

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6

内容

- ✓ PWM を DC 電圧に変換するフィルタの設計
- ✓ 電流制御ドライバの設計

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21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)



6. PWM 波形を DC に変換する RC フィルタ

前回の話を 100 文字以内で述べよ：5V でデューティ50%の PWM を RC ローパスフィルタで平滑したとき、リップルが Arduino のアナログ入力の電圧分解能 4.888mV より小さくなる時定数は？

シミュレーション VI で適当に値を追い込んでみたら、時定数が 0.51 (100k Ω と 5.1 μ F) のときにリップルが 4.854mV になった。しかし DC 電圧が予定の 97%に達するのに 1.8 秒くらいかかる (図 6-1)。

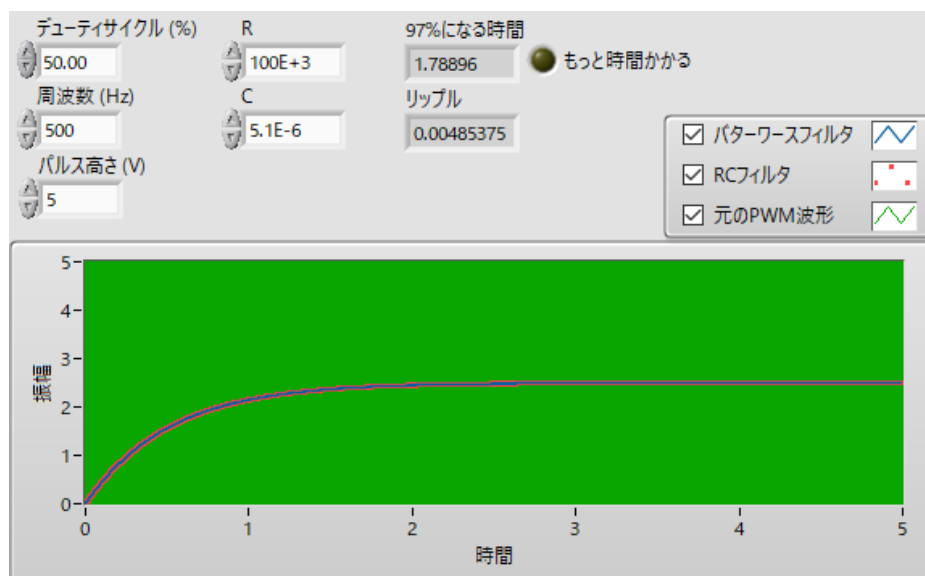


図 6-1 リップルがアナログ入力の分解能未満になる時定数

自動測定で安定時間に少し余裕を持たせて 2 秒待つことにすると、100 個のデータを取るのに 200 秒かかる計算。3分 20 秒かあ…もう少し速くしたいなあ。

デューティ50%は最悪ケースだから、実際に使う電圧範囲によってはもっと小さい時定数で間に合うかもしれない。回帰分析は多少データがばらついてもいいので分解能以下にこだわらなくてもいいだろうし、データとして必要な小数点以下の桁より小さければ無視するということも考えられる。

ちなみにリップルを 10mV くらいまで許すと、時定数は 0.25 で、97%時間も 0.88 秒くらいだ (図 6-2)。

これだと 100 個取るのに 100 秒（1 分 40 秒）だ。

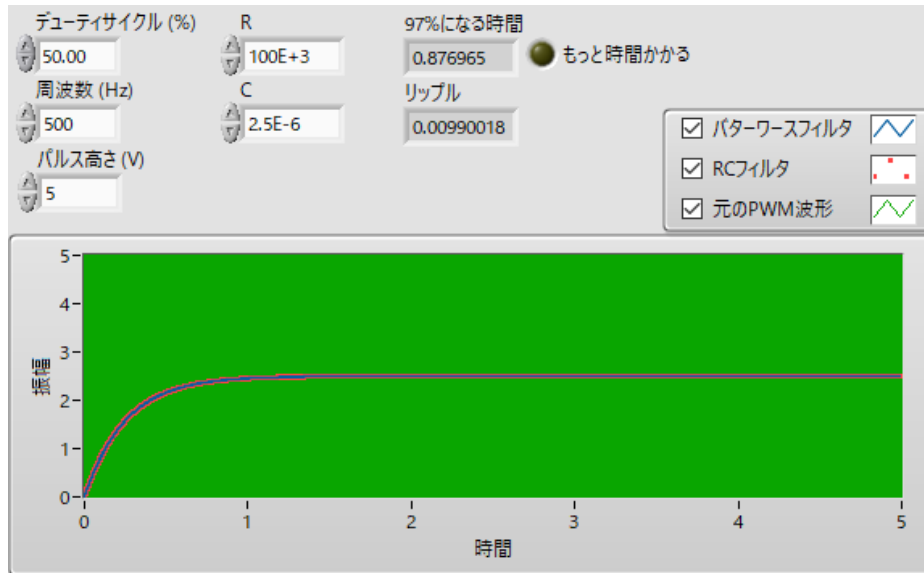


図 6-2 リップル 10mV まで許容した場合の時定数

またデューティを 25%にすると、リップル 10mV なら時定数 0.19 でいける（図 6-3）。

電圧範囲についてはこの後に続く電流ドライバの設計にも左右される。

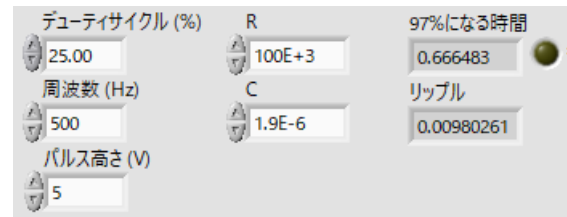


図 6-3 デューティ25%のとき

後段の入力抵抗の影響

フィルタを通して得られた DC 電圧では直接 LED を駆動できない。LED は RC フィルタ回路にとって重い負荷ということだ。だがそれだけではない。フィルタの特性が変わってしまうのだ。たとえば図 6-4 のように、出力に抵抗 (R_{in}) がつながつたとしよう。その結果、

- (1) C を充電する電圧が抵抗分割されて下がる。その分、出力電圧も下がる。

抵抗分割の式は、出力電圧 = 入力電圧 $\times R_{in} \div (R + R_{in})$ だから、 $R = R_{in}$ なら半分になる。

- (2) 時定数が変わってしまう。C を充放電する抵抗値が R と R_{in} の並列接続と等価になるので、時定数が小さくなる。従ってカットオフ周波数は高くなる。

ここでは R_{in} の影響だけしか考えていないが、周波数に関係した特性を扱い始めると、(周波数が高くなると特に) 純粋な抵抗だけでは説明できなくなってきて「インピーダンス」という概念を使うことになる。 $j\omega$ とか微分方程式とかこねくり回すのは嫌なので、実験してみよう。

まず、 $100k\Omega$ と $0.1\mu F$ (時定数 0.01) のオシロスコープ画像は、前号の図 5-8 を見てほしい。その上で、 $50k\Omega$ ($100k\Omega \times 2$ 個並列) と $0.1\mu F$ (時定数 0.005) の波形を確認する (図 6-5)。20ms 弱で定常になり、DC 電圧は 5V デューティー50%の 2.5V、リップルは 496mV となった。シミュレーションともよく合っている。

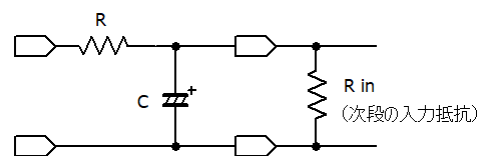


図 6-4 次段の入力抵抗の影響

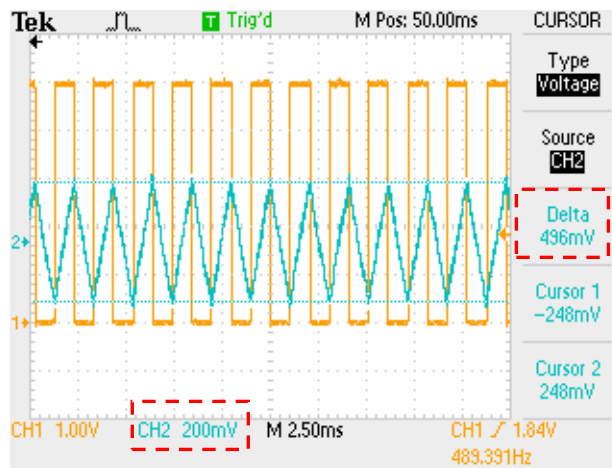
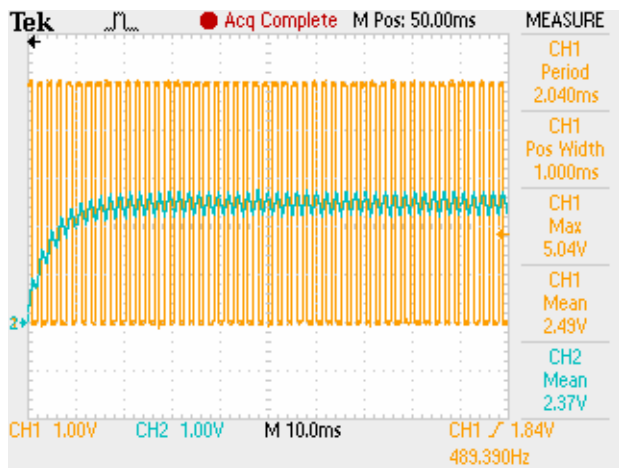
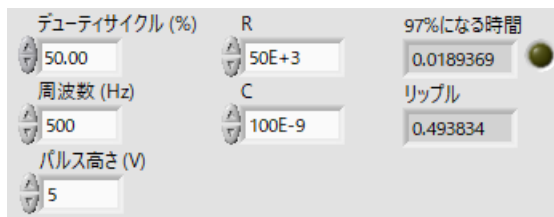
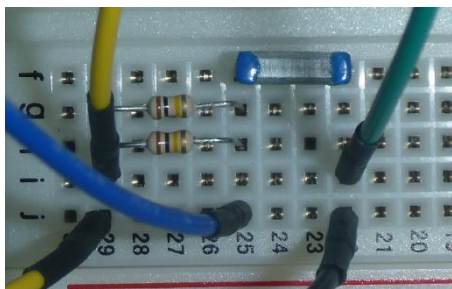
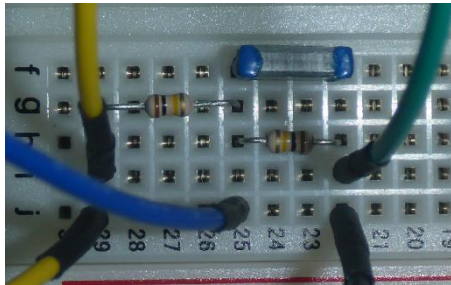


図 6-5 $50k\Omega$ と $0.1\mu F$ の実験とシミュレーション結果

さらに、図 6-4 の回路で R_{in} を $100k\Omega$ にした場合は、定常になるまで $20ms$ 弱だから時定数 0.005 と同じだ。DC 電圧は $1.25V$ で図 6-5 の半分。リップルも半分の $248mV$ になっている。シミュレーションで時定数を 0.005 、パルス高さを $2.5V$ にすると結果が合う。 R と R_{in} は抵抗分圧器を構成し、時定数上は



並列接続と等価ということだ。

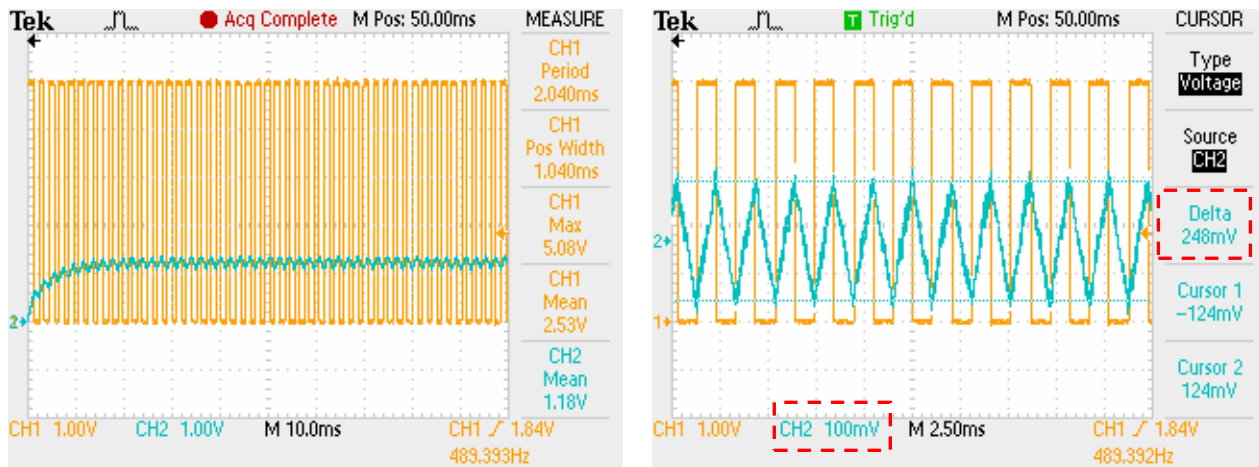
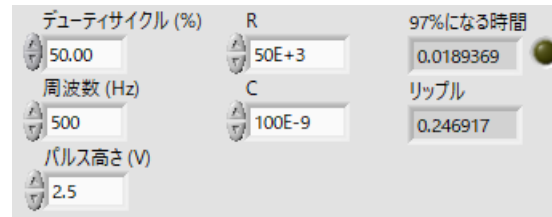


図 6-5 $100k\Omega \times 2$ 個で抵抗分圧+ $0.1\mu F$ の実験とシミュレーション結果

たとえば R_{in} が R の 100 倍以上なら影響は 1%未満に収まるので無視しても大勢に影響はないだろう。フィルタの後につながるドライバの入力抵抗を高くする必要がある。ドライバは電流を増やすついでに、入力する電圧値によって LED に流す電流をコントロールできるようにしたい。電圧-電流変換を兼ねたバッファ&ドライバ回路を設計する (図 6-6)。

電流制御ドライバの設計

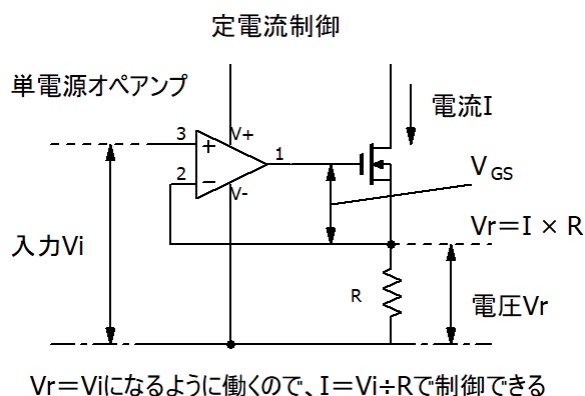


図 6-6 電流制御回路

−入力に V_r を戻せば、PWM のデューティサイクルによって LED 電流が決められる。

オペアンプはプラスとマイナスの 2 電源でもどちらか片方の電源（単電源）でも動かせるが、Arduino から出せる +5V で動き、最大出力電圧がなるべく電源電圧近くまで出せるタイプ（単電源向き、レールツーレールタイプなど）が望ましい。

手持ちに LMC662CN があったのでそれを使う。内部素子が CMOS で構成されていて省電力、入力バイアス電流が 2pA 以下なので R_{in} はほぼ無限大（基板その他の漏れ電流の影響のほうが大きい）、5V 片電源時の出力振幅は 4.78V だ。入手性が悪く値段が高いので NJU7062D あたりでもよいだろう。MOS-FET は以前の別冊で紹介した 2N7000 を使うことにする。

設計するには条件（仕様、要件）が要る。電源電圧は 5V、LED の測定範囲として I_f は最大 30mA、 V_f は最大 3V と仮定しよう。

R を決定するために、MOS-FET が動作するための V_{GS} （ゲート電圧）と V_{DS} （ドレイン電圧）を確認する。2N7000 のデータシートから、 I_D - V_{GS} と I_D - V_{DS} 特性のグラフを引用してきた（図 6-7）。

LED は電源と MOS-FET のドレインの間につながると思ってくれ。すると LED 電流 I は電流検出抵抗 R に流れ、 R の両端には $V_r = I \times R$ の電圧降下が生じる。この V_r が入力電圧 V_i に比例するよう制御すれば、 V_i によって I を調節できる。

出力側の信号を入力側に戻して制御に使うことをフィードバック回路と呼ぶ。そういう場合はオペアンプ（Operational Amplifier）を使うと便利だ。オペアンプには+と−の 2 つの入力があり、その差分を増幅するよう働く（仕組みはとにかくそういう部品だと思ってくれ）。+入力に PWM を DC 変換した信号を入れ、

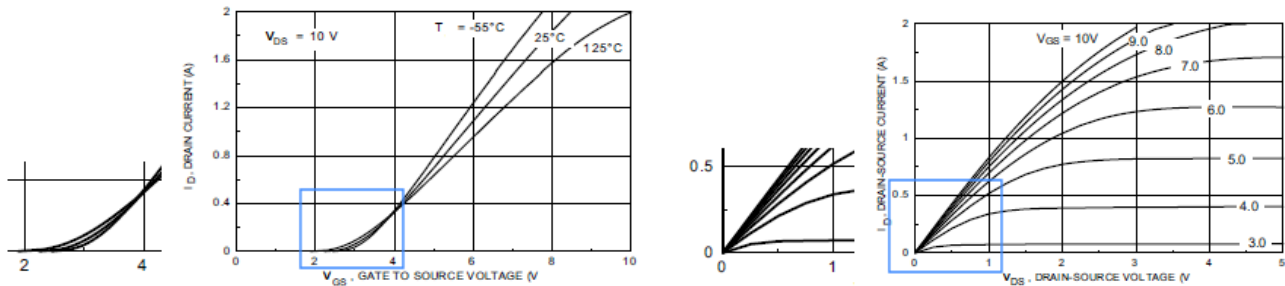


図 6-7 2N7000 の ID-VGS（左側）と ID-VDS（右側）特性曲線

かなり低い領域なので読み取りにくいですが、ID に 30mA 流すために必要なゲート (G-S 間) 電圧は 3V 強。ゲート電圧が 3V のときドレイン (D-S 間) には 0.5V くらいかかっていれば 30mA 流せそうだ。電源電圧 (5V) から Vf (3V) と VDS (0.5V) を差し引くと、Vr は最大 2.5V まで取れる。いっぽう、MOS-FET のゲート電圧はオペアンプの最大出力 4.78V 以上にはできない。そこから必要な VGS (3V) を差し引くと Vr は最大 1.78V になる。両方の条件から R にかかる電圧は 1.78V となる。30mA で 1.78V 発生させるためには、 $1.78 \div 0.03 = 59\Omega$ …E24 系列^(*)から選ぶと 51Ω で決まり。

Vi=Vr だから、最大 1.78V ってことは、PWM のデューティは $1.78 \div 5 \times 100 = 35.6\%$ までしか使わない。Arduino の PWM の分解能は 8 ビットなので、5V のとき 1 ビット当たり 0.0196V。電流分解能にすると $0.0196 \div 51 = 0.38\text{mA}$ ってところか。電流が少ない領域の制御がしにくいなあ。

図 6-5 で検討したのを逆手にとって RC フィルタで 1/2 分圧すれば、デューティ 71.2% まで使え、電流分解能は 0.19mA になる。1/3 分割が最適のはずだが、ちょっと問題があった (後ほど) のので 1/2 でいく。

R は経験的に 10kΩ ~ 100kΩ が使いやすい。前段の負荷として軽く、C の容量が大きすぎず、絶縁抵抗などに気を遣わなくていい範囲でだいたい使い勝手のいい値に落ち着く。部品として入手可能な値に限定し、セtring とリップルのよさげなところを割り出した (図 6-8)。

決定した回路定数を図 6-9 に示す。

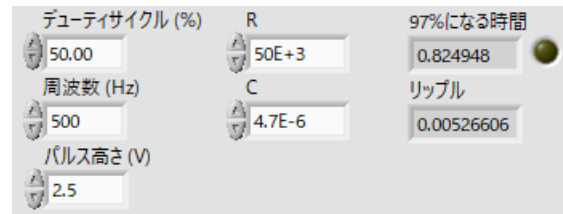


図 6-8 最終的なフィルタの定数

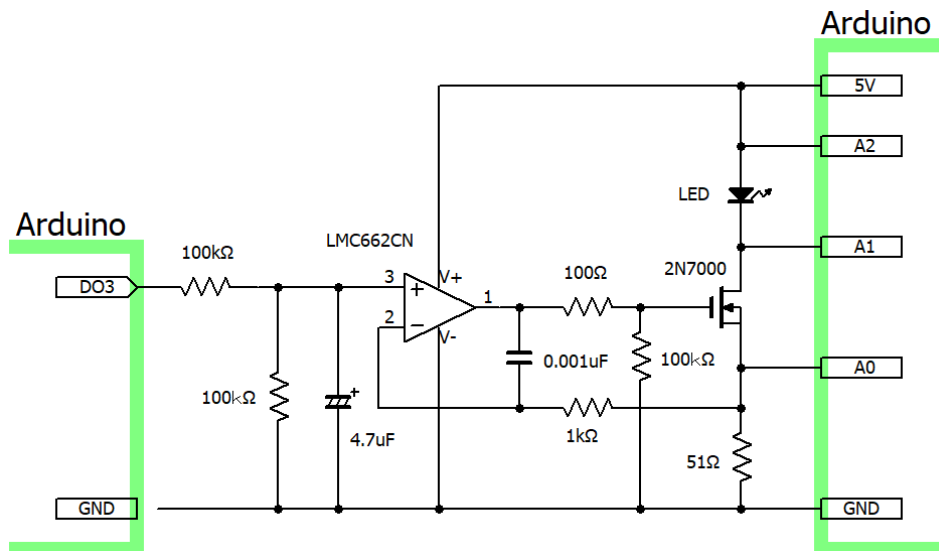


図 6-9 決定したドライバ回路と Arduino の配線

余計な部品が付いているのは安定性を高めるため（おまじないも含むがそれはまたの機会に）。部品の配線は図 6-10 のとおり。

(*1) E24 系列

入手できる抵抗やコンデンサの値は等比数列に従っていて国際規格で決まっている。抵抗で一般的なのは E24 系列（誤差 5%）か E96（誤差 1%）。ちなみに抵抗値はカラーコードで示されているが、この先チップ部品だけになると使わなくなるかも。詳しくは Web で調べてください。

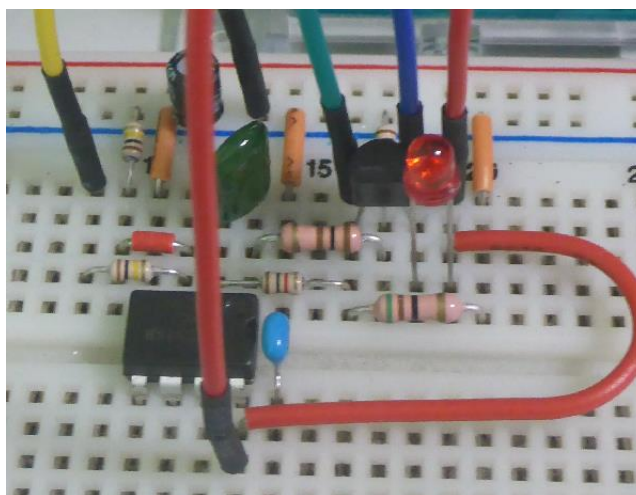


図 6-10 ブレッドボード上の配置
（部品の型は手持ちの関係でバラバラ）

次号プログラム編に続く（引っ張ってスマン）。