PICO-HDMI-PLUS-R1 (Raspberry Pi Pico/Pico W/Pico 2 用拡張ボード) 取扱説明書

マイクロファン

http://www.microfan.jp/ https://store.shopping.yahoo.co.jp/microfan/ https://www.amazon.co.jp/s?me=A28NHPRKJDC95B

> 2024 年 11 月 Copyright ⓒ 2024 MicroFan, All Rights Reserved.

目次

第1章	PICO-HDMI-PLUS-R1 の紹介					1
1.1	製品概要					1
1.2	購入・利用上の注意........................					2
1.3	マニュアルの記載内容に関して					2
第 2章	利用の準備					3
2.1	部品表					3
2.2	はんだ付けの準備.......................				•	3
	2.2.1 半田ごての状態の管理				•	3
	2.2.2 実装時のヒント				•	4
2.3	端子等のはんだ付け.......................				•	4
	2.3.1 ピンヘッダーの切り分け					4
	2.3.2 PICO へのピンヘッダーのはんだ付け					4
	2.3.3 PICO-HDMI-PLUS-R1 へのピンヘッダーのはんだ付け	۲				5
	2.3.4 PICO-HDMI-PLUS-R1 と PICO のはんだ付け					6
第3章	Arduino スケッチ環境の整備					8
第3章 3.1	Arduino スケッチ環境の整備 Raspberry Pi Pico/Pico W/Pico 2 用 Arduino 開発環境のイン	/ストー	ル.			8 8
第 3 章 3.1	Arduino スケッチ環境の整備 Raspberry Pi Pico/Pico W/Pico 2 用 Arduino 開発環境のイン 3.1.1 基本となる Arduino IDE のインストール	/ストー	ル.	 	•	8 8 8
第 3 章 3.1	Arduino スケッチ環境の整備 Raspberry Pi Pico/Pico W/Pico 2 用 Arduino 開発環境のイン 3.1.1 基本となる Arduino IDE のインストール 3.1.2 Raspberry Pi Pico/Pico W/Pico 2 用の開発機能の追加	バストー ・・・・・・ (1) ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	ル. 	 	•	8 8 8 8
第 3 章 3.1 3.2	Arduino スケッチ環境の整備 Raspberry Pi Pico/Pico W/Pico 2 用 Arduino 開発環境のイン 3.1.1 基本となる Arduino IDE のインストール 3.1.2 Raspberry Pi Pico/Pico W/Pico 2 用の開発機能の追加 Raspberry Pi Pico/Pico W/Pico 2 用 Arduino 開発環境の設定	イストー ・・・・・ 口 ・・・ 三 ・・・	ル。 ・・ ・・	· · · · · · · · · · · · · · · · · · ·		8 8 8 8 9
第3章 3.1 3.2 3.3	Arduino スケッチ環境の整備 Raspberry Pi Pico/Pico W/Pico 2 用 Arduino 開発環境のイン 3.1.1 基本となる Arduino IDE のインストール 3.1.2 Raspberry Pi Pico/Pico W/Pico 2 用の開発機能の追加 Raspberry Pi Pico/Pico W/Pico 2 用 Arduino 開発環境の設定 サンプルスケッチの実行	×ストー ・・・・・ ロ ・・・ ミ ・・・・	ル . ・・ ・・ ・・	· · · · · ·		8 8 8 9 9
第3章 3.1 3.2 3.3	Arduino スケッチ環境の整備Raspberry Pi Pico/Pico W/Pico 2 用 Arduino 開発環境のイン3.1.1 基本となる Arduino IDE のインストール3.1.2 Raspberry Pi Pico/Pico W/Pico 2 用の開発機能の追加Raspberry Pi Pico/Pico W/Pico 2 用 Arduino 開発環境の設定サンプルスケッチの実行3.3.1 BLINK:LED の単純な点滅	 ストー 	ル . ・・ ・・ ・・	· · · · · · · · · · · · · · · · · · ·		8 8 8 9 9 9
第3章 3.1 3.2 3.3 3.4	Arduino スケッチ環境の整備Raspberry Pi Pico/Pico W/Pico 2 用 Arduino 開発環境のイン3.1.1 基本となる Arduino IDE のインストール3.1.2 Raspberry Pi Pico/Pico W/Pico 2 用の開発機能の追加Raspberry Pi Pico/Pico W/Pico 2 用 Arduino 開発環境の設定サンプルスケッチの実行3.3.1 BLINK:LED の単純な点滅デジタルテレビ等への出力	バストー ・・・・・ 回・・・ 三・・・・ ・・・・・・・・・・・・・・・・・・・・・	ル。 ・・ ・・ ・・ ・・	· · · · · · · · · · · · · · · · · · ·		8 8 8 9 9 9 9
第3章 3.1 3.2 3.3 3.4	Arduino スケッチ環境の整備Raspberry Pi Pico/Pico W/Pico 2 用 Arduino 開発環境のイン3.1.1 基本となる Arduino IDE のインストール3.1.2 Raspberry Pi Pico/Pico W/Pico 2 用の開発機能の追加Raspberry Pi Pico/Pico W/Pico 2 用 Arduino 開発環境の設定サンプルスケッチの実行3.3.1 BLINK:LED の単純な点滅デジタルテレビ等への出力3.4.1 PicoDVI ライブラリのインストール	イストー 	ル . 	· · · · · · · · · · · ·	· · ·	8 8 8 9 9 9 9 9 9
第3章 3.1 3.2 3.3 3.4	Arduino スケッチ環境の整備Raspberry Pi Pico/Pico W/Pico 2 用 Arduino 開発環境のイン3.1.1 基本となる Arduino IDE のインストール3.1.2 Raspberry Pi Pico/Pico W/Pico 2 用の開発機能の追加Raspberry Pi Pico/Pico W/Pico 2 用 Arduino 開発環境の設定サンプルスケッチの実行3.3.1 BLINK:LED の単純な点滅デジタルテレビ等への出力3.4.1 PicoDVI ライブラリのインストール3.4.2 PicoDVI ライブラリの利用	 ストー ロ · · · · Ξ · · · · · · · · · · 	ル . 	· · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	8 8 8 9 9 9 9 9 9 9 9 9
第3章 3.1 3.2 3.3 3.4	Arduino スケッチ環境の整備Raspberry Pi Pico/Pico W/Pico 2 用 Arduino 開発環境のイン3.1.1 基本となる Arduino IDE のインストール3.1.2 Raspberry Pi Pico/Pico W/Pico 2 用の開発機能の追加Raspberry Pi Pico/Pico W/Pico 2 用 Arduino 開発環境の設定サンプルスケッチの実行3.3.1 BLINK:LED の単純な点滅ボジタルテレビ等への出力3.4.1 PicoDVI ライブラリのインストール3.4.3 PICO-HDMI-PLUS-R1 の機能の利用	✓ストー	ル. 	· · · · · · · · · · · · · ·	• • • • • • •	8 8 9 9 9 9 9 9 9 9 10 10
第3章 3.1 3.2 3.3 3.4 第4章	Arduino スケッチ環境の整備Raspberry Pi Pico/Pico W/Pico 2 用 Arduino 開発環境のイン3.1.1 基本となる Arduino IDE のインストール3.1.2 Raspberry Pi Pico/Pico W/Pico 2 用の開発機能の追加Raspberry Pi Pico/Pico W/Pico 2 用 Arduino 開発環境の設置サンプルスケッチの実行サンプルスケッチの実行3.3.1 BLINK:LED の単純な点滅デジタルテレビ等への出力3.4.1 PicoDVI ライブラリのインストール3.4.2 PicoDVI ライブラリの利用3.4.3 PICO-HDMI-PLUS-R1 の機能の利用CircuitPython プログラム環境の整備	✓ストー	ル. 	· · · · · · · · · · · ·	• • • • • • •	8 8 9 9 9 9 9 9 9 9 9 9 9 10 10
第3章 3.1 3.2 3.3 3.4 第4章 4.1	Arduino スケッチ環境の整備Raspberry Pi Pico/Pico W/Pico 2 用 Arduino 開発環境のイン3.1.1 基本となる Arduino IDE のインストール3.1.2 Raspberry Pi Pico/Pico W/Pico 2 用の開発機能の追加Raspberry Pi Pico/Pico W/Pico 2 用 Arduino 開発環境の設定サンプルスケッチの実行3.3.1 BLINK:LED の単純な点滅デジタルテレビ等への出力3.4.1 PicoDVI ライブラリのインストール3.4.2 PicoDVI ライブラリの利用3.4.3 PICO-HDMI-PLUS-R1 の機能の利用CircuitPython プログラム環境の整備CircuitPython ファームウエアの書き込み	 ストー ロ · · · · Ξ · · · · · · · · · · · · · · · · · · · · · · · · · · <li (·="" td="" ·="" ·<<=""><td>ル . </td><td>· · · · · · · · · · · · · ·</td><td></td><td>8 8 9 9 9 9 9 9 9 9 9 10 10 10 13</td>	ル . 	· · · · · · · · · · · · · ·		8 8 9 9 9 9 9 9 9 9 9 10 10 10 13

	4.1.2 ファームウェアの PICO への書き込み	14
	4.1.3 Python ライブラリの PICO への書き込み	14
	4.1.4 プログラムによるデジタルテレビ出力	14
第5章	資料	16
5.1	PICO-HDMI-PLUS-R1 の回路図	16
5.2	基板上の入出力....................................	17
第6章	購入および問い合わせ先	19
6.1	ご協力のお願い..................................	19
6.2	販売:ネットショップ	19
6.3	製品情報	19
6.4	問い合わせ先....................................	19
6.5	所在地	20

表目次

2.1	部品表	3
5.1	部品表	17
5.2	スイッチと圧電スピーカー...............................	18



2.1	PICO へのピンヘッダーのはんだ付け 裏面	5
2.2	PICO へのピンヘッダーのはんだ付け 表面	5
2.3	PICO-HDMI-PLUS-R1 へのピンヘッダーのはんだ付け 裏面	6
2.4	PICO-HDMI-PLUS-R1 へのピンヘッダーのはんだ付け 表面	6
2.5	PICO-HDMI-PLUS-R1 と PICO のはんだ付け 斜め上.........	6
2.6	PICO-HDMI-PLUS-R1 と PICO のはんだ付け 密着	7
2.7	PICO-HDMI-PLUS-R1 と PICO W のはんだ付け 側面	7
2.8	PICO-HDMI-PLUS-R1 と PICO W のはんだ付け HDMI コネクタ側	7
3.1	DVIGFX8 のコンストラクタ	10
3.2	スイッチと圧電スピーカーの利用	11
4.1	CircuitPython によるデジタルテレビ出力	15
5.1	PICO-HDMI-PLUS-R1 の回路図	16
59	DICO HDMI DI US D1 の郊日配署	17
0.2	LICO-IIDMI-LT02-U1 の助加的層・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	11

第1章

PICO-HDMI-PLUS-R1 の紹介



1.1 製品概要

Raspberry Pi Pico/Pico W/Pico 2 の高い性能と低コストが注目され、広く利用されるように なってきました。

PICO-HDMI-PLUS-R1 は、その Raspberry Pi Pico/Pico W/Pico 2 に HDMI コネクタを接続し、デジタルテレビなどに画像を出力できるようにする拡張ボードです。あなたの PICO を超小型のテレビゲーム機や、デジタルテレビ出力付きの Arduino/CircuitPython 実験基板に変身させることができます。

PICO-HDMI-PLUS-R1 を接続した Raspberry Pi Pico/Pico W/Pico 2 では、HDMI の画像 出力は [PicoDVI - picodvi] と呼ばれているライブラリを使用して行うことができます。PICO-HDMI-PLUS-R1 は Raspberry Pi Pico/Pico W/Pico 2 を使用し、デジタルテレビへの画像出 力なども利用した様々な応用に取り組みたい人に最適な開発ボードです。

PICO-HDMI-PLUS-R1 は以下のような特徴を持っています。

- HDMIのコネクタを装備しており、Raspberry Pi Pico/Pico W/Pico 2 に接続してデジタ ルテレビに画像を出力することができます。
- HDMI に画像出力するプログラムは、PicoDVI から派生したライブラリ等を使用し、 Arduino や CircuitPython(PICO W を除く) で作成することができます。
- 基板上にタクトスイッチを装備しており、ゲームや画面制御用の入力として利用できます。
- 基板上に圧電スピーカーを装備しており、音を出すことができます。
- GPIO0-7,22, ADC0-2 が自由に使用できるため、様々な入出力装置やセンサーを接続し、 多様な応用が可能です。

製品の利用法に関しては、本マニュアルだけでなく、以下のページ、サイトもご参照ください。 https://www.microfan.jp/2024/07/pico-hdmi-plus/

1.2 購入・利用上の注意

PICO-HDMI-PLUS-R1 をご購入の際には、下記項目をご確認ください。

- Raspberry Pi Pico/Pico W/Pico 2 は別売りです。
- ブレッドボードは別売りです。
 https://store.shopping.yahoo.co.jp/microfan/breadboard-63.html

1.3 マニュアルの記載内容に関して

Raspberry Pi Pico/Pico W/Pico 2 やそれに関連するハードウェアやソフトウェアは、機能の 追加や改良が頻繁に行われているため、本文書で提供している情報は、PICO-HDMI-PLUS-R1 の購入者の利用時にはすでに古い情報になっている可能性があります。そのため、本文書で示し ている内容と異なる部分があったり、本文書で示している手順ではうまく動作しないことがある ことと、その場合には、各自で対処方法を調査・確認していただく必要があることをご承知おきく ださい。

本マニュアルの記載内容と、ご提供するソフトウェア、ハードウェアに差異がある場合には、ご 指摘によりマニュアルの迅速な訂正を心がけますが、ご提供するソフトウェア、ハードウェアの現 品の仕様が優先されます。

お伝えする内容と本質的な問題がない場合には、本マニュアルには、旧バージョンの製品の写真 や他製品の写真などがそのまま使用されている場合がありますのでご承知おきください。

本書に記載されている内容に基づく作業、運用などにおいて、いかなる損害が生じても、弊社お よび著者をはじめとする本文書作成関連者は、一切の責任を負いません。

本文書に記載されている製品名などは、一般的にそれぞれの権利者の登録商標または商標です。

第2章

利用の準備

PICO-HDMI-PLUS-R1 の利用に先立って、PICO-HDMI-PLUS-R1 と Raspberry Pi Pico/Pico W/Pico 2 を接続するためのはんだ付けを行います。

2.1 部品表

PICO-HDMI-PLUS-R1 の部品表を表 2.1 に示します。表 2.1 には開発ボード以外の関連部品 を含めて示しています。基板が破損している場合には、ご利用になる前にマイクロファンにお問 い合わせください。

表 2.1 部品表

部品	シンボル	規格等	個数
プリント基板	PICO-HDMI-PLUS	Rev.1	1
ピンヘッダー	SV1, SV2	1x20 ピン	2 または相当品

2.2 はんだ付けの準備

2.2.1 半田ごての状態の管理

基板にピンヘッダをはんだ付けされる方のために、まず最初に、はんだ付けを行う際の一般的な 半田ごての状態の管理に関して示します。

はんだ付けを行う直前に、スポンジなどのこて先クリーナーで半田ごてをクリーニングしてフ ラックスや酸化膜などの汚れを取り除き、こて先が銀色に輝く状態ではんだ付けを行います。ま た、こて先にほとんどはんだが乗っておらず乾いていると、こて先から部品のピンや基板のパター ンなどに熱が伝わりにくいので、こて先に少し(薄く)はんだを付けてこて先がはんだで濡れた状 態にしてはんだ付けを行います。

2.2.2 実装時のヒント

ピンソケット、ピンヘッダーなどの複数の端子を持つ部品のはんだ付けは、端子の端の1ピン、 もしくは両端か対角上の2ピンをはんだ付けし、部品の取り付け姿勢などを必要に応じて修正し てから残りの端子をはんだ付けすると、部品の姿勢をきれいに整えて取り付けることができます。

PICO-HDMI-PLUS-R1 のプリント基板はベたアースになっており、熱容量が大きくなってお ります。このため、白色のシルク印刷で示されている GND 端子をはんだ付けする際には基板の 端子部分(ランド)の温度が上がりはんだが融けるまで少し時間がかかるため、他の端子と比較し て長めにはんだごてを当てておく必要がありますのでご注意ください。

2.3 端子等のはんだ付け

PICO-HDMI-PLUS-R1と PICO はピンヘッダーで接続して使用します。

PICO-HDMI-PLUS-R1 と PICO の接続部分の端子配置は、Raspberry Pi Pico/Pico W/Pico 2 すべてで共通しているので、以下のはんだ付けは、Raspberry Pi Pico/Pico W/Pico 2 の全ての共通事項となります。

- ピンヘッダーの切り分け
- 必要に応じて Raspberry Pi Pico/Pico W/Pico 2 の端子へのピンヘッダーのはんだ付け
- PICO-HDMI-PLUS-R1の Raspberry Pi Pico/Pico W/Pico 2 へのはんだ付け

2.3.1 ピンヘッダーの切り分け

PICO-HDMI-PLUS-R1 に付属するピンヘッダーを以下の様に切り分けます。

- 12 ピン 1個
- 8 ピン1個
- 10 ピン 2 個

2.3.2 PICO へのピンヘッダーのはんだ付け

PICO-HDMI-PLUS-R1 を PICO に取り付けると、PICO-HDMI-PLUS-R1 が PICO の一部 の端子を覆い隠すことになり、PICO の端子へのピンヘッダーの取り付けが難しくなります。し たがって、PICO をブレッドボード等で利用する場合には、切り分けた 12 ピンと 10 ピンのピン ヘッダーを先に PICO に取り付けてはんだ付けしてください。

ピンヘッダーをブレッドボードに挿せるように、PICO の基板の裏側に取り付けます。12 ピン のピンヘッダーを PICO の VBUS-GP22 に、10 ピンのピンヘッダーを PICO の GP0-GP7 に取 り付けます。ピンヘッダーは基板裏面からピンの短いほうをプリント基板に取り付け、プリント 基板の表面(おもてめん)ではんだ付けします。 ピンヘッダーをプリント基板にはんだ付けする際に、はんだごてが基板上の部品に接触しない ようにご注意ください。

ピンヘッダーを PICO にはんだ付けした様子を図 2.1, 2.2 に示します。



図 2.1 PICO へのピンヘッダーのはんだ付け 裏面



図 2.2 PICO へのピンヘッダーのはんだ付け 表面

2.3.3 PICO-HDMI-PLUS-R1 へのピンヘッダーのはんだ付け

ピンヘッダーを PICO-HDMI-PLUS-R1 の基板の裏側に取り付けます。10 ピンのピンヘッダー を SV1 に、8 ピンのピンヘッダーを SV2 に取り付けます。ピンヘッダーは基板裏面からピンの短 いほうをプリント基板に取り付け、プリント基板の表面(おもてめん)ではんだ付けします。

ピンヘッダーをプリント基板にはんだ付けする際に、はんだごてが基板上の部品に接触しない ようにご注意ください。

ピンヘッダーを PICO にはんだ付けした様子を図 2.3, 2.4 に示します。



図 2.3 PICO-HDMI-PLUS-R1 へのピンヘッダーのはんだ付け 裏面



図 2.4 PICO-HDMI-PLUS-R1 へのピンヘッダーのはんだ付け 表面

2.3.4 PICO-HDMI-PLUS-R1 と PICO のはんだ付け

PICO の上に PICO-HDMI-PLUS-R1 を載せて、すでに PICO-HDMI-PLUS-R1 にはんだ付けしているピンヘッダーの反対側のピンを PICO にはんだ付けします。PICO に PICO-HDMI-PLUS-R1 を搭載した様子を図 2.5 に示します。



図 2.5 PICO-HDMI-PLUS-R1 と PICO のはんだ付け 斜め上

無線機能のない PICO と PICO 2 に PICO-HDMI-PLUS-R1 を載せる場合には、図 2.5 に示

すように PICO-HDMI-PLUS-R1 を PICO の上にぴったりと重ねてピンヘッダーをはんだ付け してかまいません。はんだ付け後は、PICO の下に伸びたピンヘッダーの端子の不要部分は図 2.5 に示すように切り落としてください。



図 2.6 PICO-HDMI-PLUS-R1 と PICO のはんだ付け 密着

PICO W の場合は、PICO-HDMI-PLUS-R1 の圧電ブザーの端子が無線回路のシールド板に接触しないようにするためと、アンテナ部分の開口部を少しでも多くとるため、図 2.7, 2.8 に示すように PICO-HDMI-PLUS-R1 を浮かせて PICO にはんだ付けするようにしてください。



図 2.7 PICO-HDMI-PLUS-R1 と PICO W のはんだ付け 側面



図 2.8 PICO-HDMI-PLUS-R1 と PICO W のはんだ付け HDMI コネクタ側

第3章

Arduino スケッチ環境の整備

3.1 Raspberry Pi Pico/Pico W/Pico 2 用 Arduino 開発環境のイン ストール

Arduino の開発環境のインストールは以下の2段階の手順で行います。

- 基本となる Arduino IDE のインストール
- Raspberry Pi Pico/Pico W/Pico 2 用の開発機能の追加

この後は、必要に応じて、各種のライブラリの追加インストールを行います。

下記のインストール法がわかりにくい様であれば、WEB で検索をするとインストール法を示 したページが複数見つかるので、ご自身がわかりやすいと思うページを参照してインストールを 行ってください。

3.1.1 基本となる Arduino IDE のインストール

以下のページからダウンロードオプションで、ご自身が使用している OS 用のインストールパッ ケージを選択しダウンロードしインストールしてください。

• https://www.arduino.cc/en/software

Arduino IDE がインストールできたら起動してください。

メニュー等を日本語化するために、Arduino IDE の [File] \Rightarrow [Preferences...] \Rightarrow [Settings] タブの [Language:] を日本語に設定してください。

3.1.2 Raspberry Pi Pico/Pico W/Pico 2 用の開発機能の追加

Raspberry Pi Pico/Pico W/Pico 2 用の Arduino は以下の WEB ページで公開されています。

• https://github.com/earlephilhower/arduino-pico

インストール方法も示されているので、示されている手順に従って Raspberry Pi Pico/Pico

W/Pico 2 用の Arduino のインストールを行ってください。

3.2 Raspberry Pi Pico/Pico W/Pico 2 用 Arduino 開発環境の設定

BOOTSEL ボタンを押した状態で、PICO と PC を USB で接続してください。この操作で、 Raspberry Pi Pico/Pico W/Pico 2 がブートローダーモードに移行します。

なお、PICO に Arduino のスケッチをいったん書き込むと、以降のスケッチの書き込みでは、 ブートローダーモードに移行する操作をする必要はなくなります。

Arduino IDE のメニューの「ツール」を選択してメニューを表示してください。このメニュー の中から、まず、開発用のボードと PC と PICO の接続を行うポートを設定します。

開発ボードの選択は、[ボード:] メニューの [Raspberry Pi Pico/RP2040/R2350] の中から、初めの方に表示されている適切な選択肢を選択してください。

また、[ポート:] は、 [UF2 Board] という選択肢があるので、それを選択して設定してください。 MAC の場合には、ポートの選択方法は WEB などで確認してください。

3.3 サンプルスケッチの実行

PICO の動作確認のために、LED 点滅スケッチの実行を試してみましょう。

3.3.1 BLINK:LED の単純な点滅

電子工作界の hello world、LED の点滅スケッチを実行しましょう。

Arduino IDE の [ファイル] ⇒ [スケッチの例] ⇒ [01.Basics] から Blink を選択してください。 サンプルスケッチを開いたら、Arduino IDE の左上部の右矢印マーク [書き込み] のアイコンを クリックします。スケッチのコンパイルの後に、Arduino IDE の下部のメッセージエリアに白色 の文字で数行のメッセージが出て、スケッチの書き込みが行われます。

スケッチが PICO に正しく書き込まれたら、ボード上の LED が点滅します。

スケッチの書き込みが失敗するようであれば、ポートの設定がされているか確認してください。 いったんスケッチを書き込むと、以降はスケッチの書き込みの際にブートローダーモードに移 行する操作をする必要はなくなります。また、ポートの選択も、COMX(X は数値)の形式で表示 されたものを選択して設定します。

ブートローダーモードでの最初のスケッチの書き込みが終了したら、ポートの選択肢に [UF2 Board] がなくなり、代わりに COMX が表示されるようになるので、その COMX をポートに再 設定するようにしてください。

3.4 デジタルテレビ等への出力

3.4.1 PicoDVI ライブラリのインストール

PicoDVI ライブラリを使用してデジタルテレビに画像を出力する方法を示します。

PicoDVI ライブラリは、Arduino IDE のライブラリマネージャを利用してインストールするこ とができます。ライブラリマネージャの検索フィルタに [picodvi] を入力して絞り込むと、以下の ライブラリが表示されます。

• PicoDVI - Adafruit Fork

このドキュメントの作成時点では、ライブラリは PICO2 に対応した Ver.1.2.0 となっています。 ライブラリマネージャで、最新バージョンを選択してインストールボタンを押しライブラリを インストールします。関連するライブラリも合わせてインストールするか聞かれた場合には、そ れらもすべてインストールしてください。

このライブラリは、以下の URL で取得することもできます。

https://github.com/adafruit/PicoDVI

3.4.2 PicoDVI ライブラリの利用

ライブラリのインストール後、Arduino IDE メニューから [ファイル] ⇒ [スケッチの例] を選択 すると、リストの下の方に [PicoDVI - Adafruit Fork] フォルダが追加されているのが確認できま す。[PicoDVI - Adafruit Fork] フォルダの中を確認するといくつかのサンプルスケッチがあり、 選択して実行することができます。

例として [PicoDVI - Adafruit Fork] \Rightarrow [screensavers] \Rightarrow [aquarium] を選択します。このス ケッチは、昔流行したスクリーンセーバーと同様な表示を行うものです。

このサンプルスケッチの 11 行目付近に図 3.1 の上の行があるので、その最後の引数を下の行 のように pico_sock_cfg に書き換えてください。これの変更を行うことにより、デモスケッチを PICO-HDMI-PLUS-R1 用に変更できたので、コンパイルして実行できるようになります。

```
DVIGFX8 display(DVI_RES_320x240p60, true, adafruit_feather_dvi_cfg);
```

DVIGFX8 display(DVI_RES_320x240p60, true, pico_sock_cfg);

図 3.1 DVIGFX8 のコンストラクタ

他のデモスケッチも同様な書き換えを行うことにより、PICO-HDMI-PLUS-R1 で実行できる ようになります。

3.4.3 PICO-HDMI-PLUS-R1 の機能の利用

PICO-HDMI-PLUS-R1 には、スイッチや圧電スピーカーが搭載されているので、デジタルテ レビ出力に加えてそれらの使用法を確認するための簡単なスケッチを図 3.2 に示します。

```
#include <PicoDVT.h>
// display()の最後の引数を pico_sock_cfg に設定
DVIGFX16 display(DVI_RES_320x240p60, pico_sock_cfg);
void setup() {
  display.begin();
  // スイッチのプルアップ機能を有効化
  pinMode(9,OUTPUT) ;
  digitalWrite(9,HIGH) ;
  // 右側の2つのスイッチを有効化
  pinMode(21,INPUT) ;
 pinMode(20,INPUT) ;
}
void loop() {
  if (digitalRead(21)) // スイッチ SW4
    display.setTextColor(0xFFFF) ;
  else
    display.setTextColor(0xFF00); // 押されたら文字色が黄色に
  if (digitalRead(20)) // スイッチ SW3
   noTone(8) ;
  else
    tone(8, 1000); // 押されたら音が鳴る
  display.fillScreen(0);
  display.setFont();
  display.setCursor(0, 0);
  display.setTextSize(3);
  display.println(F("MicroFan"));
  display.setTextSize(2,4);
  display.println(F("PICO-HDMI-PLUS"));
  display.print(analogReadTemp()); // MCUの内部温度
  display.print(F("'C, "));
  display.print(millis()/1000);
  display.println(F(" sec."));
  delay(300);
  for (int y = 70; y < 240; y += 8)
    display.drawLine(0, 90, 319, y, 0x07E0) ;
  delay(300);
  display.fillCircle(80, 180, 50, 0xF800);
  display.drawCircle(80, 180, 50, 0xFF00);
  display.fillRect(200, 100, 100, 100, 0x001F);
  display.drawRect(200, 100, 100, 100, 0xFF00);
  delay(400);
}
```

このスケッチには以下のような機能が組み込まれています。

- SW4 を押すと、表示されている文字の色が黄色に変わります。
- SW3 を押すと、圧電スピーカーの音が鳴ります。

なお、このスケッチの loop() 関数の実行サイクルが1秒になっているので、スイッチの操作が 画面や音に反映されるには少し遅延が発生します。

第4章

CircuitPython プログラム環境の整備

PicoDVI は Python でプログラムすることもできますが、現時点で対応しているのは Circuit-Python で MicroPython は対応していません。ここでは、CircuitPython で PicoDVI を利用す る方法を示します。

4.1 CircuitPython ファームウエアの書き込み

CircuitPython のサイトを以下に示します。 https://circuitpython.org/

4.1.1 ファームウェアのダウンロード

CircuitPython のファームウェアは以下のページから対象の開発ボードを選択し、ダウンロードすることができます。

https://CircuitPython.org/download/

PICO-HDMI-PLUS-R1 でデジタルテレビ出力ができる PICO/PICO 2 のファームウェアは、 それぞれ以下のページからダウンロードできます。現在のところ、CircuitPython の PICO W は デジタルテレビ出力に対応していません。

- https://circuitpython.org/board/raspberry_pi_pico/
- https://circuitpython.org/board/raspberry_pi_pico2/

このドキュメントの記述時点では、それぞれファームウェアのバージョンは V9.2.0 となっています。

メッセージが日本語されたファームウェアも用意されているので、それをダウンロードして使用 するとよいでしょう。例えば、PICO 2 用のファームウェアの名前は以下のようになっています。

 $adafruit-circuit python-rasp berry_pi_pico2-ja-9.2.0.uf2$

また、下記のページから [Bundle for Version 9.x] という Python ライブラリの圧縮ファイルを ダウンロードしてください。

https://circuitpython.org/libraries

ダウンロードするファイルの名前は現時点では次のようなものです。 adafruit-circuitpython-bundle-9.x-mpy-**日付**.zip

4.1.2 ファームウェアの PICO への書き込み

ファームウェアをダウンロードしたら PICO に書き込みます。

BOOTSEL ボタンを押した状態で、PICO と PC を USB で接続してください。この操作で、 Raspberry Pi Pico/Pico W/Pico 2 がブートローダーモードに移行し、PC にマウントされた PICO のファイルシステムが表示されます。

このファイルシステムのルートに、先にダウンロードした CircuitPython のファームウェアを ドラッグアンドドロップします。ファームウェアのコピーが終わると、PICO へのファームウェ アの書き込みが実行され再起動するとともに、PC にマウントされていたファイルシステムはいっ たん解除されます。

PICO が CircuitPython のファームウェアで再起動すると、CircuitPython で読み書きできる PICO 上のファイルシステムが PC にマウントされ利用できるようになります。

4.1.3 Python ライブラリの PICO への書き込み

先にダウンロードした下記の Python ライブラリを解凍してください。

adafruit-circuitpython-bundle-9.x-mpy-日付.zip

解凍したライブラリフォルダの中にある lib フォルダを PICO のファイルシステムのルートに ドラッグアンドドロップしてコピーしてください。lib ディレクトリの中にはたくさんのライブラ リが収められているため、コピーには結構時間がかかります。

これで、CircuitPython の環境設定は終了です。

4.1.4 プログラムによるデジタルテレビ出力

PICO の CircuitPython プログラミングは、色々な端末ソフトや IDE を使用して行うことがで きます。Thonny を使用してプログラミングを行う場合には、以下のページを参考にして Thonny をインストールしてください。

https://www.microfan.jp/2023/04/thonny/

プログラミング環境が整備でき、PICO で CircuitPython のプログラムを実行できるように なったら、CircuitPython でデジタルテレビに文字と画像を出力するプログラムを入力し、実行 してみましょう。プログラムを図 4.1 に示します。

```
# 表示システムの初期化を行う
import displayio, picodvi, digitalio, board, framebufferio, simpleio
displayio.release_displays()
fb = picodvi.Framebuffer(
    width=320, height=240, color_depth=8,
    clk_dp=board.GP14, clk_dn=board.GP15, # 端子の配置は PICO-HDMI-PLUS 用
    red_dp=board.GP12, red_dn=board.GP13,
    green_dp=board.GP18, green_dn=board.GP19,
    blue_dp=board.GP16, blue_dn=board.GP17)
display = framebufferio.FramebufferDisplay(fb)
group = displayio.Group()
display.root_group = group
# 文字列の表示を行う
from adafruit_display_text import label
import terminalio
group.append(label.Label(terminalio.FONT, text="MicroFan",
                        x=0, y=10, scale=2, color=0xFFFFF))
group.append(label.Label(terminalio.FONT, text="RP2040-UNO-HDMI:PicoDVI",
                        x=0, y=30, scale=2, color=0xFFFFF))
# 図形の表示を行う
from adafruit_display_shapes.line import Line
from adafruit_display_shapes.rect import Rect
from adafruit_display_shapes.circle import Circle
from adafruit_display_shapes.triangle import Triangle
group.append(Line(0, 80, 320, 80, 0xFF00FF))
group.append(Rect(0, 120, 100, 100, fill=0xFF0000, outline=0xFFFF00))
group.append(Circle(160, 170, 50, fill=0x00FF00, outline=0xFFFF00))
group.append(Triangle(260, 120, 210, 220, 310, 220, fill=0x0000FF, outline=0xFFFF00))
# スイッチを利用できるように設定する
pullup = digitalio.DigitalInOut(board.GP9)
pullup.direction = digitalio.Direction.OUTPUT
pullup.value = True
sw4 = digitalio.DigitalInOut(board.GP21)
# sw4 が押されたら音を鳴らす
while True:
    if sw4.value == 0:
       simpleio.tone(board.GP8, 1000, 0.05)
```

第5章

資料

5.1 PICO-HDMI-PLUS-R1の回路図

PICO-HDMI-PLUS-R1の回路図を図 5.1、部品配置を図 5.2、部品表を表 5.1 に示します。



図 5.1 PICO-HDMI-PLUS-R1の回路図

部品	シンボル	規格等	個数
プリント基板	PICO-HDMI-PLUS	Rev.1	1
抵抗	R1-R4	10K Ω	4
	R5	1Κ Ω	1
	R6-R13	220 Ω	8
タクトスイッチ	SW1-SW4	4 端子	4
HDMI	CN1	HDMI 19PIN コネクタ	1
圧電スピーカー	X1		1
ピンヘッダー	SV1, SV2	20 ピン	2

表 5.1 部品表



図 5.2 PICO-HDMI-PLUS-R1の部品配置

5.2 基板上の入出力

PICO-HDMI-PLUS-R1の基板上のスイッチと圧電スピーカーを表 5.2 に示します。

SW1-SW4 は MCU の内部プルアップでプルアップすることもできますが、内部プルアップは 抵抗値が大きく、利用環境によってはノイズ等の影響を受けることがあります。このため、外部プ ルアップ用回路を組み込んでいます。プルアップには電源が必要ですが、PICO-HDMI-PLUS-R1 の接続対象となる Raspberry Pi Pico/Pico W/Pico 2 の端子には、電源端子がありません。この ため、信号線 GP9 を HIGH にしてプルアップの電源として利用するように構成しています。

このため、PICO-HDMI-PLUS-R1 のタクトスイッチを使用する場合には、GP9 を出力にして HIGH に設定してご利用ください。

シンボル	信号線	備考
PU	GP9	スイッチプルアップ用電源
SW1	GP10	負論理
SW2	GP11	負論理
SW3	GP20	負論理
SW4	GP21	負論理
X1	GP8	圧電スピーカー

表 5.2 スイッチと圧電スピーカー

第6章

購入および問い合わせ先

6.1 ご協力のお願い

製品をより良くし、多くの方々にお楽しみいただけるよう、製品の向上に努めて参ります。問題 点やお気づきの点、あるいは製品の企画に対するご希望などございましたら、microfan_shop@ yahoo.co.jp までご連絡いただけますようよろしくお願いいたします。末永くご愛顧いただけま すよう、お願いいたします。

6.2 販売:ネットショップ

製品の販売はネットショップで行っています。対面販売は行っておりません。

- マイクロファン Yahoo!ショップ
 WEB アドレス: https://store.shopping.yahoo.co.jp/microfan/
- ・ アマゾン WEB アドレス:https://www.amazon.co.jp/s?merchant=A28NHPRKJDC95B

6.3 製品情報

マイクロファン ラボ

WEB アドレス:http://www.microfan.jp/ マイクロファンの製品情報や活用情報を紹介しています。

6.4 問い合わせ先

株式会社ピープルメディア マイクロファン事業部 E-Mail: microfan_shop@yahoo.co.jp TEL: 06-6767-6588 お問い合わせは基本的にメイルでお願いいたします。

6.5 所在地

株式会社ピープルメディア マイクロファン事業部 〒542-0061 大阪市中央区安堂寺町 2-3-5 第 18 松屋ビル 501