

演習 1: 環境のセットアップ

この演習では、ハードウェアを準備して、ワークショップのコンテンツをローカルにダウンロードし、クラウドコンポーネントを初期化します。また、アマゾン ウェブ サービス (AWS) のユーザー アカウントにログインし、このワークショップで使用するザイリンクス ハードウェアの認証情報を設定し、ザイリンクス デバイスのデータとデバイス シャドウを追跡するためのインターフェイスを用意します。

クレジットの適用

このワークショップに関連して発生する AWS クラウド利用料金分として事前にクレジットを受け取っている場合は、カードに記載の手順に従って、クレジットを適用してください。

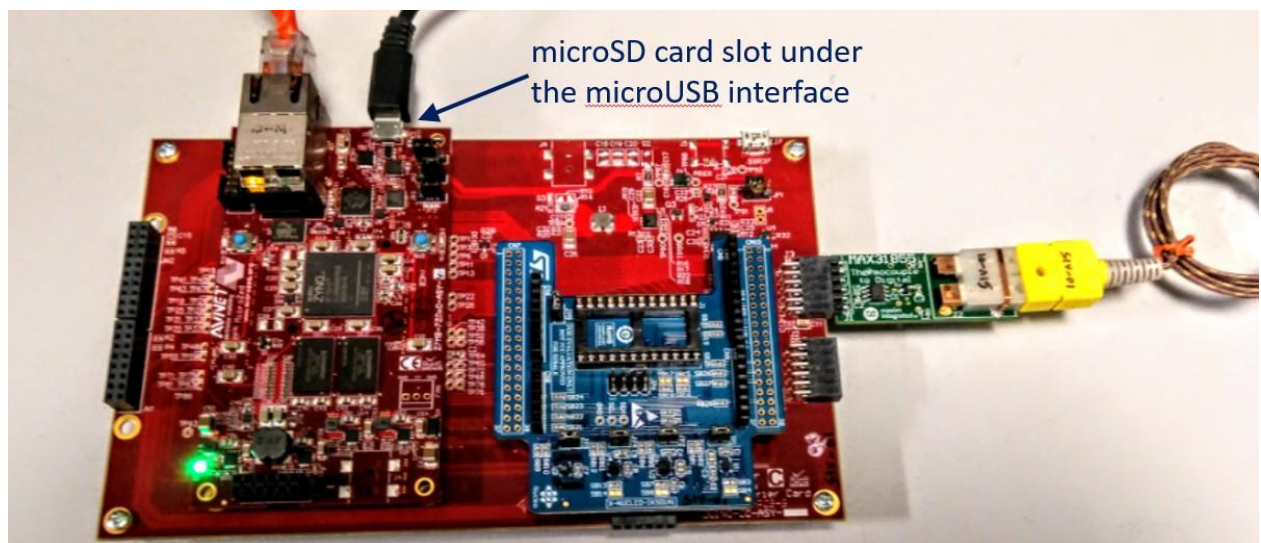
ザイリンクス ハードウェアのセットアップ

ハードウェアの物理的な設定のほとんどはワークショップ開始前に完了しています。このセクションでは、各自がハードウェアを簡単にチェックできるように、ハードウェアの物理的な設定手順について説明します。

Ultra96 と MicroZed には、いずれもワークステーションとの接続に使用するシリアル コネクタが 1 つあります。このワークショップでは主に Ultra96 のシリアル ポートを使用しますが、デバッグが必要になったときのために、両方のプラットフォームのセットアップ手順について説明します。

Avnet 社 MicroZed IIoT キット

1. Arduino キャリア カードに STMicroelectronics X-NUCLEO シールドを取り付けます。これにより、センサーが FPGA プログラマブル ロジックに接続されます。
2. Maxim 31855 PMOD 熱電対を Arduino キャリア カードの「PL_PMOD」と書かれた J3 コネクタに接続します。J3 の上側の列に、1 ピンの位置を合わせて挿入します。これにより、センサーが FPGA プログラマブル ロジックに接続されます。
3. Arduino キャリア カードに MicroZed SoM (System on Module) を取り付けます。ここまでの手順が完了すると、システムは次のようになります。



- ブートモードジャンパー (JP1 ~ JP3) が SD カードブート (JP1: ピン 1-2、JP2: ピン 2-3、JP3: ピン 2-3) に設定されていることを確認します。次に、正しいジャンパー設定を示します。

SD Card



- MicroZed SoM の RJ45 コネクタと、テーブル上のイーサネット スイッチをイーサネット ケーブルで接続します。

MicroZed はこの時点ではこのままにし、Ultra96 プラットフォーム経由で microSD カードに内容を書き込んだ後、演習 2 の最後に電源を投入します。MicroZed ボードには、J2 および J7 の microUSB コネクタから電源を供給します。

MicroZed は Silicon Labs CP2104 USB-to-UART Bridge を使用します。OS でドライバーが自動検出されない場合は、次のページからダウンロードできます。 <https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcv-drivers>

シリアル ポートのインストールと接続: Windows

PuTTY や Tera Term などのシリアル ポート ターミナルをインストールしてあることを確認してください。

- [デバイス マネージャー] を開きます。現在利用可能な COM ポートを確認します。
- MicroZed デバイスを接続します。
- どの COM ポートが変化するかを確認します。
- その COM ポートに、PuTTY や Tera Term などのシリアル プログラムを使用して 115200-N-1 で接続します。

シリアル ポートのインストールと接続: MacOS/Linux

ほとんどすべての Linux ディストリビューションには screen がインストールされています。MacOS にはデフォルトで screen がインストールされています。

- MacOS の場合はターミナル ウィンドウ、Linux の場合は XTerm などのターミナル プログラムを開きます。
- 次のコマンドを実行して、現在のシリアル デバイスを確認します。

```
ls /dev/cu.usb*
```

- MicroZed を接続します。
- もう一度次のコマンドを実行して、MicroZed デバイスを見つけます。

```
ls /dev/cu.usb*
```

5. リストのどの部分が変わったかを確認して、次のコマンドを実行します。

```
screen /dev/<device> 115200 -L
```

たとえば、`cu.usbmodem0003` の部分が違う場合、コマンドは次のようになります。

```
screen /dev/cu.usbmodem0003 115200 -L
```

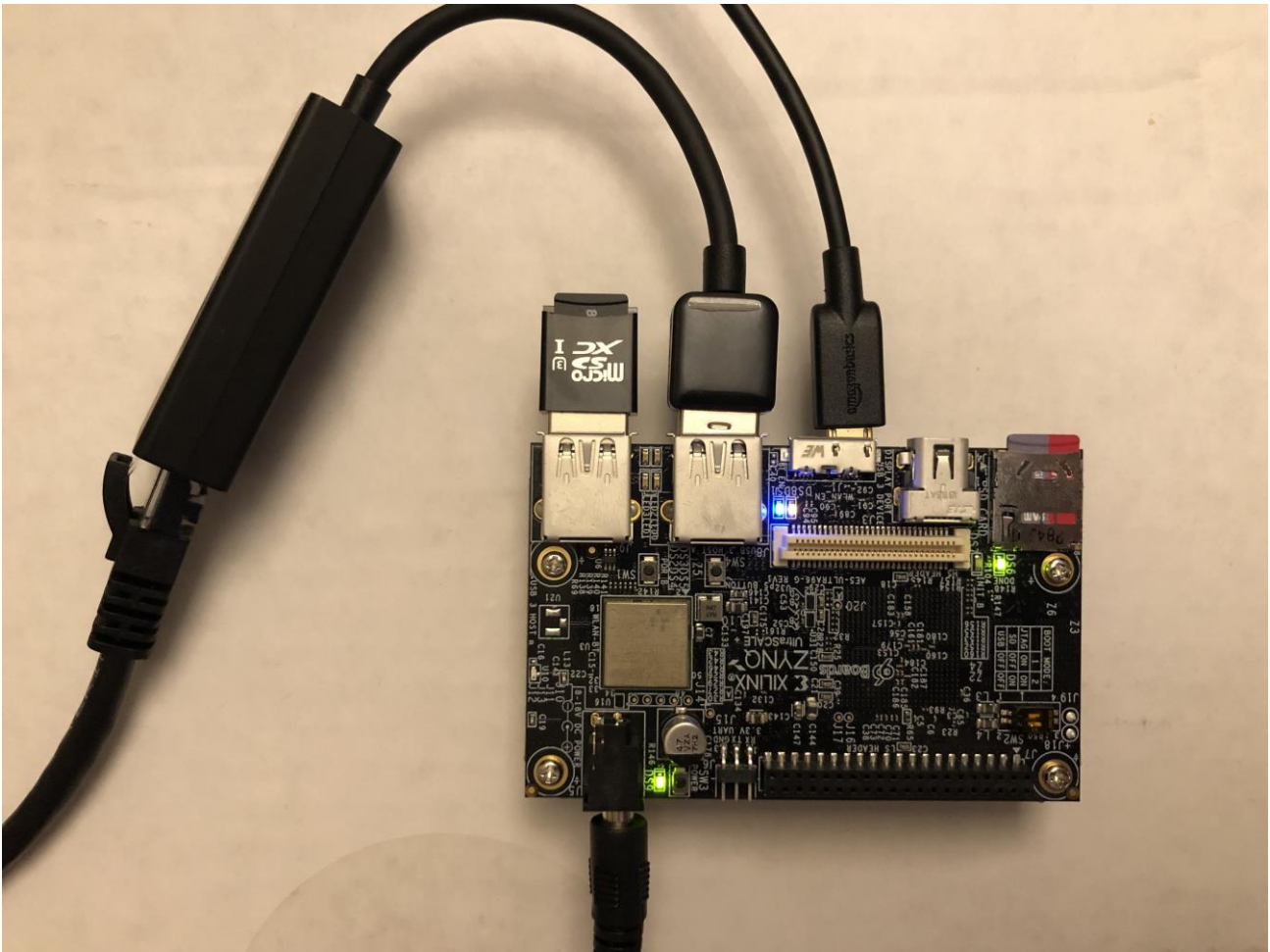
MacOS の場合は、ターミナルの代わりに **Serial** の使用も推奨します。このアプリケーションは有料ですが、MacOS 上で動作するアプリケーションによくある便利な機能が豊富に揃っています。

Avnet Ultra96

このワークショップ用に Ultra96 のビルド済みイメージが用意されています。まず、このイメージをダウンロードして、Ultra96 キットに付属の 16GB microSD カードに書き込み、ボードをブートします。手順は次のとおりです。

1. [ここ](#)から Ultra96 AWS re:Invent 2018 の SD カード イメージをダウンロードします。ファイルの MD5 チェックサムは、E385BE22AD5506A33F4B5D5CCA0FD3DF です。
2. ファイルを解凍し、Win32DiskImager などのプログラムを使用してイメージを microSD カードに書き込みます。
3. microSD カードを挿入します。
4. J5 に 12V 電源を接続します。
5. J9 に USB イーサネット アダプターを接続し、アダプターとテーブル上のイーサネット スイッチをイーサネット ケーブルで接続します。
6. microUSB ケーブルで USB ハブと Ultra96 の J1 を接続します。この USB インターフェイスを介して PC には次の 3 つのサービスが提供されますが、このワークショップで使用するのは UART インターフェイスのみです。
 - i. デバッグ UART インターフェイス
 - ii. 「PYNQ-USB」という名前のポータブルドライブ (Ultra96 のファイル システムへのアクセス)
 - iii. RNDIS (USB イーサネット) インターフェイス
7. 8GB の microSD カードを挿入した状態で USB-microSD アダプターを USB 端子に接続します。これは、演習 2 で Ultra96 を使用して MicroZed のブートイメージを書き込む際に使用します。

次に、ここまでの Ultra96 の完成図を示します。



シリアルポートのインストールと接続

Ultra96 は Gadget Serial Driver を使用します。OS でドライバーが自動検出されない場合は、「PYNQ-USB」と表示される Ultra96 ポータブルドライブの「serial_driver」フォルダーからドライバーをロードします。

オペレーティング システムの種類 (Windows または MacOS/Linux) に応じて次の手順を実行し、Ultra96 のターミナル エミュレーターを開きます。

シリアルポートのインストールと接続: Windows

1. [デバイス マネージャー]を開きます。現在利用可能な COM ポートを確認します。
2. MicroZed デバイスを接続します。
3. どの COM ポートが変化するかを確認します。
4. その COM ポートに、PuTTY や Tera Term などのシリアル プログラムを使用して 115200-8-N-1 で接続します。

シリアルポートのインストールと接続: MacOS/Linux

1. MacOS の場合はターミナル ウィンドウ、Linux の場合は XTerm などのターミナル プログラムを開きます。
2. 次のコマンドを実行して、現在のシリアル デバイスを確認します。

```
ls /dev/cu.usb*
```

3. MicroZed を接続します。

- もう一度次のコマンドを実行して、MicroZed デバイスを見つけます。

```
ls /dev/cu.usb*
```

- リストのどの部分が変わったかを確認して、次のコマンドを実行します。

```
screen /dev/<device> 115200 -L
```

たとえば、`cu.usbmodem0004` の部分が違う場合、コマンドは次のようになります。

```
screen /dev/cu.usbmodem0004 115200 -L
```

ハードウェアの設定とデプロイ

重要: このワークショップでは、VPN ソフトウェアは実行しないでください。

- 必要であればログインします。ログイン、および後々の `sudo` アクセスが必要なコマンドに使用するユーザー名は `xilinx`、パスワードは `xilinx` です。`sudo` パスワードも `xilinx` です。
- コマンド `ip a` を実行し、すべてのイーサネット インターフェイスを確認します。次のものが表示されます:
 - `usb0` (アドレス `192.168.3.1/24`)
 - `eth0` (アドレスは DHCP サーバーで決定)。これは、AWS Greengrass を実行する Ultra96 のアドレスです。
 - その他のインターフェイスは、このワークショップでは使用しません。
- コマンド `ping -c 3 aws.amazon.com` を実行し、インターネットに接続されていることを確認します。

AWS クラウドのセットアップ

前提条件

bash シェルを使用した Linux コマンドラインの実行に関する基礎知識が必要です。

このワークショップでは AWS クラウドで関数を実行するため、Git と AWS コマンドライン インターフェイス (CLI) をインストールしておく必要があります。Ultra96 のルート ファイル システムには、これらのコマンドが含まれます。演習で使用するシェル コマンドはすべて、デフォルト ユーザーの `xilinx` でログインした後に、Ultra96 のターミナル エミュレーターで実行することを前提としています。

この後のスクリプトを正しく実行するには、IAM ユーザーにアクセス許可の境界 (Permissions Boundary) のない `AdministratorAccess` ポリシーが付与されている必要があります。このポリシーでは非常に広い権限が与えられます。これよりも権限の少ないポリシーでもスクリプトを実行できることがあります。

新規 IAM ユーザーの作成

次のウェブ ページにアクセスします。

https://docs.aws.amazon.com/IAM/latest/UserGuide/getting-started_create-admin-group.html

「Creating an Administrator IAM User and Group (管理者 IAM ユーザーおよびグループの作成 (コンソール))」を参照します。

手順 1 ~ 14 を実行します (ただし手順 4 と 13 は次の手順に差し替えます)。

手順 4: [Programmatic access] をオンにします (これにより、アクセス キーとシークレット キーを生成できます)。

手順 13: この手順はスキップします。

ワークショップ リポジトリのクローン作成

このセクションでは、ワークショップの Git リポジトリのクローンを作成します。Git リポジトリには、ワークショップで使用するすべてのコードとスクリプトが含まれます。

1. Ultra96 デバイスへの UART 接続を 115200-8-N-1 として開き、ユーザー名 xilinx としてログインします。
2. \$HOME ディレクトリに移動します。

```
cd $HOME
```

3. 次のコマンドを実行して、リポジトリのクローンを作成します。

```
git clone https://github.com/rpcme/aws-cloud-and-xilinx-workshop
cd aws-cloud-and-xilinx-workshop
```

リポジトリのサイズは全体で 30MB 以上あるため、ダウンロードには少し時間がかかります。そのまま待ってください。

ダウンロードが完了したら、次のセクションに進みます。

AWS IoT コンソールへのサインイン

このセクションでは、AWS IoT コンソールにログインします。このインターフェイスは、ザイリンクス エッジ デバイスのモニターとテストに使用するため、演習が終わるまで開いたままにしておいてください。

[AWS コンソールのサインイン](#) ページにアクセスし、右上にある [Sign In] をクリックします。AWS 認証情報を使用してログインし、IoT コア ページへ移動します。この後のいくつかの演習では、IoT コアで提供されるサービスを使用します。

AWS コマンドライン インターフェイス (CLI) の設定

このセクションでは、Ultra96 ボードで AWS CLI を設定します。AWS CLI には、AWS IoT コンソールのクラウド アクションをエッジ ターゲット (この場合は Ultra96) 上の CLI から駆動するメカニズムが用意されています。設定の詳細は、「[Configuring the AWS CLI \(AWS CLI の設定\)](#)」を参照してください。このページに記載されている手順 1 ~ 5 を実行し、自身 (のユーザー名) に AdministratorAccess ポリシーを追加し、アクセス キー ID とシークレット アクセス キーをキャプチャします。これらは、この後の手順で必要になります。

AWS CLI は既に Ultra96 にインストールされています。

次のコマンドを実行して、設定を開始します。

```
aws configure
```

このコマンドを実行すると、次の情報の入力が必要です。

1. 上で取得した AWS アクセス キー ID をタイプまたはコピー アンド ペーストして、Enter キーを押します。
2. 上で取得した AWS シークレット アクセス キーをタイプまたはコピー アンド ペーストして、Enter キー

を押します。

3. デフォルトリージョン名: `us-west-2`
4. デフォルト出力形式: `json`

最初の 2 つは暗号化されずに `~/.aws/credentials` ファイルに保存され、それ以外は `~/.aws/config` に保存されます。セキュリティ保護のため、認証情報ファイルはワークショップが終わったら削除してください。

AWS Artifact のデプロイ

このセクションでは、AWS Cloudformation を使用して AWS アカウントに AWS Artifact をデプロイします。

1. クラウド オブジェクトをデプロイするスクリプトのあるディレクトリに移動します。

```
cd $HOME/aws-cloud-and-xilinx-workshop/cloud/script
```

2. 次に示すスクリプトを実行すると、アカウントに S3 バケットが作成され、この後の演習全体を通じて使用する AWS Artifact が追加されます。このコマンドには、ユーザーが定義した一意の接頭辞を付ける必要があります。この接頭辞は、このワークショップを通じて使用します。

```
./deploy-s3-objects.sh <prefix>
```

接頭辞が既に使用されています、というエラーが表示された場合は、別の接頭辞を選んでください。

このスクリプトを実行すると、`<prefix>-aws-cloud-and-xilinx-workshop` という名前の Amazon S3 バケットが作成されます。

まとめ

この演習では、ワークステーションに必要なものをインストールした後、AWS アカウントにログインして演習に必要なものを AWS クラウドにインストールしました。

[次の演習](#)

[トップ ページ](#)

この資料は英語版を翻訳したもので、内容に相違が生じる場合には原文を優先します。資料によっては英語版の更新に対応していないものがあります。日本語版は参考用としてご使用の上、最新情報につきましては、必ず最新英語版をご参照ください。