

# LabVIEW Community Edition でプログラミングを楽しもう

## 別冊：電気の話 シリーズ その5

### 内容

- ✓ LED の電圧・電流特性を測定するプログラムを自動化したい
- ✓ Arduino の PWM 出力を DC 電圧に変換する

### この本について

- ✓ LabVIEW コミュニティ版を活用するための情報を書き記した e-Book です
- ✓ 「プログラミングを楽しもう」本文から派生した内容+筆者の思いつきからできています
- ✓ わからないところは知っていそうな人に聞くか Web で調べてください（他力本願）



2020 年 7 月 15 日 初版発行

著者：渡島浩健（日本 LabVIEW ユーザ会）  
ワカリヤスイ セツメイハ ムズカシイ

クリエイティブコモンズライセンスにて配布します  
（詳しくは右のバナーをクリック）



## 5. LED の電圧-電流測定を自動化したい

第 6 章で LED の電圧-電流特性を測定するプログラムが紹介されていて、グラフに特性曲線を描いて回帰分析までできます（図 5-1）。ただし必要なデータを集めるために、手で可変抵抗を調節してデータ追加ボタンをクリックする作業を何回も繰り返さなくてはなりません。これを何とか自動的にやってくれるようにしたいと思います。

プログラムからロボット（モータなど）を制御して可変抵抗を回すという方法もありますが、メカを作るのが面倒そうなので、電氣的に LED に流す電流を制御することを考えましょう。（以下口語）

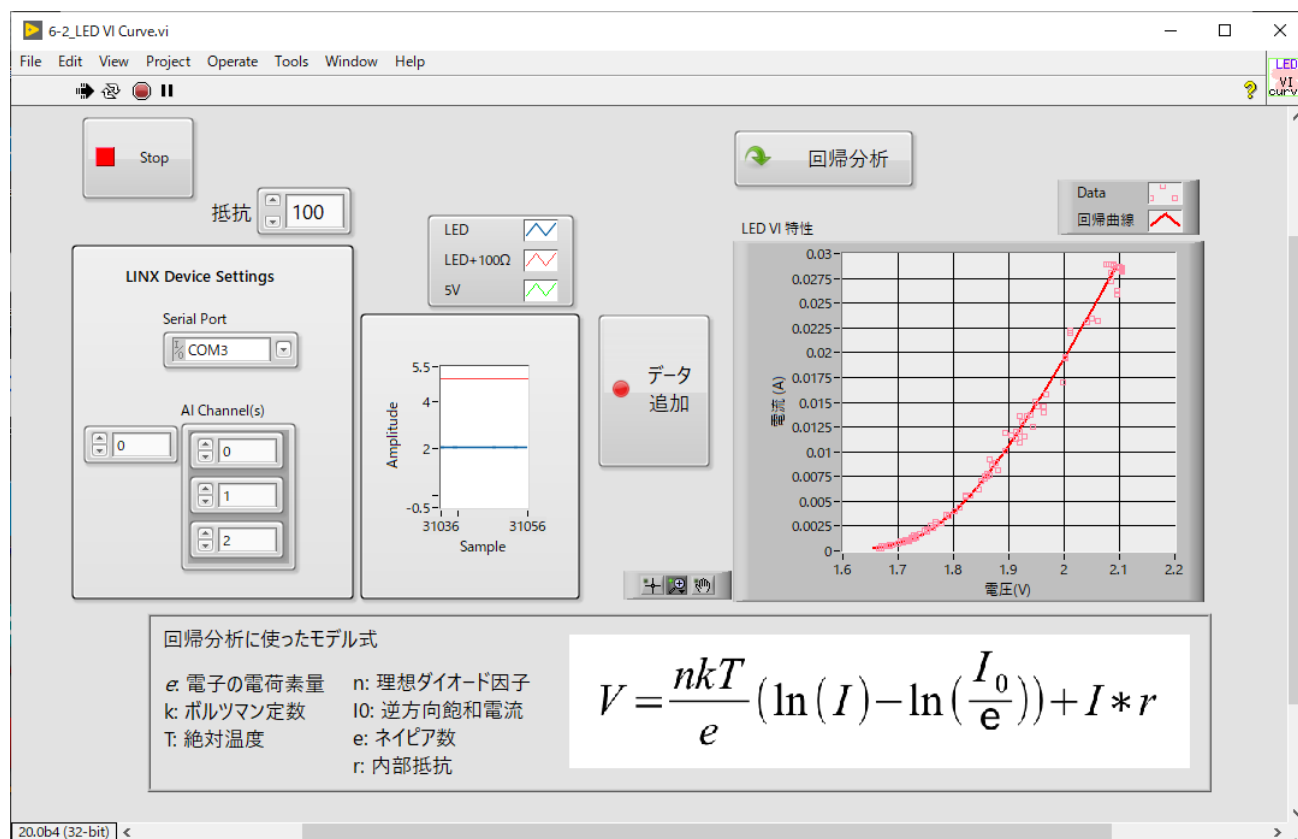


図 5-1 LED の電圧電流特性を測定する VI

「別冊 電気の話シリーズ その4」で、RGB LED を PWM 駆動すると、パルス幅で明るさが変わることを確認した。LED は約 500Hz または 1kHz (Arduino の出力ピンによって違う) の速さで ON と OFF を繰り返して、周期に対する ON の時間幅によって明るさが決まることがわかった。

PWM 波形の平均値は取り込んだデジタルデータから計算できる (図 5-2) が、Arduino UNO のアナログ入力のサンプリング周波数は最大でも 10kHz 程度で、500Hz や 1kHz の PWM だと 1 周期に 20 とか 10 ポイントしかサンプリングできず、正しい平均値が求まらない。さらに「LED をパルス駆動すると DC 駆動するより明るい」という話 (ホンマか?) もあり、手動測定とは違う結果になりそうな気がする。

PWM でない正当 (?) なアナログ出力があればいいけれど、Arduino UNO には備わってない。DAC (D/A コンバータ) モジュール (図 5-3) を追加する手もあるが買うと数 100 円するし、たいてい I2C や SPI インタフェースで制御する仕様だから Arduino のスケッチを組まないとうまく動かせない。DAC の出力は電圧値なので LED の電流を制御するための電流ドライバが必要になる。以上のことから、Arduino の PWM 出力を電子回路で平均化して、LED を電流制御で駆動できるドライバを設計することにしよう。

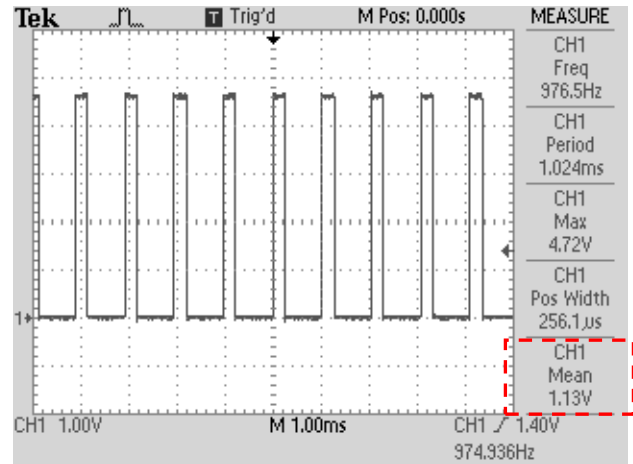


図 5-2 PWM 波形の平均値

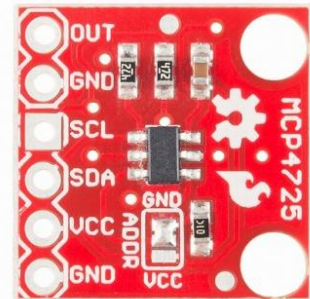


図 5-3 DAC の例

## PWM 波形を DC 電圧に変換する

矩形波をローパスフィルタに通すとエッジがなまる (図 5-4)。さらにフィルタのカットオフ周波数を低くするとなまりがひどくなって、ついには平均値を中心とした DC に近くなる。ただしパルスの ON/OFF でコンデンサの充放電が繰り返されることになるので、リップル (さざ波) が乗る (図 5-5)。

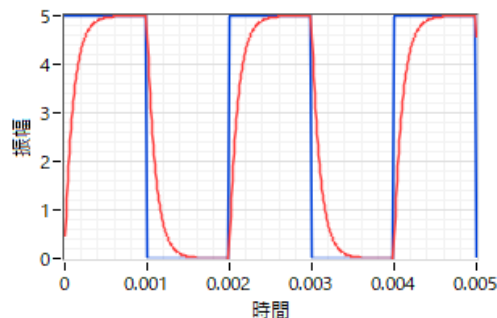


図 5-4 エッジが“なまる”様子

DC（直流）は電池、AC（交流）は家庭のコンセントが代表的だが、DC も細かく見ると変動していて、周期的な変動が混じってれば AC 成分と考えることができる。変動の速さ（業界によって速さの感覚は違う）によって言い分けたりするし、要するに DC と AC の境界はあいまいなのだ…刻を戻そう。

ここで使ったのは抵抗  $R$  とコンデンサ  $C$  だけの最も簡単なローパスフィルタ。「RC フィルタ」で Web 検索すればたくさんヒットするので、性質はそちらで調べてください。ただほとんどがステップ応答にしか言及していないし、すぐラプラス変換

とか持ち出してきやがる（敗北感）。それでも  $R$  と  $C$  の積を時定数と呼び、その値が大きいほどカットオフ周波数は低くなり、出力の変化がゆっくりになることはわかる。

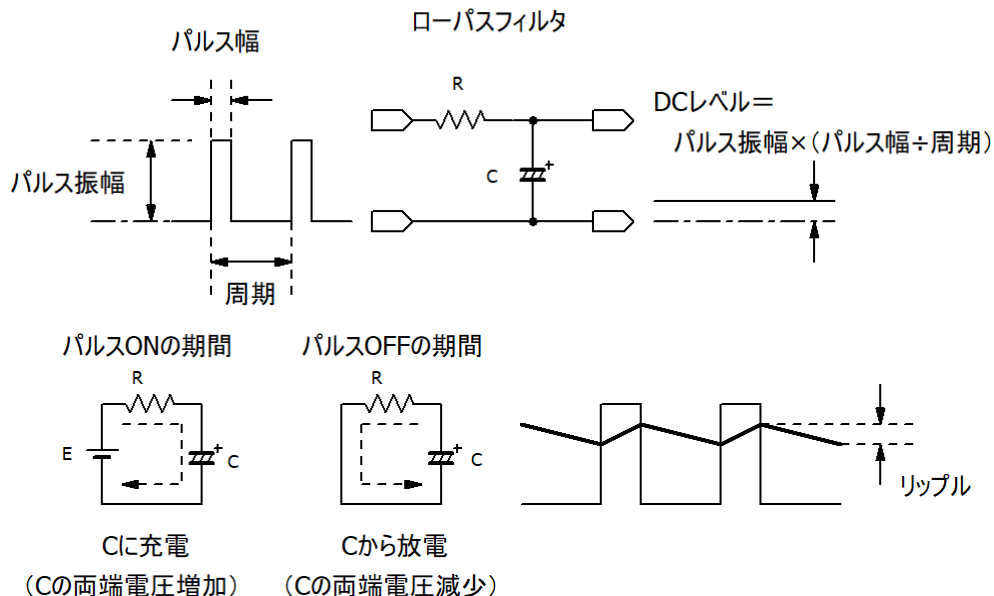


図 5-5 RC ローパスフィルタによる PWM-DC 変換（カットオフ周波数が低い場合）

PWM 波形の生成やフィルタ、グラフ表示は LabVIEW で簡単にシミュレーションできる（図 5-6）。

100k $\Omega$ と 100nF (0.1 $\mu$ F) のとき、出力が定常になるまでに 35ms、リップル 247mV となった。

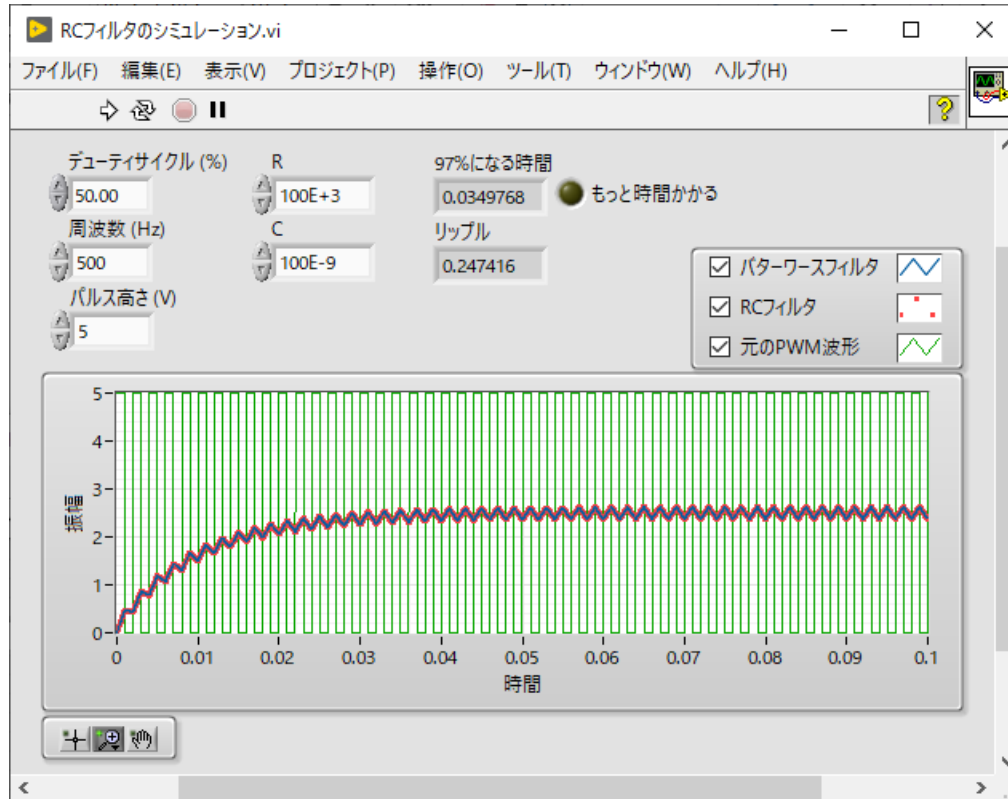


図 5-6 PWM とローパスフィルタのシミュレーション VI

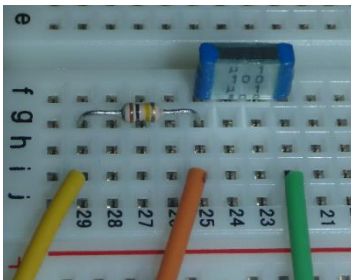


図 5-7 RC フィルタ

実際にブレッドボード上で組み立てて、Arduino の 3 番ピンから PWM を入れ、オシロスコープで観測してみた (図 5-7)。

立ち上がり部分 (左側の図) はシミュレーションの結果とよく合っている。リップルを観測するために垂直レンジを 100mV/DIV にしてカーソルを当てる (右側の図) と、ピークツーピークで 248mV と読める。シミュレーションのリップルは 247mV だからカンペキと言っていいだろう。

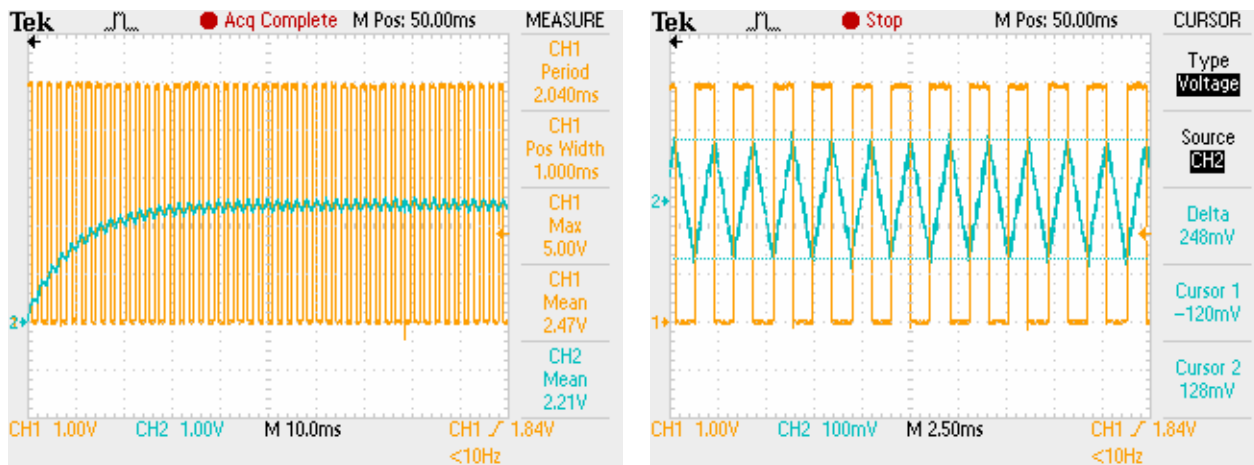


図 5-8 左：時定数 0.01 の立ち上がり部分 右：リップルの測定結果

シミュレーション VI のダイアグラムは図 5-7。

最初に Square Waveform.vi (方形波形) 関数で PWM を作る。サンプリング周波数は 100kHz、ポイント数は 500000 個だか 5 秒間の波形データになる。フィルタは LabVIEW のバターワースフィルタを流用して 1 次ローパスフィルタに設定した (RC フィルタのカットオフ周波数が  $1/2\pi RC$  っていうのはかろうじて憶えてた)。念のため RC フィルタの差分方程式を探してきて計算させてみたら、青 (バターワース) と赤 (RC) のプロットが完全に一致した。1 次フィルタならベッセルでも同じだった。

VI は同梱しておくので適当にいじってもらって OK。

ちなみに、リップルはデューティサイクルが 50% のときに一番大きくなる。シミュレーションでも、実物でもそうなので、フィルタの設計はこの条件下で行うことにする。

時定数を大きくするとリップルは減るが、DC 電圧が定常になるまでの時間 (整定時間とかセトリングタイムと言えなくもない) が長くなる。自動測定では LED 電流を変えてデータを追加することを繰り返すので、毎回の待ち時間が長いと全体の測定時間がかかってしまう。リップルによる測定値のばらつきをある程度許容してなるべく速くしたい。

リップルの条件として考えられるのは Arduino のアナログ入力の分解能だ。UNO は 10 ビットだから

$2^{10}=1024$  レベル。それを入力レンジ  $5V$  に割り当ててあるので、電圧分解能は  $5 \div 1023 = 4.888mV$ 。リップルがこれ未満なら無視できるのではなかろうか。そのときの整定時間をシミュレートして…

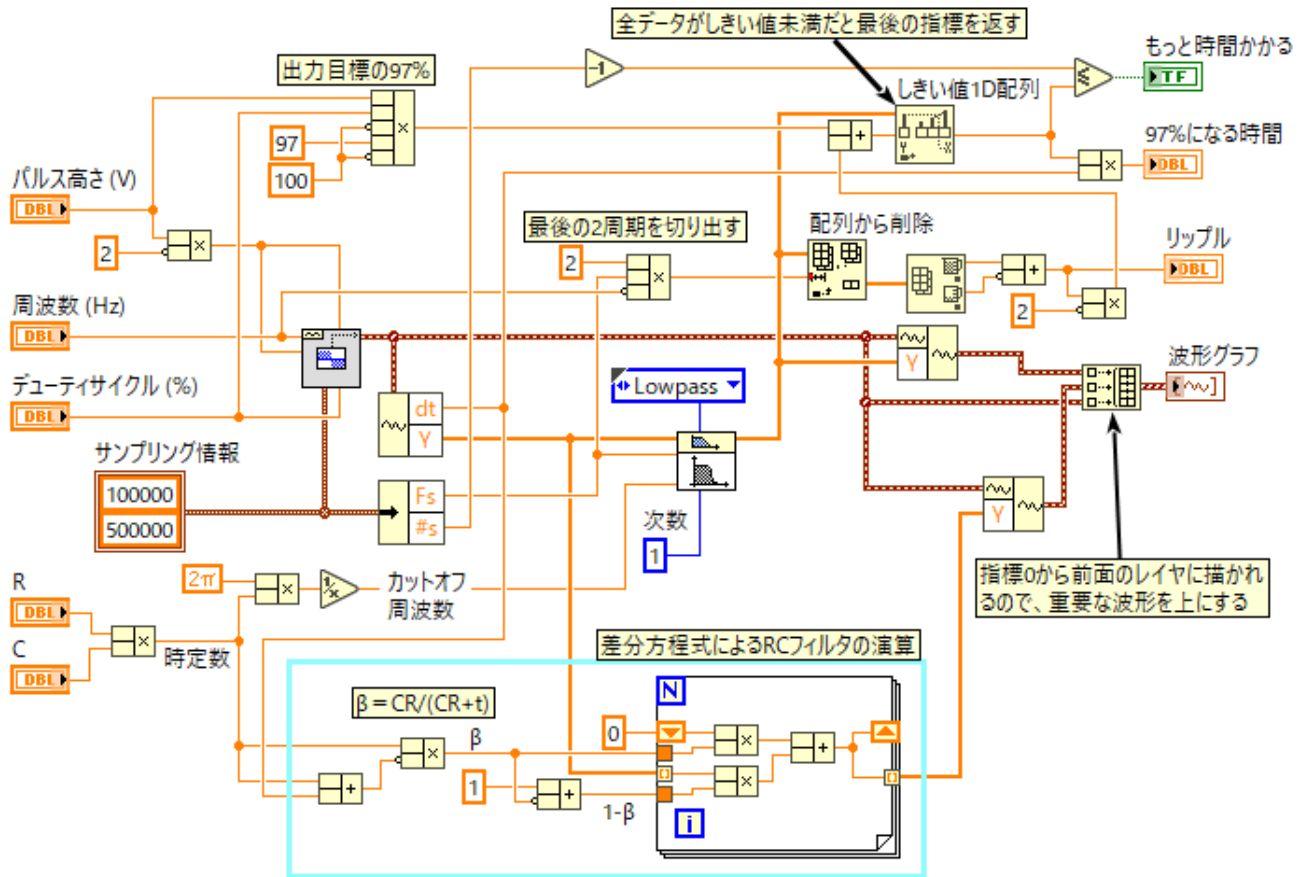


図 5-7 シミュレーション VI のダイアグラム

次号に続く。