# INTRODUCTION

# SOFTWARE DEVELOPMENT KITS

2.1 Azure Kinect Sensor SDK

2.2 Body Tracking SDK

2.3 Azure Cognitive Services

# SYSTEM REQUIREMENTS

3.1 Supported operating systems and architectures

3.2 Host PC hardware requirements

3.3 Known compatibility issues

# HARDWARE SPECIFICATION

4.1 Product dimensions and weight

4.2 Operating environment

4.3 Depth camera supported operating modes

4.4 Color camera supported operating modes

4.5 Motion sensor

4.6 Microphone array

4.7 Indicators

4.8 Powering Device

4.9 Power consumption

4.10 Calibration

4.11 External synchronization

# AZURE KINECT VS. KINECT FOR WINDOWS

5.1 Hardware

5.2 Sensor access

5.3 Feature mapping
1 Introduction

This document provides an overview of the different software development kits (SDKs) as well as hardware specifications for Azure Kinect DK. The full SDK documentation will be available soon.

2 Software Development Kits

2.1 Azure Kinect Sensor SDK

The Azure Kinect Sensor SDK provides cross-platform low level sensor access for Azure Kinect DK hardware sensors and device configuration, including:

- Depth camera access and mode control (a passive IR mode, plus wide and narrow field-of-view depth modes)
- RGB camera access and control (e.g. exposure and white balance)
- Motion sensor (gyroscope and accelerometer) access
- Synchronized Depth-RGB camera streaming with configurable delay between cameras
- External device synchronization control with configurable delay offset between devices
- Camera frame meta-data access for image resolution, timestamp and temperature
- Device calibration data access

Tools provided as sample code include:

- Viewer tool for checking device data streams and configuring different modes
- Sensor recording tool and playback reader API using Matroska container format
- Firmware update tool

The Sensor SDK has a Win32 C-API with preview support for Linux. The SDK will be open sourced and made available in GitHub.

Microphone specific functionality can be accessed using Speech SDK or directly through Windows Media Foundation.

2.2 Body Tracking SDK

The Body Tracking SDK includes a Windows library and runtime to track bodies in 3D using Azure Kinect DK hardware, providing:

- Body segmentation
- An anatomically correct skeleton for each partial or full body in FOV
- A unique identity for each body
- The ability to track bodies over time

Tools provided as sample code include,

- A viewer tool to track bodies in 3D

The Body Tracking SDK with Win32 C-API will be in open preview at Azure Kinect DK availability and in final release in the second half of 2019. Linux support will arrive after the Windows release.
2.3 Azure Cognitive Services
With Azure Kinect DK and Azure AI services you have all the power to build complete solutions. For example, you can use voice controls to complete dimensional analysis of an object while also identifying key labels.

For more information about Azure Cognitive Services:

- **Speech Services** enabled Azure connected speech services (e.g. speech-to-text and translation)
- **Azure Vision Services** provide several useful services to enhance your application capabilities (e.g. face detection and optical character recognition)

3 System Requirements

3.1 Supported operating systems and architectures

- Windows 10 April 2018 release (x64) or later
- Linux Ubuntu 18.04 (x64) with OpenGLv4.4 or later GPU driver

Sensor SDK is available for the Windows API (Win32) for native C/C++ Windows applications and is not currently available to UWP applications. Azure Kinect DK is not supported for Windows 10 in S mode.

3.2 Host PC hardware requirements

The PC host hardware requirement is very dependent on the application/algorithm/sensor frame rate/resolution processed on host PC.

The recommended minimum Sensor SDK configuration for Windows is:

- 7th Gen Intel® Core™ i3 Processor (Dual Core 2.4 GHz with HD620 GPU or faster)
- 4GB Memory
- Dedicated USB3 port

Lower end and older CPUs may also work depending on your use case/scenario. The Body Tracking minimum hardware requirement will be higher and published later.

3.3 Known compatibility issues

There are known compatibility issues with following USB Host controllers:

- ASMedia USB 3.1 eXtensible Host Controller (e.g. ASM1142)
- Texas Instruments USB 3.0 xHCI Host Controller
4 Hardware Specification

Azure Kinect DK integrates Microsoft latest sensor technology into single USB connected accessory.

![Azure Kinect DK](image)

*Figure 1. Azure Kinect DK*

4.1 Product dimensions and weight

Dimensions: 103 x 39 x 126 mm

Weight: 440 g
4.2 Operating environment

Azure Kinect DK is intended for developers and commercial businesses operating under the following ambient conditions:

- Temperature: 10-25°C,
- Humidity: 8-90% (non-condensing) RH,
- **Light Levels: 10-2500 lux**

NOTE: Use outside of the ambient conditions could cause the device to fail and/or function incorrectly.

These ambient conditions are applicable for the environment immediately around the device under all operational conditions. When used with an external enclosure, active temperature control and/or other cooling solutions are recommended to ensure the device is maintained within these ranges. The device design features a cooling channel in between the front section and rear sleeve. In any implementation of the device, this cooling channel should not be obstructed.
4.3 Depth camera supported operating modes

Azure Kinect DK integrates a Microsoft designed 1 Megapixel Time-of-Flight depth camera using the Yeats image sensor presented at ISSCC 2018. The depth camera supports the modes indicated below.

Table 1. Depth camera supported operating modes

<table>
<thead>
<tr>
<th>MODE</th>
<th>RESOLUTION</th>
<th>FOI</th>
<th>FPS</th>
<th>OPERATING RANGE*</th>
<th>EXPOSURE TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFOV unbinned</td>
<td>640x576</td>
<td>75°x65°</td>
<td>0, 5, 15, 30</td>
<td>0.5 - 3.86 m</td>
<td>12.8 ms</td>
</tr>
<tr>
<td>NFOV 2x2 binned (SW)</td>
<td>320x288</td>
<td>75°x65°</td>
<td>0, 5, 15, 30</td>
<td>0.5 - 5.46 m</td>
<td>12.8 ms</td>
</tr>
<tr>
<td>WFOV 2x2 binned</td>
<td>512x512</td>
<td>120°x120°</td>
<td>0, 5, 15, 30</td>
<td>0.25 - 2.88 m</td>
<td>12.8 ms</td>
</tr>
<tr>
<td>WFOV unbinned</td>
<td>1024x1024</td>
<td>120°x120°</td>
<td>0, 5, 15</td>
<td>0.25 - 2.21 m</td>
<td>20.3 ms</td>
</tr>
<tr>
<td>Passive IR</td>
<td>1024x1024</td>
<td>N/A</td>
<td>0, 5, 15, 30</td>
<td>N/A</td>
<td>1.6 ms</td>
</tr>
</tbody>
</table>

*15% to 95% reflectivity, 2.2 uW/cm^2/nm, σ <= 17 mm. Depth provided outside of indicated range depending on object reflectivity

4.4 Color camera supported operating modes

Azure Kinect DK includes an OV12A10 12MP CMOS sensor rolling shutter sensor. The native operating modes are listed below.

Table 2. Color camera supported operating modes

<table>
<thead>
<tr>
<th>RGB CAMERA RESOLUTION (HXV)</th>
<th>ASPECT RATIO</th>
<th>FORMAT OPTIONS</th>
<th>FRAME RATES (FPS)</th>
<th>NOMINAL FOV (HXV)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3840x2160</td>
<td>16:9</td>
<td>MJPEG</td>
<td>0, 5, 15, 30</td>
<td>90x59</td>
</tr>
<tr>
<td>2560x1440</td>
<td>16:9</td>
<td>MJPEG</td>
<td>0, 5, 15, 30</td>
<td>90x59</td>
</tr>
<tr>
<td>1920x1080</td>
<td>16:9</td>
<td>MJPEG</td>
<td>0, 5, 15, 30</td>
<td>90x59</td>
</tr>
<tr>
<td>1280x720</td>
<td>16:9</td>
<td>MJPEG/YUY2/NV12</td>
<td>0, 5, 15, 30</td>
<td>90x59</td>
</tr>
<tr>
<td>4096x3072</td>
<td>4:3</td>
<td>MJPEG</td>
<td>0, 5, 15</td>
<td>90x74.3</td>
</tr>
<tr>
<td>2048x1536</td>
<td>4:3</td>
<td>MJPEG</td>
<td>0, 5, 15, 30</td>
<td>90x74.3</td>
</tr>
</tbody>
</table>

The RGB camera is USB Video class compatible and can be used without the Sensor SDK.

4.5 Motion sensor

The embedded Inertial Measurement Unit (IMU) is an LSM6DSMUS and includes both an accelerometer and a gyroscope, simultaneously sampled at 1.6 kHz and reported to the host at a 208 Hz rate.

4.6 Microphone array

Azure Kinect DK embeds a high quality 7-Microphone circular array that identifies as a standard USB audio class 2.0 device. The performance specification is

- Sensitivity: -22 dBFS (94 dB SPL, 1 kHz)
- Signal to noise ratio > 65 dB

4.7 Indicators

The device has a camera streaming indicator on the front of the device that can be disabled programmatically using the Sensor SDK.
The status LED behind the device indicates device state:

- Solid white – the device is powered and operating normally
- Blinking amber – the device does not have enough power to operate
- Blinking white – the device is powered but does not have a USB 3.0 data connection

4.8 Powering Device
The device can be powered using the in-box power supply and USB Type-C to A cable, or by using a Type-C to C cable for both power and data. A type-C to C cable is not included.

4.9 Power consumption
Azure Kinect DK consumes up to 5.9W; specific power consumption is use-case dependent.

4.10 Calibration
Azure Kinect DK is calibrated at the factory. The calibration parameters for visual and inertial sensors may be queried programmatically through the Sensor SDK.

4.11 External synchronization
The device includes 3.5mm synchronization jacks that can be used to link multiple units together to achieve coordinated Depth/RGB camera triggering. There are specific sync-in and sync-out jacks on the device, enabling easy daisy chaining. Compatible cable is not included in box and must be purchased separately, cable requirements:

- 3.5mm tip male-to-male cable ("3.5mm audio cable")
- Maximum cable length < 10m
- Both stereo and mono cable are supported
5 Azure Kinect vs. Kinect for Windows

The Azure Kinect DK hardware and Software Development Kits have differences from Kinect for Windows v2. Any existing Kinect for Windows v2 applications will not work directly with Azure Kinect DK and require porting.

5.1 Hardware

A high-level summary of the differences between Azure Kinect DK and Kinect for Windows v2 is given in the table below.

*Table 3. Azure Kinect DK vs. Kinect for Windows V2 hardware*

<table>
<thead>
<tr>
<th>FEATURE</th>
<th>AZURE KINECT DK</th>
<th>KINECT FOR WINDOWS V2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audio Details</td>
<td>7-mic circular array</td>
<td>4-mic linear phased array</td>
</tr>
<tr>
<td>Motion sensor Details</td>
<td>3-axis accelerometer + 3-axis gyro</td>
<td>3-axis accelerometer</td>
</tr>
<tr>
<td>RGB Camera Details</td>
<td>3840 x 2160 px @30 fps</td>
<td>1920 x 1080 px @30 fps</td>
</tr>
<tr>
<td>Depth Camera Method</td>
<td>Time-of-Flight</td>
<td>Time-of-Flight</td>
</tr>
<tr>
<td>Resolution/FOV</td>
<td>640 x 576 px @30 fps</td>
<td>512 x 424 px @ 30 fps</td>
</tr>
<tr>
<td>Connectivity Data</td>
<td>USB3.1 gen 1 with Type-C connector</td>
<td>USB 3.1 gen 1</td>
</tr>
<tr>
<td>Power</td>
<td>External PSU or USB-C</td>
<td>External PSU</td>
</tr>
<tr>
<td>Synchronization</td>
<td>RGB &amp; Depth and IMU internal, external device-to-device</td>
<td>RGB &amp; Depth internal only</td>
</tr>
<tr>
<td>Mechanical Dimensions</td>
<td>103 x 39 x 126 mm</td>
<td>249 x 66 x 67 mm</td>
</tr>
<tr>
<td>Mass</td>
<td>440 g</td>
<td>970 g</td>
</tr>
<tr>
<td>Mounting</td>
<td>One ¼-20 UNC Four internal screw points</td>
<td>One ¼-20 UNC</td>
</tr>
</tbody>
</table>

5.2 Sensor access

A low-level device sensor access capability comparison is given in the table below.

*Table 4. Sensor access comparison*

<table>
<thead>
<tr>
<th>FUNCTIONALITY</th>
<th>AZURE KINECT DK</th>
<th>KINECT FOR WINDOWS V2</th>
<th>NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depth</td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>IR</td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Color</td>
<td>✔</td>
<td>✔</td>
<td>Color format support differences, Azure Kinect DK support camera controls like: Exposure, white balance, brightness, contrast, saturation, sharpness and gain control</td>
</tr>
<tr>
<td>Audio</td>
<td>✔</td>
<td>✔</td>
<td>Azure Kinect DK access is through Windows native API</td>
</tr>
<tr>
<td>IMU</td>
<td>✔</td>
<td>Partial (1-axis)</td>
<td></td>
</tr>
</tbody>
</table>
The Azure Kinect sensor SDK relies on WinUSB/libUSB to access device and does not have a service implemented to enable sharing device access with multiple processes.

The Sensor SDK is under development and new features are expected to be made available over time; the SDK will be open sourced to enable contributions from outside Microsoft.

### 5.3 Feature mapping

The Azure Kinect SDK feature set is different from Kinect for Windows v2, as illustrated below

<table>
<thead>
<tr>
<th><strong>KINECT V2 FEATURE</strong></th>
<th><strong>KINECT V2 DATA TYPE</strong></th>
<th><strong>AZURE KINECT SDK/SERVICE</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor Data Access</td>
<td>DepthFrame</td>
<td>Azure Kinect Sensor SDK</td>
</tr>
<tr>
<td></td>
<td>InfraredFrame</td>
<td>Azure Kinect Sensor SDK</td>
</tr>
<tr>
<td></td>
<td>ColorFrame</td>
<td>Azure Kinect Sensor SDK</td>
</tr>
<tr>
<td></td>
<td>AudioBeamFrame</td>
<td>Not currently supported</td>
</tr>
<tr>
<td>Body Tracking</td>
<td>BodyFrame</td>
<td>Azure Kinect Body Tracker</td>
</tr>
<tr>
<td></td>
<td>BodyIndexFrame</td>
<td>Azure Kinect Body Tracker</td>
</tr>
<tr>
<td>Coordinate Mapping</td>
<td>CoordinateMapper</td>
<td>Azure Kinect Sensor SDK</td>
</tr>
<tr>
<td>Face Tracking</td>
<td>FaceFrame</td>
<td>Cognitive Services: Face</td>
</tr>
<tr>
<td>Speech Recognition</td>
<td>N/A</td>
<td>Cognitive Services: Speech</td>
</tr>
</tbody>
</table>