スマートフォンを利用した
ワイヤレスエレキーパドルの制作

Building wireless ele-key paddle for smartphone

AC22108　矢野立樹

AC22108　Tatsuki YANO

芝浦工業大学　無線研究部

Shibaura Institute of Technology, Ham radio club

1. 動機

本研究の動機は主にアマチュア無線の移動運用の際、パドルは重く、ケーブルが長いため持ち運びに向かないことから、普段持ち運ぶスマートフォン（以下スマホ）をパドルにできないかと考えたことである。

2. 目的

通常アマチュア無線で使用されるパドルは、本体に直接 3.5mm、もしくは 6.3mm プラグで接続するものが多く、運用の際にはどうしてもケーブルが長く、多くなってしまう。

また、CW を始めようとした際パドルは高価なものが多く、手が出しづらかった経験もある。

そこで本研究では、現代ではもはや生活必需品となったスマホにインストールできるパドルと、無線機を動作させるモジュールを制作し、手軽に CW ができるようになる装置の制作を目指す。

3. 設計

パドルの構造としては左右両方のキー（レバー）を操作し、無線機から接続された+端子と GND 端子が導通することで無線機に信号を送るというものである。

これらを踏まえ、設計はスマホから Wi-Fi を経由して導通の動作を制御し、CW の送出を行う方向性にした。

動作はまずスマホからの制御で 3.3V が抵抗に入力され、トランジスタのベース電流となる。これによりエミッタの無線機からの+5V とコレクタの GND 端子が導通し、CW 送出が行われる、というものである。これを長点側と短点側で 2 系統用意する。

スマホと無線機側端末の接続、制御には Wi-Fi アクセスポイント機能を搭載し、Arduino IDE からプログラムの書き込み、4 系統の定格 3.3V 出力を持ち単体で動作可能な小型のモジュール ESP-01 を選定した。

4. 製作

まずESP-01 に各プログラムを書き込んだ(別資料１)。プログラムの設計には Wi-Fi の設定、接続したスマホ側アプリへのボタンの配置と自動生成、そのボタンと ESP-01 の動作の紐づけを一括したプログラムを自動生成できるフリーソフト RemoteXY を使用した。

動作用の電源は単 3 型ニッケル水素電池 3 本から取り、無線機と端末間の接続には 3.5mm ケーブル、制御には hFE300 程度、耐電圧 40V 程度のトランジスタ 2N2222、抵抗は 220Ωのものを使用した。

これらをブレッドボード上に配置したところ ESP-01 からのノイズをトランジスタが拾ってしまい予期しない送出が行われることが多かったため、トランジスタに 5.1kΩ抵抗を挿入してデジタルトランジスタと同様の回路とし、ノイズをカットした。(別資料２)

これにより動作できるようになったため、これらの回路をユニバーサル基板上にまとめ、3D プリンターでケースを作成して完成とした。

5. 結果

かさばる機器と長いケーブルを必要としないパドルとして使用可能な状態になった。

しかし結局ケーブルを使わなければならず、単 3 電池 3 本を使用することで重量が増え、またスマートフォンと端末の間で通信にラグが生じることがあった。

6. 考察

電源の重量の問題は小型電子機器用のリチウムイオンバッテリーを用いることで解決できる。

動作のラグは Wi-Fi などの無線接続においては必ずと言っていいほどついて回る課題であり、より高速通信が可能な規格を用いることで解決できる可能性がある。

7. まとめ・展望

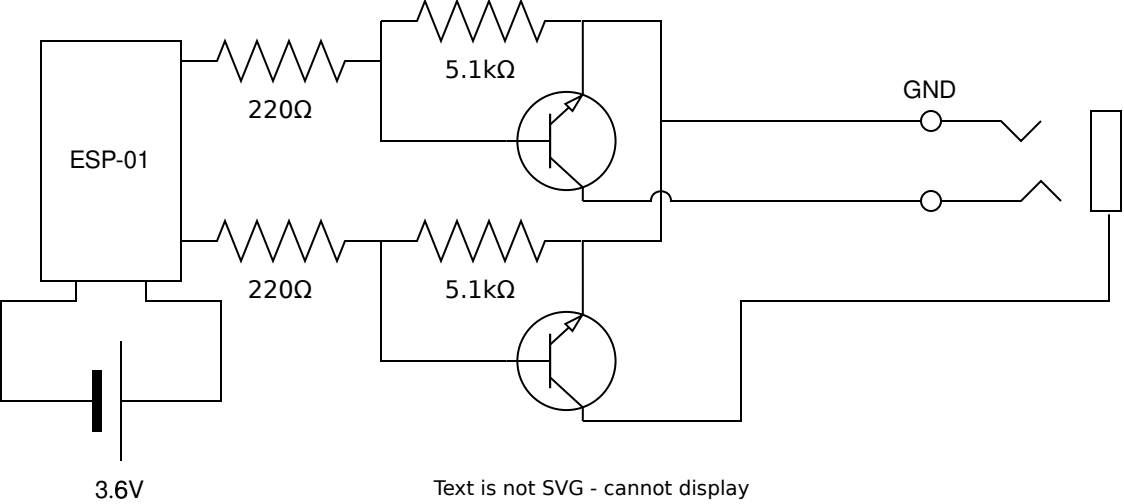
パドルのスマートフォンによるワイヤレス化は手軽に CW に手を出せる一つの手段として有効と考えられる。以降は本研究で解決しきれなかったラグ、電源の問題に加え、シリアル通信によるキーボードからの文字直接入力や縦振り電鍵モードの追加に挑戦したい。

8. 参考文献

CQ Ham Radio 2021 年 2 月号,CQ 出版社

ぶらり Web 走り書き,https://burariweb.info/electronic-work/esp01-esp8266-how-to-use.html(最終閲覧 2023/11/30)

基礎からの IoT 入門, https://iot.keicode.com/esp8266/esp-01-program.php(最終閲覧 2023/12/02)



Text is not SVG - cannot display

```
/*
   -- New project --

   This source code of graphical user interface
   has been generated automatically by RemoteXY editor
   To compile this code using RemoteXY library 3.1.11
or later version
   download by link http://remotexy.com/en/library/
   To connect using RemoteXY mobile app by link
http://remotexy.com/en/download/
     - for ANDROID 4.11.4 or later version;
     - for iOS 1.9.1 or later version;

   This source code is free software; you can
redistribute it and/or
   modify it under the terms of the GNU Lesser General
Public
   License as published by the Free Software
Foundation; either
   version 2.1 of the License, or (at your option) any
later version.
*/

//////////////////////////////////////////////
//        RemoteXY include library          //
//////////////////////////////////////////////

// RemoteXY select connection mode and include library
#define  REMOTEXY_MODE__ESP8266WIFI_LIB_POINT
#include <ESP8266WiFi.h>

#include <RemoteXY.h>
```

```
// RemoteXY connection settings
#define REMOTEXY_WIFI_SSID "SmartPaddle"
#define REMOTEXY_WIFI_PASSWORD "12345678"
#define REMOTEXY_SERVER_PORT 6377


// RemoteXY configurate
#pragma pack(push, 1)
uint8_t RemoteXY_CONF[] =   // 42 bytes
  {
255,6,0,0,0,35,0,16,62,1,1,1,35,46,22,40,8,31,100,97,

115,104,0,1,1,6,46,22,40,8,31,100,111,116,0,8,0,17,9,28
,
  28,16 };

// this structure defines all the variables and events
of your control interface
struct {

    // input variables
  uint8_t button_1; // =1 if button pressed, else =0
  uint8_t button_2; // =1 if button pressed, else =0
  float compass_1; // from 0 to 359.999

    // other variable
  uint8_t connect_flag;  // =1 if wire connected, else
=0

} RemoteXY;
#pragma pack(pop)

/////////////////////////////////////////////
```

```
//            END RemoteXY include          //
/////////////////////////////////////////////

#define DASH_BUTTON 2
#define DOT_BUTTON 3


void setup()
{
  RemoteXY_Init ();

  pinMode (DASH_BUTTON, OUTPUT);
  pinMode (DOT_BUTTON, OUTPUT);

  // TODO you setup code

}

void loop()
{
  RemoteXY_Handler ();

   digitalWrite(DASH_BUTTON, (RemoteXY.button_1==0)?
LOW:HIGH);
   digitalWrite(DOT_BUTTON, (RemoteXY.button_2==0)?
LOW:HIGH);

  // TODO you loop code
  // use the RemoteXY structure for data transfer
  // do not call delay(), use instead RemoteXY_delay()


}
```