



# Generic Developer Hardware Set-up Guide

## Streamlined instructions for activating imp development kit

This guide is intended to help you get any imp-based development hardware online, whatever imp it contains and by whatever means it uses to connect to the Internet. The guide will refer to all such imp-enabled development hardware as 'imp kits', so whenever you see this term, you should just substitute in the name of the breakout board or device you'll actually be using. We also provide a number of hardware-specific getting-started guides; they are listed [here](#).

**Note** This guide only applies to development devices. Commercial imp-enabled products have their own configuration apps, and you should follow the product manufacturer's own guidelines.

In addition to your imp kit, you will need a suitable power source. This depends on what power options your imp kit offers, so check the device's documentation if you're unsure which to use. You'll also need a mobile phone or tablet running iOS or Android. This is used to activate the imp kit and provide it with network access credentials.

## 1. Create a Free Electric Imp Account

The first step to getting your imp kit up and running is to create an Electric Imp account. This is free and gives you access to Electric Imp's online code development and product management application, impCentral™, which you will use to program and monitor your imp kit. Code is deployed from impCentral to imp kits via the Internet. If you already have an Electric Imp account, you can skip to Step 2.

Electric Imp provides two impClouds: one hosted on Amazon Web Services (AWS) and one hosted on Microsoft Azure. You can create an account on either impCloud, but development devices are bound to specific impClouds, so you must ensure you either create an account on the impCloud your device is tied to, or buy development hardware that matches the impCloud that's hosting your account. For example, if you create an account on the Azure impCloud, you will need to use development hardware marked as Azure compatible. An AWS account cannot 'see' devices bound to the Azure impCloud, and vice versa.

Currently, the Electric Imp Store provides Azure-compatible [imp004m breakout boards](#) and [impAccelerator™ Fieldbus Gateways](#) — all other hardware offered by the Store and by third-parties is tied to the AWS impCloud.

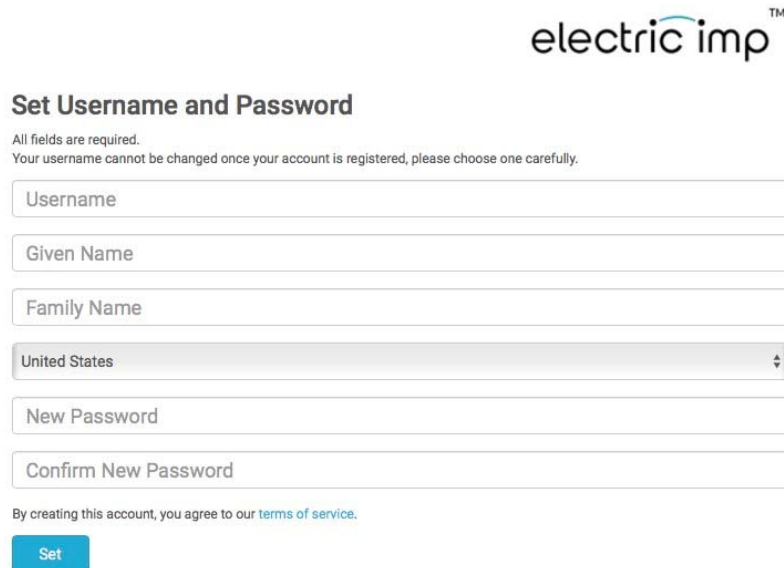
Create your account now: visit the [AWS impCloud](#) or the [Azure impCloud registration page](#) and enter your email address in the **Sign Up** section:



The screenshot shows the Electric Imp logo at the top right. Below it, the "Sign In" section has two yellow input fields for email and password, a blue "Sign In" button, and a "Forgot Password?" link. The "Sign Up" section has a white input field for "Your Email" and a "Sign Up" button.

*The impCentral sign-in and registration screen*

You'll now receive a message asking you to verify the email address you supplied. When you've done so, by clicking the **Confirm** button in the email, you'll be automatically returned to the impCentral login page and asked to choose an account username and password, and provide some basic information about yourself. Your password must contain between eight and 20 characters. Your username must also contain 8-20 characters, and must not be your email address. Click **Set** to apply your choices:



The screenshot shows the Electric Imp logo at the top right. Below it, the "Set Username and Password" section has a note: "All fields are required. Your username cannot be changed once your account is registered, please choose one carefully." There are input fields for "Username", "Given Name", "Family Name", a dropdown menu for "United States", "New Password", and "Confirm New Password". At the bottom, there is a "By creating this account, you agree to our [terms of service](#)." and a blue "Set" button.

*The account completion screen*

**Note** By creating an account, you agree to Electric Imp's Terms of Service, which you can [read here](#). Please also take a moment to view our [Privacy Policy](#).

## 2. Activate Your imp Kit

In order to program your imp kit, you need to add it to your account. This process is called activation. If your imp kit will connect to the Internet via WiFi, you'll also need to tell it which network to connect to. Both of these tasks are accomplished using Electric Imp's BlinkUp™ technology.

BlinkUp uses a mobile device's display to transmit activation data to the imp kit. You use the Electric Imp app to activate development devices. BlinkUp is also used to activate commercial devices based on the Electric Imp platform, but using their own mobile apps. By default, commercial products can't be used for development.

Download the Electric Imp app from either the [iTunes App Store](#) or [Google Play](#) according to what make of mobile device you use. When the app has downloaded, launch it and sign in using the account username and password you set up in Step 1:

Email or Username

Password

### Select Your impCloud

Make sure the correct impCloud — AWS or Azure — is selected for your imp kit (*see above*). Tap on the switch to select your account and hardware's specific impCloud:

Select your impCloud™

Amazon AWS  Microsoft Azure

### Select Your Network Mode

The Electric Imp app walks you through the process of activating your imp kit. After signing in, tap **Configure a Device** and then select your network type: **Wireless** or **Ethernet**. All imps support WiFi, but only the imp005 supports Ethernet, though a given imp005-based imp kit may not necessarily include an Ethernet port.



Use Wireless if you plan to connect your imp Kit by either WiFi or Ethernet. Only select Ethernet, if you don't intend to connect your imp kit by WiFi. If you choose Ethernet but later want to connect wirelessly, just re-configure the imp kit, this time using the the Wireless option. You can also re-configure your imp kit whenever you need to change the imp kit's network settings.

## Wireless Network Requirements

All imps use **2.4GHz 802.11n WiFi**. The imp005 can also connect 802.11n networks in the 5GHz band. The Electric Imp app defaults to the WiFi network your mobile device is connected to. This may not be a network that is compatible with your imp kit, but you can change the target network in the app by tapping the **Network Name** field.

If your imp kit was purchased in the US, it will only be able to connect to certain WiFi channels, eg. 2.4GHz channels 1-11, for regulatory reasons. Make sure your wireless router or access point is operating on a compatible channel. This is especially important if it chooses channels automatically.

All imps are able to detect what kind of security — WEP, WPA or WPA2 — a WiFi network uses, so just enter the correct password. However, the imp doesn't support enterprise-level authentication (802.11x), which requires that you log in with a username as well as the customary SSID and password, or networks which require login via a web page. If you're connecting to an unsecured network, leave the **Password** field blank.

- You can read more about imp network requirements [here](#).

**Hint** You can save your network settings in the Electric Imp app for future use. To retrieve them, tap the **triangle** in the **Network Name** field: you'll be presented with a list of stored networks — just tap the one you want to use.

## BlinkUp

Tap **Next** in the app, and keep doing so until you see the **BlinkUp** page:

 **Do not look at the screen during BlinkUp**

START BLINKUP

If required, connect an Ethernet cable to your imp kit. Power up the kit and when you see its BlinkUp status LED flashing (this will be orange if this is the first time the imp kit has been used), tap the app's **Start BlinkUp** button. Quickly lay the mobile device's screen face down over the status LED. After a three-second countdown, BlinkUp commences; when BlinkUp is complete, the mobile device will signal this with a beep and a buzz.

**Always run a BlinkUp with the screen facing away from you and towards the device being configured, especially if you suffer from a medical condition that makes you susceptible to strobing light. In the iOS app there are audio prompts both at the start and end, so you can close your eyes during the process. The Android app vibrates the handset after BlinkUp is done.**

Modern imp kits signal a successful BlinkUp by keeping the status LED lit green for **three seconds**. After that, various **color combinations** may be seen as the imp connects and activates, culminating in a steady green blinking. If the LED doesn't blink green, the device has not been able to connect, and you should check out the next section.

Connected or not, after a short time the imp kit's LED will turn off to save power. This is normal, and the device is still running. But if you need to configure your imp kit again, you'll need to power-cycle it: an active LED tells you your imp kit is ready for BlinkUp.

## Getting BlinkUp Right First Time


A **fast flashing red LED** tells you that the device wasn't able to read the BlinkUp transmission — perhaps you held it too far from the phone's screen. BlinkUp can also be adversely affected by unusual lighting conditions, especially very bright lighting such as direct sunlight. For best results, we recommend that you:

- Try to block out any direct bright lights, or perform a BlinkUp in the shade.
- Hold the device still during the process.

## Troubleshooting

If your imp kit doesn't connect, try BlinkUp again. Keep an eye on the LED after BlinkUp: the **pattern of colors and flashes** will tell you where any problem is occurring. You may have entered the WiFi password incorrectly, for example, or selected a network your imp kit is not compatible with.

### 3. Programming Your imp Kit

At this point, your imp kit should be online and activated. It will be now listed in impCentral: log in to [impCentral on the AWS impCloud](#) or [impCentral on the Azure impCloud](#) on your computer, click on the  at the top and select **My Development Devices**. Your imp kit is currently 'unassigned', which simply means that it is not yet associated with any impCentral Device Group.


impCentral organizes devices and code as follows:

- **Product** This is the top-level project container. It's typically named for the product you're creating.
  - **Device Group** These group devices according to the code they run. All devices that are assigned to a given Device Group run the same code, which can be viewed and modified the code editor. There are four types of Device Group:
    - **Development** Where you develop and test application code.
    - **Factory Test** Where you develop and test factory code, if your account has permission to do so.
    - **Production** Where you deploy production application code, if your account has permission to do so.
    - **Factory** Where deploy production factory code, if your account has permission to do so.

Development devices are assigned to Development Device Groups and, once assigned, will download and run any code deployed to that Device Group. Each Development Device Group has a code editor — changing the code and clicking the editor's **Build and Force Restart** button will automatically cause all of the group's devices to restart and receive the new software.

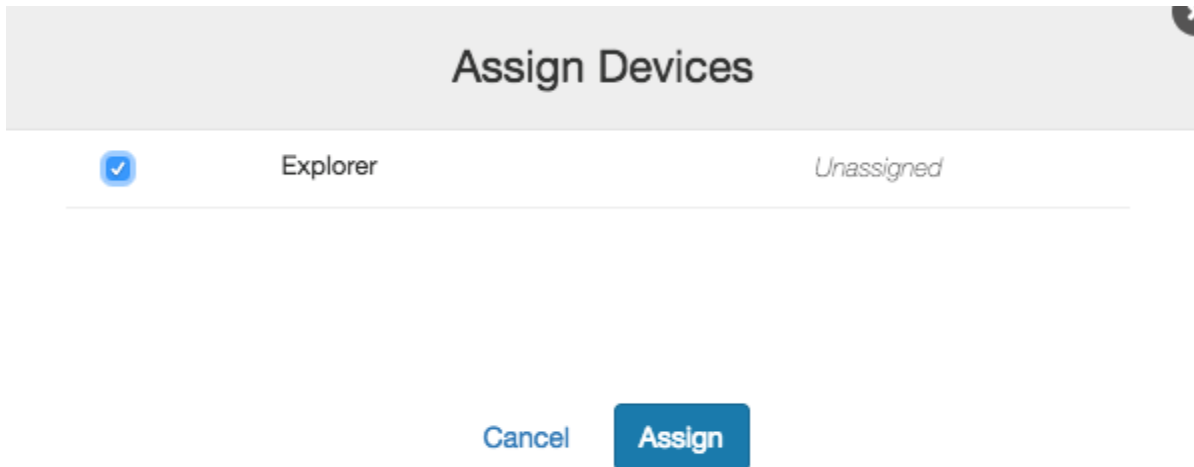
Find your account name in the impCentral navigation bar, then click on the **Account** menu beneath it and select your personal account. This will display the selected account's Products. Click **Create New Product**. The **Create Product** panel will now appear for you to give your first Product a name and to name its first Development Device Group too. Enter the two names then click **Create**.

impCentral takes you straight to the code editor, which you'll use to develop application code. To test that code, you need hardware, such as your imp kit. For the kit to run the code you work on in the Device Group, it needs to be 'assigned' to that Device Group. Look in the **log pane** at the bottom of the window and you'll see an **Assign devices** button:



Assign devices

Click this and a panel will appear which allows you to select one or more of the development devices linked to your Electric Imp account and assign them to the current Device Group:



Find your imp kit by name or device ID, select it and click **Assign**. It will now be added to the Development Device Group.

## Agents and devices

Electric Imp application firmware comprises two parts. The first, the device code, runs on the imp kit itself. The second runs within the Electric Imp impCloud and is called the device's agent. Each device has its own, unique agent, so if you have ten imp kits assigned to models, there will be ten agents running in the cloud, all using the model's agent code. Each device has a one-to-one relationship with its agent.

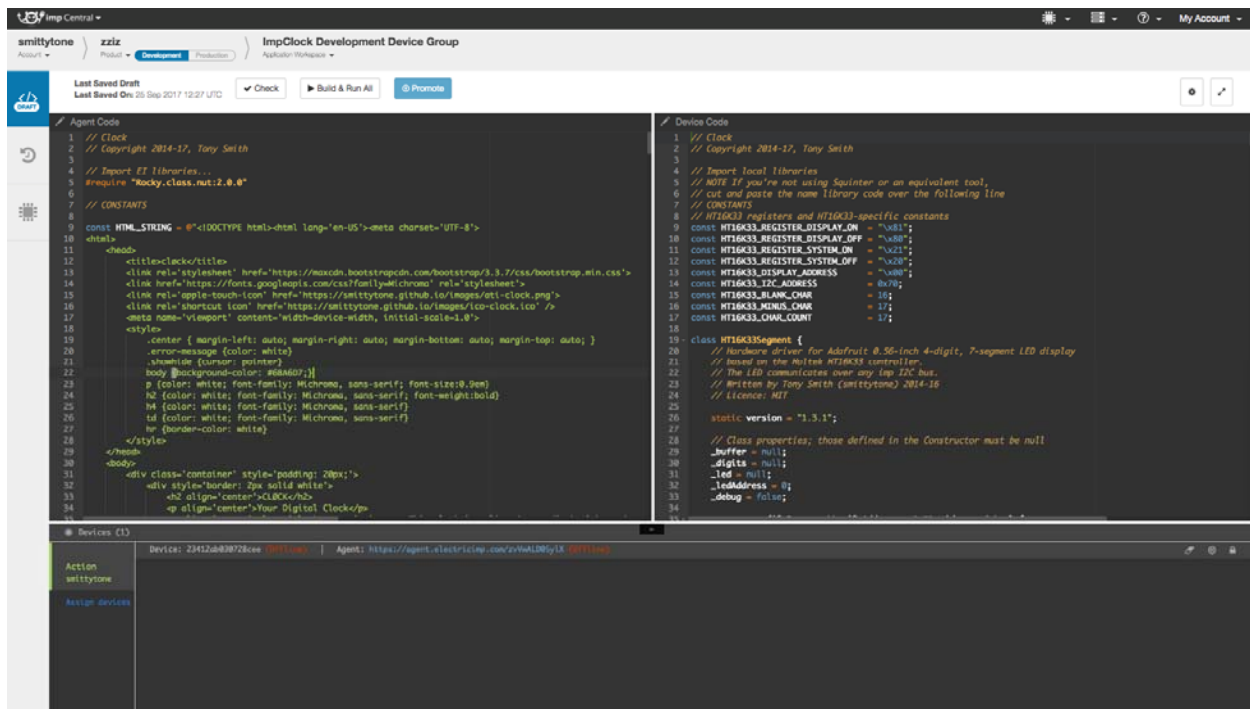
The agent keeps the device safe: all communications between the device and external services are mediated by the device's agent. To maintain security, there is no direct access to devices. This is the case even when the external service is on the same physical network as the device. The impCloud is built for performance, so any latency this approach introduces is negligible.

You might use an agent to receive commands from a mobile app and relay them to the device. The device might ask the agent to fetch data, such as a weather forecast, from an online service. The agent makes the outgoing request, handles the returned data and sends the relevant part back to the device.

Agents run continually, allowing devices to be accessible even if they are sleeping to conserve power. The agent can check the device's state and hold any received data until the device is awake again.

## First Code

The code editor presents areas for the firmware's two parts, a list of assigned devices (just your imp kit for now) and a logging area, which is populated when you select a device:



Click on the **device code** pane and type in the following:

```
server.log("Hello from your imp kit");
```

If your imp kit is powered up and connected to the Internet, select it from list then click **Build and Force Restart** (above the **agent code** pane). The imp kit will new restart, and the code editor log should show:

```
[Device] Hello from your imp kit
```

## Squirrel

You write your agent and device code in Squirrel, a modern, object-oriented language similar to JavaScript, Swift and C++. Squirrel is not difficult to learn, and we have a full guide to the language [here](#) plus a [cheat sheet](#).

## The imp API

The imp API provides a series of functions (like `server.log()`, used above) which can be included in your code to interact with peripherals connected to the imp, either within the imp kit or externally; to apply device settings; to control communication between the agent and its device; to control communication between the agent and external Internet resources; and much more besides. The functions made available by the imp API are listed and documented [here](#).



## Electric Imp Libraries

Building on the imp API, your code can include libraries provided by Electric Imp to simplify communication between an agent and popular web services, and between the device's imp and connected hardware. You can find a full list of available libraries [here](#).

## 4. Next Steps

The Electric Imp Dev Center offers [sample code](#) to help you learn Squirrel and understand how to write application firmware. This sample code includes the imp API cookbook: a series of recipes for common imp application tasks.

The examples may not have been written for the imp in your imp kit, but imps are software compatible. You may need to make small changes to the code if it makes use of the imp hardware features, but this is straightforward. Use the [imp pin mux](#) pages to help you &mdash; it lists all of the available GPIO pins and buses available to each type of imp.

You should also take a look at the following pages to help you fully understand the Electric Imp application and platform architecture:

- [Network State Diagram](#)
- [Event-driven Programming](#)
- [Effective Internet-agent-device Communications](#)
- [How to Run an imp-enabled Device Offline](#)
- [How to Write Loop Structures](#)
- [Data Persistence in Electric Imp Applications](#)