

# // Build/2015 – AllJoyn Z-Wave DSB Sample

The objective of the Z-Wave AllJoyn Device System Bridge (DSB) is to bridge AllJoyn to Z-Wave and to allow controlling Z-Wave devices via AllJoyn.

This document provides an overview of the Z-Wave DSB sample.

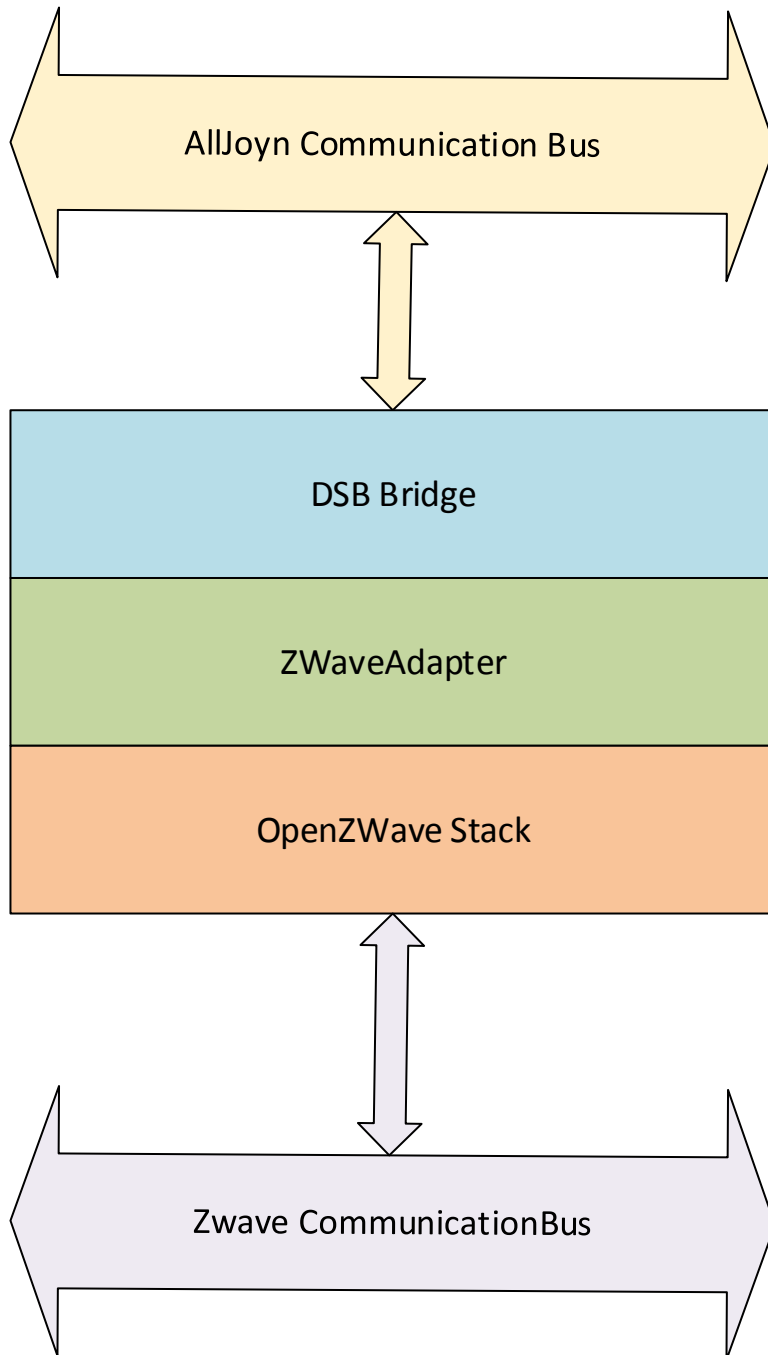
## Description:

The ZWave DSB sample uses DSB Platform components (aka Bridge). The sample shows how to implement a DSB Adapter that is the glue code between AllJoyn and some other network. In this sample, an open source implementation of the Z-Wave stack called 'OpenZWave' will be used to talk to the underline ZWave devices. The stack is available at <http://www.openzwave.com>.

Adapter allows to enumerate Z-Wave devices and support properties, method and signal calls from AllJoyn by implementing the interfaces defined by the Bridge. Interfaces are exposed from the Bridge via .winmd, which makes it possible to implement Adapter as any Windows Runtime supported project type.

Bridge and Adapter need to be hosted in UWP. For Desktop, AdapterHostApp UWP is included in the sample. This is a simple UWP with minimal UI to show how Bridge and adapter should be instantiated. For Windows 10 IoT Core, a Background Service type of UWP is included in the sample. This shows how a DSP should be instantiated when run as a background service UWP with no UI.

## Architecture Overview



## Projects:

### **Common\**

Common.lib : Static library that contains the common utility classes and functions that can be shared across modules.

### **BridgeRT\**

BridgeRT.dll : Dynamic library that hosts the ASB Bridge WinRT component.

#### **Openzwave-1.2.919\**

An open source implementation of the Z-Wave stack called 'OpenZWave' will be used to talk to the underline ZWave devices. The stack is available at <http://www.openzwave.com>.

#### **ZWaveStackRuntime\**

OpenZWaveStackLibrary.lib: open source OpenZWave stack library. The static library would be updated to support WinRT APIs.

#### **AdapterLib\**

AdapterLib.dll: library that contains the adapter core logic. It uses the openZWave stack library to discover the Z-Wave devices and exposes them onto the AllJoyn bus using the BridgeRT component.

#### **AdapterHostApp\**

AdapterHostApp.exe : A UWP app to host the Adapter and Bridge for Desktop Windows 10.

#### **ZWaveBackgroundService\**

ZWaveBackgroundService.dll: A continuous background task, also known as a startup task on Windows 10 IoT Core device.

## Building

To build using Visual Studio 2015, open the supplied **ZWaveAdapterUAP.sln** and Build Solution for **x86, AMD64 or ARM**. Note that warning LNK4264 is expected.

## Setup and Deployment

To get the sample up and running with RPi2 or MBM Athens boards, see //Build/2015 AllJoyn ZWave Setup Guide. You can also deploy the sample on Windows 10 desktop from Visual Studio by following similar steps.