10.100

(3) [3]のボックスに無線 LAN ルーターの SSID を入力します。

※指定できる無線 LAN ルーターの SSID の条件:

- ・2.4GHz 帯を使用する Wi-Fi 規格「IEEE 802.11b/g/n」に対応していること。
間違えて 5GHz の周波数帯の SSID を指定しないように気を付けて下さい。
- ・暗号化方式は WPA2(AES)であること。

(4) [4]のボックスに無線 LAN ルーターのパスワードを入力します。

(5) [5]のボタンを押すと、設定データは Pi Pico W の Flash メモリ内の後方に保存されます。

(無線 LAN 設定が行われます。)

※PicoJig-WL をもう使用しない場合は、＜メイン画面＞の[9]のボタンで Flash メモリ内の後方に保存されている設定データを消去することをお勧めします。

(6) Pi Pico W の LED が点滅ではなく点灯になっていることを確認します。(Pi Pico W が無線 LAN ルーターと接続で

きていることを確認します。)

無線 LAN 設定が終わっていると、Pi Pico W は無線 LAN ルーターへの接続を試みます。

無線 LAN ルーターと接続できた場合、LED が点滅ではなく点灯になります。

※LED が点滅のままで点灯にならない場合、以下を行って下さい。

- ・電波干渉が起きるような機器が Pi Pico W の近くにはないか確認して下さい。
- ・無線 LAN 設定に間違いがないか確認して下さい。

8 GPIO

8.1 GPIO で使用しているピン

GPIO で使用しているピンは以下です。

GPIO 入力:

- GP3=5 番ピン
- GP4=6 番ピン
- GP5=7 番ピン
- GP8=11 番ピン
- GP9=12 番ピン
- GP10=14 番ピン
- GP11=15 番ピン

GPIO 出力:

- GP12=16 番ピン
- GP13=17 番ピン
- GP14=19 番ピン
- GP15=20 番ピン
- GP20=26 番ピン
- GP21=27 番ピン
- GP22=29 番ピン

8.2 GPIO 画面

GPIO 画面は、＜メイン画面＞の[8]の中の[GPIO]ボタンを押すと表示されます。

GPIO - COM8 - Monitoring

Input GPIO monitoring [1]
GP3 GP4 GP5 GP8 GP9 GP10 GP11
High High High High High High High

Output GPIO Monitoring [2]
GP12 GP13 GP14 GP15 GP20 GP21 GP22
Low Low Low Low Low Low Low

Changing the output GPIO value
Select High or Low and click "Change output value" [3]
GP12 GP13 GP14 GP15 GP20 GP21 GP22
Low Low Low Low Low Low Low
Change output value [4]

setting

Input GPIO - Pull-Up/Down [5]
GP3 GP4 GP5 GP8 GP9 GP10 GP11
Up Up Up Up Up Up Up

Output GPIO - initial value when power on [6]
GP12 GP13 GP14 GP15 GP20 GP21 GP22
Low Low Low Low Low Low Low
setting change [7]

(1) [1]に、現在の GPIO 入力の High/Low が表示されます。

(2) [2]に、現在の GPIO 出力の High/Low が表示されます。

(3) 以下の手順で GPIO 出力の High/Low を変更できます。

- ①[3]の中の GP12～GP22 の High/Low を選択します。
- ②[4]のボタンを押します。

(4) 以下の手順で GPIO の設定を変更できます。

- ①[5]で入力 GPIO の Pull-Up/Pull-Down を選択します。
- ②[6]で出力 GPIO の電源 ON 時初期値を選択します。
- ③[7]のボタンを押します。

「7」のボタンを押すと、設定データは Pi Pico の Flash メモリ内の後方に保存されます。

※PicoJig をもう使用しない場合は、＜メイン画面＞の[9]のボタンで Flash メモリ内の後方に保存されている設定データを消去することをお勧めします。

9 ADC

9.1 ADC で使用しているピン

ADC で使用しているピンは以下です。

- ・ADC0=GP26=31 番ピン
- ・ADC1=GP27=32 番ピン
- ・ADC2=GP28=34 番ピン

ADC4 は温度センサ

9.2 ADC 画面

ADC 画面は、<メイン画面>の[8]の中の[ADC]ボタンを押すと表示されます。



The screenshot shows a window titled "ADC - COM8 - Monitoring" with standard window controls (minimize, maximize, close). Inside the window, there are four rows of data, each with a label and a numerical value in a text box:

Label	Value
ADC0:[V]	0.652
ADC1:[V]	0.633
ADC2:[V]	0.619
ADC4(temperature sensor):[degrees Celsius]	27.607

ADC0～ADC2 の電圧値[V]と ADC4 の温度センサ値「℃」が表示されます。

10 UART

10.1 UART で使用しているピン

UART で使用しているピンは以下です。

- ・UART0 TX=GP0=1 番ピン
- ・UART0 RX=GP1=2 番ピン

10.2 UART 画面

UART 画面は、<メイン画面>の[8]の中の[UART]ボタンを押すと表示されます。

(1) 以下の手順で UART の設定を変更できます。

- ①[1]でボーレートを選択します。
 - ②[2]でストップビットを選択します。
 - ③[3]でパリティを選択します。
- ※データビットは 8 固定です。
- ④[4]のボタンを押します。

[4]のボタンを押すと、設定データは Pi Pico の Flash メモリ内の後方に保存されます。

※PicoJig をもう使用しない場合は、<メイン画面>の[9]のボタンで Flash メモリ内の後方に保存されている設定データを消去することをお勧めします。

(2) 以下の手順で UART 送信できます。

①[5]に送信データを 16 進数 2 桁、セパレータ=スペースまたはカンマで入力します。

※送信データサイズは 1～256byte にして下さい。

②[6]を押します。

(3) [7]に送受信データのログが表示されます。

(4) [8]を押すと送受信データのログがクリアされます。

11 SPI

11.1 SPI で使用しているピン

SPI で使用しているピンは以下です。

- SPI0 RX=GP16=21 番ピン
- SPI0 CSn=GP17=22 番ピン
- SPI0 SCK=GP18=24 番ピン
- SPI0 TX=GP19=25 番ピン

11.2 SPI の注意事項

(1) PicoJig は SPI マスタになります。

(2) CS について:

- ① PicoJig が SPI のクロックを送信している間は CS は Low です。それ以外(アイドル時)は CS は High です。
- ② SPI のクロックを送信する 5us 前から CS は Low になります。
- ③ SPI のクロック送信完了後に 5us 経過してから CS は High になります。
- ④ CS は、SPI CSn の機能を使わずに GPIO で制御しています。

理由:

Raspberry Pi Pico のマイコンの RP2040 を SPI マスタにして SPI CSn の機能を使う場合、モード 0 とモード 2 においては、下図の通り、バイトごとに CS を Low/High するという一般的でない仕様であり、そのような CS の動作は避けたいためです。

このため、PicoJig では、SPI クロック送信中に CS は Low を維持するようにするために、CS については SPI CSn の機能を使わずに GPIO で制御し、上記の①～③のように出力しています。

Figure 91. Motorola
SPI frame format,
single transfer, with
SPO=0 and SPH=0

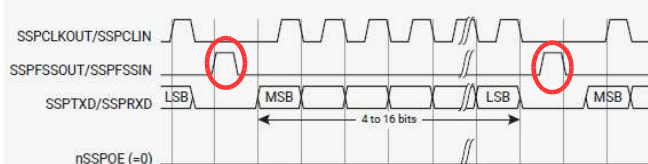
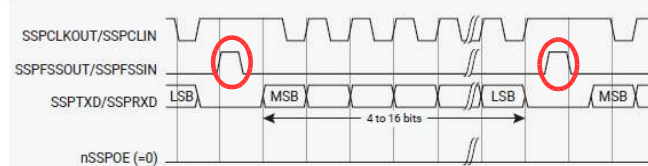


Figure 94. Motorola
SPI frame format,
continuous transfer,
with SPO=1 and
SPH=0



⑤ PicoJig の SPI の通信相手の SPI スレーブが Raspberry Pi Pico のケース:

(i) このケースでは、モード 1 かモード 3 を使用して下さい。

理由:

Raspberry Pi Pico のマイコンの RP2040 が SPI スレーブの場合、モード 0 とモード 2 においては、マスタ側が④の下線部のような一般的でない CS 出力を行う必要があるためです。

(ii) このケースで塩町ソフトウェア(PicoJig の作成者)が動作確認した時は、

SPI スレーブ側の Raspberry Pi Pico は、DREQ を使用して SPI と DMA を連携させました。

(スレーブ側は、pico-sdk の spi_write_read_blocking()のような CPU を使う関数は使用しませんでした。)

これは、スレーブ側がマスタの SPI クロックに対して処理遅れが発生しないか心配だったためです。

- ⑥ SPI 関連のピンに接続するリード線が長いとデータ化けすることがあったので、
SPI 関連のピンに接続するリード線は短い方がよいかもしれません。(原因未調査)

11.3 SPI 画面

SPI 画面は、<メイン画面>の[8]の中の[SPI]ボタンを押すと表示されます。

(1) 以下の手順で SPI の設定を変更できます。

- ①[1]で周波数(Hz)を入力します。
- ②[2]で SPI のモードを選択します。
※データビット=8 固定、バイトオーダー=MSB ファースト固定です。
- ③[3]のボタンを押します。

[3]のボタンを押すと、設定データは Pi Pico の Flash メモリ内の後方に保存されます。

※PicoJig をもう使用しない場合は、<メイン画面>の[9]のボタンで Flash メモリ内の後方に保存されている設定データを消去することをお勧めします。

(2) 以下の手順で SPI 送信できます。

- ①[4]に送信データを 16 進数 2 桁、セパレータ=スペースまたはカンマで入力します。
※送信データサイズは 1～256byte にして下さい。

②[5]を押します。

マスタの SPI 送信なので、送信と同時に受信します。

(3) [6]に送受信データのログが表示されます。

(4) [7]を押すと送受信データのログがクリアされます。

12 I2C

12.1 I2C で使用しているピン

I2C で使用しているピンは以下です。

- I2C1 SDA=GP6=9 番ピン
- I2C1 SCL=GP7=10 番ピン

12.2 I2C の注意事項

(1) PicoJig は I2C マスタになります。

12.3 I2C 画面

I2C 画面は、<メイン画面>の[8]の中の[I2C]ボタンを押すと表示されます。

The screenshot shows the 'I2C - COM8' window with the following elements:

- Communication setting**
 - frequency(Hz): 100000 [1] 100000 or more [2] setting change
- 7bit slave address:** 17 [3] Hex(00-7F)
- send**
 - send data:Hex(00-FF) separator:space or comma e.g. 00,01,FE,FF
 - Input field: 1 2 FE FF [4]
 - send size:1-256 [5] send
- receive**
 - receive size 4 [6] 1-256 [7] receive
- send/receive log**
 - Log entries: [S]01 02 FE FF [8], [R]01 02 FE FF
 - [9] clear

(1) 以下の手順で I2C の設定を変更できます。

- ①[1]で周波数(Hz)を入力します。
- ②[2]のボタンを押します。

[2]のボタンを押すと、設定データは Pi Pico の Flash メモリ内の後方に保存されます。

※PicoJig をもう使用しない場合は、<メイン画面>の[9]のボタンで Flash メモリ内の後方に保存されている設定データを消去することをお勧めします。

(2) 以下の手順で I2C 送信できます。

- ①[3]に 7bit スレーブアドレス(16 進数)を入力します。
- ②[4]に送信データを 16 進数 2 桁、セパレータ=スペースまたはカンマで入力します。
※送信データサイズは 1～256byte にして下さい。
- ③[5]を押します。

(3) 以下の手順で I2C 受信できます。

- ①[3]に 7bit スレーブアドレス(16 進数)を入力します。
- ②[6]に受信データサイズを入力をします。
※受信データサイズは 1～256byte にして下さい。
- ③[7]を押します。

(4) [8]に送受信データのログが表示されます。

(5) [9]を押すと送受信データのログがクリアされます。

13 PWM

13.1 PWM で使用しているピン

PWM で使用しているピンは以下です。

- ・GP2=4 番ピン

13.2 PWM 画面

PWM 画面は、<メイン画面>の[8]の中の[PWM]ボタンを押すと表示されます。

PWM - COM8

PWM frequency = 125MHz / ((wrap+1) * clkdiv)
duty ratio = high_period / (wrap + 1)

e.g. PWM frequency = 100Hz, duty ratio = 50%

clkdiv:
250.000000 [1] 1-259.999999

wrap:
4999 [2] 0-65535

high_period:
2500 [3] 0-65535

[4] PWM start [5] PWM stop

(1) 以下の手順で PWM 出力できます。

①[1]にクロック分周器を入力します。

②[2]に分解能を入力します。

③[3]に High の期間を入力します。

PWM 周波数 = 125MHz ÷ ((分解能+1) × クロック分周)

デューティー比 = High の期間 ÷ (分解能+1)

④[4]を押します。

(2) [5]を押すと、PWM 出力を中止します。

14 PicoJigApp のプロジェクトとライブラリ

(1) PicoJigApp は Visual Studio2022 で作成されています。

(2) PicoJig/PicoJig-WL の FW が書き込まれた Pi Pico/Pi Pico W とコマンドをやりとりするライブラリの cs ファイルは、
以下です。

- JigCmd_Common.cs

- JigCmd_Recv.cs

- JigCmd_Send.cs

- JigCmd_Virtual.cs

- JigSerial.cs

- JigTcpClient.cs

※ライブラリの説明はまだ記載できていません。