

TOPPERS 活用アイデア・アプリケーション開発 コンテスト

部門 : アプリケーション開発部門

作品のタイトル : ラズパイ Pico 大判電光掲示板

作成者 : 石岡之也 (個人)

共同作業 : -

対象者 : 展示会出展者、イベント企画者

使用する開発成果物 : TOPPERS/ASP 1.9.3
ARM Cortex-M0 アーキテクチャ・GCC 依存部パッケージ
コンフィギュレータ Release 1.9.6

目的・狙い

昨年移植した RaspberryPi Pico 用 TOPPERS/ASP 用のアプリケーションがなかったことから Pico を使った作品を開発しようと考えた。
ネットで公開されている Pico-example には PIO を使ったマトリクス LED のサンプルプログラムがあり、これを使った電光掲示板を開発することとした。

アイデア/アプリケーションの概要

Raspberry Pi Pico に 64x64 ドットのマトリクス LED 4枚と SD カードを接続し、SD カード上の画像データファイルとテキストファイルを読み取り、画像表示と漢字コードを含むテキストファイルをスライドさせながら表示する 128x128 ドットの大判の電光掲示板システム。

1. 開発の背景

昨年、TOPPERS/ASP を Raspberry Pi Pico (以後、ラズパイ Pico) へ移植し、TOPPERS/ASP を使えるようにしたが、自身でもラズパイ Pico と TOPPERS/ASP を組み合わせたアプリケーションや作品の製作を行っていませんでした。

ラズパイ Pico は安価でありながら搭載されている SRAM が 264KB とワンチップマイコンでは大容量で小サイズであれば画像データも扱える。利用できる GPIO も多数あり PIO という機能で CPU の能力を使わずに GPIO を制御することができます。

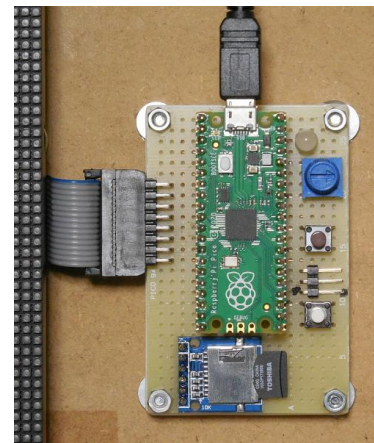
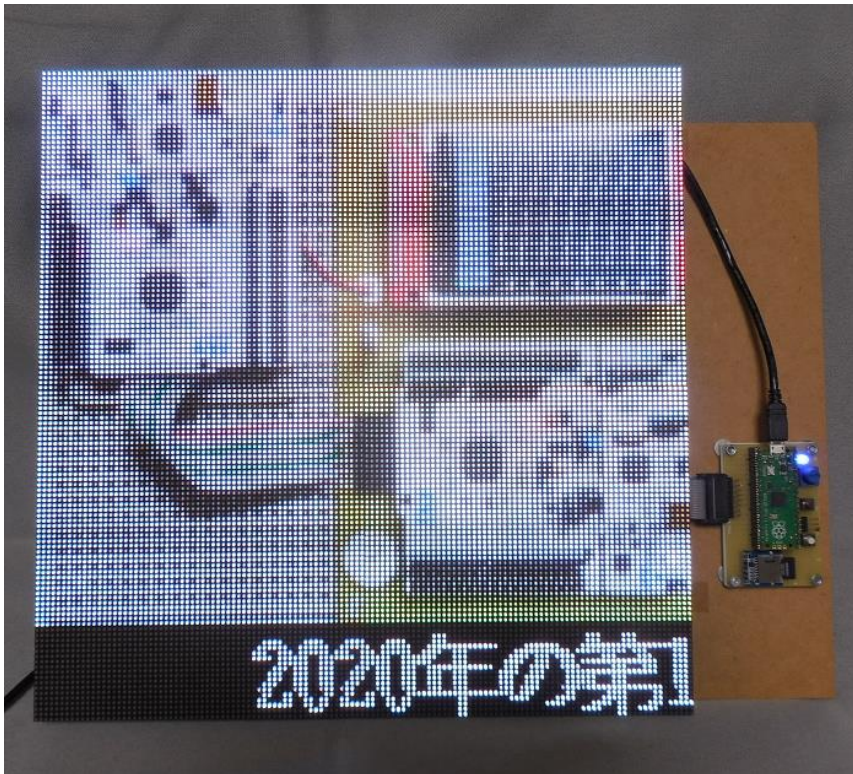
また、ラズパイ Pico には pico-examples という各種機能のサンプルコードがネットで公開されていて、この中には PIO を使ったマトリクス LED を制御するプログラムが含まれていた。このプログラムを流用することでマトリクス LED へ RGB の 3 原色を使った 7 色より多くの発色を容易に出力することが可能なの分かりました。

以前に FM3 や STM32 マイコンボードを使った文字表示の電光掲示板を開発していたが、ラズパイ Pico を用いることで画像表示も行える電光掲示板を開発してみようと考えました。

2. 開発作品の紹介

作品の全体構成は以下のようになります。

4 枚のマトリクス LED パネルで 128 × 128 ドット、32 cm × 32 cm のディスプレイとなります。制御は 1 枚のラズパイ Pico でマトリクス LED パネルの他に SPI で SD カードと接続しています。



写真の基板には SD カードやリセットスイッチ以外の部品も搭載されていますが、今回は使用しないため、後述の回路図には記載がありません。

実際の動作動画を以下へアップしました。

動画中でディスプレイ面にちらつきがありますが、これはダイナミック表示方式で縦方向を 1 ライン単位での表示を繰り返しているためです。実機ではちらつきはほとんどきになりません。

https://youtu.be/33s-l_L08Eg

SDカード上の画像データとテキストデータは以下の仕様で記録されている想定です。

- 画像データ

以下の8つのファイル名を順に読み出し、表示します。

ファイル形式は1ピクセル16ビット、横128ドット、縦110ドットのBMPファイルです。

```
image1. bmp  
image2. bmp  
image3. bmp  
image4. bmp  
image5. bmp  
image6. bmp  
image7. bmp  
image8. bmp
```

- テキストデータ

以下の8つのファイル名を順に読み出し、ドットフォントに変換して表示します。

文字データはSJISコードの2バイトコード、および1バイトのアスキーコードを扱えます。

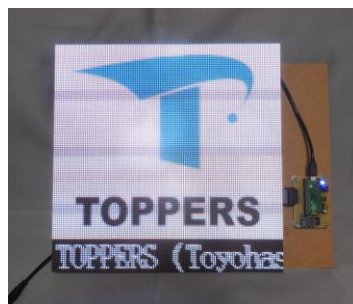
表示は全角200文字まで表示することができます。

```
text1. txt  
text2. txt  
text3. txt  
text4. txt  
text5. txt  
text6. txt  
text7. txt  
text8. txt
```

画像データとテキストデータはファイル名末尾の数字が同じファイル同士が連携して表示されます。
image1.bmp を表示する際には text1.txt の内容がディスプレイ面下部にスライドしながら表示されます。

今開発では以下のような内容を記録し、動作確認に使用しました。

1)



TOPPERS (Toyohashi OPen Platform for Embedded Real-time Systems) プロジェクトは、組み込みシステム技術と産業の振興を図ることを目的としたプロジェクトです。

2)



TOPPERS 公式マスコット「とばめ」です。小柄ながら空高く自由に舞う機敏な姿に TOPPERS プロジェクトのイメージを重ねた「ツバメ」が採用されています。

3)



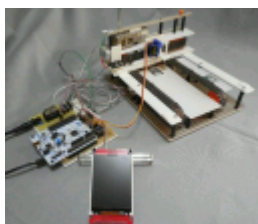
とばめ夏休みバージョン。麦わら帽子とスイカで夏をイメージしているようです。

4)



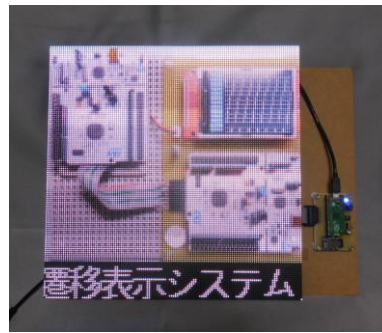
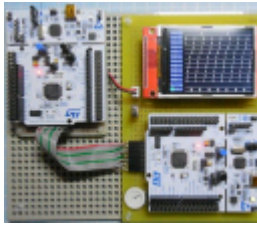
とばめ夏休みバージョン 2。水中メガネ、シュノーケル、浮き輪で海水浴から夏をイメージしているようです。

5)



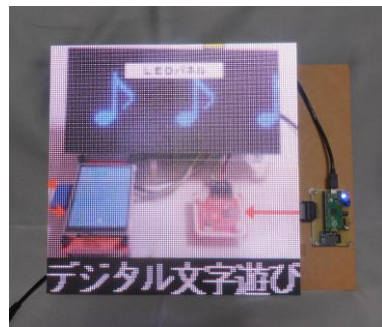
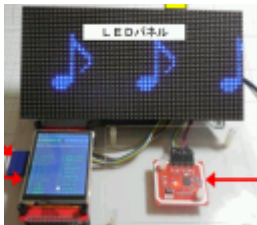
2021 年の第 11 回 TOPPERS 活用アイデア・アプリケーション開発コンテストで銀賞を受賞した「自動 ROM 書き込み装置デモ機」です。

6)



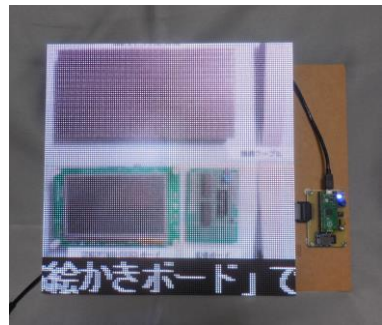
2020年の第10回 TOPPERS 活用アイデア・アプリケーション開発コンテストで金賞を受賞した「TOPPERS/ASP のタスク遷移表示システム」です。

7)



2019年の第9回 TOPPERS 活用アイデア・アプリケーション開発コンテストで銀賞を受賞した「デジタル文字遊び」です。

8)



2018年の第8回 TOPPERS 活用アイデア・アプリケーション開発コンテストで銅賞を受賞した「デジタルお絵かきボード」です。

3. ソフトウェア構成

(1) 流用ソフトウェア

今作品では、ネット上の以下のソフトウェアを流用しています。

- TOPPERS/ASP カーネル ターゲット非依存部パッケージ
<https://www.toppers.jp/download.cgi/asp-1.9.3.tar.gz>
- ARM Cortex-M0 アーキテクチャ・GCC 依存部パッケージ
https://www.toppers.jp/download.cgi/asp_arch_arm_m0_gcc-1.9.4.tar.gz
※インターフェース 2021年8月号のラズパイ Pico 移植記事のダウンロードデータ
<https://www.cqpub.co.jp/interface/download/2021/08/IF2108TOPP.zip>
- ラズパイ Pico pico-sdk
<https://github.com/raspberrypi/pico-sdk.git>
- ラズパイ Pico pico-examples
<https://github.com/raspberrypi/pico-examples>
- Fatfs
http://elm-chan.org/fsw/ff/00index_e.html
<http://elm-chan.org/fsw/ff/arc/ff14b.zip>
<http://elm-chan.org/fsw/ff/ffsample.zip>
- 16x16 ドット、8x16 ドット FONTX フォントデータ
jiskan16.bdf
8x16rk.bdf

(2) 改造概要

- TOPPERS/ASP
今作品の TOPPERS/ASP はインターフェース 2021年8月号のラズパイ Pico 移植記事で公開されているダウンロードデータを用いました。
インターフェース誌で公開されている TOPPERS/ASP では以下のピン操作がされているため、削除、改造して利用しました。
 - 外部 LED 制御 (GP10、GP11、GP12、GP13)
 - 外部ボタン制御 (GP15)
 - UART ピン (GP0 → GP20、GP1 → GP21)
 - UART ポート番号 (uart0 → uart1)
- pico-examples の hub75 サンプルコード
マトリクス LED を制御するため、pico-examples/pio/hub75 サンプルコードを流用します。
ダウンロードだけではコード内の変換前コード pico-examples/pio/hub75/hub75.pio がソースコードとして利用できないため、pico-examples を一度ビルドしてヘッダファイルへ変換します。
hub75.c と hub75.pio.h を流用します。
- hub75.c の改造
流用元の hub75.c はサンプルコード内の mountains_128x64_rgb565.h を出力するコードになっているため、これを通常の BMP ファイルから 128x110 ドットの BMP データとして取り出し、hub75 へ出力するコードへ変更します。
また、文字データを表示領域下部へ出力するため、文字データをビットイメージへ展開したデータを使うコードへ変更します。

- Fatfs のラズパイ Pico への対応

Fatfs の <http://elm-chan.org/fsw/ff/ffsample.zip> 内にある STM のコードを改造してラズパイ Pico の以下のピン設定でアクセスするよう改造します。

SPI0-RX	-	GP1019
SPI0-CSn	-	GP1018
SPI0-SCK	-	GP1017
SPI0-TX	-	GP1016

- sample1 サンプルコード

TOPPERS/ASP 内のサンプルコード `sample1` を改造し、以下の機能を組み込みます。

- `main_task0`

マトリクス LED への出力機能を初期化します。

`task10`、`task20` を起動します。

マトリクス LED を定期的に更新するためループで処理を繰り返します。

- `task10`

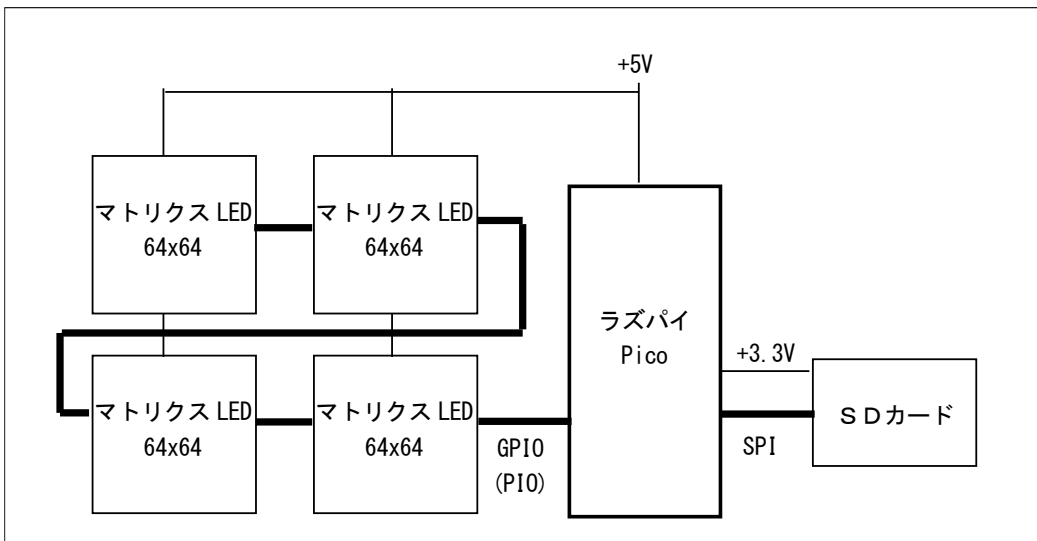
文字データをスライドさせるため、文字データの出力位置の情報を更新します。

文字データの表示が終端に達したらイベントフラグを使い新たなデータの読み出しを起動します。

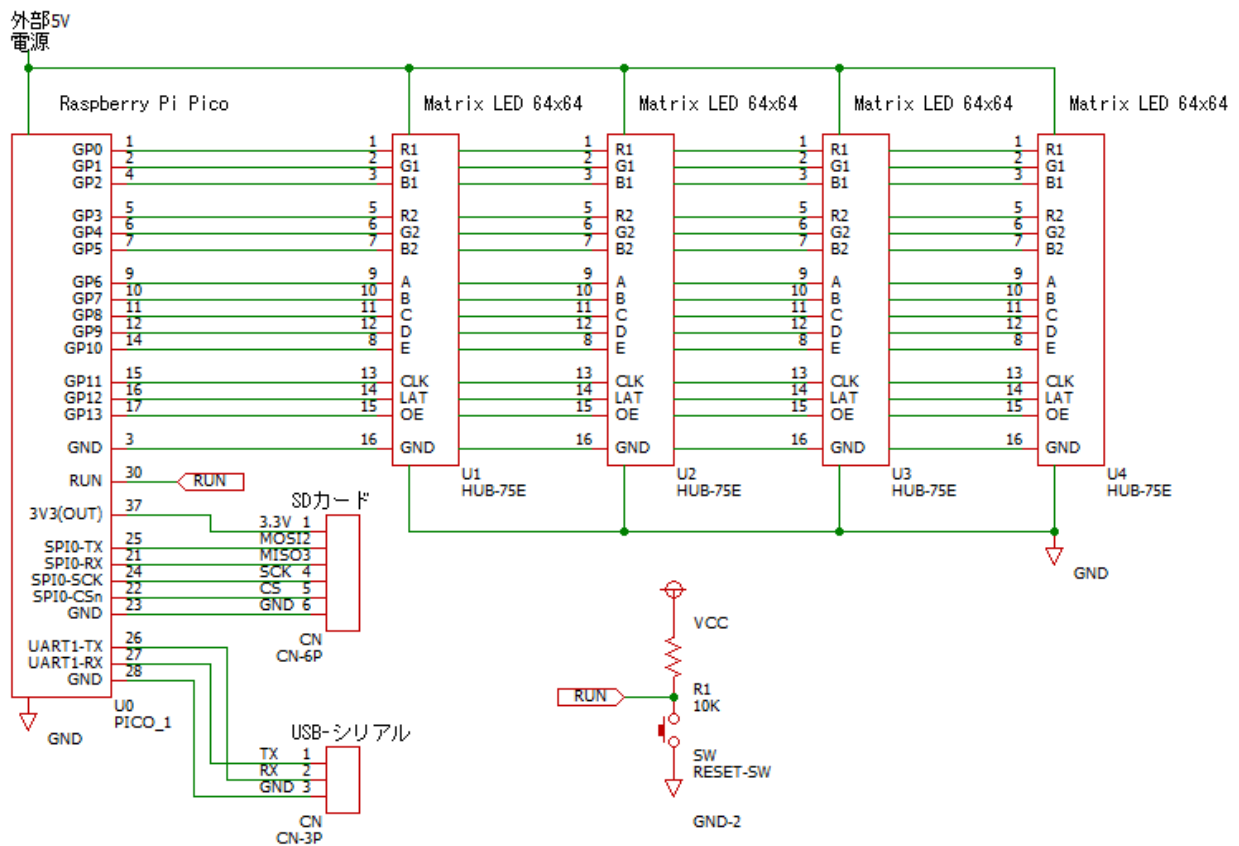
- `task20`

イベントフラグで待ち状態へ移行し、イベントフラグがセットされたら SD カードから画像データと文字データを読み出し、マトリクス LED へ出力するためのデータへの変換を行います。

4. ハードウェア構成



今作品の回路は以下になります。
マトリクス LED は IN/OUT のデジチェーンでの接続です。



5. 最後に

1) 画像データについて

今作品はまだ試作段階のソフトウェアのため、SDカードに収める BMP ファイルの形式が以下の条件に合う必要があります。今後はもう少し柔軟な画像サイズやヘッダデータに対応できるような改善が必要と考えています。

- ・ヘッダサイズは **0x42** バイト
- ・1 ピクセルは **16** ビット
- ・画像サイズは横 **128** ビット、縦 **110** ドット

2) ファイル名、個数について

SDカードに書き込む画像ファイルとテキストファイルのファイル名、個数を固定にしてあり、表示するファイルの個数の変更にはソースコードを書き換えてビルドしなおす必要があります。

アクセスするファイル名や個数もSDカード上のデータにすることで任意のファイル名や個数のデータを出力できるよう改善が必要と考えています。

3) テキストデータの表示色

マトリクス LED はカラー表示が可能ですが、テキストデータは白色表示のみにしてあり、表示色の変更にはソースコードを書き換えてビルドしなおす必要があります。

テキストデータの表示色もSDカード上のデータとして設定できるようにすることで任意の表示色で出力できるよう改善が必要と考えています。

4) 応募コードについて

今作品では TOPPERS/ASP の改造を行っているため、コンテスト応募コードには TOPPERS/ASP のソースコードを含みます。

また、コンフィギュレーターのバイナリが同梱してありますが、開発環境に応じて利用者がコンフィギュレーターをダウンロードしてビルドしなおし生成後、TOPPERS/ASP のビルドを行う必要があります。

5) 応募コードについて その2

応募コード内には pico-sdk や elf2ut2 のコード、バイナリは含まれていません。

利用者が必要なコードをダウンロードし、ビルドを行う必要があります。

以上