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	：	TOPPERS/ASP のタスク遷移表示システム
作成者	：	石岡之也（個人）
共同作業者	：	
対象者	：	TOPPERS/ASP 利用者、組込みソフト開発者
使用する開発成果物	：	TOPPERS/ASP 1.9.3 ARM Cortex-M4 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

マルチタスクでプログラムを実行した場合に、各タスクが意図した順番で動作しているか、特定のタスクが動作していないなど確認したくなることがある。TOPPERS/ASP にはコンソールへのデバッグ情報の出力機能である syslog()を各タスク内に埋め込むことで大まかなタスクの流れを確認することもできるが、プログラムの修正や syslog()そのもののオーバーヘッドが大きくなることから多用するのが難しい。
このため、TOPPERS/ASP カーネルのディスパッチ処理から直接 GPIO を操作して各タスクの状態を表示する機能を実現しようと考えた。

アイデア/アプリケーションの概要

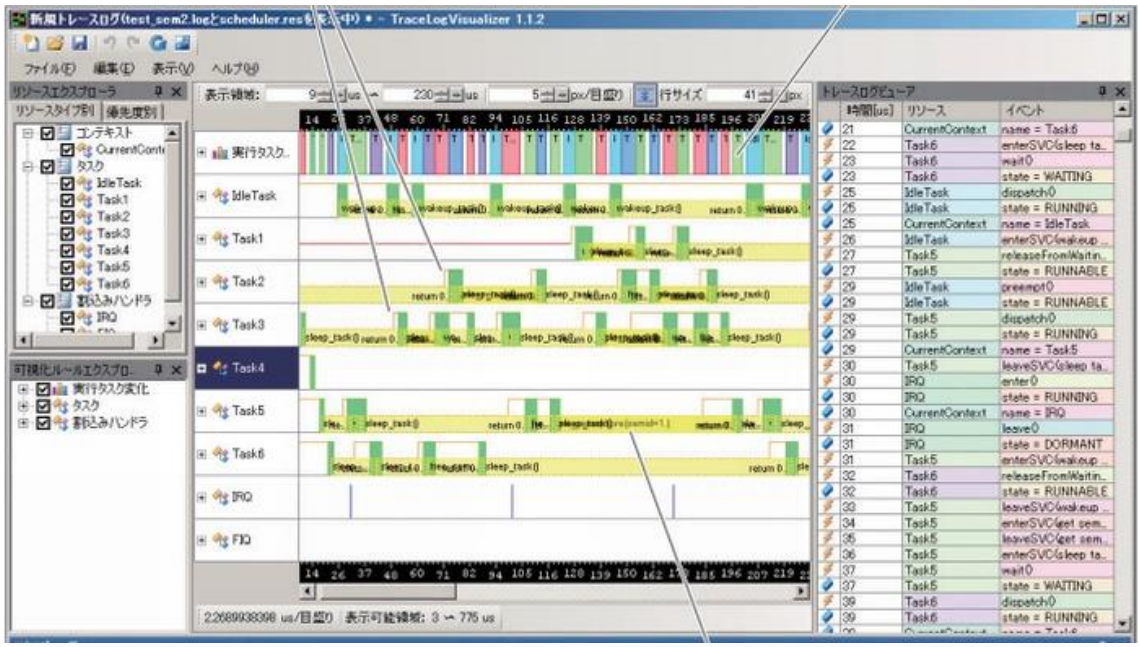
タスク遷移表示はターゲット装置上の TOPPERS/ASP 側の状態出力プログラムと、出力された状態を受信して結果をグラフィック LCD へ表示する受信・表示装置の２つで構成される。
状態出力プログラムはタスク ID ごとに割り当てた GPIO への 0/1 でタスクが動作中か否かを STM32F401-Nucleo ボードを用いて出力する。
受信・表示装置は STM32F446-Nucleo ボードを用いてタスク ID どごに割り当てられた信号を GPIO で受け、その結果をグラフィック LCD へグラフ状に表示を行う。

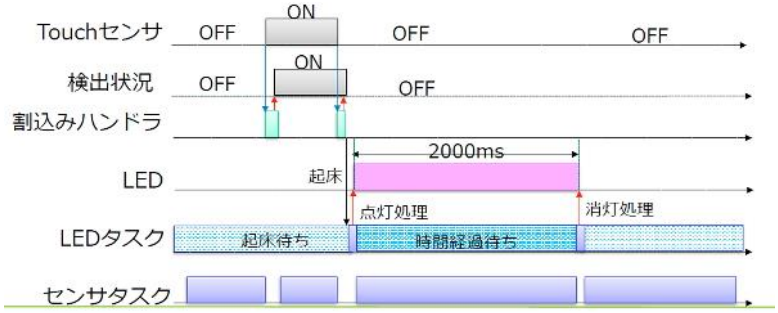
１．開発の背景

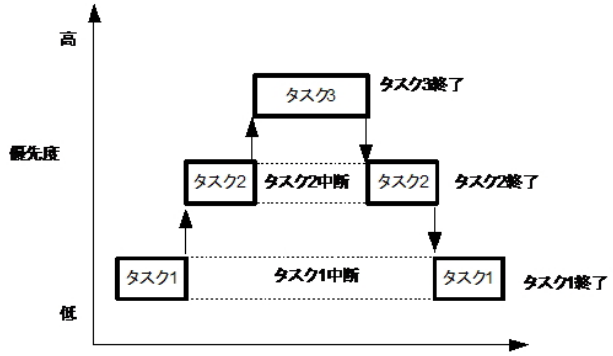
RTOS を使ってマルチタスクでプログラムを開発して動作確認やデバッグなどを行う際に、各タスクが意図した順番で動作しているか、特定のタスクが動作していないなどの状況を確認したいことが多々ある。

syslog()などのロギング処理をソースコードに埋め込むことでタスクや処理の流れを大まかに確認することができるが、ソースコードに修正を加えなければならないこと、ロギング処理を追加したルートしか流れがつかめないこと、コンソールへの出力となるとオーバーヘッドが大きくなるなどの問題があり、多用するのが難しい。

また、RTOS やマルチタスクの参考資料などにもタスクの流れを説明するものもあるが、基本的に開発者の頭の中でイメージしながらになり、実際に流れを目視する機会はほとんどない。市販のツールやデバッガなどではタスクの動作状況を表示する機能のあるものもあるが、個人での利用を考えると高価であるため、手ごろなボードや部品を使って安価に実現してみたいと考えた。



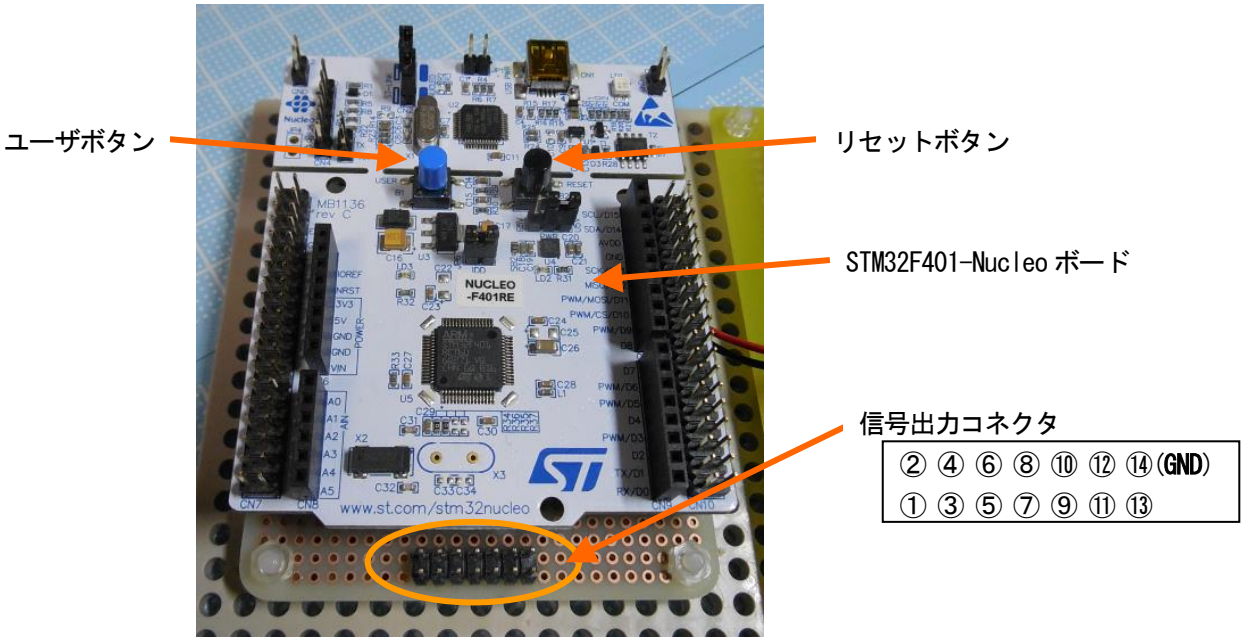




２．開発機器紹介

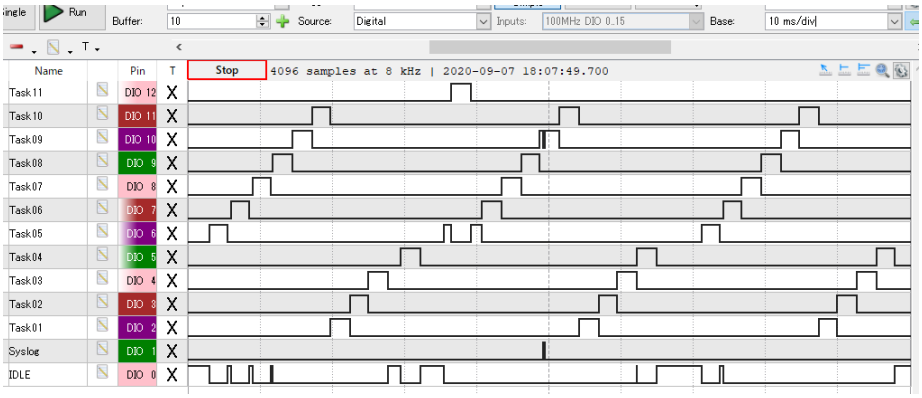
（１）ターゲット装置側

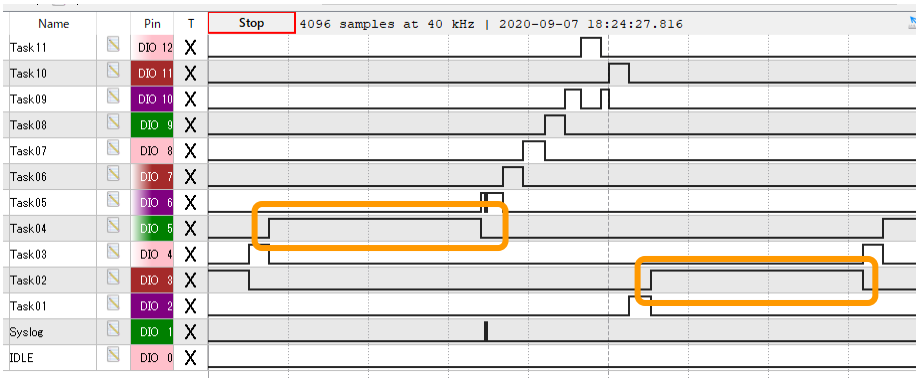
TOPPERS/ASP カーネルのディスパッチ処理に改造を加え、タスクスイッチ時にスイッチしたタスク ID に応じた GPIO を「1」にする。タスクが動作していないアイドル状態１本（①）とタスク ID 0～ID 11（②～⑬）までの１２本、計１３本の GPIO を制御してタスクの動作状態を出力する。

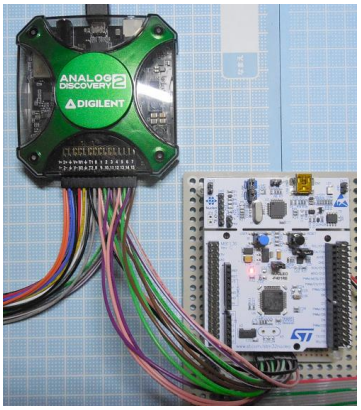


今開発ではカーネルの改造のほか、タスクの動作状況を分かり易くするデモとして sample1.c を改造してメインタスクの他にタスク１０個を生成し、それぞれが２ミリ秒程度のビジーループと slp_tsk()を繰り返すプログラムを作成した。さらにタスクの状態変化を見られるようユーザボタンの押下で TASK02、TASK04 のビジーループの時間を２０ミリ秒程度へ変化する処理した。

信号出力をロジックアナライザで受けることでタスクの動作状況を可視化することができる。







（2）表示装置側

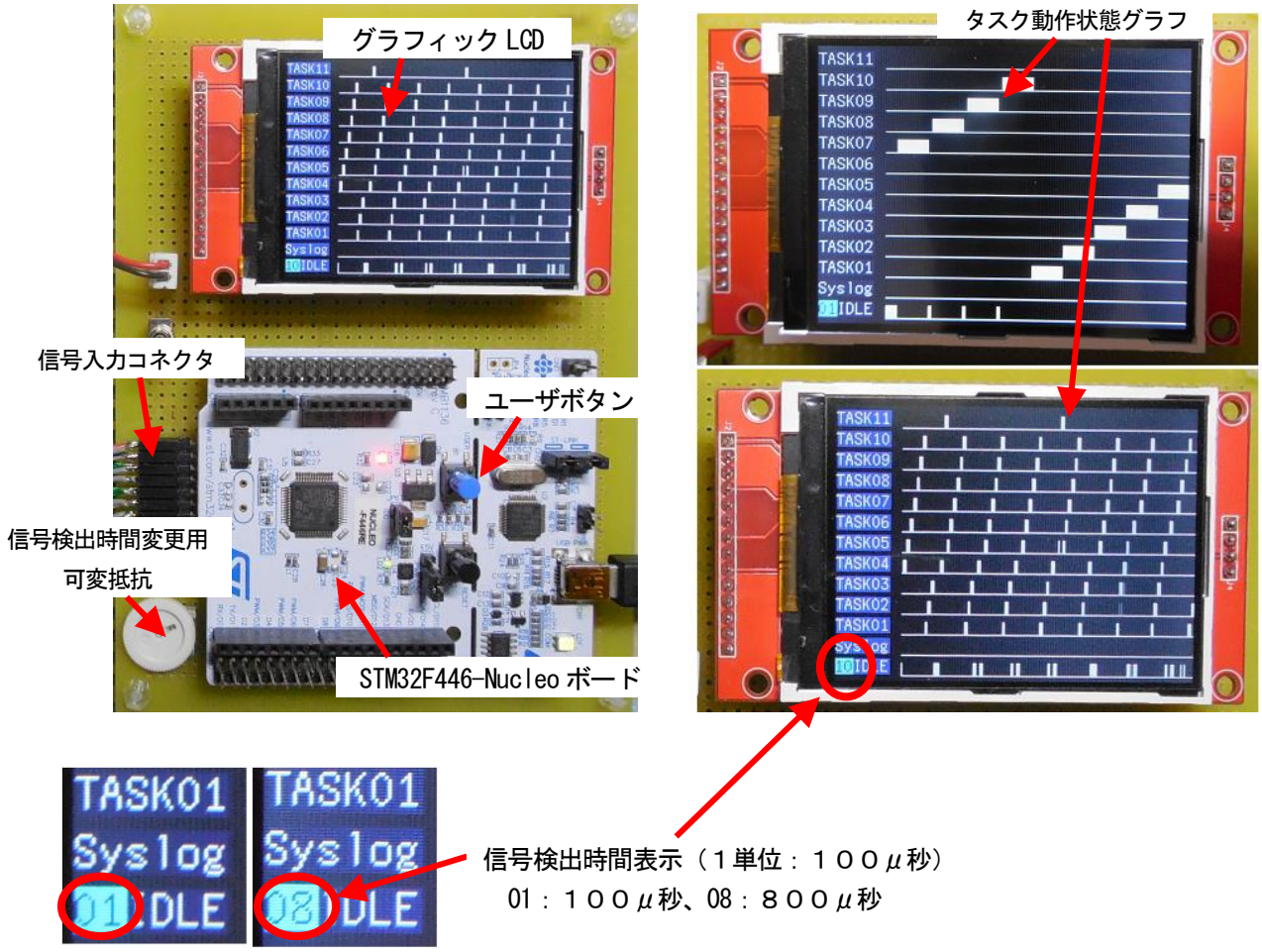
ターゲット装置側から出力されたタスクの動作状態信号を GPIO で受け、100μ秒ごとに GPIO の状態を取得しグラフィック LCD へタスク動作状態グラフとして表示する。

グラフィック LCD の縦方向がアイドルと各タスクの動作状態で、横方向が時間になる。時間は１ドットがデータ取得間隔の時間単位を意味する。

取得間隔は、TOPPERS/ASP のミリ秒単位の SysTick では荒くなるため、STM32F401 に内蔵されているタイマを使い 100μ秒周期で割り込みが発生するよう設定し、割り込みハンドラで定期的に GPIO の値を取得する。

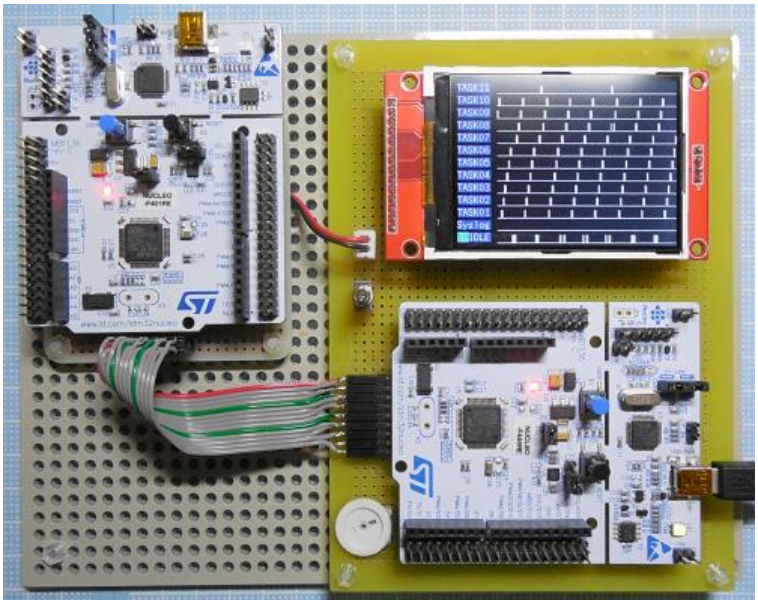
間隔が 100μ秒だけだと取得できる範囲が狭くなるため、可変抵抗と AD 変換を用いて信号検出時間を変更できる。AD 変換値から 100μ秒の ×１～×１６の１６段階を設定可能。

グラフ表示は１画面分のデータを取得したら逐次更新されるが、ユーザボタンを押下することで表示更新を停止することができる。



信号検出時間表示（１単位：１００μ秒）
01：１００μ秒、08：８００μ秒

装置全景



３．動作環境

（１）ターゲット装置側

●マイコンボード
ＳＴマイクロエレクトロニクス社 STM32F401RE-NUCLEO ボード
<https://akizukidenshi.com/catalog/g/gM-07723/>

（２）表示装置側

●マイコンボード
ＳＴマイクロエレクトロニクス社 STM32F446RE-NUCLEO ボード
<http://akizukidenshi.com/catalog/g/gM-10176/>

●2.8 インチ液晶モジュール
ILI9341 コントローラを搭載した SPI 接続の 320x240 ドットのカラーグラフィック LCD
検出信号のグラフ表示用
<http://www.aitendo.com/product/16038>

●可変抵抗器
５０ＫΩ
信号検出時間変更用、マイコンのＡＤ変換で抵抗値を検出

４．開発環境

Raspberry Pi / Ubuntu 20.04 上に ARM クロスコンパイラのインストールと TOPPERS 成果物を展開してプログラムの作成とビルドを行う。
以下に開発環境に必要なソフトウェアとそのダウンロード元を記す。

●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 >

●TOPPERS/ASP カーネル ターゲット依存部
ARM Cortex-M4 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_m4_gcc-1.9.6.tar.gz >
ターゲット装置側：stm32f401nucleo_gcc
表示装置側　　：stm32f446nucleo64_gcc

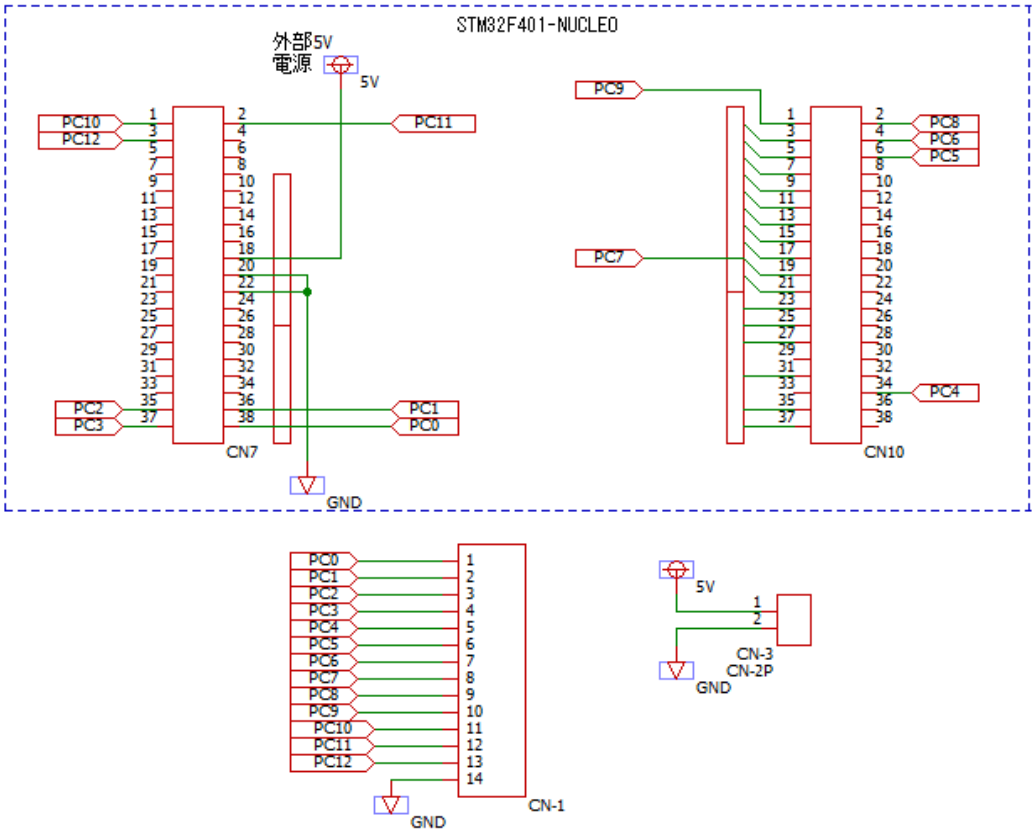
●コンフィギュレータ Release 1.9.6
< https://www.toppers.jp/download.cgi/cfg-1.9.6.tar.gz >

●ビルド用プラットフォーム
Ubuntu 20.04.1 / Raspberry Pi aarch64 用
< https://ubuntu.com/download/raspberry-pi/thank-you?version=20.04.1&architecture=arm64+raspi >

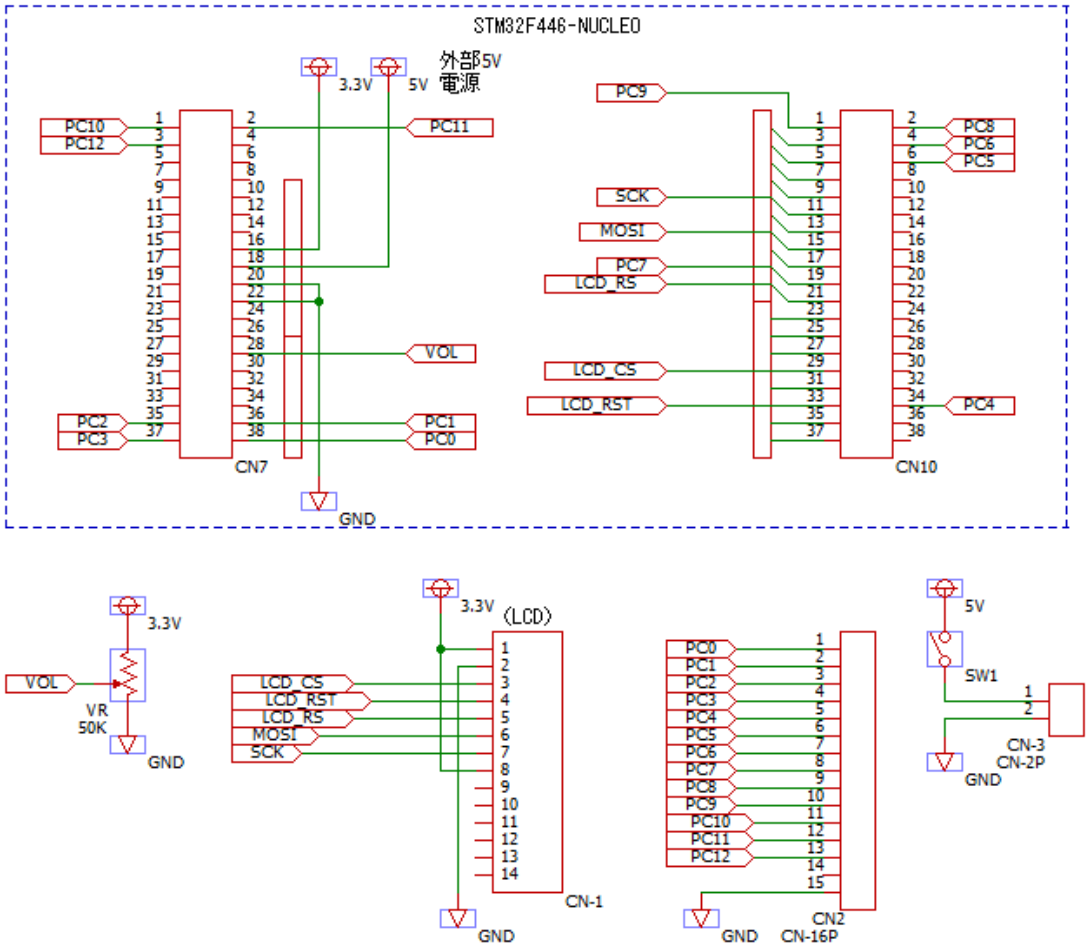
●クロスコンパイラ
GNU Arm Embedded Toolchain: 10-2020-q2-preview June 29, 2020 (Linux AArch64 用)
<
https://developer.arm.com/-/media/Files/downloads/gnu-rm/10-2020q2/gcc-arm-none-eabi-10-2020-q2-preview-aarch64-linux.tar.bz2 >

5．回路図

（1）ターゲット装置側



（2）表示装置側



６．ソフトウェア構成

（1）ターゲット装置側

●ディレクトリ、ソースファイルの構成

```
|--con2020_TaskDisp
|  |--outboard
|  |  |--asp    ・・・・・・・・・・・・・ASP カーネル用ディレクトリ
|  |  |--diff-con2020-1-asp.patch  ・・ASP カーネル修正パッチ１（ビルド不具合対応）
|  |  |--diff-con2020-2-asp.patch  ・・ASP カーネル修正パッチ２（タスクの動作状態信号出力修正）
|  |  |--out_f401  ・・・・・・・・・・・タスクの動作状態信号出力試行タスク
|  |  |  |--obj
|  |  |  |  |--Makefile
|  |  |  |  |--WaitMsec.S  ・・・・・・微小時間待ち、ミリ秒サブルーチン
|  |  |  |  |--dev_head_out.h  ・・・・デバイスアクセス用定義ファイル
|  |  |  |  |--sample1.c   ・・・・・・タスク状態出力用お試しタスクソース
|  |  |  |  |--sample1.cfg
|  |  |  |  |--sample1.h
```

● ソフトウェア追加・修正概要

TOPPERS/ASP カーネルビルド用のＧＣＣ不具合対応として以下のファイルの修正を行う。

```
asp/arch/arm_m_gcc/common/start.S
asp/target/stm32f401nucleo_gcc/stm32f4xx_rom.ld
asp/target/stm32f446nucleo64_gcc/stm32f4xx_rom.ld
```

ASP カーネルの dispatcher_0 と dispatcher_2 へタスクの動作状態を GPIO へ出力するルーチンを呼び出す処理を追加する。

```
asp/arch/arm_m_gcc/common/core_support.S
```

ディスパッチャ内でタスク ID へのアクセスを容易にするため、TCB へタスク ID のメンバを追加する。

```
asp/kernel/task.h
```

タスクの動作状態を GPIO へ出力する処理の初期化ルーチンを呼び出す処理と TCB へタスク ID を設定する処理を追加する。

```
asp/kernel/task.c
```

タスクの動作状態を GPIO へ出力する処理の初期化ルーチンやタスク ID から該当する GPIO を「1」にするルーチンを新作する。

```
asp/kernel/user_monitor.c
```

（2）表示装置側

●ディレクトリ、ソースファイルの構成

```
|--con2020_TaskDisp
|  |--inboard
|  |  |--asp    ・・・・・・・・・・・・・ASP カーネル用ディレクトリ
|  |  |--diff-con2020-1-asp.patch  ・・ASP カーネル修正パッチ１（ビルド不具合対応）
|  |  |--in_f466
|  |  |  |--dev
|  |  |  |  |--WaitMsec.S  ・・・・・・微小時間待ち、ミリ秒サブルーチン
|  |  |  |  |--WaitNsec.S  ・・・・・・微小時間待ち、ナノ秒サブルーチン
|  |  |  |  |--WaitUsec.S  ・・・・・・微小時間待ち、マイクロ秒サブルーチン
|  |  |  |  |--dev_gpio.c  ・・・・・・デバイスアクセスソース
|  |  |  |  |--fnt_shnm7x14a.inc ・・・shinonome フォントデータ
|  |  |  |  |--lcd_ili9341.c   ・・・・グラフィック LCD 制御ソース
|  |  |  |  |--spi.c   ・・・・・・・・SPI アクセスソース
|  |  |  |  |--spi.h
|  |  |  |  |--stm32f446xx.h   ・・・・STM32F446 用アクセス定義
|  |  |  |  |--stm32f4xx.h
|  |  |  |  |--system_stm32f4xx.h
|  |  |  |--include
|  |  |  |--obj
|  |  |  |  |--Makefile
|  |  |  |--src
|  |  |  |  |--out_pulse.c  ・・・・・・タスク動作状態グラフ表示ソース
|  |  |  |  |--sample1.c    ・・・・・・タスク制御、タイマ割込みハンドラソース
|  |  |  |  |--sample1.cfg
|  |  |  |  |--sample1.h
```

● ソフトウェア追加・修正概要

ASP カーネルビルドの GCC 不具合対応として以下のファイルの修正。

```
asp/arch/arm_m_gcc/common/start.S
asp/target/stm32f401nucleo_gcc/stm32f4xx_rom.ld
asp/target/stm32f446nucleo64_gcc/stm32f4xx_rom.ld
```

タスクの動作状態をグラフ表示するための処理ルーチンを作成。

```
inboard/in_f466/src/out_pulse.c
```

タイマ割込みハンドラやデバイス初期化ルーチンの呼び出し、タスク動作状態をグラフ表示するルーチンの呼び出しを追加。

```
inboard/in_f466/src/sample1.c
```

GPIO、タイマ、SPI、LCD などのデバイスへアクセスする処理や定義用のファイルを追加、作成。

```
inboard/in_f466/dev ディレクトリ
```