

# PythonまたはMATLAB環境でACEを用いてデータ・コンバータを短時間で評価する方法

Lluis Beltran Gil、製品アプリケーション・エンジニア

## 概要

評価用ボード（EVB）と付属のソフトウェアを用いると、プラグ・アンド・プレイ機能によりアナログ・デバイス製品の性能評価を容易に実行できます。グラフィカル・ユーザ・インターフェース（GUI）により、デバイスの設定と通信を手動で直感的に実行できます。しかし、より複雑な製品では、これらの反復タスクが自動化されていない場合、利用可能なすべての機能を評価しながら、製品のすべてのオプション機能まで掃引すると、長い時間が必要になることがあります。

本稿では、マクロを記録する方法と、Python®およびMATLAB®環境でマクロを使用し、複雑なソフトウェア・コントローラ・コードの開発を必要とせずに、特定の評価タスクを自動化する方法について説明します。本稿で使用するAD7380とAD7606C-16の例では、すべてのユーザ操作が自動化されるため、様々な設定での掃引の実行、変換の起動、結果のエクスポートができます。これらの例を本稿で用いた目的は、オーバーサンプリング・デジタル・フィルタがA/Dコンバータ（ADC）にもたらす性能上の利点を評価するタスクを自動化するためです。本稿で焦点を当てているのはAD7380とAD7606C-16ですが、本稿の情報はアナログ・デバイスその他の製品やアプリケーションにも適用できます。

## はじめに

アナログ・デバイスでは、新製品のリリースごとに、達成可能な性能を製品の機能として容易に評価できる手段を提供しています。まず、当社の評価用ボードは、主要デバイス（ADCなど）と、必要な周辺製品（電圧リファレンス、ドライバ・アンプ、クロック源、電源ソリューションなど）を搭載したプリント回路基板（PCB）で構成されています。次に、これらのボードには、ソリューションが達成可能な実際の性能を明示するソフトウェアが

付属しています。そのソフトウェアは、このプラットフォームで取得できる実際のデータを表示します。ここ数年、これを目的とするプラットフォームとして、ACEソフトウェア<sup>1</sup>が使用されてきました。

ACEプラットフォームでは、各製品にプラグインが搭載されており、このプラグインを用いて、すっきりとした操作しやすいGUIからデバイス・レジスタ・マップにアクセスすることで、デバイスの設定を容易に実行できます。

ADCの例にならって、ACEは通常、生データを波形またはヒストグラム形で表示し、FFTによりS/N比やTHDのようなAC性能指標を計算します。本稿では、ソフトウェア開発を必要とせずに、これらの手順をすべて自動化する方法について説明します。

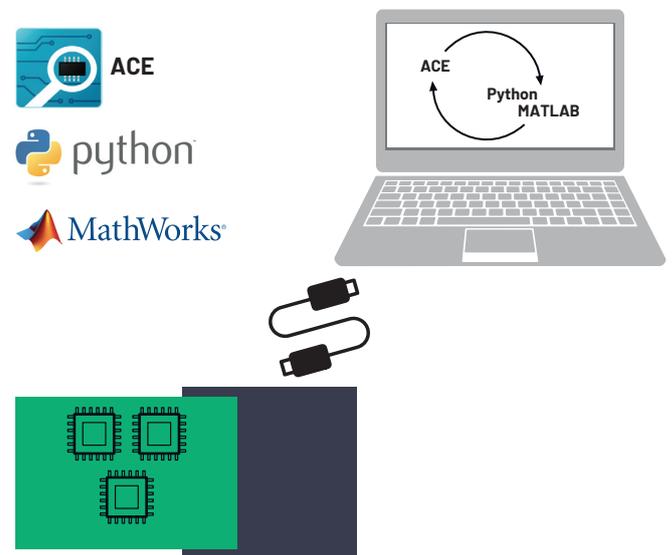


図1 ACEとPython/MATLABのやり取りを含む、評価用ボードのハードウェアとソフトウェア間のインタラクションを示すブロック図。



## ACEプラグイン

図2は、AD7380プラグインでのチップ概略図で、デバイスを容易に設定するためのいくつかのボタンが備わっています。

1. REFでは、内部または外部リファレンスの選択が可能です。
2. OSCでは、オーバーサンプリングを有効にし、オーバーサンプリング比を設定できます。

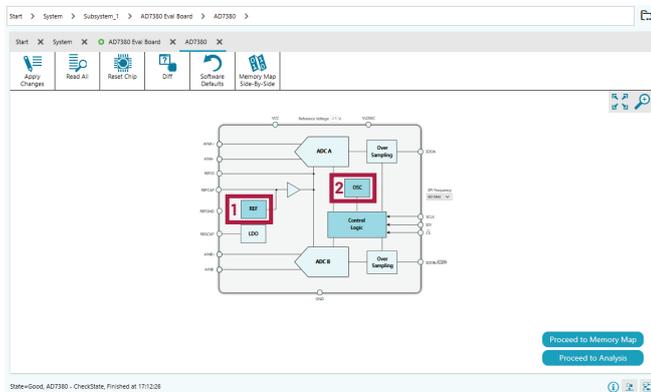


図2 AD7380のACEプラグインでのチップ概略図。

図3は、AD7606C-16プラグインでのチップ概略図で、濃い青色で強調表示されているいくつかのボタンを使用すると、以下のことができます。

1. PGAシンボルのクリックによる、チャンネルごとのアナログ入力レンジの選択。
2. 内部か外部かを問わず、リファレンス・ソースの選択。
3. オーバーサンプリング比の選択。
4. オフセット/ゲイン/位相キャリブレーション機能の設定。
5. 診断の設定。

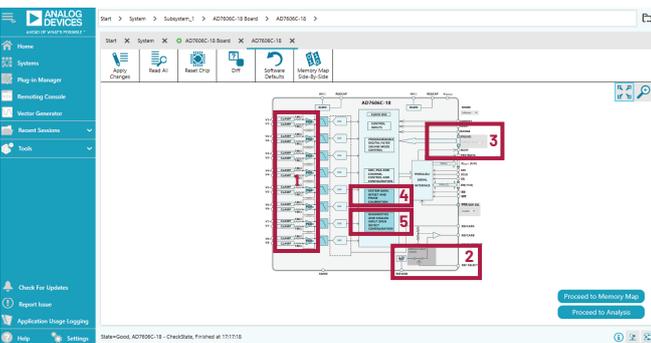


図3 AD7606C-16のACEプラグイン・チップの概略図。

これまで、AD7380とAD7606C-16を例として使用していますが、本稿が対象とする範囲は、これらの製品だけでなく、ACEでの作業と反復タスクの自動化に関連するすべての製品です。すべてのACEプラグインには、独自のGUIレイアウトがあります。しかし、製品の複雑さが増し、追加される機能（診断、キャリブレーションなど）が増えるにつれて、これらすべての機能を手動でテストする作業は、より複雑でうんざりするものになります。

ACEリモート・コントロールには、例えばオーバーサンプリング比を掃引し、ノイズ性能の改善の程度を測定するためのパラメータを設定または掃引する機能があります。

## ACEリモート・コントロールのセットアップ

ACEをリモート制御するコードの作成を開始する前に、次の手順に従って、ACEとの通信を確立する必要があります。

1. ACEを開き、[Settings]に進む。
2. [IPC Server]タブに進み、[Server enabled]が有効になっていることを確認する。
3. ポートが割り当てられていることを確認する（2357が望ましい）。

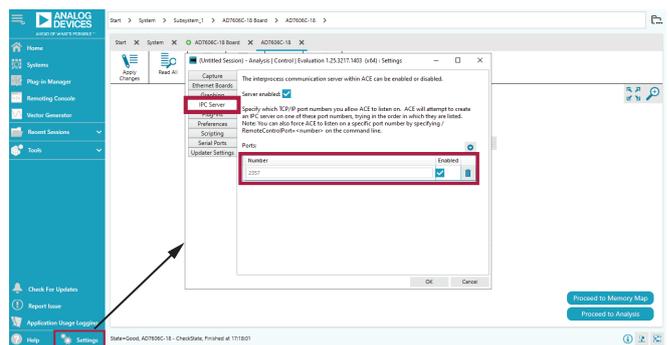


図4 ACEとの通信のセットアップ。

## マクロの記録とスクリプトの生成

ACEには [Macro Tools] メニューがあり、設定とデバイス間のインタラクション・ステップをマクロファイルに入力して記録します。マクロを生成すると、その後はマクロを用いて、すべてのステップを一度に自動的に繰り返すことができます。マクロを生成する手順を以下に示します。

1. ACEを開き、[Tools]、[Macro Tools]の順に進む。
2. 図5で強調表示されている青い領域内のアイコンの右側にある、[Record Macro] コマンド・ボタンをクリックします。
3. パラメータの変更、メモリ・マップ上のレジスタへの値の入力、データの取り込みなど、デバイス間のインタラクションを開始します。GUIを操作し対話するにつれ、Commandウィンドウが一杯になっていきます。
4. 入力すべきアクション/設定値がすべて記録されたら、[Stop Recording Macro] コマンド・ボタンをクリックします。場所は上記と同じです。すると、Commandウィンドウが自動的にポップ・アップ表示されます。

一例として、マクロの記録中に、ユーザは、図2で説明したボタンを用いて、オーバーサンプリング比と分解能モードを設定します。Commandウィンドウは図5のようになります。



## Pythonを用いた評価タスクの自動化

前のセクションで生成された.pyファイルを使用すると、以前に記録された一連のアクションを容易に起動して繰り返すことができます。更に、これらのスクリプトを編集し、特定のタスクを自動化すると、得られるメリットが一層大きくなります。AD7380 ACEプラグインでオーバーサンプリング比と分解能を設定した上記の例にならって、図7に示すように、生成されたマクロを編集して、入力パラメータを使用できるようにします。デフォルトでは、これらのマクロの名前は「execute\_macro()」となっていますが、より説明的な名前に変更できます。

```
def execute_macro(client, RES, OSR, sampleCount, DataSet):
    client.set_ContextPath("\\System\Subsystem_1\AD7380 Eval Board\AD7380")
    client.SetIntParameter("virtual-parameter-chkd-sample-count", str(sampleCount), "-1")
    client.SetIntParameter("RES", str(RES), "-1")
    client.SetIntParameter("virtual-parameter-OSR_Ratio", str(OSR), "-1")
    client.Run("@AsyncDataCapture"+"DataSet+")
    client.PullAllCaptureDataToStore("-1", "2000", "False", "True", DataSet,
        "Root::System.Subsystem_1.AD7380 Eval Board.AD7380 Analysis", "None")
    client.set_ContextPath("\\System\Subsystem_1\AD7380 Eval Board\AD7380\AD7380 Analysis")
    client.ExportStoreDataToFile(join(exportDatapath, DataSet), "acesamples", "", "")
    client.ClearStore()
    client.set_ContextPath("\\System\Subsystem_1\AD7380 Eval Board\AD7380")
```

図7 入力パラメータを使用するためのPython関数の編集。

このマクロは、ループ内で呼び出して、オーバーサンプリング比の値を掃引し、オーバーサンプリング比の増加につれてノイズ性能がどのように改善されるかを確認することや、2ビットの分解能を追加可能にする利点を確認することができるようになります。

```
def main():
    manager = ClientManager.Create()
    client = manager.CreateRequestClient("localhost:2357")
    vRef=3.3
    sampleCount=4096
    DataSet="DataSet_99"
    resolutionMode={0:"Normal Resolution",1:"2 extra bit Higher Resolution"}
    nbits={0:15,1:17}

    for RES in range(2):
        print('\n\n-----'+str(resolutionMode[RES])+ '-----')
        for OSR in range(6):
            execute_macro(client, RES, OSR, sampleCount, DataSet)
```

図8 これらの命令は、オーバーサンプリング比を掃引し、2ビットの追加の分解能をオン/オフするループのためのものです。

その結果、Pythonコードの例では、エクスポートされたデータファイルにアクセスし、ACEプラグインによって測定されたS/N比のデータを読み取り、シリアル・モニタを介してそのデータを報告します。表1に示すように、オーバーサンプリングはノイズ性能<sup>3</sup>を改善しますが、AD7380が16ビットではなく18ビットワードとして示した方が利点大きいことを実質的に確認するためには、ある時点で2ビットの追加の分解能が必要になります。

表1 AD7380におけるS/N比とオーバーサンプリング比 (OSR) の関係

OSR	ダイナミック・レンジ (通常の分解能)	ダイナミック・レンジ (分解能の増強、18ビット)
1	90.8dB	90.8dB
2	92.6dB	93.6dB
4	94.3dB	96.5dB
8	95.8dB	99.2dB
16	96.3dB	100.4dB
32	96.5dB	100.5dB

## MATLABを用いた評価タスクの自動化

Pythonの例と同様に、自動生成されたコードをMATLAB用にエクスポートすることもでき、この場合、.mファイルが生成されます。ここでは、マクロは関数として定義されていますが、入力パラメータの使用を可能にするよう編集することもできます。以下のMATLABの例では、代わりにAD7606C-16ボードを使用しています。

ここでも、AD7606C-16のオーバーサンプリングの利点が、ACEの自動化により容易に評価できます。図9はMATLAB関数を示しており、スループット、サンプル数、インターフェース・タイプ (シリアルまたはパラレル)、リファレンス・ソース (内部または外部)、オーバーサンプリング比、デジタル・ライン数、アナログ入力レンジなどのパラメータを受け取るように編集されています。

```
function configure_ad7606x(client, generic, throughput, no_samples, par_serb, ref_sel, OSR, sdo_lines, range)
    % Configure AD7606x interface reference source
    client.ContextPath["\\System\Subsystem_1\generic Board\generic"];
    client.SetDoubleParameter("virtual-parameter-throughput-display", num2str(throughput), '-1');
    client.SetIntParameter("virtual-parameter-sample_size", num2str(no_samples), '-1');
    client.SetBoolParameter("virtual-parameter-par_serb", par_serb, '-1');
    client.SetBoolParameter("virtual-parameter-reference_sel", ref_sel, '-1');
    client.SetIntParameter("virtual-parameter-os_sw", num2str(OSR), '-1');
    client.SetIntParameter("virtual-parameter-doutsel", num2str(sdo_lines), '-1');
    client.SetIntParameter("virtual-parameter-channel1_range_sw", num2str(range), '-1');
    client.SetIntParameter("virtual-parameter-channel2_range_sw", num2str(range), '-1');
    client.SetIntParameter("virtual-parameter-channel3_range_sw", num2str(range), '-1');
    client.SetIntParameter("virtual-parameter-channel4_range_sw", num2str(range), '-1');
    client.SetIntParameter("virtual-parameter-channel5_range_sw", num2str(range), '-1');
    client.SetIntParameter("virtual-parameter-channel6_range_sw", num2str(range), '-1');
    client.SetIntParameter("virtual-parameter-channel7_range_sw", num2str(range), '-1');
    client.SetIntParameter("virtual-parameter-channel8_range_sw", num2str(range), '-1');
end
```

図9 入力パラメータを使用するためのMATLAB関数の編集。

この特定の関数をメイン・コード内で使用すると、様々な構成のデータキャプチャを素早く起動して、AD7606C-16について次のようないくつかの比較を容易に実行することができます。

- ▶ 各オーバーサンプリング比でのノイズ性能の比較
- ▶ 各スループットでの性能の比較
- ▶ 各アナログ入力レンジでの性能の比較
- ▶ 内部リファレンスと外部リファレンスの比較
- ▶ シリアル・インターフェースとパラレル・インターフェースの比較

その他、多数の比較が可能です。これはAD7606C-16の例に過ぎません。評価対象の製品によっては、様々な性能評価のニーズが生じることがあります。

AD7606C-16の参考資料については、[wiki.analog.com](http://wiki.analog.com)<sup>4</sup>を参照してください。ソースコードも入手できます。ここでは、マクロのコーディング方法だけではなく、データの後処理の例も参照できます。OversamplingSweep.mの例では、オーバーサンプリング比を掃引し、各値についてコードのヒストグラムをプロットしています。図10に示すように、オーバーサンプリング比が大きいくほど、コードのヒストグラムの幅は狭くなります。表2では、コードのピークtoピーク分布がオーバーサンプリング比によってどのように減少するかも示しています。

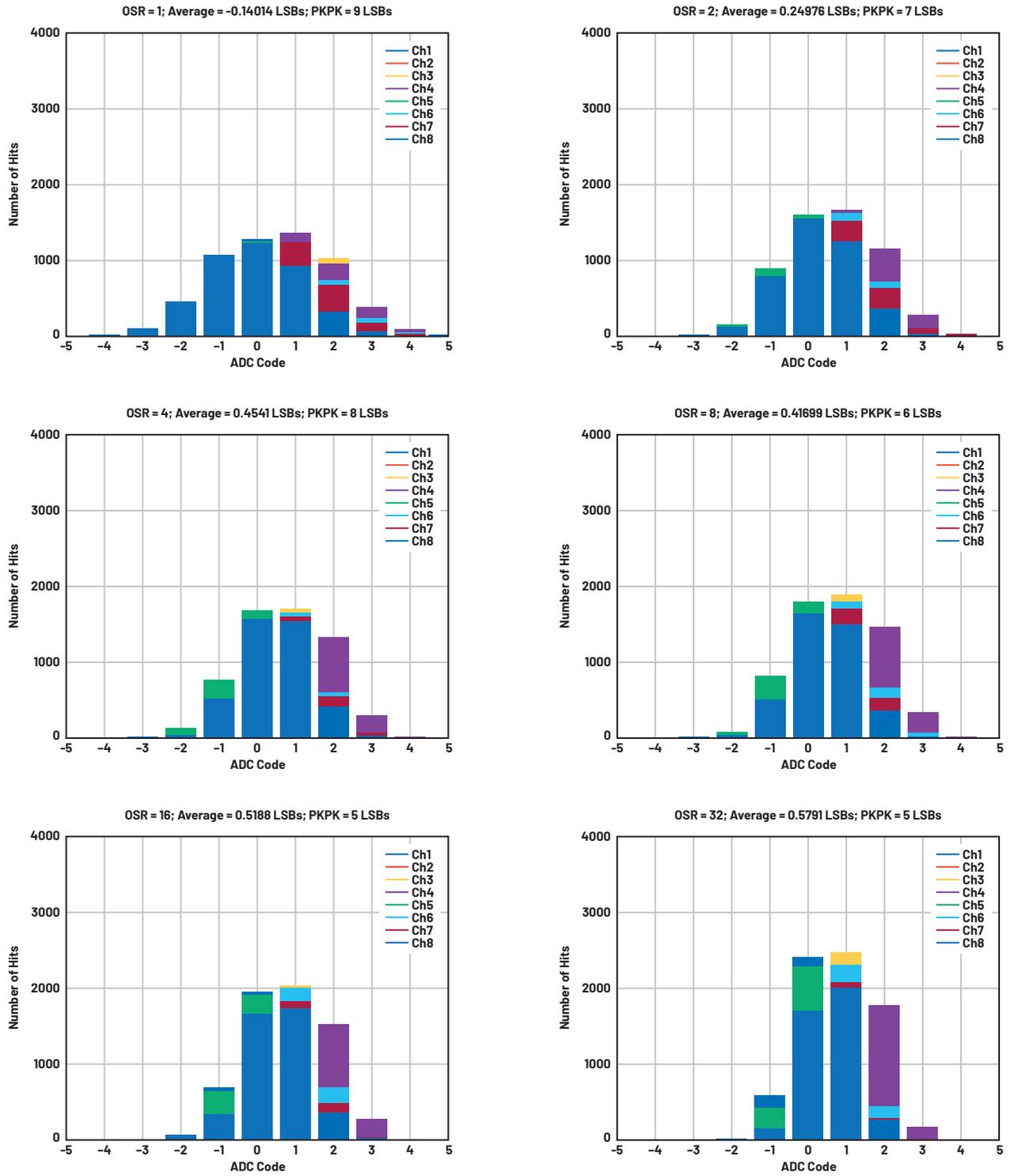


図10 コードのヒストグラムは、オーバーサンプリング比によってノイズ性能がどのように改善されるかを示しています。簡単なスクリプトにより、オーバーサンプリング比の多くの選択肢（最大256まで利用可）を掃引することが可能になります。

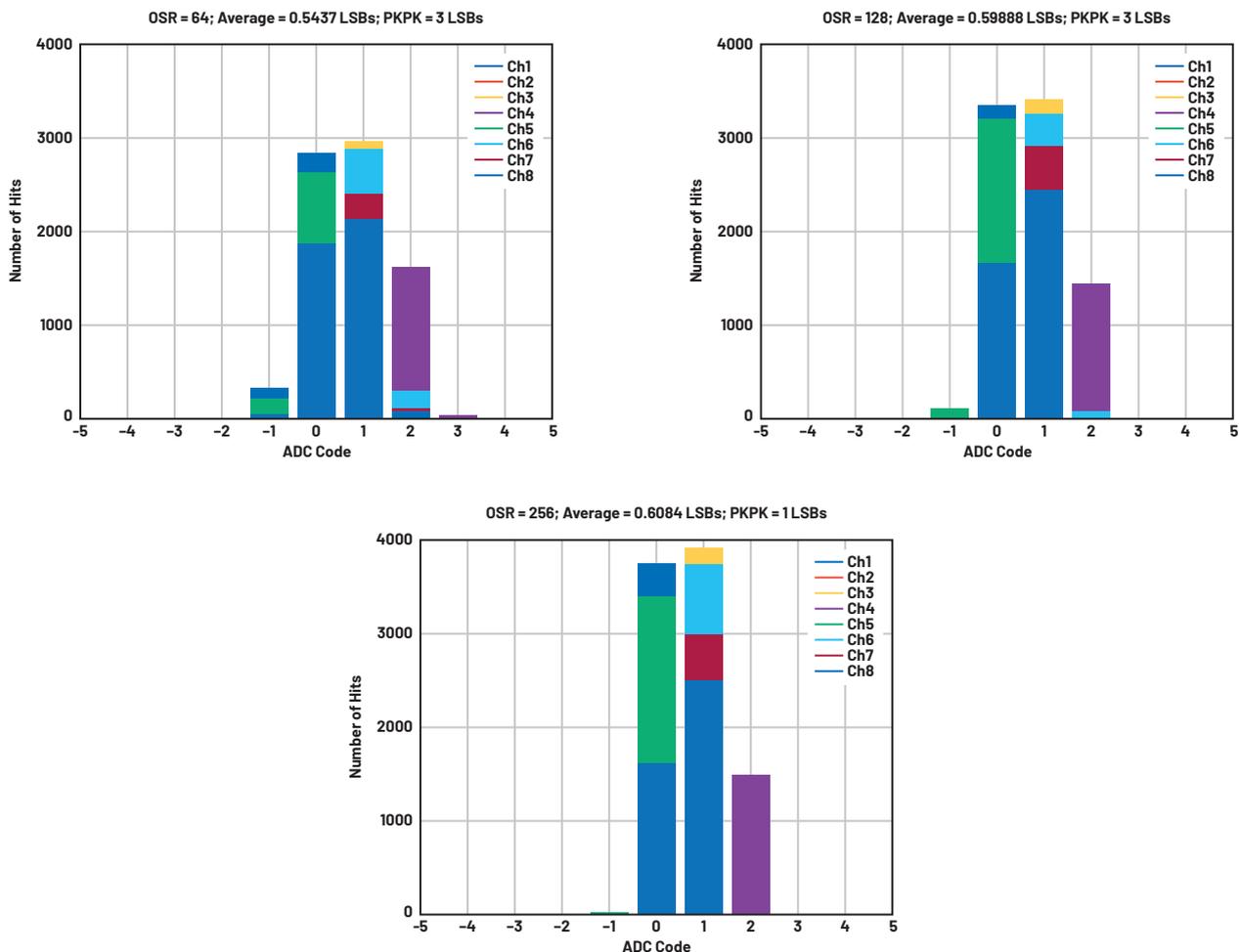


図10（続き） コードのヒストグラムは、オーバーサンプリング比によってノイズ性能がどのように改善されるかを示しています。簡単なスクリプトにより、オーバーサンプリング比の多くの選択肢（最大256まで利用可）を掃引することが可能になります。

表2 AD7606C-16のコード分布とオーバーサンプリング比（OSR）の関係

OSR	ピークtoピーク(LSB)
1	16
2	13
4	12
8	10
16	7
32	6
64	4
128	4
256	2

## まとめ

最新の評価用ボードのソフトウェア向けのフレームワークである新しいACEプラットフォームは、マクロを記録する機能があるため、アナログ・デバイセズ製品の評価作業を短時間化したいと考えているハードウェア設計者にとってメリットの大きなツールです。これらのマクロにより、PythonやMATLABのような

業界で広く知られているプログラミング言語を介したテストの自動化が可能になります。同様に、これらのタスクを自動化することで、製品の評価や選定の作業を高速化できます。その結果、市場投入までの時間とハードウェア設計サイクルが短縮されます。このトピックや他のトピックについての更なるサポートは、EngineerZone™でいつでも受けられます。他のリモート・コントロールの例は [wiki.analog.com](http://wiki.analog.com) で参照できます。

## 参考資料

- 1 「分析！制御！評価用（ACE）ソフトウェア」 アナログ・デバイセズ
- 2 「AD5791 ACE Remote Control」 アナログ・デバイセズ、2020年2月
- 3 Jonathan Colao 「モータ駆動用の小型フィードバック・システム、高速応答の光学式エンコーダに対応」 Analog Dialogue, Vo. 54, No. 2, 2020年4月
- 4 「AD7606B/C ACE Remote Control」 アナログ・デバイセズ、2021年2月

## 著者について

Lluis Beltran Gilは、バレンシア工科大学（UPV）で、2009年に電子工学、2012年に生産管理工学の修士号を取得しています。卒業後の2013年に、リムリックの高精度コンバータ・グループ所属のアプリケーション・エンジニアとして、アナログ・デバイセズに入社しました。現在、計測器BU内のSAR ADCアプリケーション・チームに所属して、スペインのバレンシアに居住しています。バレンシア大学（UV）で電子工学の修士号も取得しています。

## EngineerZone®

### オンライン・サポート・コミュニティ

アナログ・デバイセズのオンライン・サポート・コミュニティに参加すれば、各種の分野を専門とする技術者との連携を図ることができます。難易度の高い設計上の問題について問い合わせを行ったり、FAQを参照したり、ディスカッションに参加したりすることが可能です。



Visit [ez.analog.com](https://ez.analog.com)

\*英語版技術記事は[こちら](#)よりご覧いただけます。