

# StudentZone——2022年2月 ADALM2000による実習： A/Dコンバータの原理と動作

著者：Andreea Pop、システム設計／アーキテクチャ・エンジニア  
 Antoniu Miclaus、システム・アプリケーション・エンジニア  
 Mark Thoren、システム設計／アーキテクチャ・エンジニア  
 Doug Mercer、コンサルティング・フェロー

## 目的

今回は、A/Dコンバータ（ADC）を取り上げます。  
 「ADALM2000」とブレッドボード、実習用のサンプルを活用し、  
 その原理や動作について詳細に解説します。

## 背景

ADCは、アナログ信号をデジタル・データに変換するコンポーネントです（図1）。変換の対象となるアナログ信号としては、温度、圧力、電圧、電流、距離、光の強度などを表す実世界の信号が挙げられます。それをデジタル・データとして表現することによって、操作、演算、伝送、保存などの処理を容易に実現できるようになります。

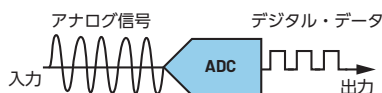


図1. A/D変換の概念

ADCは、アナログ信号を一定の周期（時間間隔）でサンプリングします。そして、得られた個々のサンプルに対し、適切なデジタル値を割り当てます。それらのデジタル値は、バイナリ・コー

ド形式でADCから出力されます。図2に示すように、デジタル値は、サンプリングされたアナログ入力電圧をリファレンス電圧で割り、デジタル・コードの数を掛けることで得ることができます。ADCの分解能は、出力コードのバイナリ・ビット数で決まります。

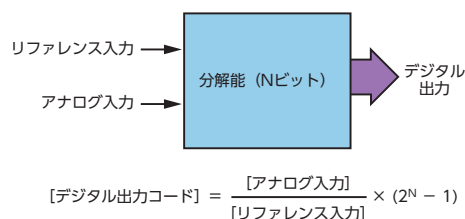


図2. アナログ信号と  
デジタル・コードの関係

より詳しく言えば、ADCによる変換処理は、サンプリングと量子化という2つのプロセスから成ります。ADCでは、無限大の分解能でなければ表現できないはずのアナログ信号を、分解能が有限のデジタル値として表現します。デジタル・データのとり得る値は、 $2^N$ 個となります。ここで、NはADCの分解能（バイナリ出力におけるビット数）です。

ADCの分解能が有限であるということは、常に図3に示すような状態が発生するということを意味します。すなわち、実際のアナログ信号は2つの量子化レベルの間のどこかに存在し、その範囲内にあるアナログ信号はすべて同じデジタル・コードで表現されるということです。言い換えれば、有限の分解能で量子化を行うと、必ず誤差（固有の不確定性）が生じます。この量子化誤差によって、ADCの最大ダイナミック・レンジが決まります。

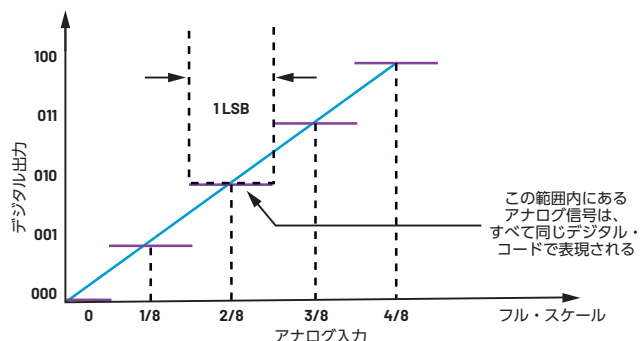


図3. 量子化のプロセス

サンプリングのプロセスでは、連続時間領域のアナログ信号を、均等な時間間隔で測定した離散的な値で表現するというを行います（図4）。このプロセスにより、ナイキスト定理に従って、サンプリングされた信号の最大帯域幅が決まります。同定理によれば、エイリアシングを防止するためには、サンプリングの対象となる信号の周波数はサンプリング周波数の1/2以下でなければなりません。エイリアシングとは、帯域外の周波数信号が、サンプリング・プロセスによって帯域内に現れる現象のことです。ちなみに、通信システムでは、このエイリアシングを積極的に利用するケースがあります。すなわち、高い周波数の信号を低い周波数にダウンコンバートするということが行われます。この手法はアンダーサンプリングと呼ばれています。アンダーサンプリングに求められる要件は、ADCが、対象とする最高周波数の信号を取得できるだけの十分な入力帯域幅とダイナミック・レンジを備えていることです。

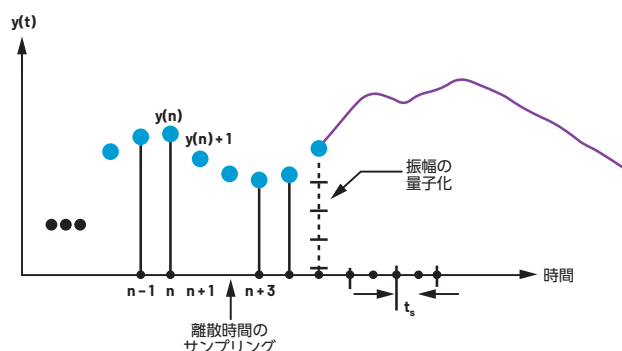


図4. サンプリングのプロセス

サンプリングと量子化は、非常に重要な概念です。これらによって、理想的なADCの性能の限界が決まります。図5に示すように、理想的なADCでは、コードの遷移は1LSBを単位として正確に行われます。NビットのADCには $2^N$ 個のコードがあり、 $1\text{LSB} = \text{FS}/2^N$ となります。ここで、FSはフル・スケールのアナログ入力電圧です。但し、現実のADCの動作は、理想的ではない要因からの影響を受けます。その結果、分解能とサンプリング・レートによって決まる理論値に対して、現実のADCの出力にはズレ（誤差）が生じます。それらの誤差は、現実のADCのAC性能／DC性能に反映されます。

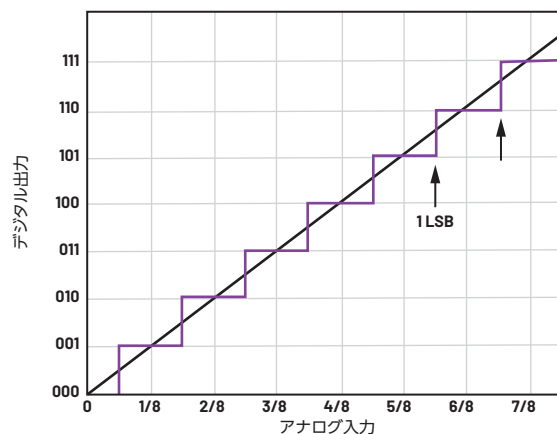


図5. 理想的なADCの伝達関数。  
1LSBに相当する範囲内のすべてのアナログ信号は、すべて同じデジタル・コードに変換されます。

## 準備するもの

- ▶ アクティブ・ラーニング・モジュール「[ADALM2000](#)」
- ▶ ソルダレス・ブレッドボード
- ▶ ジャンパ線のキット
- ▶ オペアンプ：「[OP482](#)」（1個）
- ▶ V/Fコンバータ：「[AD654](#)」（2個）
- ▶ 抵抗：1kΩ（3個）、10kΩ（5個）
- ▶ コンデンサ：1nF（1個）、1μF（1個）
- ▶ ANDゲート：「[SN74HC08](#)」（1個）
- ▶ ORゲート：「[SN74HC32](#)」（1個）
- ▶ インバータ：「[SN74HC04](#)」（1個）
- ▶ 12ビットのADC：「[AD7920](#)」（1個）

## フラッシュ型のADC

ADCの基本的な概念は上述したとおりです。ただ、その機能の実現手段はいくつも存在します。つまり、ADCの変換方式（アーキテクチャ）は何種類も存在するということです。ここでは、最も基本的なものとして、フラッシュ型のADCを取り上げます。

### 背景

フラッシュ型のADCは、並列型のADCとも呼ばれます。この方式のADCを使えば、非常に高速にアナログ信号をデジタル・データに変換することができます。つまり、フラッシュ型のADCは、非常に広い帯域幅を必要とするアプリケーションに適しています。代表的な用途の例としては、データ・アキュイジション・システム、衛星通信、レーダー・システム、サンプリング・オシロスコープ、高密度のディスク・ドライブなどが挙げられます。その一方で、フラッシュ型のADCには、他の方式のADCと比べて消費電力が多いという欠点が存在します。また、一般的には分解能が8ビット程度に制限されます。

フラッシュ型のADCは、図6に示すような形で高速コンパレータを使用することによって実現されます。NビットのADCでは、 $2^N - 1$ 個のコンパレータを使用します。また、リファレンス電圧を $2^N$ 個の抵抗によって分割します。各コンパレータは、印加されたリファレンス電圧と比べてアナログ入力電圧の方が高い場合には1を出力します。そうでなければ、0を出力します。入力信号が各コンパレータのリファレンス電圧よりも低くなるポイントで、コードは1から0に変化します。

図6の回路は、分解能が2ビットのフラッシュ型ADCのアナログ側を表しています。このアーキテクチャのエンコーディングは、温度計コード（ユーナリー・コード）と呼ばれています。ただ、

ユーナリー・コードは使い勝手が良いとは言えません。そのため、論理回路を追加し、アプリケーションにとって都合が良い形式にデコードすることが行われます。例えば、AND、OR、INVの論理ゲートを使用すれば、プライオリティ・エンコーダを構築することができます。図7のように回路を構成すれば、その出力はMSBが0から始まる元の数値のバイナリ表現になります。

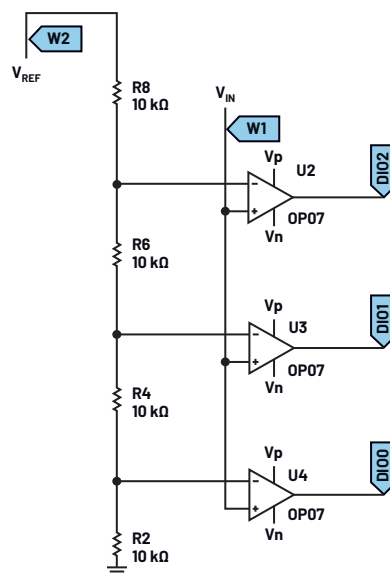


図6. フラッシュ型のADC。  
アナログ側の回路を示しました。

先述したように、実際のフラッシュ型ADCは高速コンパレータを使用して構成されます。ただ、本稿では、回路を簡素化するために、コンパレータの代わりにクワッド・オペアンプであるOP482を使用して解説を進めることにします。もちろん、「AD8561」のようなコンパレータを4個使用することで、同等の回路を構成することも可能です。

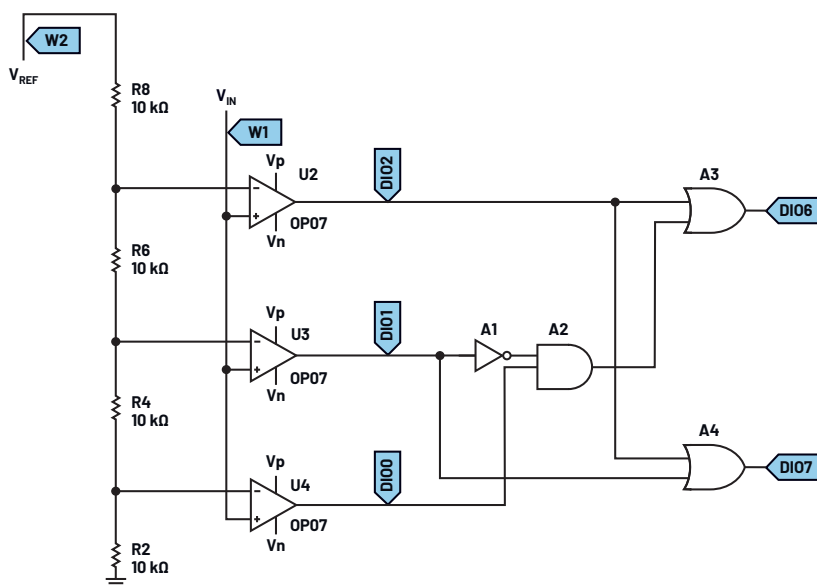


図7. エンコーダを追加したフラッシュ型のADC

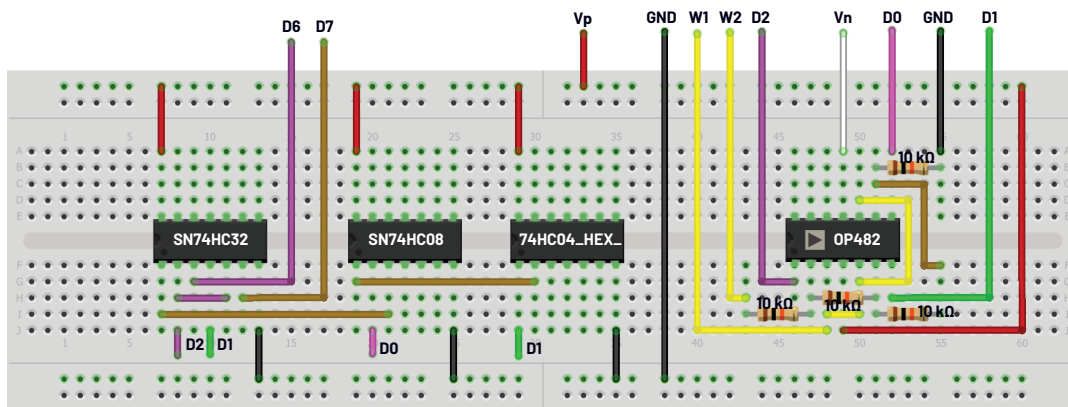


図8. 図7の回路を実装したブレッドボード

## ハードウェアの設定

図7に示した回路を、図8に示すようにブレッドボードに実装してください。この回路は2ビットのフラッシュ型ADCであり、エンコード済みの出力が得られます。

## 手順

図7の回路には、±5Vの電源を供給します。任意波形ジェネレータ (AWG1) は、ピークtoピークの振幅が5V、オフセットが2.5V、周波数が100Hzののこぎり波を生成するように設定してください。AWG2は、ADC用の5Vのリファレンス電圧を生成するために使用します。

ここでは、ADALM2000 (通称、M2K) が備えるロジック・アナライザ機能を使用します。デジタル・チャンネルDIO0、DIO1、DIO2は、ユニナリー・コードとしてデコードされるチャンネル・グループとして扱います。一方、DIO6、DIO7はパラレル出力としてデコードされるチャンネル・グループとして表示します。図9に、出力データの表示結果を示しました。なお、ロジック・アナライザ機能による信号の表示にはソフトウェア・パッケージ「Scopy」を使用します。

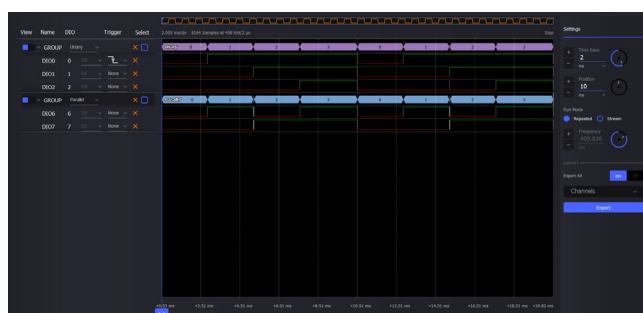


図9. 図7の回路の出力データ

ここでは、有効な入力範囲 (0V～5V) 全体にわたってアナログ電圧を変化させます。ユニナリー・コードのグループ・チャンネルは、2ビットのフラッシュ型ADCにおける温度計コードのすべてを出力します。つまり、出力される可能性のあるすべての値が表示されています。パラレル出力のチャンネル・グループについても、ADCの出力がバイナリ値として表示されています。

## V/F コンバータをADCとして使用する

続いては、V/F (電圧 - 周波数) コンバータをADCとして使用する例を紹介します。

## 背景

ここでは、V/F コンバータとしてAD654を使用します。同ICを使って図10のような回路を構成します。変換を行うために、同ICの出力を、インターバル・タイマー/イベント・カウンタを備えたマイクロコンピュータに接続します。

この回路では、入力電圧に比例して、カウント期間中に信号エッジ (立上がり、または立下がり) の総数がカウントされます。また、この回路では、1Vのフル・スケール入力電圧によって100kHzの信号が生成されます。カウント周期が100ミリ秒である場合、総カウント数は1万になります。この最大値を基にスケールリングを行うことで、入力電圧を算出できます。例えば、カウント数が5000であれば入力電圧は0.5Vであるということになります。

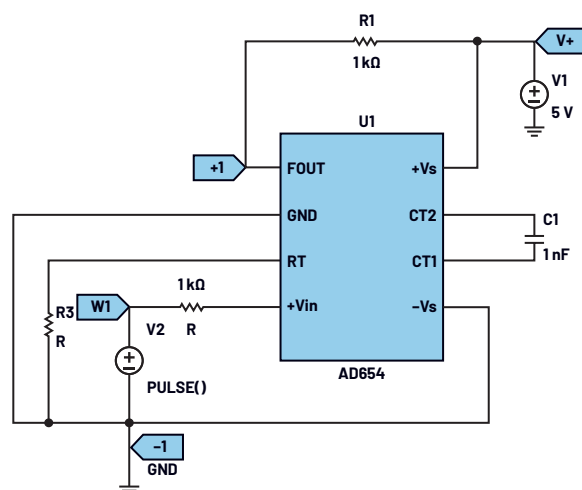


図10. V/F コンバータをADCとして使用する回路

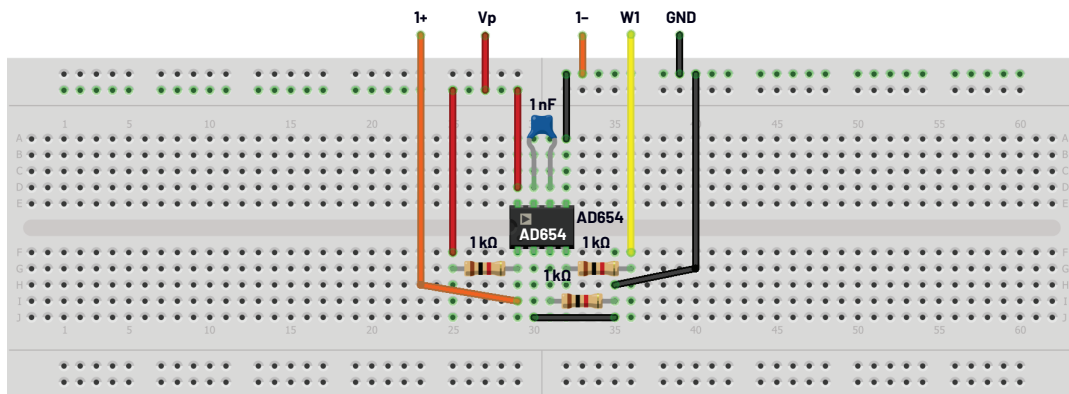


図 11. 図 10 の回路を実装したブレッドボード

## ハードウェアの設定

図 10 に示した回路を、図 11 に示すようにブレッドボードに実装してください。

### 手順

図 10 の回路には、電源として 5V を印加します。AWG1 は、1V の定電圧を生成するように設定してください。

Scopy を使用し、出力信号がチャンネル 1 に表示されるようにオシロスコープを設定します。また、チャンネル 1 の測定タブで周波数測定をイネーブルにしてください。

図 12 に示したのが、出力信号の例です。これは、1V のフル・スケール入力電圧に対する V/F コンバータの出力信号を表しています。出力信号の周波数が 100kHz である点に注目してください。

ここで、入力電圧を 0.5V (ハーフ・スケール) に変更します。すると、出力信号が図 13 のように変化します。出力信号の周波数が 50kHz に変化していることがわかります。

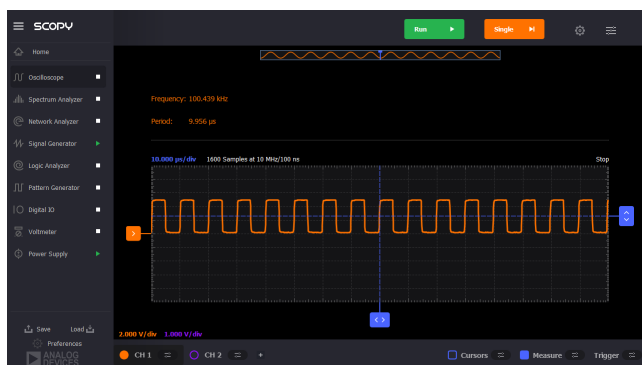


図 12. 1V (フル・スケール) の電圧を入力した場合の出力信号



図 13. 0.5V (ハーフ・スケール) の電圧を入力した場合の出力信号

## SAR ADC の動作

続いて、逐次比較型の ADC (SAR ADC) を取り上げます。

### 背景

SAR ADC は、連続したアナログ信号波形を離散的なデジタル表現に変換するために、とり得るすべての量子化レベルに対するバイナリ・サーチを実施します。その結果として、最終的なデジタル出力に収束するように変換処理が行われます。

通常、SAR ADC の回路は、以下に示す 4 つのサブ回路から構成されます。

- ▶ サンプル&ホールド回路 (S/H)：入力電圧  $V_{IN}$  の値を取得／保持します。
- ▶ アナログ電圧コンパレータ：ADC が内蔵する D/A コンバータ (DAC) の出力と入力電圧  $V_{IN}$  を比較し、結果を逐次比較レジスタ回路 (SAR) に出力します。
- ▶ 逐次比較レジスタ回路：  $V_{IN}$  の近似値に相当するデジタル・コードを ADC の内蔵 DAC に供給します。
- ▶ リファレンス用の DAC：逐次比較レジスタ回路のデジタル・コード出力に相当するアナログ電圧をコンパレータに供給します。

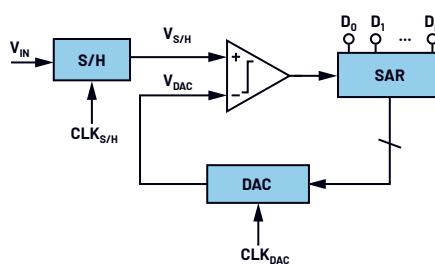


図 14. SAR ADC の標準的なアーキテクチャ

逐次比較レジスタ回路は、MSBがデジタル値の1になるように初期化されます。そのコードがリファレンス用のDACに入力されます。DACは、このデジタル・コードに相当するアナログ電圧 ( $V_{REF}/2$ ) をコンパレータに供給します。コンパレータは、その電圧とサンプリングされた入力電圧を比較します。 $V_{REF}/2$ が  $V_{IN}$  より高い場合、コンパレータの動作に応じて逐次比較レジスタ回路はこのビットをリセットします。そうでない場合には、このビットは1のままになります。続いて、次のビットが1に設定されて同様の処理が行われます。このバイナリ・サーチは、逐次比較レジスタ回路のすべてのビットに対する処理が完了するまで繰り返されます。その結果、サンプリングされた入力電圧に相当するデジタル値のコードが得られます。変換が終了 (EOC: End of the Conversion) したら、最終的に逐次比較レジスタ回路からデータが出力されます。

ここで、図15をご覧ください。これは、4ビットのSAR ADCで行われる処理の例です。Y軸は、DACの出力電圧を表しています。この例では、最初の比較では  $V_{IN} < V_{DAC}$  となっています。そのため、ビット3は0に設定されます。次に、DACの入力は0100に設定されます。その上で、2回目の比較処理が実行されます。その結果、 $V_{IN} > V_{DAC}$  であることから、ビット2は1のまま保持されます。続いて、DACの入力が0110に設定され、3回目の比較が行われます。その結果、ビット1は0に設定されます。更に、DACの入力が0101に設定され、最後の比較が行われます。 $V_{IN} > V_{DAC}$  であることから、ビット0は1のままになります。

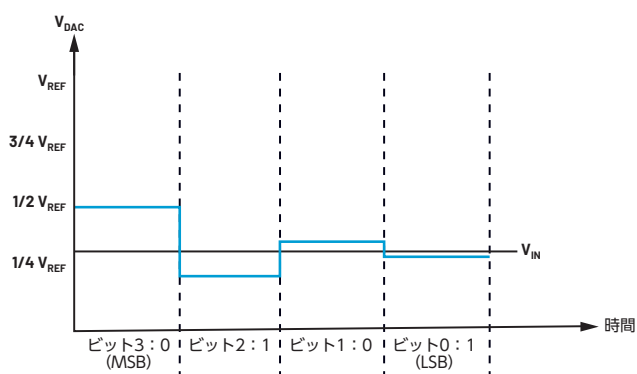


図15. 4ビットのSAR ADCで行われる処理

## ハードウェアの設定

続いて、ADALM2000を使用し、SAR ADCに相当する回路を構築して動作を確認してみます (図16)。この回路では4ビットのDACを使用することにします (DACについては今後取り上げる予定です)。そのDACの出力をコンパレータに接続します。ただ、逐次比較レジスタ回路については、スクリプトを使ってシミュレーション・レベルで動作を再現することにします。すなわち、コンパレータの出力に基づいてバイナリ・サーチを実行し、適切なバイナリ値を生成するスクリプトを用意するということです。

図16の回路を、図17に示すようにブレッドボードに実装してください。

ここでは「OP484」が備える2つのオペアンプを利用します。両オペアンプは精度が高くレールtoレールの入出力に対応しています。1つはR-2Rラダー型DAC向けに使用し、もう1つはDACの出力と入力電圧を比較するコンパレータとして使用します。

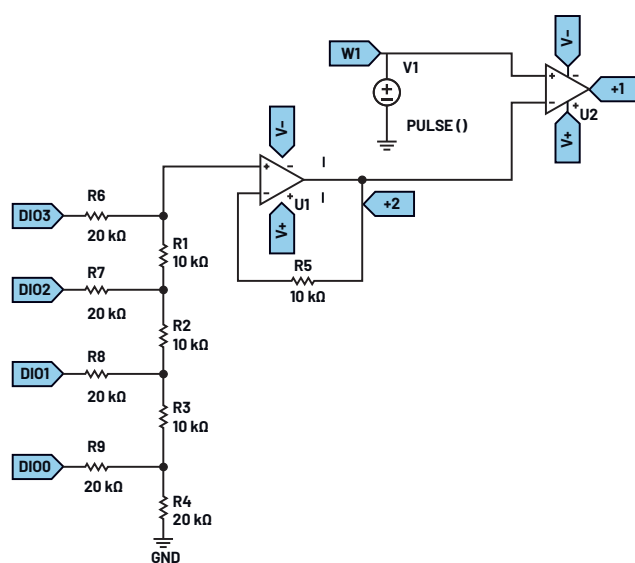


図16. SAR ADCの回路図

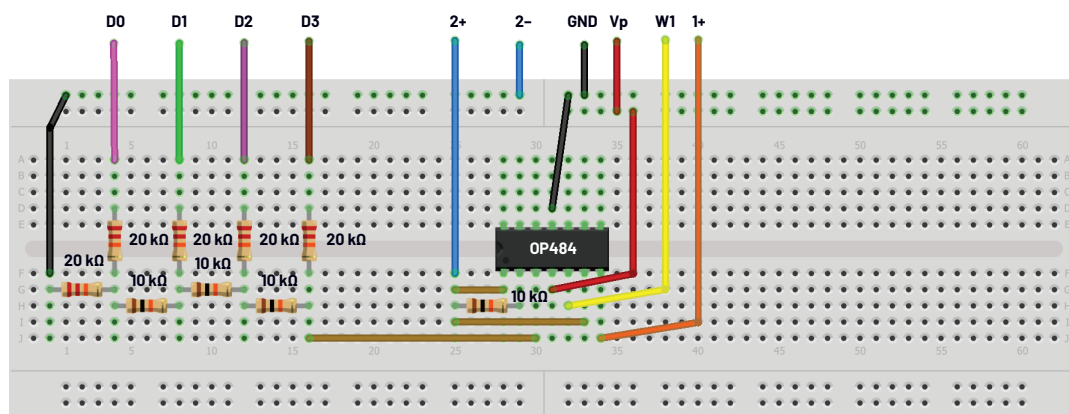


図17. 図16の回路を実装したブレッドボード

## 手順

図16の回路には、 $\pm 5V$ の電源を供給します。オシロスコープは、コンパレータの出力がチャンネル1に、DACの出力がチャンネル2に表示されるように設定してください。ロジック・アナライザについては、最初の4つのデジタル・チャンネルをグループ化します。また、デコードをパラレルに設定してください。

こちらから、[SAR ADCのスクリプト](#)をダウンロードしてください。これは、Scopyのインターフェースを使用することで実行できます。

デジタル・コードは、コンパレータの出力から受け取ったフィードバックに基づき、逐次近似法を使用して更新されます。

オシロスコープを使って、DACの出力を時間領域で表示してください。すると、図18に示すようなプロットが得られるはずです。ご覧のように、出力の値は数回の近似ステップを経て、入力値（2V）に近づいていきます。

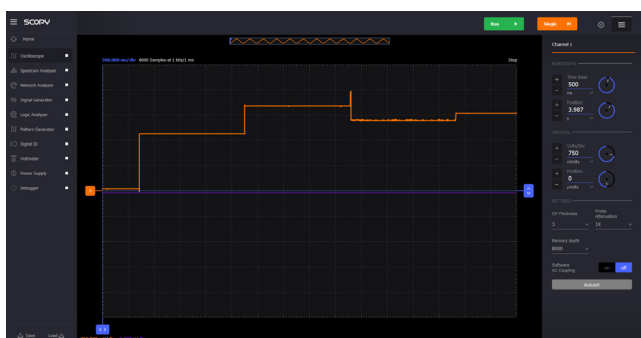


図18. SAR ADCが内蔵するDACの出力信号

## SAR ADC 製品の実例

続いて、実際に市販されているSAR ADC製品の例を紹介します。具体的には、分解能が12ビットのAD7920を取り上げます。

### 背景

AD7920は高速、低消費電力のSAR ADCであり、 $2.35V \sim 5.25V$ の単一電源で動作します。このADCはシリアル・インターフェースを備えており、シリアル・クロックを変換クロックとしても使用します。このクロックは、変換の実行中にAD7920からのデータを転送するためにも使用されます。シリアル・クロックと/CSを使用することにより変換フェーズとアキュイジション・フェーズを制御できるので、マイクロプロセッサやDSPと接続して利用することが可能です。入力されたアナログ信号は、/CSの立下がりエッジでサンプリングされます。また、その時点で変換動作が開始されます。

図19に示した同ADCのブロック図をご覧ください。それぞれ、アキュイジション・フェーズと変換フェーズの状態を簡略化して表しています。アキュイジション・フェーズではSW2が閉じており、SW1は位置Aにあります。この状態で、コンパレータはバランスのとれた状態に保たれており、サンプリング用のコンデンサに $V_{IN}$ の信号が取り込まれます。ADCが変換を開始する際にはSW2が開き、SW1は位置Bに移動します。これによって、コンパレータはアンバランスな状態になります。ここで、制御ロジックと電荷再配分方式のDACにより、サンプリング・コンデンサに対して一定量の電荷の加算／減算が行われます。その結果、コンパレータがバランスのとれた状態に戻り、変換が完了します。

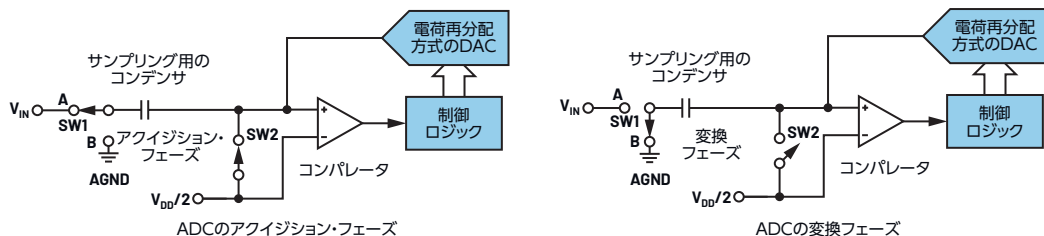


図19. AD7920のアキュイジション・フェーズと変換フェーズ

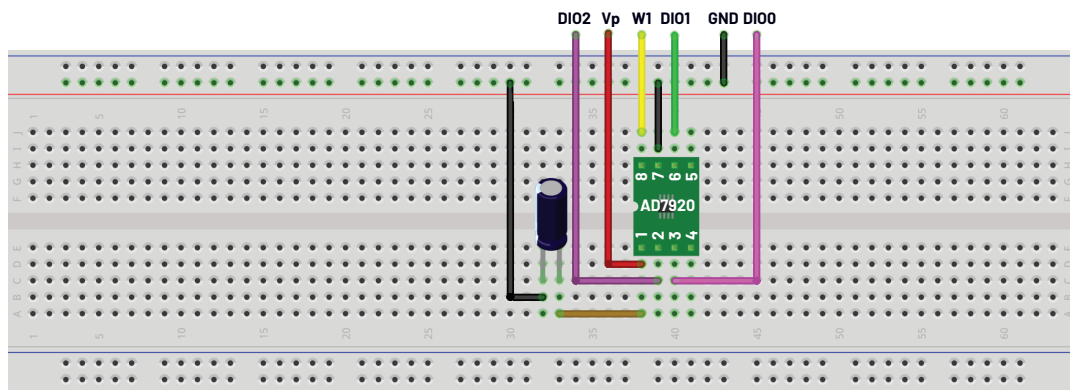


図20. 図21の回路を実装したブレッドボード

## ハードウェアの設定

図21に、AD7920を使用する場合の標準的なピン接続の例を示しました。同ADCの場合、リファレンス電圧 ( $V_{REF}$ ) は内部で  $V_{DD}$  を基に生成します。そのため、 $V_{DD}$  は十分にデカップリングしなければなりません。アナログ入力範囲は  $0V \sim V_{DD}$  となります。変換結果は、16ビットのワードとして出力されます。4つの0の後に、12ビット（または10ビット）のデータがMSBファーストで続きます。

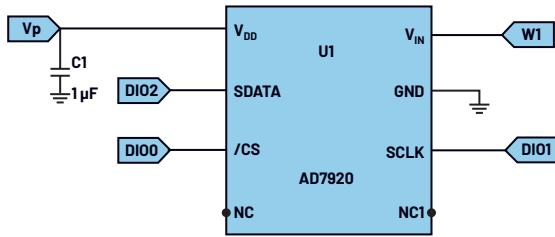


図21. AD7920の標準的なピン接続

## 手順

Scopyを使用し、正の電源電圧を3Vに設定します。AWGのチャンネル1を「Constant」に設定すると共に、 $0V \sim 3V$ の範囲の値（例えば1.5V）を設定してください。これらの電圧の値は、オシロスコープを使うことで実際に確認することができます。

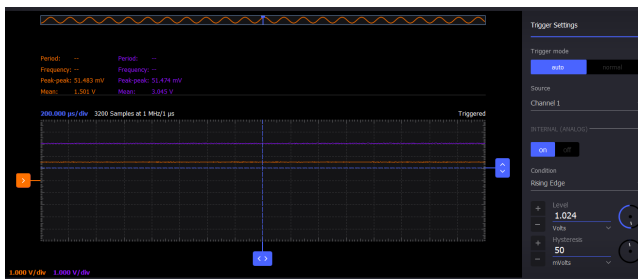


図22.  $V_{IN}$ （チャンネル1）と  $V_{REF}$ （チャンネル2）の電圧

ロジック・アナライザにおいて、DIO0、DIO1、DIO2をグループ・チャンネルとして設定します。また、このグループ・チャンネルはSPI（Serial Peripheral Interface）として設定します。すなわち、DIO0をCS#、DIO1をCLK、DIO2をMISOとして扱います。データの転送は、CS#の立下がりエッジで始まります。そこで、DIO0の立下がりエッジをトリガとして使用します。また、DIO1のトリガをローに設定すると共に、トリガの設定メニューでトリガのロジックとしてANDを選択します。DIO2はADCの出力信号なので、トリガの設定は必要ありません。ロジック・アナライザをイネーブルにすると、トリガの信号を待っている状態になります。

クロック信号は、ADALM2000のパターン・ジェネレータで生成します。DIO1のチャンネルをイネーブルにし、そのパターンを周波数が5MHzのクロックとして設定した上で「Run」をクリックします。CS#は、デジタルIOツールによって制御すること

ができます。出力ピンとして設定されているDIO0ピンをトグルすると、変換が始まります。すなわち、CS#の立下がりエッジとCLKのローの状態が同時に発生すると変換が開始されます。出力信号とMISOの16進のデータは、ロジック・アナライザによって確認することができます（図23）。

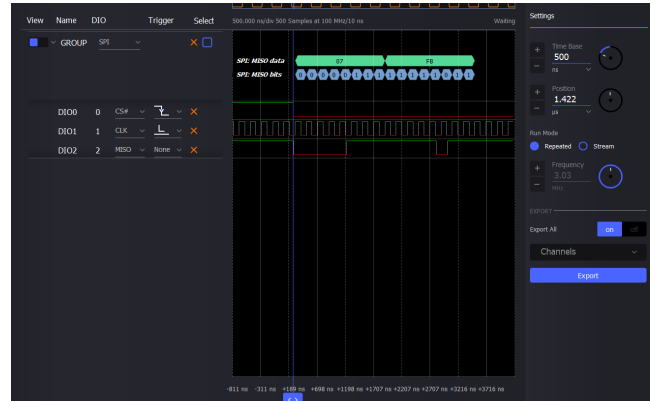


図23. SPIの各信号

この結果については、以下に示すADCの伝達関数を用いることで確認することができます。

$$[\text{デジタル出力コード}] = \frac{([\text{アナログ入力}])}{([\text{リファレンス入力}])} \times ((2^N) - 1)$$

$$V_{in} = \left( \frac{[\text{デジタル・コード}]_{10}}{(2^{12}) - 1} \right) \times V_{ref}$$

$$V_{in} = \left( \frac{(07\text{ FB})_{10}}{4095} \right) \times V_{ref}$$

$$V_{in} = \left( \frac{2043}{4095} \right) \times 3.04\text{ V}$$

$$V_{in} = 1.5\text{ V}$$

ここで、デジタル・コードはMISOのデータ、アナログ入力はオシロスコープにおけるチャンネル1の電圧の読み取り値、リファレンス入力は同チャンネル2の電圧の読み取り値、NはAD7920のビット数です。上記のとおり、ADCの入力電圧は1.5Vと計算できます。つまり、オシロスコープにおけるチャンネル1の読み取り値と同じ値になります。

## 追加の実習——デュアルスロープADC

デュアルスロープADC（またはその派生品）は、最高の精度を達成する多くのデジタル電圧計の心臓部として使われています。そのアーキテクチャは、いくつかの有用な特性を備えています。例えば、誤差要因のほとんどは相殺されるため、高精度の部品はわずか数個しか必要ありません。また、50Hz/60Hzのライン・ノイズといった特定周波数のノイズを除去するように設定することができます。更に、高周波ノイズの影響を受けません。

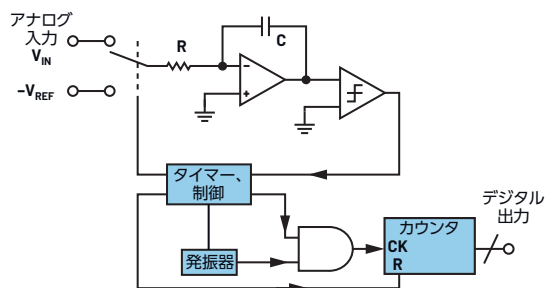


図24. デュアルスロープADCのブロック図

この種のADCでは、値が未知の入力電圧を一定期間積分器に印加します（ランアップ）。その後、入力と逆極性の既知のリファレンス電圧を積分器に印加する（ランダウン）ことで動作します。このようにすることで、リファレンス電圧と、ランアップ時間に対するランダウン時間の比を基にして入力電圧を算出することができます（以下参照）。

$$V_{in} = V_{ref} \times \left( \frac{T_{rundown}}{T_{runup}} \right)$$

様々な検証が行われた結果、デュアルスロープADCの精度は、ほとんどの部品の許容誤差からの影響を受けないことがわかっています。その理由は以下のとおりです。

- ▶ 積分器の抵抗とコンデンサの許容誤差は、出力の傾きに影響を与えます。しかし、ランアップとランダウンの両方に対して等しく影響を与えます。
- ▶ ランアップ時間の決定と、ランダウン時間の測定に使用されるタイム・ベースの誤差は、両方の時間に等しく影響を与えます。

リファレンス電圧は測定値に直接影響を及ぼすので、高い精度が必要になります。また、積分器においては、コンデンサの誘電吸収が誤差の要因になります。したがって、ポリプロピレン・コンデンサやポリスチレン・コンデンサを使用するのが理想的です。アルミ電解コンデンサは適していません。

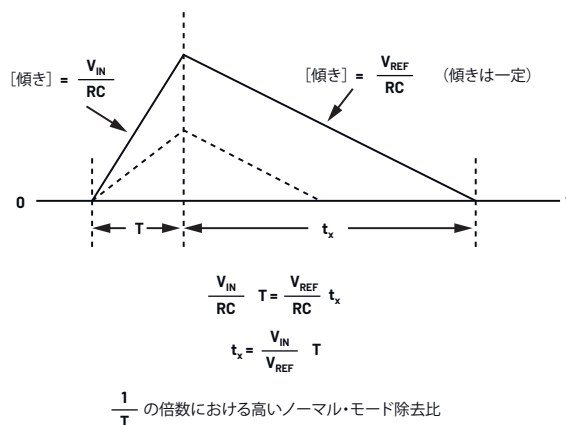


図25. デュアルスロープADCの積分器の出力波形

図26に示したのは、デュアルスロープADCの周波数応答です。デュアルスロープADCでは、一定の期間（ランアップ）にわたって入力がサンプリングされます。ランアップの開始時の電圧は、変換結果に対して、ランアップの終了時の電圧と同程度の影響を与えます。これはボックスカー平均と呼ばれることもあり、 $1/T$ 、 $2/T$ 、 $3/T$ などの周波数で発生する外乱（ノイズ）を除去する効果があります。200ミリ秒という積分時間は、50Hzのノイズの10サイクル分、60Hzのノイズの12サイクル分に相当します。この積分時間を使えば、世界中のどの国でもライン・ノイズを除去できます。そのため、この値がランアップ時間としてよく選択されます。

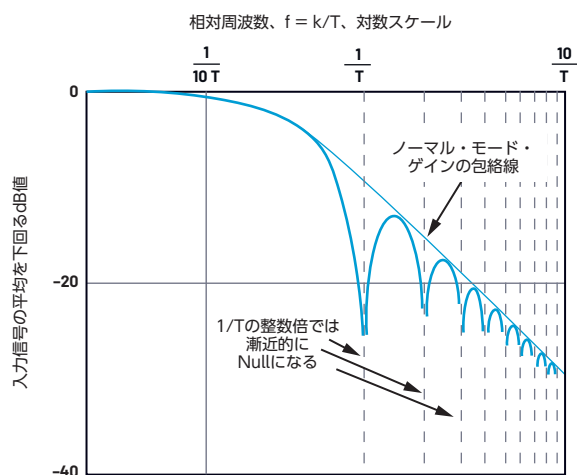
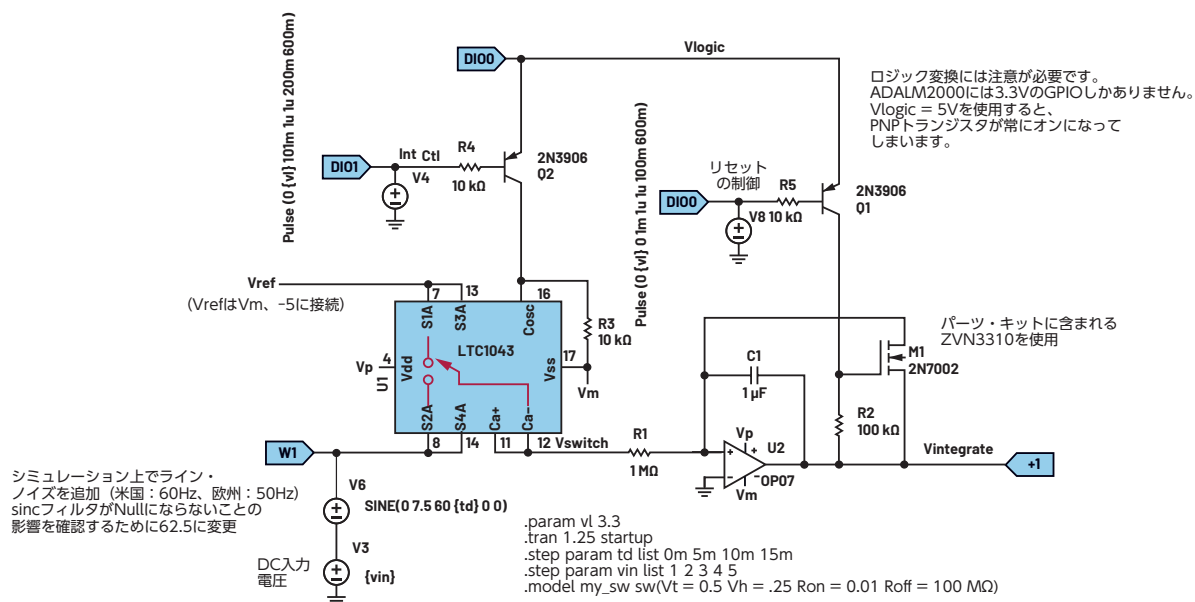


図26. デュアルスロープADCの積分器の周波数応答



## シミュレーション

こちらから、DualSlope.ascというファイルをダウンロードしてください。このファイルを使用してLTspice®用によるシミュレーションを実行しましょう。その際には、Vintegrateノードにプローブを当てます。

このシミュレーションでは、DC入力電圧に60Hzのライン・ノイズを加えています。stepコマンドを使うことで、いくつかのケースについてシミュレーションを実施します。具体的には、入力電圧が1V、2V、3V、4V、5Vの場合を取り上げています。また、60Hzのライン・ノイズについては位相をいくつかの値に変更しています。200ミリ秒のランアップ時間は、60Hzのライン・サイクルの整数倍に相当します。そのため、周波数応答においてノイズはNullの部分にあたり、ランダウン時間は位相の値の影響を受けません。また、周波数を62.5Hzに変更した場合のシミュレーションも行います。62.5Hzは、周波数応答のピークに位置します。

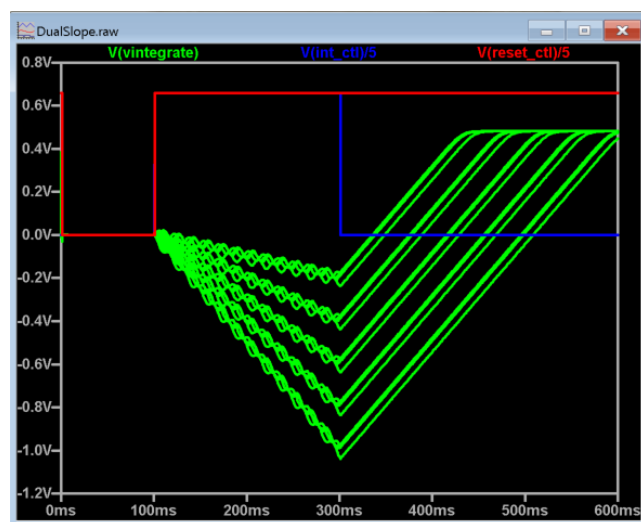


図29. デュアルスロープADCの積分器のシミュレーション結果 (その2)

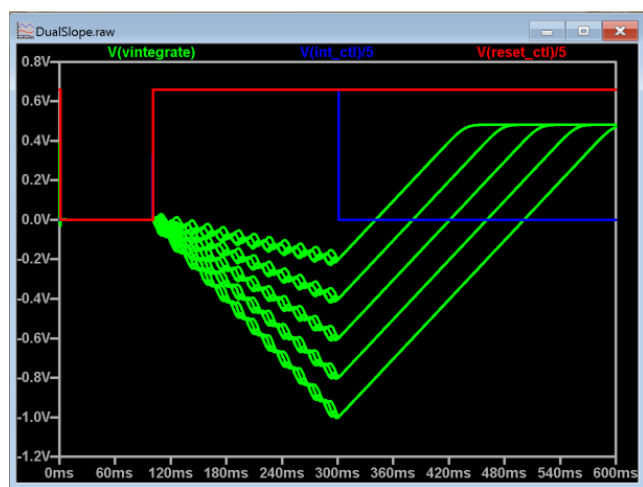


図28. デュアルスロープADCの積分器のシミュレーション結果 (その1)

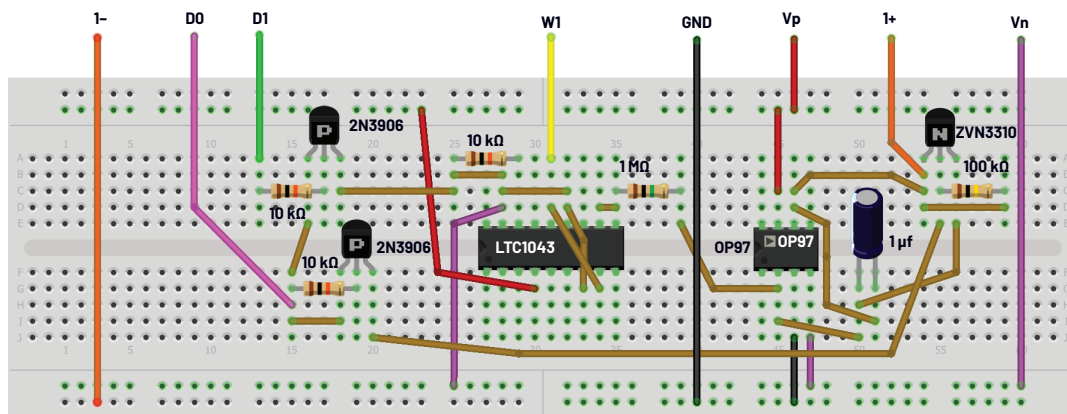


図 30. 図 27 の回路を実装したブレッドボード

## ハードウェアの設定

図 27 の回路を、図 30 のようにブレッドボードに実装します。デュアルスロープ ADC を構築すると共に、ADALM2000 への接続を行います。

## 手順

Scopy を開き、初期化ファイルである [Dual\\_slope\\_scopy\\_setup.ini](#) をインクルードします。それにより、設定が容易になります。各種の設定は以下のように行ってください。

- ▶ 電源：トラッキングをイネーブルにし、± 5V に設定します。
- ▶ デジタル IO：DIO2 を出力に設定し、1 を設定します。
- ▶ パターン・ジェネレータ：DIO0、DIO1 をグループ化します。パターンについては [dual\\_slope\\_pattern.csv](#) をインポートしてください。周波数は 5kHz に設定します。
- ▶ AWG：チャンネル 1 については「Constant」を選択し、2.5V を設定します。
- ▶ オシロスコープ：タイム・ベースを 200 ミリ秒、チャンネル 1 を 400mV/div に設定します。トリガは立下がりエッジに設定し、そのレベルは 200mV とします。それにより、積分器のリセット期間が始まる際に ADALM2000 がトリガされます。

リファレンスを -5V の電源に接続し、入力電圧を 2.5V に設定してください。すると、ランダウン時間が 2 目盛分（400 ミリ秒）になり、ランアップ時間が 1 目盛分（200 ミリ秒）になります。つまり、次式が成り立ちます。

$$V_{IN} = 5V \times (200 \text{ ミリ秒} / 400 \text{ ミリ秒}) = 2.5V$$

入力電圧を変化させると、ランアップ時間が変化することがわかります。取得されるプロットの例を図 32 に示しました。



図 31. デュアルスロープ ADC の積分器の出力波形

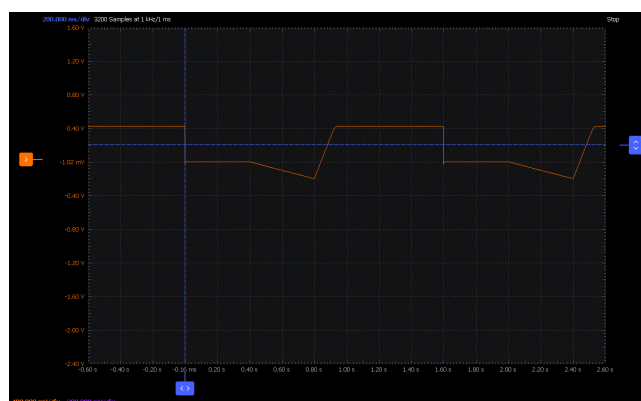


図 32. 異なる入力電圧に対するデュアルスロープ ADC の積分器の出力波形

実際のデュアルスロープ ADC では、マイクロコントローラを使用して、積分器の制御、ランアップ時間の設定、ランダウン時間の測定を行います。ほとんどのマイクロコントローラは、ペリフェラルとしてカウンタを備えているので、そうした処理を簡単に実施できます。

## 問題

ADC を利用するアプリケーションの例を挙げてください。

答えは [StudentZone](#) で確認できます。



#### 著者について

Andreea Pop ([andreea.pop@analog.com](mailto:andreea.pop@analog.com)) は、アナログ・デバイセズのシステム設計／アーキテクチャ・エンジニアです。2019年より現職。クルジュナポカ工科大学で電子工学と通信工学の学士号を取得しています。また、同校でIC／システムに関する修士課程を修了しました。



#### 著者について

Antoniu Miclaus ([antoniu.miclaus@analog.com](mailto:antoniu.miclaus@analog.com)) は、アナログ・デバイセズのシステム・アプリケーション・エンジニアです。アカデミック・プログラムや、Circuits from the Lab<sup>®</sup> 向けの組み込みソフトウェア、QAプロセス・マネジメントなどに携わっています。2017年2月から、ルーマニアのクルジュナポカで勤務しています。現在、バベシュボヨイ大学においてソフトウェア・エンジニアリングに関する修士課程にも取り組んでいます。また、クルジュナポカ技術大学で電子工学と通信工学の学士号を取得しています。



#### 著者について

Mark Thoren ([mark.thoren@analog.com](mailto:mark.thoren@analog.com)) は、アナログ・デバイセズのシステム設計／アーキテクチャ・エンジニアです。メイン大学で農業／機械工学の学士号、電気工学の修士号を取得しています。壊れた機器を入手して使用可能なレベルまで修理することで、実験室の機材を充実させています。



#### 著者について

Doug Mercer ([doug.mercer@analog.com](mailto:doug.mercer@analog.com)) は、1977年にレンセラー工科大学で電気電子工学の学士号を取得しました。同年にアナログ・デバイセズに入社して以来、直接または間接的に30種以上のデータ・コンバータ製品の開発に携わりました。また、13件の特許を保有しています。1995年にはアナログ・デバイセズのフェローに任命されました。2009年にフルタイム勤務からは退きましたが、名誉フェローとして仕事を続けており、Active Learning Programにもかかわっています。2016年に、レンセラー工科大学 電気／コンピュータ／システム・エンジニアリング学部の Engineer in Residence に指名されました。