

3種の CPU基板による Lチカの目的

今回 何故、3種の CPU基板による Lチカの
実験を行うのか不思議に思われた方も、おら
れるのではないかと思います。その目的は 2
つ あります。

① それぞれのマイコンの開発環境を使って、
1-1 プロジェクト(プログラム開発の仕事の単
位)を作成する。

1-2 コンパイル作業を行う。

1-3 マイコンにプログラムを 書き込みする。

1-4 マイコンプログラムを 実行する。

これらの手順を 視聴者の皆様に
お見せする事。

② Lチカのプログラムの動作は、LEDを点灯
または、消灯させた後、しばらく間を置いて、ま
た、LEDを 反転状態で 点灯、消灯させ、これ
を 繰り返します。で、今回の実験では、
しばらく間を置いて の 部分が、重要になりま
す。 ということかということ、通常 Lチカでは
タイマー機能を使って指定した待ち時間を作り
ます。

今回は、そうではなくて、単純にプログラムの
ループで、時間待ちを 作ります。

具体的には、空の for 文を 60000回、回して
時間稼ぎを行います。

```
for( i=0; i<60000; i++ );
```

上記のコーディングで時間待ちをさせます。

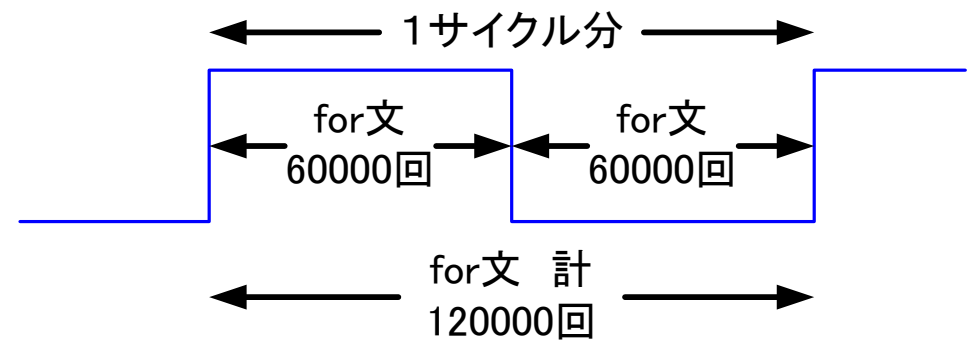
こうする事により、各CPUの処理能力の差が
待ち時間の差となって現れます。

その時間を、LED出力端子で 計測します。

LED出力端子で 計測した時間を 元に、各CPUのクロック周波数の違いを 計算で補正して、同一クロック周波数の、待ち時間に変換して、各CPUの処理能力を、比較したいと思います。 但し、今回の実験で得られたデータは、各CPUが持つ処理能力の 一部分と考えて下さい。

それと、オシロのリードアウト測定に関わる事ですが、LED出力を測定すると デューティ50の 方形波となります。 今回 周期と、周波数を 計っていますが、方形波の 1サイクル分の 値となります。（ 右図を参照の事 ）

よって計測値は、for 文が 120,000回、回った時の時間です。



では、for 文 60000回 × 2の 時間待ちによる Lチカの周期、周波数を計る動画に移ります。

Lチ力周期、周波数測定 1

R8C/M110A

クロック = 20 MHz

周期 = 125.6 ms

周波数 = 7.962 Hz

Lチ力周期、周波数測定 2

R8C/M110A

クロック = 20 MHz

周期 = 125.6 ms

周波数 = 7.962 Hz

PIC12F635

クロック = 4 MHz

周期 = 1.96 sec

周波数 = 0.5102 Hz

クロック = 8 MHz

周期 = 956 ms

周波数 = 1.046 Hz

20 MHz 補正計算値
(8MHz × 2.5倍)

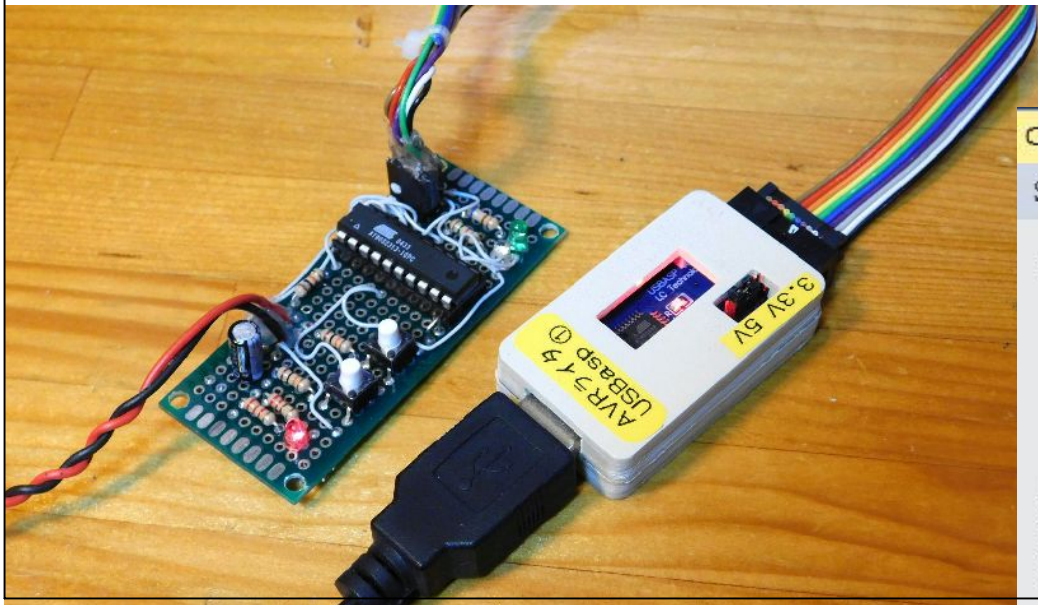
周期 = 382 ms

周波数 = 2.615 Hz

R8Cの方が 約 3倍速い

★ AVR書き込み器に関わる作業_1

あるサイトで「**Atmel Studio + USBasp + avrdude**で**AVRに書き込む**」を見つけて、この記事参考に作業を行いましたが、USBaspで書き込む事は出来ませんでした。USBaspのドライバーは、正常にインストール出来たようです。AtmelStudio上で書き込むとすると、右下のエラーが出ます。vid、pidが認識出来ないというようなエラーです。？



```
Output
Show output from: USBasp ATtiny2313

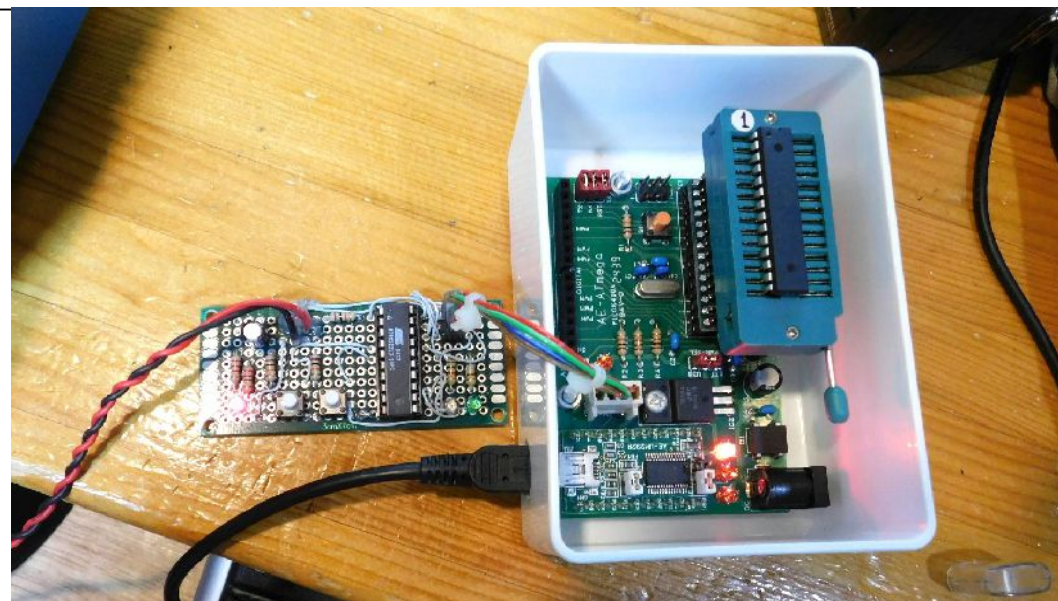
avrdude.exe: Version 5.10, compiled on Dec 19 2010 at 22:24:28
Copyright (c) 2000-2005 Brian Dean, http://www.bdmicro.com/
Copyright (c) 2007-2009 Joerg Wunsch

System wide configuration file is "G:\AVRdude\bin\avrdude.conf"

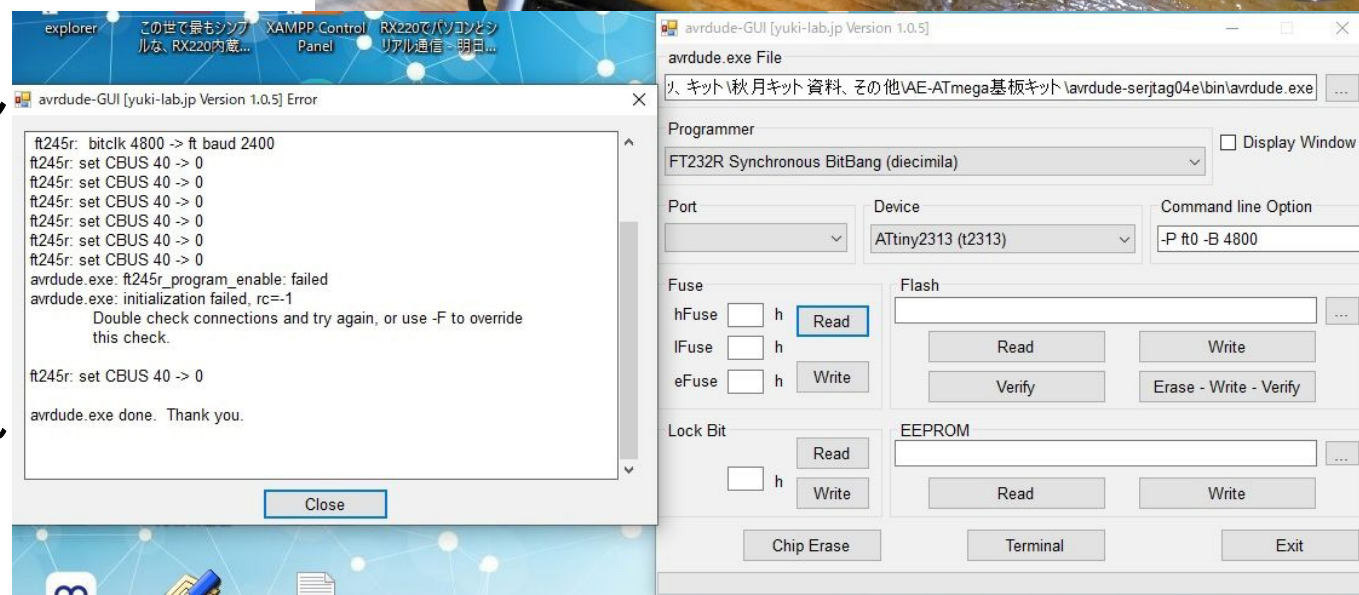
Using Port                : usb
Using Programmer           : usbasp
USBASP: vendor www.fischl.de prod USBasp
USBASP: vendor www.fischl.de prod USBasp
avrdude.exe: error: could not find USB device "USBasp" with vid=0x16c0 pid=0x5dc
|
```


★ AVR書き込み器に関わる作業_2

遥か昔、秋月電子で **AVRキット基板** を購入して組み立てました。そのキット基板は **Arduino UNO 基板** と互換性を持たせた物であると、同時に **FTDI USB-シリアルの高機能モジュール小基板** を付けてあり、シリアル通信以外に、数ビットの **ビットIOの機能** を持っています。その機能を使って **BitBang Mode AVR-Writer** なる AVR 書き込み器を実装しています。



この機能を使って、AVRマイコン **AT90S2313** に、書き込みを行う事にしました。結果からいうとまた 失敗でした。右の画像の左側のフォームが エラー表示で **program_enable: failed** と表示されています。



★ ATMEGA328Pに 書き込んだら どうか？

しばらく茫然と 秋月の Arduino UNOの基板を見ていたら、上に乗っているATMEGA328Pに、目が とまりました。

今回の Lチカプログラムを、ATMEGA328Pに書き込んだらどうなるか、試してみよう。という事に なりました。 ATMEGA328Pは秋月電子で 340円で、百円マイコンでは ありませんが、データシートを見ると、周辺回路は多数実装されてますが、CPUコアは、同じ物と思われるので、Lチカの 待ち時間比較が、出来ると、考えました。

という事で、AVRマイコンの代表として、ATMEGA328Pの動画を 取る事にしました。

Lチカ周期、周波数測定 3

R8C/M110A

クロック = 20 MHz
周期 = 125.6 ms
周波数 = 7.962 Hz

PIC12F635

クロック = 4 MHz
周期 = 1.96 sec
周波数 = 0.5102 Hz

クロック = 8 MHz
周期 = 956 ms
周波数 = 1.046 Hz

20 MHz 補正計算値
(8MHz × 2.5倍)
周期 = 382 ms
周波数 = 2.615 Hz
R8Cの方が 約 3倍速い

ATMEGA328P

クロック = 16 MHz 水晶
周期 = 30 ms
周波数 = 33.33 Hz

20 MHz 補正計算値
(16MHz × 1.25倍)
周期 = 24 ms
周波数 = 41.7 Hz

R8Cより 約 5.2倍 速い

