

Design Azure AI services using Azure Functions

6 minutes

In this module, you'll build a translator application with an IoT device by deploying Azure Function in the cloud. You'll use Visual Studio Code to create and deploy a function. The function will receive an HTTP request. This request will trigger an Azure function and initiate the code execution. The code will use Speech APIs to translate the voice into the target language. The function receives voice data from the IoT device and interactively translates speech from a foreign language into the target language. You'll specify the target language in the module and display the results on the screen of the IoT device.

You'll use the following components in the design of the solution:

- **Azure Functions and serverless:** Serverless computing enables developers to build applications without worrying about application infrastructure. Serverless computing has three parts:
 - Serverless Kubernetes
 - Serverless Functions
 - Serverless Application Environments

In this module, we use Azure functions only. Azure Functions enable you to develop serverless applications. You run small pieces of code (called "functions"). Specific events trigger the function to run the code.

- **Cognitive Speech Service:** The Speech service is an API that provides you speech to text, text to speech, and speech translation functionality into a single Azure subscription. In this module, you'll use the Cognitive Speech service to integrate real-time speech translation.
- **Visual Studio Code:** is a source-code editor. You use Visual Studio Code as a development tool for the IoT device.
- **Azure IoT Device Workbench:** is a Visual Studio Code extension. You use the Azure IoT Device Workbench to open a project, create and deploy Azure functions, and debug IoT Devkit.

- **C#:** You'll develop Azure Function by using C# programming language in .NET class libraries.
- **Function app:** is the container that hosts the execution of individual functions.
- **Devkit Translator sample:** is a project sample using Azure Functions and Azure AI services that are available on [GitHub](#). You'll use this code as a basis and modify it for the scenario outlined above
- **Device code:** is Arduino Sketch File (.ino file extension) that you can use for controlling an Arduino circuit board.

The overall module steps are:

1. Configure IoT Devkit to the cloud
2. Create a Cognitive Speech Service in Azure
3. Build a translator sample using Visual Studio Code
 - a. Open "Devkit Translator" sample
 - b. Provision Azure IoT Hub and IoT device and Azure Functions
 - c. Configure Speech API key in the Azure function
 - d. Deploy Azure function
 - e. Configure and compile the code

After completing the module, your device will be configured to the IoT Hub. You'll have the Azure function app deployed in the cloud. You'll also implement the device code in the IoT device. The device code will use the Azure function and will execute the code in a serverless environment. The resulting solution will perform speech translation using Azure AI services for the Digital signage scenario.

Next unit: How to configure IoT device to the IoT Hub

[Continue >](#)