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	:	二字熟語問題生成器
作成者	:	石岡之也（個人）
共同作業者	:	ー
対象者	:	高齢者介護者、教育関係者、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

高齢者社会に向けて、認知症やボケ予防を意識した脳トレ問題を日常の中や、高齢者の集会などのスキマ時間に、トライする環境を用意することで予防の助けになるのではないかと思いました。
装置を電光掲示板の形態にして、目につく場所に設置することで、スキマ時間に繰り返し答えを考え、頭を使うことができるようにしたいと考えました。

アイデア/アプリケーションの概要

制御マイコンにラズパイ Pico、表示デバイスに 64x64 ドットのマトリクス LED パネルを用い、二字熟語の穴埋め問題の自動生成と表示を連続して行う装置です。
問題生成はプログラム内に埋め込んだ約７０００語、もしくはＳＤカードにテキストファイルで書き込んだ二字熟語を使い、ランダムに選択した「漢字」から適合する二字熟語をランダムに抽出して問題として表示します。

１．開発の背景

高齢者の認知症やボケの予防の一つに脳の刺激による認知機能の向上、達成感を味わうことが効果的と言われています。
また、家に引き込まらずに他人との交流によるコミュニケーションも脳の刺激に役立つことから、地域で高齢者同士の集会などが定期的に行われています。

開発者の親も高齢になり地域の集会へ参加して『脳トレ』になるパズルや漢字クイズなどを参加者と行ったり、問題を家へ持ち帰って解いたりしています。
持ち帰った問題は少なく１～２日程度で終ってしまい、集会も２週間程度の間隔で開催されるため、ほとんどの日常で『脳トレ』は実施していません。
このため、自宅でいつでも『脳トレ』が実施できるよう、また、同じ問題が繰り返しにならないよう、プログラムでランダムに問題を作成して表示できる装置を開発してみようと考えました。

使用場所は自宅だけでなく、高齢者同士の集会へも置いて使用できるようにすることで利用価値が上がると考えました。このため、煩雑な操作なしで使用できるようマイコンボードを用いて電源オンから操作なしで『脳トレ』問題をランダムに自動で出題し、表示に視認性の良いドットマトリクス LED パネルを使った装置にしようと考えました。

ドットマトリクス LED パネルはドット数が多く、大きい割に入手性がよいことから 64x64 ドットのパネルを用いることにしました。
このドット数の表示装置で、自動で作問できる『脳トレ』問題という条件から二字熟語の共通する漢字を当てる『二字熟語の穴埋め問題』を作問して表示する装置を開発することとしました。

・二字熟語の穴埋め問題例

＜問＞		＜答＞
題		題
↓		↓
注→■→録	➡	注→目→録
↓		↓
撃		撃

２．開発作品の紹介

まず、「二字熟語穴埋め問題」ですが、二字熟語の２つの漢字のうち一方が欠けた状態でヒントを元に二字熟語を完成させる問題です。
ヒントは表示された４つの漢字から欠けている同じ漢字を入れるとそれぞれで二字熟語が完成させられます。
表示される漢字は以下のように十字に４つ配置され、その中心が欠けている漢字を示します。

```
        ②題
         ↓
①注→■→③録
         ↓
        ④撃
```

配置位置に応じて欠ける部位が変わり、①、②が２文字目が欠けていて、③，④が１文字目が欠けていることになります。表示中の矢印の始点側が１文字目、終点側が２文字目を表しています。

```
① 注■
② 題■
③ ■録
④ ■撃
```

ここからそれぞれの漢字を使う二字熟語から共通の■部分の漢字を探し、その漢字を答えるという問題になります。

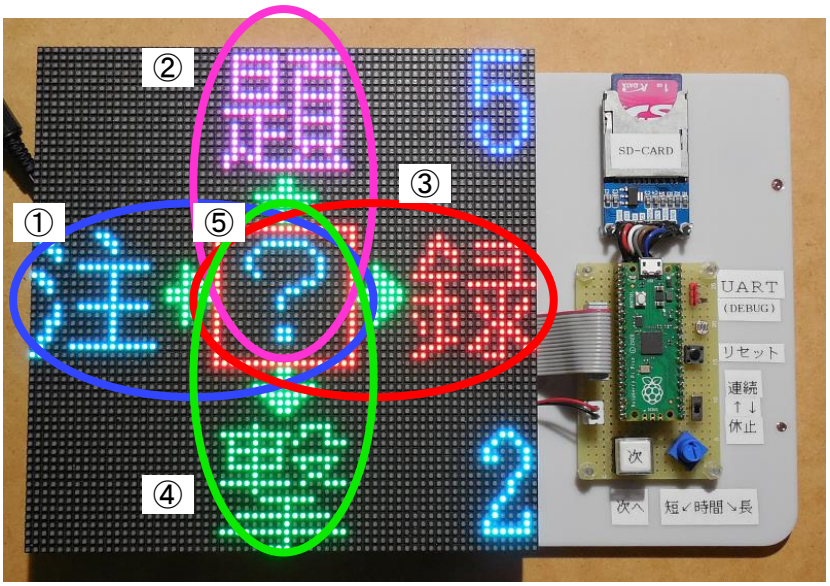
今回開発した作品は、この問題をマイコンボードを使って作問、マトリクス LED パネルへ表示、時間経過後に答えを表示という動作を繰り返します。
作問のネタとなる二字熟語をプログラム内に収録していますが、さらに後からネタとなる二字熟語をＳＤカード上のデータを使うことができます。
マイコンボードとしてラズパイ Pico を使いマトリクス LED パネルとＳＤカードを制御します。
マトリクス LED パネルへ二字熟語の穴埋め問題と回答が表示されるまでの秒数が表示されます。

プログラム内には約７０００語の二字熟語が収録されていて、プログラムが起動すると二字熟語からランダムに文字を選択し、その文字を含む二字熟語を抽出後にランダムに選択して作問と表示を行います。
図１の例では周りにある４つの漢字①②③④がそれぞれ４つの二字熟語の一文字となります。
この問題では、①、②が「注■」、「題■」と熟語の１文字目に、③、④が「■録」、「■撃」と二文字目となります。問題の解答となる漢字が⑤の「？」として表示されます。
設定した時間が経過後に図２の⑥のように答えが表示されます。

ランダムな文字選択から作問、表示、解答までを繰り返して延々と繰り返します。

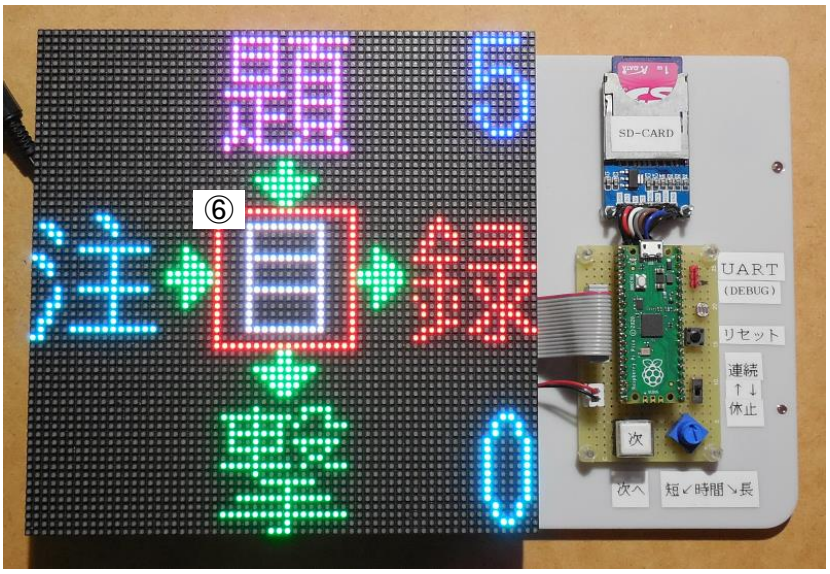
動作動画を以下へアップしました。併せてごらんください。

ラズパイ Pico で二字熟語問題作成！
https://youtu.be/XUUGabcCLz0



① 注■
② 題■
③ ■録
④ ■撃

図１　問題表示状態



① 注目
② 題目
③ 目録
④ 目撃

図２　回答表示状態

３．作品仕様

（１）二字熟語穴埋め問題生成

プログラム内に収録した二字熟語データを使い二字熟語問題を自動生成して表示します。

（２）解答表示時間設定

問題表示後から解答を表示するまでの待ち時間を変更できます。
時間は「１秒」「２秒」「５秒」「１０秒」「１５秒」「３０秒」「６０秒」から選択できます。
設定時間はパネル右上の『解答表示待ち時間の設定値』部分に秒数が表示されます。
解答表示までの残り時間はパネル右下の『解答表示までの残り時間』部分に秒数が表示されます。
工作品の各部の説明を図３に示します。

（３）解答表示停止設定

モードスイッチの状態により、解答表示待ち時間経過後も解答を表示せずに利用者の操作待ちとするか、時間経過後に解答を表示して次の問題へ移るか設定できます。
停止設定の場合、「次へ」ボタンを押すことで解答の表示と次の問題へ移ることができます。
連続設定の場合、操作せずに時間経過後に解答を表示して次の問題へ移ります。
デモは常時動作させる場合には「連続」を選択し、クイズ形式の場合「停止」を選択する想定です。

（４）時間経過スキップ設定

解答表示待ち時間を長く設定した場合などに、すぐに解答を見て次の問題へ移りたい場合、「次へ」ボタンを押すことで解答表示までの時間経過をスキップすることができます。

（５）外部問題利用

二字熟語をプログラム内に収録してありますが、問題の難易度を変更したい場合などのためにＳＤカード上のテキストデータから二字熟語を読み出し、これを使って二字熟語穴埋め問題を生成することができます。この場合、プログラム内に収録されている二字熟語は利用しません。
１００００語までの熟語を読み込むことができます。

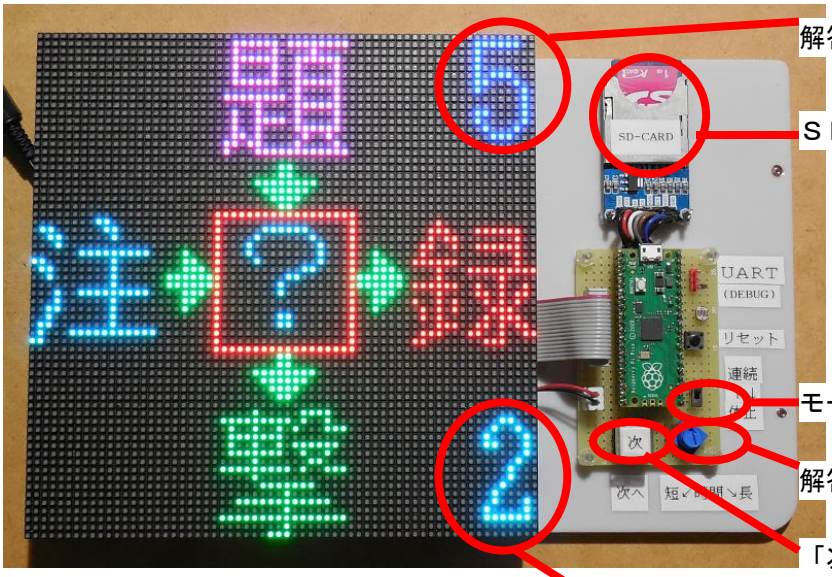


図３　各部の説明

（６）ＳＤカード

外部から二字熟語を定義するのに使用するＳＤカードは FAT ファイルシステムフォーマットに対応しています。

ファイルはテキストファイルでファイル名 niji.dat、漢字コードは Shift-JIS です。

ファイル内のフォーマットは以下のように二字熟語の漢字２文字と改行コードは<CR><LF>です。

```
＜１文字目＞＜２文字目＞<CR>＜LF＞
```

１００００語（１００００行）までを有効なデータとして使用します。

テキストファイル niji.dat の記述例を以下に記します。

```
母屋<CR><LF>
家屋<CR><LF>
小屋<CR><LF>
長屋<CR><LF>
屋号<CR><LF>
屋根<CR><LF>
  :
```

（７）乱数シード生成

当作品では、二字熟語穴埋め問題の作問時に乱数を使って解答となる漢字や用いる二字熟語をランダムに選択しています。

単純にＣ言語の rand()ライブラリを使うと常に同じ乱数列を使って動作するため、電源投入やリセット後に常に同じ問題を生成することになります。このため、当作品では明るさにより抵抗値が変化する CdS 素子を ADC へ接続し、周囲の微妙な明るさの違いを ADC 値で取得できるようにしました。この値で乱数のシード設定 srand()を行うことで起動後に異なる問題を作問しています。

３．ソフトウェア構成

（１）構成

当作品は、TOPPERS/ASP 上の３つのタスクと１つの周期ハンドラでプログラムを実行しています。タスクなどの構成を図４に記します。

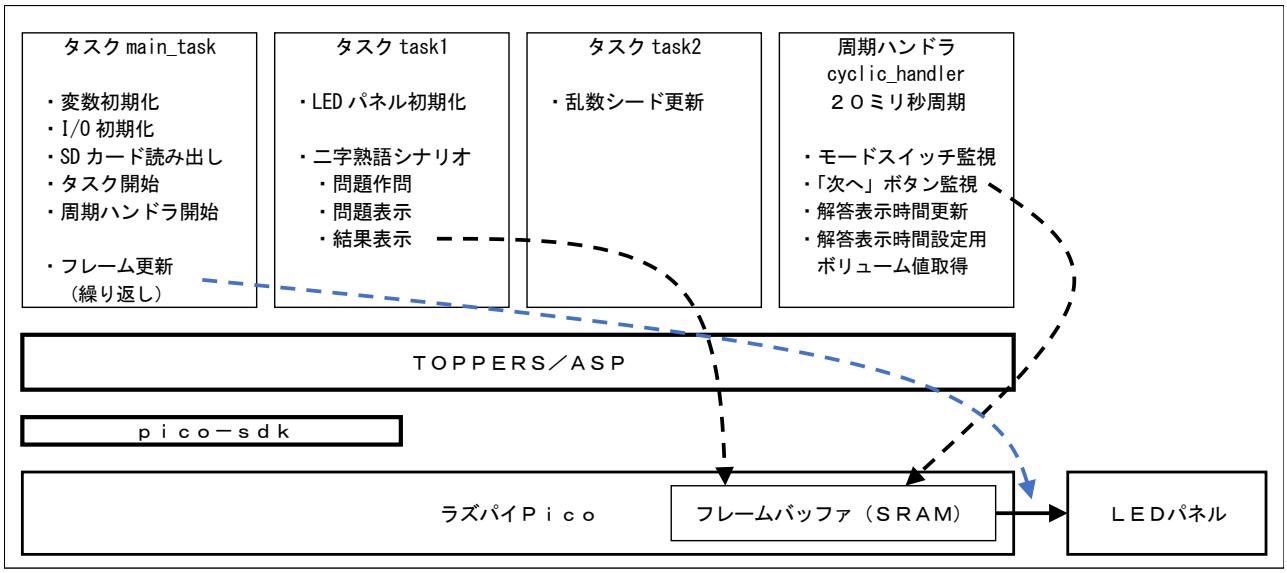


図４　ソフトウェア構成

（２）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利用して開発しています。

・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

（２－１）ラズパイ Pico 版 TOPPERS/ASP の改造

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）

（２－２）pico-exsamples の hub75 サンプルコードの流用、改造

マトリクス LED を制御するため、pico-exsamples/pio/hub75 サンプルコードを流用しています。
ダウロードした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-exsamples 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。

また、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 SRAM 上に割り当てたフレームバッファから PIO へ出力する処理へ改造を行いました。

（２－３）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。

```
SPIO-RX  - GPIO19
SPIO-CSn - GPIO18
SPIO-SCK - GPIO17
SPIO-TX  - GPIO16
```

（２－４）sampole1 サンプルコード

TOPPERS/ASP 内のサンプル sample1.c を改造し、以下のタスクや周期ハンドラを組み込みます。

・タスク main_task
変数を初期化します。
GPIO、PIU、ADC、SPI の I/O を初期化します。
SD カードの有無を判断し、SD カードがあれば二字熟語データの読み出しを行います。
CdS の ADC 値を読み出し、乱数のシードを設定します。
タスク task1、task2 を開始します。
周期ハンドラ cyclic_handler を開始します。
マトリクス LED パネルを定期的に更新するためループで処理を繰り返します。

・タスク task1
マトリクス LED パネルのフレームバッファを初期表示状態にします。
二字熟語のシナリオ動作を開始します。
乱数で二字熟語データ群から漢字１文字を取得します。
取得した漢字を使う熟語を１文字目、２文字目の各グループで抽出し、リスト化します。
１文字目のグループから乱数で２つの熟語を取得します。
２文字目のグループから乱数で２つの熟語を取得します。
ただし、２つ以上の熟語がなければ漢字１文字の取得からリトライします。
問題表示をフレームバッファへ出力します。
時間経過フラグを監視し、時間が経過したら解答表示をフレームバッファへ出力します。
問題、解答を消去後、シナリオ動作の先頭から繰り返します。

・タスク task2
６０秒ごとに CdS の ADC 値を読み出し、乱数のシードを更新します。

・周期ハンドラ cyclic_handler
モードスイッチの状態をチェックし、結果を変数へ設定します。
「次へ」ボタンの状態をチェックし、結果を変数へ設定します。
残り時間表示をフレームバッファへ出力します。
解答表示までの残り時間をカウントしていれば１秒ごとにカウントダウンします。
解答表示待ち時間の設定用ボリュームの ADC 値を取得します。
解答表示待ち時間をフレームバッファへ出力します。

４．ハードウェア構成

（１）モジュール構成

当作品で使用しているラズパイ Pico とマトリクス LED パネルなどのハードウェアデバイスなどのハードウェアモジュールの関係を図５に記します。

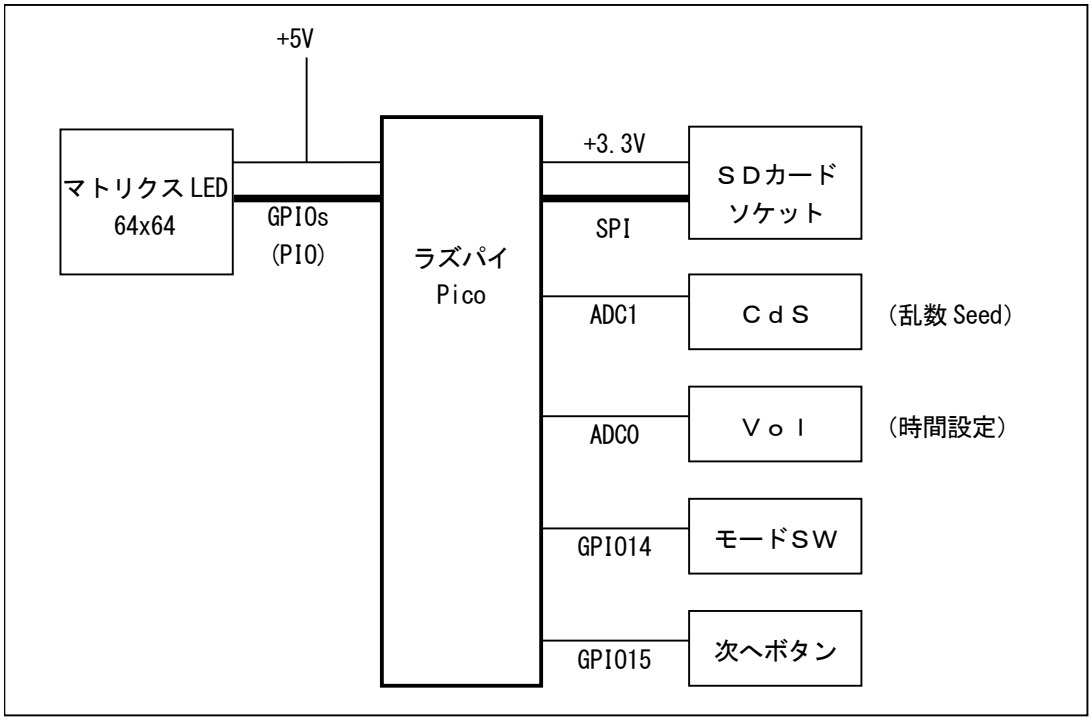


図５　ハードウェアモジュール構成

（２）回路図

当作品の回路図を図６に記します。

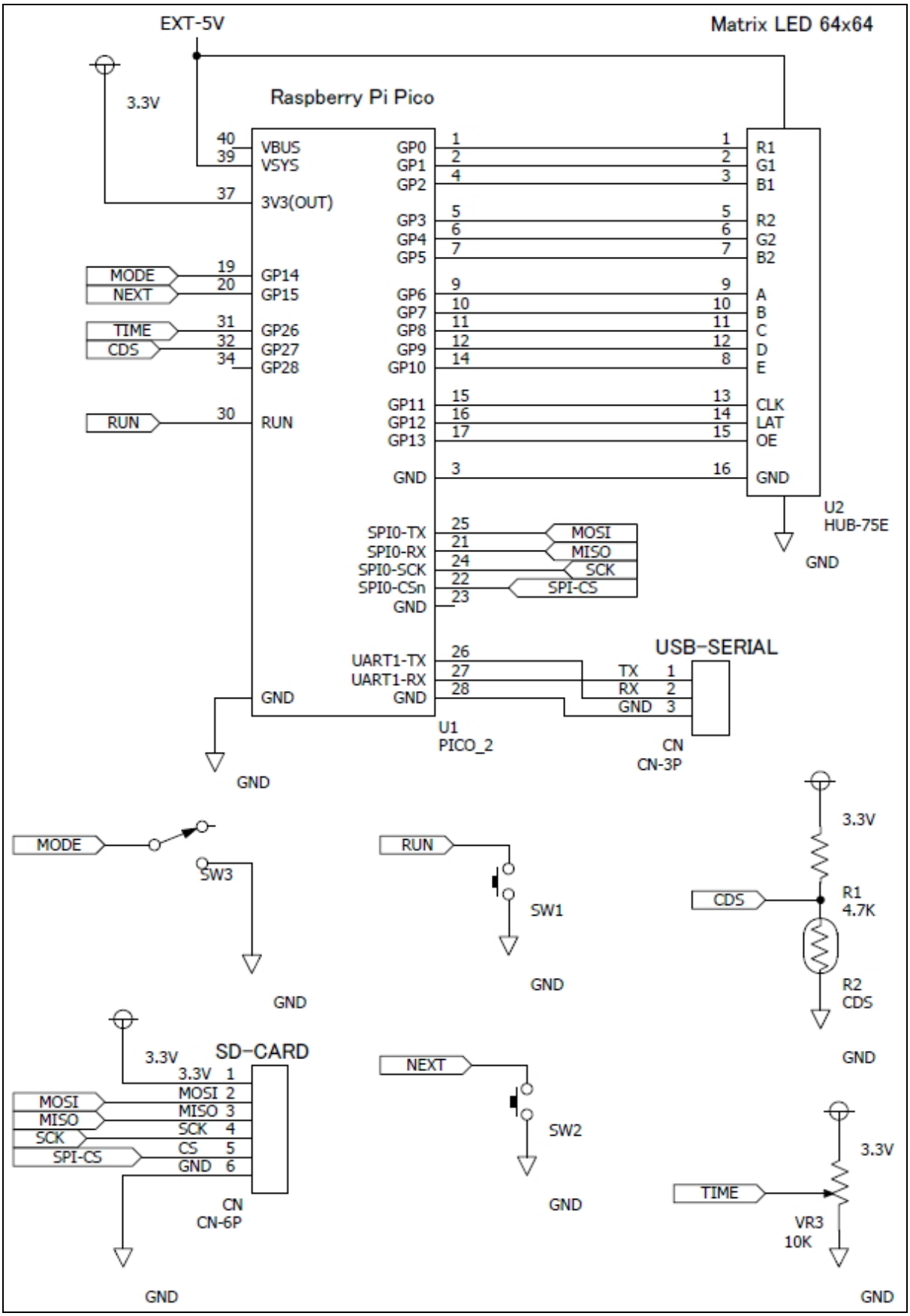


図６　回路図

５．最後に

１）利用範囲について

目的や背景にも記しましたが、作品開発当初は開発者自身の高齢の親の利用や、親が参加している集会での利用を考えて企画、開発しました。しかし、二字熟語のデータ作成のアドバイスをもらうため開発者の友人の国語教師へデモを見せたところ、教育関係者にも需要があるかもしれないとの助言をもらいました。

作品のデモ先、利用先を検討する場合には、高齢者向けだけでなく、教育関係、個人的には小学生や中学生など新たな漢字を勉強する人たちが集まる場も含めることで、利用範囲が広がりそうだということが分かってきました。

２）ＳＤカード上の二字熟語データについて

現時点ではＳＤカード上のファイルの読み込みは niji.dat というファイル名の１ファイルに限られていています。このため異なる難易度のデータを作成して試す場合、ＳＤカード上の元のファイルを上書きするか、複数のＳＤカードでの運用となります。

ここで二字熟語を１００００個収録しても niji.dat のファイルサイズは６０ＫＢ程度に収まり、ＳＤカードの容量に十分な空きがある状態です。

今後は複数の二字熟語データのファイルを切り替えて読み込みできるようにすることで、ＳＤカード１枚で複数の難易度のデータを使えるようにしていきたいと考えています。

３）使用するマトリクス LED パネルについて

今回は使用するパネル１枚で実現可能な 64x64 ドットのマトリクス LED パネルを使い、開発を行いましたが、HUB-75 形式のインターフェースを持つマトリクス LED パネルであれば他のサイズのマトリクス LED パネル、例えば 64x32 ドットを２枚使った構成でも表示データを展開しているフレームバッファから PIO への出力処理を少し工夫するだけで対応が可能です。

マトリクス LED パネルの製品をネットで検索すると 64x32 ドットのパネルは複数の LED サイズの製品が入手可能です。

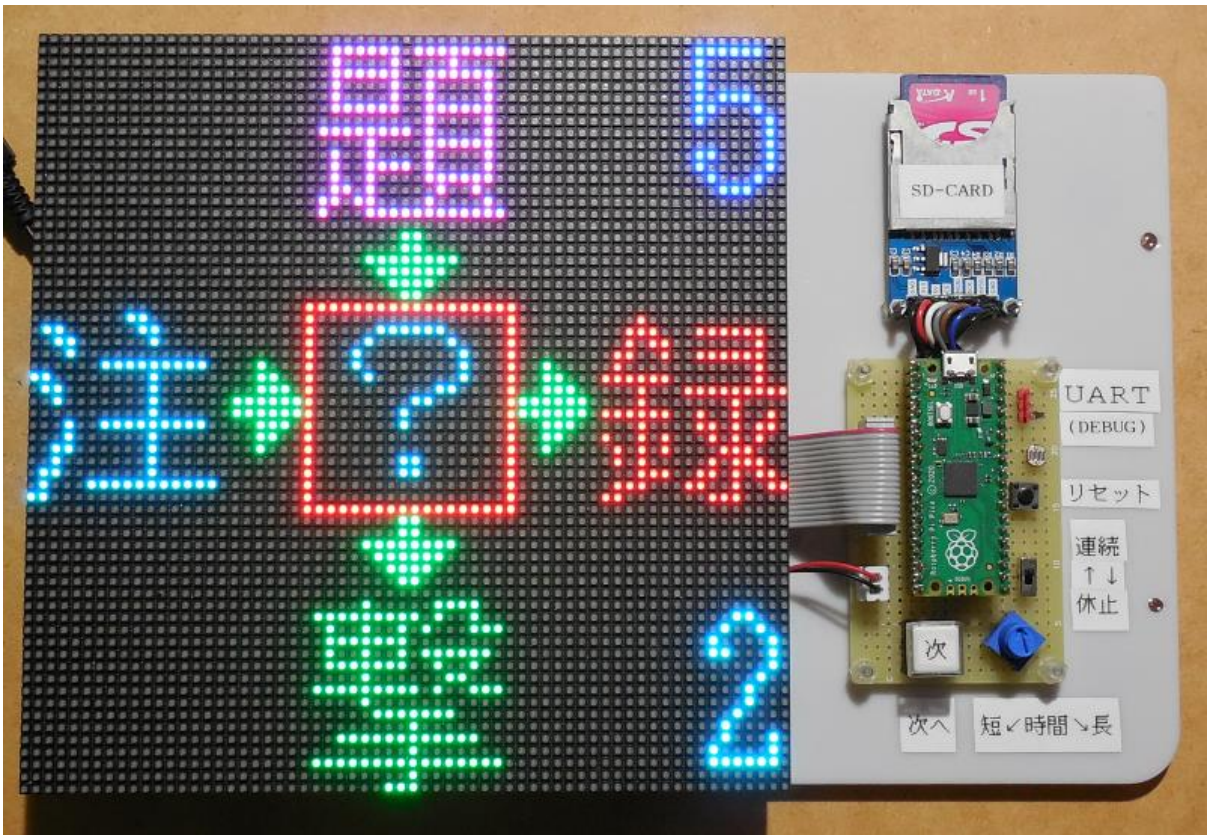
今後は 64x32 ドットのパネル２枚使いの構成にもスイッチ操作で対応できるよう、使用可能なデバイスのバリエーションを増やす方法も考えていきたいと思います。

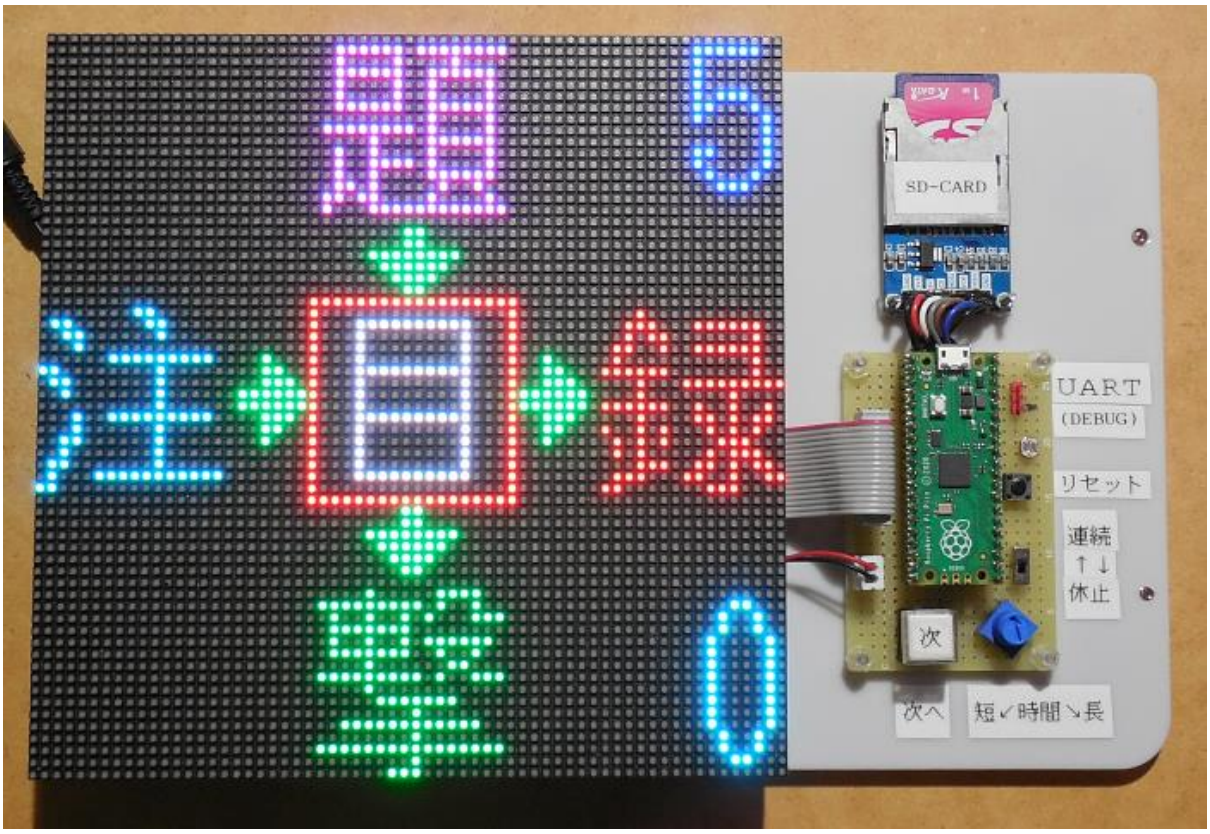
以上

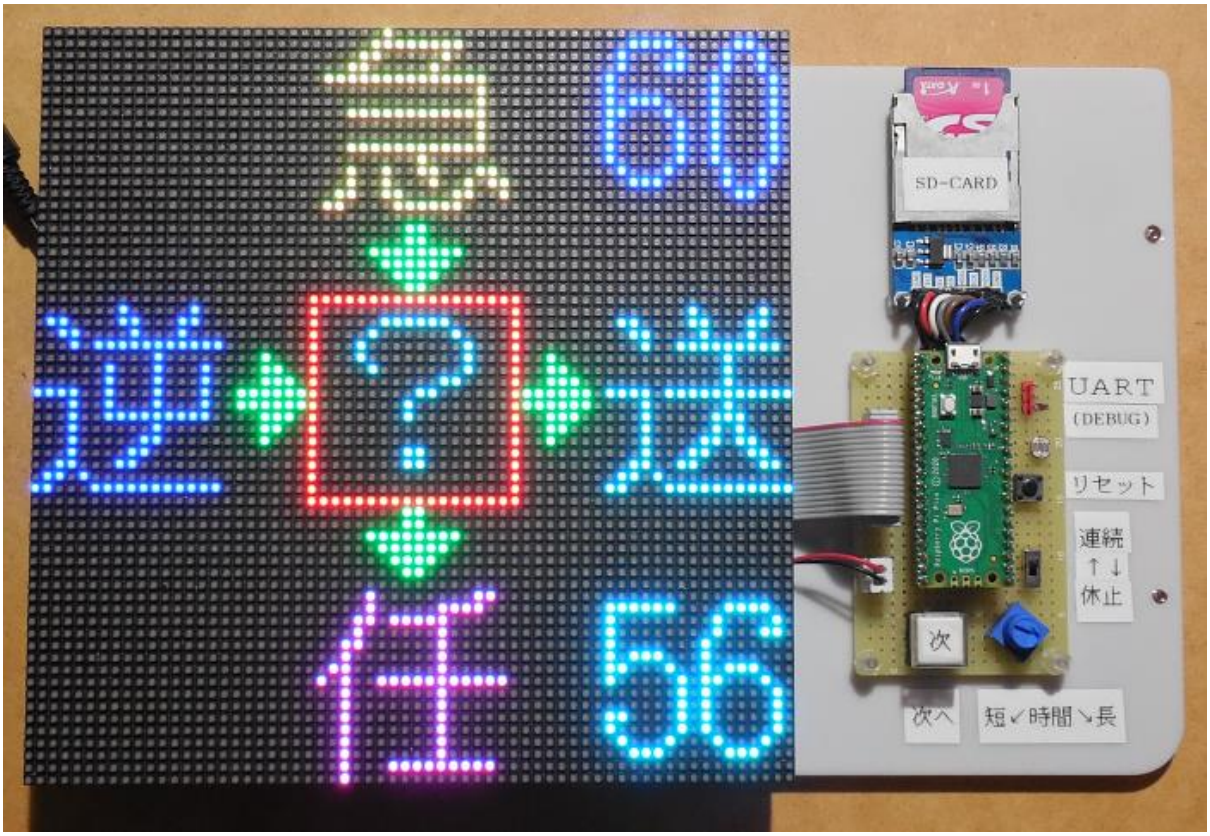
付録：動作画像

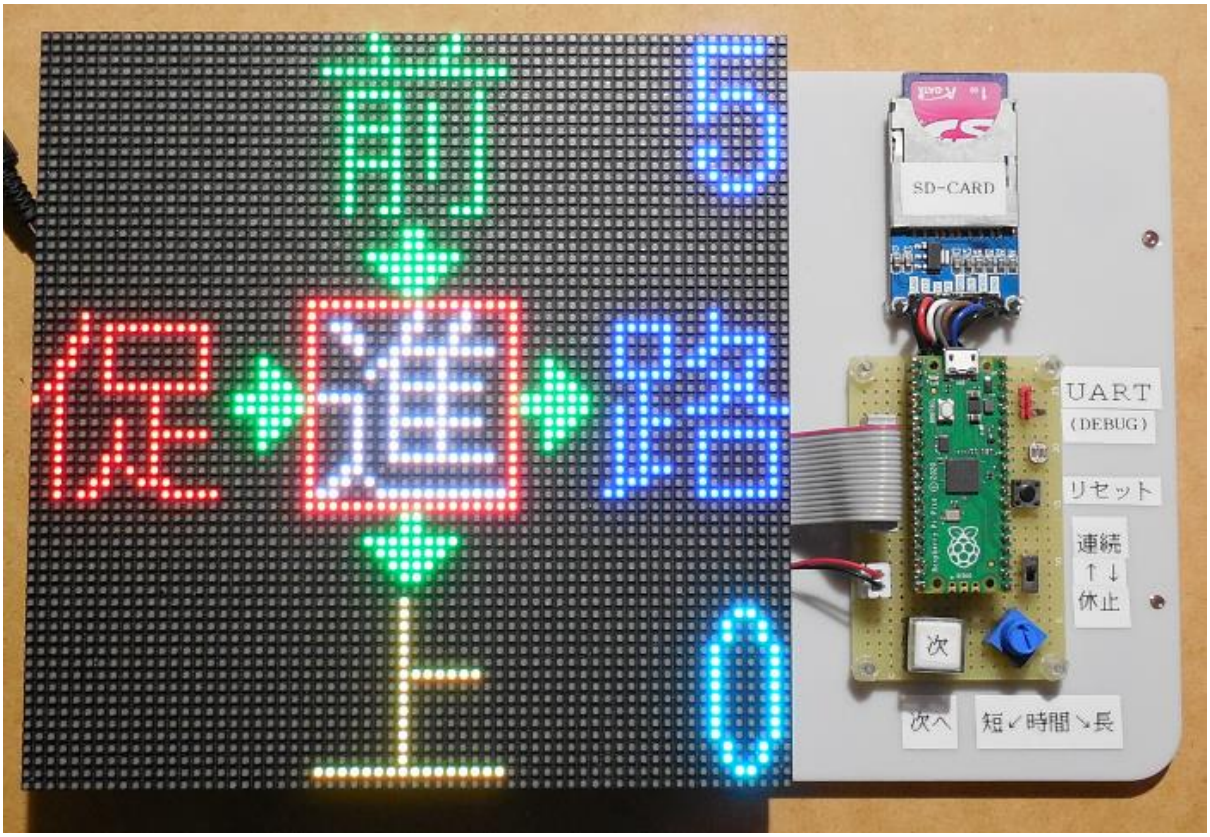
応募資料外の付録として動作画像をいくつか付けておきます。

応募資料内では画像にコメントが付加されているため作品画像を使う場合にはこちらの画像を利用ください。






SD-CARD
UART
(DEBUG)
リセット
連続
↑↓
休止
次
次へ
短↙時間↘長


SD-CARD
UART
(DEBUG)
リセット
連続
↑↓
休止
次
次へ
短↙時間↘長