

動作確認の方法

- テスト環境

- ・ STM32CubeIDE 1.14.1
- ・ STM32G030F6P6およびST-LINK V3MINIE
- ・ Nucleo STM32F446RE

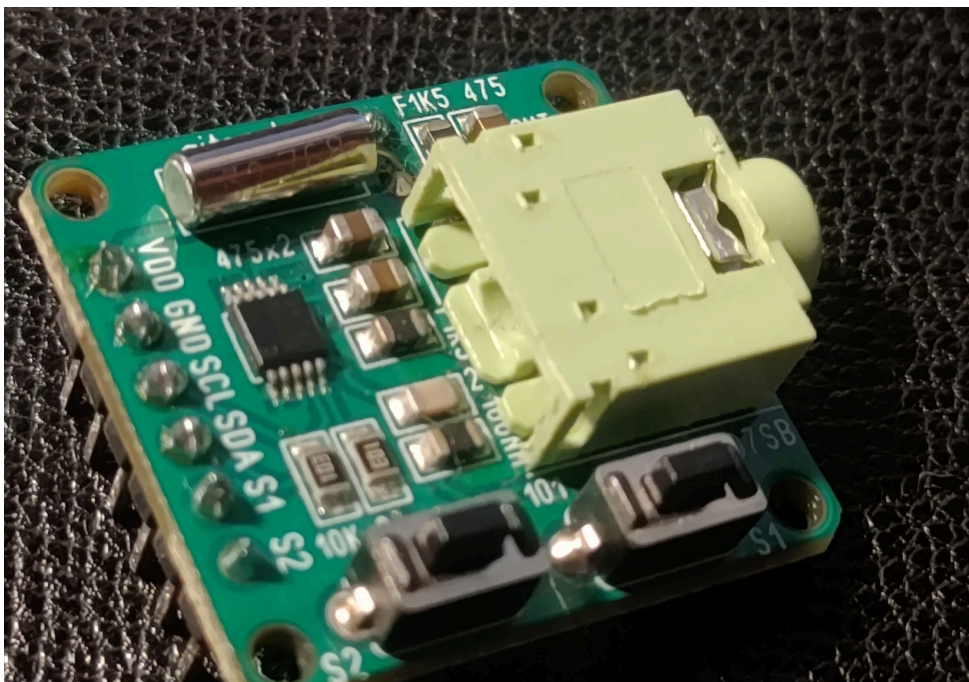
- 使用部品

- ・ aitendo ステレオFMラジオモジュール [M5807SB]

<https://www.aitendo.com/product/21202>

DIP部品のハンダ付けのみで動作が可能な小型のモジュールです。

イヤホンジャックや共振回路(100nHと24pF)も組み込まれており、これ1つで十分なテストができます。



- ・プルアップ抵抗について

このモジュールにはI2CのSCL、SDAが10KΩでプルアップされています。

しかしこのプルアップはモジュール裏面のPULL-UPラインをショートさせないと機能しないと思われます。

私はこのPULL-UPラインの存在を今になって知りました。

なおこのプルアップは利用しなくても問題ありません。

その場合はSCL、SDAそれぞれを5~10KΩでプルアップしてください。

●代替品

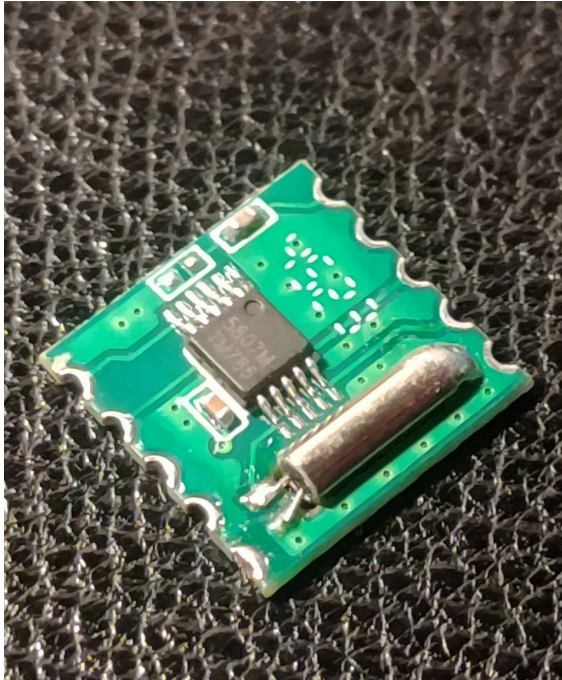
- ・秋月電子 FMラジオモジュール RDA5807M

<https://akizukidenshi.com/catalog/g/g117315/>

パソコンや水晶が実装済みの最低限のモジュールです。

注意点として**ピンピッチが2mm**な点です。ブレッドボードは通常**2.54mmピッチ**な為、ピンヘッダをそのままハンダ付けすることは困難です。

2mmピッチを変換する基板などもあるようですが、上記のaitendoのモジュールであれば単体で完結しているのでそちらをおすすめします。



- ・FMレシーバ [RDA5807FP]

<https://www.aitendo.com/product/4797>

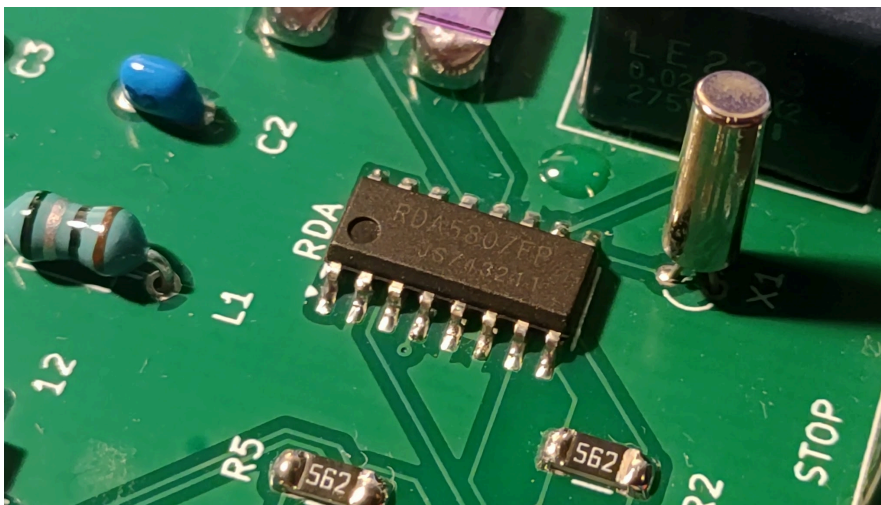
GPIOピンを持っており、シークの完了などをパルスとして出力できます。

特筆すべきは**SOP16**パッケージである点です。

単体で購入して自分で基板を作りたいという際にもおすすめです。

なお余談ながら**税込み99円**と今回紹介したものの中ではぶっちぎりで最安です。

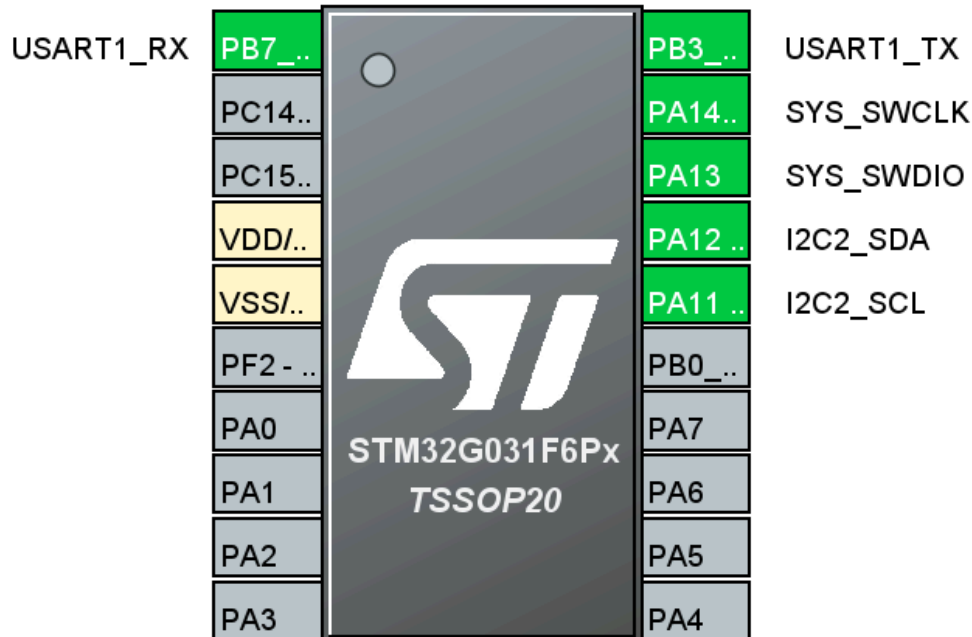
パソコン(22nF、0.022uFとして売られていることが多いです)や水晶振動子(32KHz)、SOP16→DIP変換基板と必要な物は多くなりますが、それでも最安で試せると思います。



●ソフトウェアの設定

STM32CubeIDEとHALライブラリの使用を想定しています。

プロジェクトファイル全体をアップロードすること考えたのですが、動作環境が異なることを想定してあえてCoreフォルダの内容のみを上げています。



.iocファイルからの抜粋です。

基本的には上記のようにUART、I2Cを有効化してもらえれば利用できると思います。

また、Nucleoボードの場合は**UART**、**SWCLK**、**SWDIO**がデフォルトで有効になっているはずです。I2Cのみ有効化してください。

UARTは特に設定値の変更をしていません。ボーレートは115200bit/s、8bit、パリティなし、StopBit 1です。

TeraTermを使用する場合、これらの設定がTeraTermと同一かどうか**必ずチェックしてください**。

I2Cもそのままの設定で問題ありません。

RDA5807MがSCL400KHzまで対応しているのでI2C Speed ModeをFast Modeにしていますが、大した差はないです。

・main.cの編集箇所

インクルードの設定

```
/* USER CODE BEGIN Includes */
#include <string.h>
#include "Radio.h"
#include "menu.h"
/* USER CODE END Includes */
```

strcpy関数(バッファに文字列をコピー)とstrlen関数を使用する為、string.hをインクルードしています。

main関数で使用する変数の宣言

```
int main(void)
{
    /* USER CODE BEGIN 1 */
    char message[30];
    uint8_t status = 0;
    char key;
    /* USER CODE END 1 */
```

初期設定

```
/* USER CODE BEGIN 2 */
RadioInit(&hi2c2);
strcpy(message, "RDA5807 Radio\r\nHelp->'?' \r\n");
HAL_UART_Transmit(&huart1, (uint8_t*)message, strlen(message),
HAL_MAX_DELAY);
/* USER CODE END 2 */
```

メインループ

```
/* USER CODE BEGIN 3 */
    status = HAL_UART_Receive(&huart1, (uint8_t*)&key,
sizeof(key), HAL_MAX_DELAY);
    if(status == HAL_TIMEOUT)
    {
        continue;
    }
    else if(status == HAL_OK)
    {
        SerialHome(&huart1, &hi2c2, &key);
    }
}
```

特に注意していただきたいのが**UART、I2Cの名前**です。

今回テストしたMCUではUART1、I2C2を利用しています。

特にmain.cをそのままコピーして使う際、仮に**I2C1**を利用しているとエラーになります。

個人的にはむしろmain.cをそのままコピーし、あえてエラーを出させて修正箇所を絞り込んでもらうことをおすすめします。

なお運良くUART1、I2C2が空いていればそっくりそのままビルドが通るはずです。

おわりに

今回のファイルはSTM32をあまり触ったことがないという方にも、手軽にラジオの実験ができるように という思いで作りました。

私もまだまだ浅学の身ですので、より汎用的かつ手軽に試せる環境を提供できるようにしていきたいと思っています。

このプログラムを通して、あなたの電子工作ライフがより楽しいものになれば幸いです。

wataoxp

