

DSCW 用 Arduino UNO R3 を使ったインタフェースの作り方

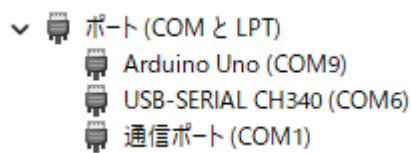
Arduino UNO R3 は、プログラム次第でモニター音も発生できます。

USB ケーブル接続するだけで CW 送信が出来ないトランシーバーをお使いの局長さんとか、PTT 制御をして、プリアンプ、リニアアンプなどを別途使いたい局長さん、その他、CW フルブレイクインにしている、送信中はリレーがバタバタしなくて、送信が終われば素早く受信に切り替えられる、、、などの使い方が楽しめるインタフェースです。

Arduino UNO を使うには、最初に Arduino IDE をダウンロードしてインストールします。インターネット上で、ダウンロードの仕方、インストールの仕方、そして USB ドライバーのインストールなどを多くの方が詳しく書いて居られますので検索してみてください。あるいは、Arduino の入門書もたくさん出ていますから、1 冊お買い求めの上勉強してみてください。

USB ドライバーに関しては、Arduino UNO R3 正規版をお使いの場合は、名前: Arduino USB Driver、発行元: Arduino SA」をインストールください。
正規品なら、Arduino UNO(COM*)と表示されます。

中国/台湾などの製品は CH340 とデバイスマネージャで表示されることもあります。



CH340 は、主に中国の WCH (Nanjing QinHeng Electronics) が製造している USB-シリアル変換チップです。このドライバーをダウンロードするためには、WCH の公式サイトを訪れるのが最も確実です。以下に、具体的な手順を示します。ただし、最近では、Windows に認証されたのか、USB を差しただけで、ドライバーが自動的に入ることもあります。入らねば、、、

WCH 公式ダウンロードページ

[search CH340 - NanjingQinhengMicroelectronics \(wch-ic.com\)](http://search.CH340-NanjingQinhengMicroelectronics(wch-ic.com))

「CH340 シリーズ」のドライバーがリストに表示されます。

自分の使用しているオペレーティングシステム (Windows) に対応するドライバーを選びます。

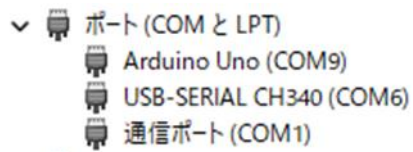
ダウンロードリンクをクリックしてドライバーをダウンロードします。

ダウンロードしたファイルを解凍し、インストーラを実行してドライバーをインストールします。

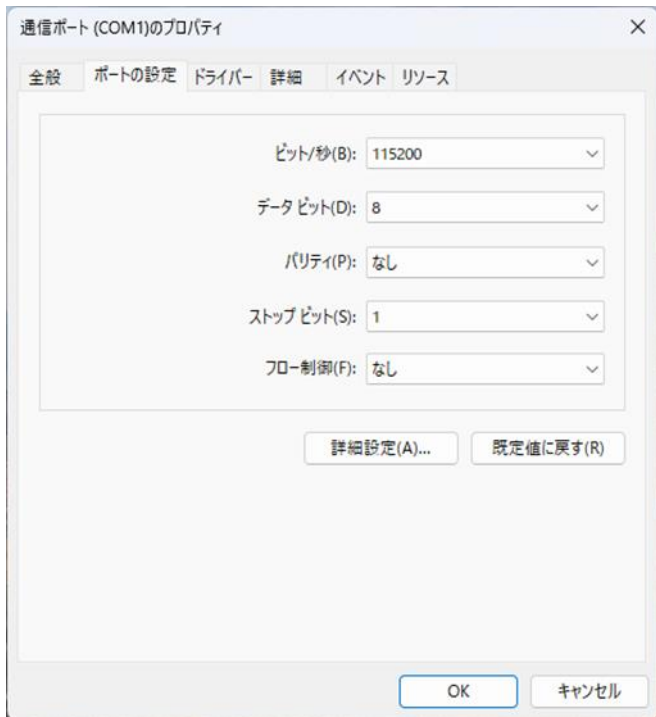
インストールが完了したら、Arduino UNO を PC に接続し、正しく認識されるか確認します。

上記の手順に従うことで、CH340 ドライバーを正しくインストールし、Arduino UNO を使用できるようになるはずです。

次に PC のスタートボタンを右クリックしてデバイスマネージャを選択し、



上記の Arduino Uno(COM*)または USB-SERIAL CH340(COM*)を右クリックしてプロパティを選択します。 画面のポートの設定タブをクリックして



上記の様にセット出来たら、OK を押して、デバイスマネージャは閉じておきます。

スケッチの書き込み

Arduino UNO R3 にスケッチ（プログラム）を書き込むには IDE を起動してください。

そして、以下のスケッチを総てコピーして、そのまま IDE のスケッチ枠に張り付けてください。

なお、COM* の COM 番号を覚えておいてください。後で IDE 設定に必要です。

```
#include "Arduino.h"
```

```
//definePort
```

```
#define PortKey 9
```

```
#define PortBuzzOut 11
```

```
#define PortCWOut 7
```

```
#define PortPTTOut 5
```

```
#define LED 13
```

```
#define SpaceBase 5000
```

```
//define User Memory
```

```
static unsigned char FlagBuzz = false;
```

```
static unsigned int SigLen = 0x0000;
```

```
static unsigned int SpaceLen = 0x0000;
```

```
static unsigned char SigFlag = false;
```

```
//int Value;
```

```
char input;
```

```
//_____
```

```
// Buzzer 1msec
```

```
//_____
```

```
void buzz_1msec(void) {
```

```
    static boolean sts = LOW;
```

```
    if (FlagBuzz) {
```

```
        sts = !sts;
```

```
        digitalWrite(PortBuzzOut, sts);
```

```

    } else {
        digitalWrite(PortBuzzOut, LOW);
    }
}

//-----
// Here be interrupt function
//-----
void interrupt_1msec(void) {

    buzz_1msec();
}

void setup() {
    // put your setup code here, to run once:
    Serial.begin(115200); //,SERIAL_8N1);
    while (!Serial)
        ; // 準備が終わるのを待つ
    Serial.println("開始"); // シリアル通信でメッセージを PC に送信
    pinMode(PortBuzzOut, OUTPUT);
    pinMode(LED, OUTPUT);
    pinMode(PortKey, INPUT_PULLUP);
    pinMode(PortCWOOut, OUTPUT);
    pinMode(PortPTTOut, OUTPUT);
    digitalWrite(PortCWOOut, LOW);
    digitalWrite(PortPTTOut, LOW);
    FlagBuzz = false;
    noTone(PortBuzzOut);
}

boolean ChkKeyPort(void) {
    if (digitalRead(PortKey) == LOW) {
        return true;
    } else {
        return false;
    }
}

```

```
}  
}
```

```
void loop() {  
  // put your main code here, to run repeatedly:  
  if (Serial.available() > 0) {  
    //digitalWrite(LED, HIGH);  
    input = Serial.read();  
    if (input == 'a') {  
      tone(PortBuzzOut, 600);  
      digitalWrite(PortCWOOut, HIGH);  
    } else {  
      if (input == 'b') {  
        digitalWrite(LED, LOW);  
        noTone(PortBuzzOut);  
        digitalWrite(PortCWOOut, LOW);  
      } else {  
        if (input == 'c') {  
          tone(PortBuzzOut, 780);  
          digitalWrite(PortCWOOut, HIGH);  
        }  
        else {  
          if (input == 'd'){  
            digitalWrite(PortPTTOut, HIGH);  
          }  
          else {  
            if (input == 'e'){  
              digitalWrite(PortPTTOut, LOW);  
            }  
          }  
        }  
      }  
    }  
  }  
}
```

```

if (ChkKeyPort()) { // キーが押されていた
    digitalWrite(LED, HIGH);
    if (SigFlag == false) { // その前はキーは押されていなかった
        if (SpaceLen > 4) {
            Serial.println(SpaceLen + SpaceBase); //信号データーかスペースなのか判定のため
5000 加算
            Serial.flush();
            SigFlag = true;
            SigLen = 1;
            FlagBuzz = true;
            tone(PortBuzzOut, 600);
            digitalWrite(PortCWOOut, HIGH);
        } else {
            SpaceLen++;
        }
    } else { // キーは連続して押されている
        SigLen++;
        if (SigLen > 5000) SigLen = 4000;
    }
} else { // キーは押されていなかった
    digitalWrite(LED, LOW);
    if (SigFlag == true) { // その前は押されていた
        if (SigLen > 4) {
            Serial.println(SigLen);
            Serial.flush();
            SigFlag = false;
            SpaceLen = 1;
            FlagBuzz = false;
            noTone(PortBuzzOut);
            digitalWrite(PortCWOOut, LOW);
        } else {
            SigLen++;
        }
    } else {

```

```

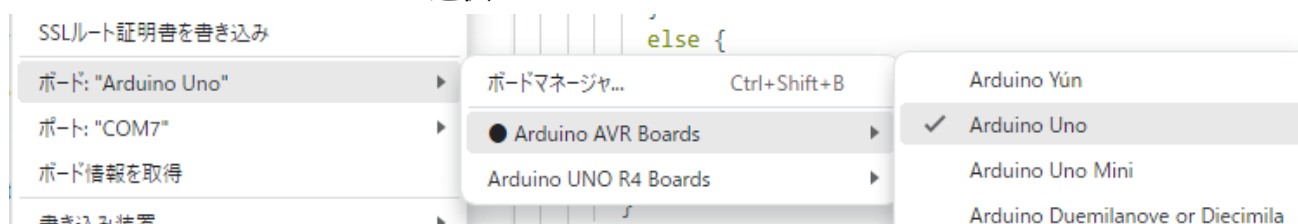
    SpaceLen++;
    if (SpaceLen > 5000) SpaceLen = 4000;
  }
}
delay(2);
}

```

貼り付けたら、Arduino UNO R3 を USB 接続してください。

次に IDE の設定を行います。

IDE のメニューからツールを選択し



上記の様に Arduino Uno にチェックマークを付けます。

次に、ボード“Arduino Uno”の下のボードを選択し、シリアルポートの画面から、今繋いでいる COM 番号を選択します。

次いで、IDE メニューのツールから、シリアルモニターをクリックします。



上記右側のプルダウンコンボボックス下向き矢印をクリックして、上記の様に 115200baud を選択します。

次に、メニューのスケッチから書き込みを選択して、あらかじめコピー済みのスケッチ（プログラム）をコンパイルして、Arduino UNO に書き込みが始まります。

書き込み完了が出たら、IDE は終了してかまいません。

以上で、スケッチ（プログラム）が Arduino UNO R3 の準備が出来ました。

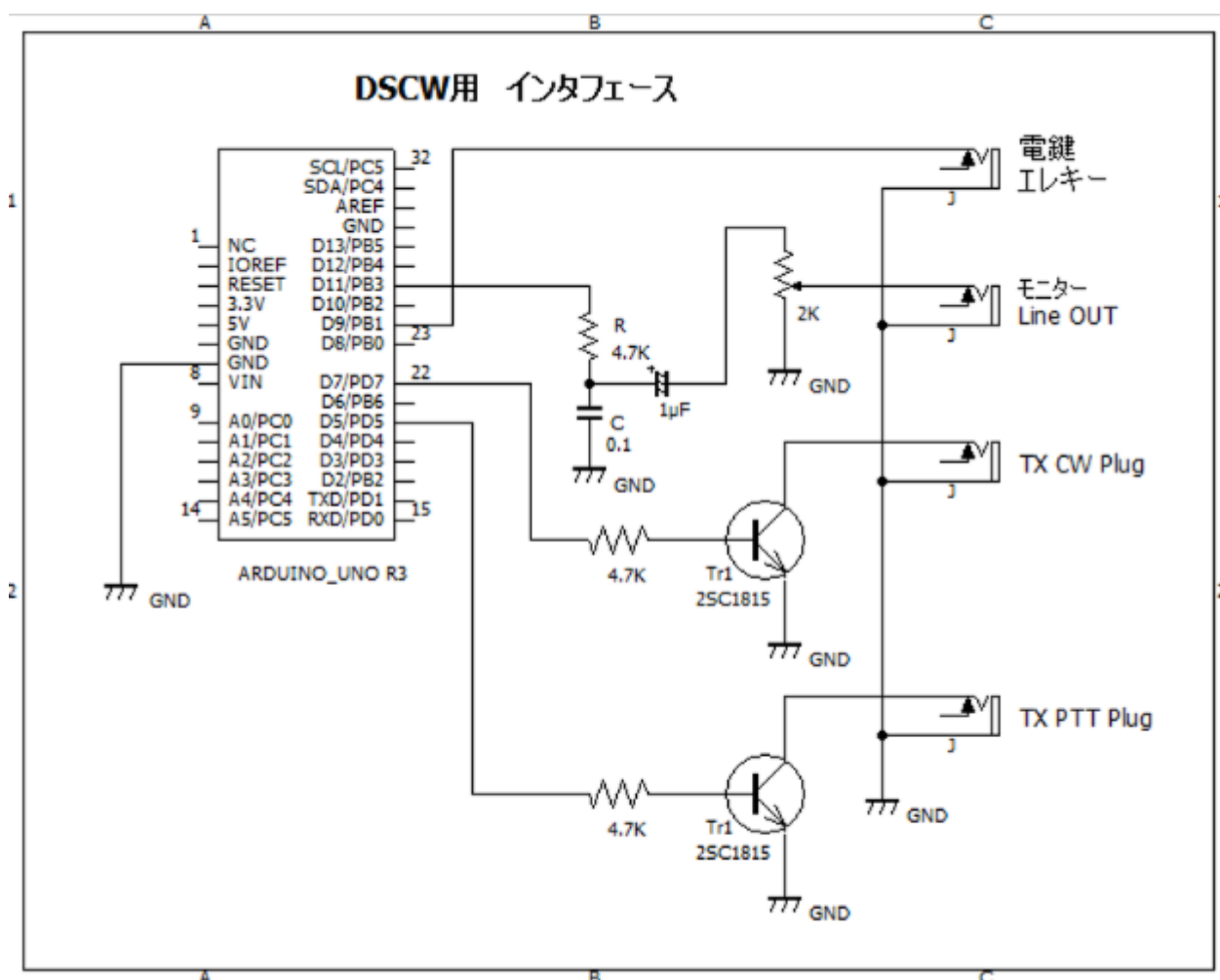
次に、配線作業に移ります。

配線図は、以下の様に簡単です。 この通り配線してください。

なお、PTT 制御が要らない方は、一番下の TX PTT Plug への配線は不要です。

既にお気づきかとは思いますが、PTT 制御は、DSCW 側のオプション画面で PTT を設定して、DSCW 側から送信操作した時だけ有効です。

下図の電鍵操作をした時は PTT の時間差動作をしませんので、ご注意ください。



以上。

By JA3CLM
May 2024