ATTiny202をArduinoで使用する

ATTiny202というAVRマイコンが秋月電子で40円で売っていた（今は値上がりしている□□ESP8266とかのSoCはともかく、昔ながらのライターで書き込む系のものはPICマイコンしか基本使ったことが無いが、これをArduinoでプログラムできる方法があるようだ。電源は1.8V□5Vなので何にでも使えるし、簡単なものならこれで良いのかもしれない。

使い方については小坂先生が懇切丁寧に記事にしてある。 Arduino IDE に ATtiny202 他の開発環境を組み込む | Make | kosakalab

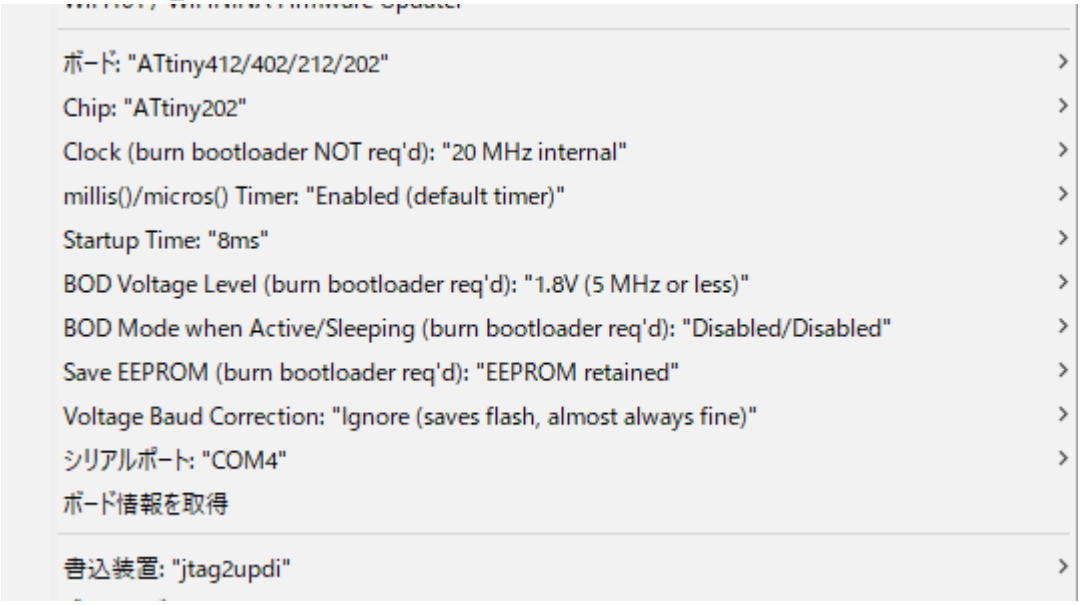
jtag2updi□ArduinoをUPDIライターに改造する）の方法と、ATTiny202のピン配について Arduino勉強会/36-jtag2updiで最新のATtiny202を使う - PukiWiki

マイコン自体をソフトリセットできないか、検証する。 Arduinoでソフトウェアからリセットをかける方法 | なんでも独り言

24時間ごとに電源をリセット

USB電源のWebカメラがあるが、どうも暫く運用していると固まって動かなくなる。対処方法として、一度コンセントを抜いて、数秒経った後に再度接続することで解決する。が、実家に設置している場合、そもそもコンセントから抜くという行為ができない。そこで□USB電源をMOSFETで制御して、毎日1回リセットを実施する。

SFRというか□Arduinoの設定は以下の通り。



AT202_24hReset.ino

```
int LED = 0;
int SW = 3;
int var;

void setup() {
  // put your setup code here, to run once:
  pinMode(LED, OUTPUT);
```

```
  pinMode(SW, OUTPUT);
 }

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(SW,LOW);
  digitalWrite(LED, LOW);

  //30 seconds
  var = 0;
  while(var < 30){
     delay(850);
     digitalWrite(LED, HIGH);
     delay(150);
     digitalWrite(LED, LOW);
     var++;
  }

  digitalWrite(SW,HIGH);
  //24 hour minus 30 second
  var = 0;
  while(var < 8637){
    delay(9500);
    digitalWrite(LED, HIGH);
    delay(500);
    digitalWrite(LED, LOW);
    var++;
  }
}
```

書き込み時に下記のエラーが表示されるが、正常に書き込めている模様(2021/12/04)

```
avrdude: jtagmkII_initialize(): Cannot locate "flash" and "boot" memories in
description
```

起動時間を1分遅延させる

家庭用のインターネットに接続しているルーターなどの通信機器は、電源を入れた時には最も早い通信速度で通信を行い、何度かエラーをしたり、通信できなかったりすると段階的に通信速度を落として再試行する（これをベストエフォートと呼ぶ）。この何度かエラーしたりというのは、普段は発生しない一時的なノイズで通信速度を低下することも含まれるので、本来安定して通信できる速度以下になったとしても、電源を入れ直さない限り、低下した通信速度を戻すことはない。 なのでインターネットを常に快適に使用したい場合は、通信機器の電源を定期的にOFF/ONする必要がある。昔はオーディオタイマーという高級なものを使わないと実現できなかったけど、最近では節電志向の影響か、プログラムタイマーと呼ばれるものがいくつも市販されているので、その点はどうとでもなる。

ところがインターネットの接続が、ごくまれに不安定な時があって困っていた。繋がらないならまだしも、繋がるけど自宅サーバーが見えない（外から内が見えない）という、なんとも不思議な不具合である。それを通信機器に詳しい方に相談したところ、もしかしたらモデムとルーターを同時に電源を入れていて、ハンドシェイクに失敗しているのが原因かもしれない、だからルーターの電源だけを1分程度

遅らせれば改善するかもしれないとの事だった。そのためだけに新しくプログラムタイマー買って入れるのも何なので、当基板で代用しようと考えている。

Arduinoの設定は前回の記事を参照。24時間リセットとやっていることは同じだが、動作の識別ができるように、無駄にデバッグ用LEDを点灯・消灯したりするようにしている。本当はPWMで蛍のような明滅をさせたかったが上手くいかず、結局データシートよりATTiny202の2ピンはPWMの指定ができない事に気が付いた。適当に作った罰である。

AT202_BootDelay.ino

```
int LED = 0;
int SW = 3;
int var;

void setup() {
  // put your setup code here, to run once:
  pinMode(SW, OUTPUT);
  pinMode(LED, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(SW, LOW);
  digitalWrite(LED, LOW);
  var = 0;
  //60 seconds
  while (var < 60) {
   delay(850);
   digitalWrite(LED, HIGH);
   delay(150);
   digitalWrite(LED, LOW);
   var++;
  }

  digitalWrite(SW, HIGH);

  //Lights off and on as identification during operation.
  while (1) {
    digitalWrite(LED, HIGH);
    delay(2850);
    digitalWrite(LED, LOW);
    delay(100);
    digitalWrite(LED, HIGH);
    delay(100);
    digitalWrite(LED, LOW);
    delay(100);
  }
}
```

LEDのPWM点灯制御

LEDを使用する場合、何らかの電流制限を行わないと、過電流でLEDが焼けて点灯しなくなる。一般的には電圧に合わせた正しい抵抗をつけるか、CRD（定電流ダイオード）で電流そのものを一律で制御することになるが、COB等の照明に使用する高輝度LEDの場合は、抵抗だと熱量が多すぎて無駄が生じ、CRDでは小電流のみなので大量に付けるとコストがかかるので、PWMによる制御が一般的となる。

なのでAliExpressで探すと色々な照明用のPWMコントローラが取り扱っているが、RFリモコン操作専用だったり、赤外線の距離センサを使ったスワイプによる非接触スイッチとかで、単に線でスイッチを伸ばしてONOFFしたい時用のものが無い。またボリュームがついて明るさを調整できるものならあるが、使用するには都度回さなくてはいけない。輝度は一定で良いので、設定も頭打ちで良い。

・・・と色々文句を言っていると、結局自分で作るしかなくなったので、当基板で実現することにした。使用するCOBによって流せる電流量がまちまち（暗すぎたり、放熱が追い付かず焼けたりする）なので、可変抵抗をADCで読み取り、デューティー比に変換する事で、PWMを適当に調整できるようにした。スイッチは押しボタンスイッチではなく波動スイッチにした。それのほうが自分にとって自然だからだ。

例によってSFRは前の記事を参照すること。

AT202_LightSW.ino

```
int LED = 0;
int POT = 1;
int SW = 3;  //MOSFET
int TRIG = 2; //Switch

void setup() {
  // put your setup code here, to run once:
  pinMode(POT, INPUT);
  pinMode(LED, OUTPUT);
  pinMode(SW, OUTPUT);
  pinMode(TRIG, INPUT_PULLUP);
}

void loop() {
  // put your main code here, to run repeatedly:
  if ( digitalRead(TRIG) == LOW) {
    digitalWrite(LED, LOW);
    float v = analogRead(POT);
    analogWrite(SW, v / 1023.0 * 255.0);
    } else {
     analogWrite(SW, 0);
     digitalWrite(LED, HIGH);
    }
}
```

ところで、たったこれだけの処理でATTiny202は以下のリソースを消費する。これはArduinoの関数を使用しているからオーバーヘッドが過ぎるのか、megatinycoreが最適化されていないからか、そもそもATTiny202の性能はこの程度なのかは不明だが、まあ大した事をする予定も無いので、良しとしよう。それよりもMPLABXで同じプログラムを作成した場合、どれだけリソースを消費しているのか気になる。

(2021/12/10)

最大2048バイトのフラッシュメモリのうち、スケッチが1686バイト（82%）を使っています。
最大128バイトのRAMのうち、グローバル変数が10バイト（7%）を使っていて、ローカル変数で118バイト使うことができます。

追伸：2025年10月現在、ATTiny202は70円まで値上がりしたが、ATTiny402というのが発売され、こちらも同じく70円で発売してる。中国製のCH32Vシリーズも安くて気になるが、量産しない限り、こちらで作ったほうがまだ早いと考えられる。