SoftKinetic™
The Interface is You

DepthSense® SDK 1.9.0

Release Notes

Project Name – Reference	DepthSense® SDK 1.9.0 - Release Notes
Package Build Number	5
Document Version	2.0
Last Change – Revision	2015-10-07
Author	SoftKinetic

CONFIDENTIALITY

COPYRIGHT

TABLE OF CONTENTS

1 Introduction

The DepthSense® Software Development Kit provides camera driver components and defines an Application Programming Interface for user applications and middleware to interface with the SoftKinetic DS311, DS32x and DS536 cameras, as well as with the TICDK Camera Development Kit hardware.

This section briefly describes the contents and scope of this document. It also defines the various acronyms and abbreviations used throughout the document. Finally, it provides a quick overview of the software package contents.

1.1 Purpose

The purpose of this document is to provide information about the 1.9.0 Public Release of the DepthSense® Software Development Kit.

1.2 Scope

This document applies to the DepthSense® SDK software package from SoftKinetic.

1.3 SDK version

All information in this document applies to the DepthSense® SDK 1.9.0 Public Release.

1.4 Definitions, acronyms and abbreviations

ABI: Application Binary Interface

API: Application Programming Interface

CDK: Camera Development Kit

COM: Component Object Model

FW: Firmware

GUI: Graphical User Interface

HW: Hardware

IPC: Inter-Process Communication

MW: Middleware

OS: Operating System

SDK: Software Development Kit

SW: Software

ToF: Time of Flight

UAC: USB Audio Class

USB: Universal Serial Bus

UVC: USB Video Class

WMF: Windows Media Foundation

1.5 Overview

The DepthSense® SDK software package provides library components and a programming interface which allows applications or middleware to control one or more DepthSense® ToF cameras and access the data streams produced by the device sensors and microphones.

It also allows the basic operation of the TICDK Camera Development Kit platform (hardware detection and depth data streaming using the existing configurations stored in the device).

The package contains a set of applications which can be used to evaluate a DepthSense® ToF camera on a functional level; these applications consist of a depth/color image viewer GUI with image registration capabilities and of a general-purpose command-line tool.

Finally, a simple console demo application is provided in source form.

2 Release Notes

2.1 Camera compatibility

This release is compatible with the following cameras:

	Depth			**Color**		
Model	**VID**	**PID**	**Revisions**	**VID**	**PID**	**Revisions**
DS311	0x2113	0x0137	07.08 – 07.09	0x2113	0x8000	00.01
DS325	0x2113	0x0145	21.00 – 21.07	(same as depth node)		
VF0780 (Creative)	0x041E	0x4096	21.00 – 21.07	(same as depth node)		
DS325B	0x2113	0x3252	00.03	(same as depth node)		
DS326	0x2113	0x0146	00.03	(same as depth node)		
DS327 (Imprivata)	0x2113	0x0147	21.07	(same as depth node)		
DS536	0x2113	0x5360	00.05	(N/A)		
DS536A	0x2113	0x5361	00.05 – 00.06 10.05 – 10.06	(N/A)		
DS536B	0x2113	0x5362	00.05 – 00.06 10.05 – 10.06	(N/A)		
TICDK	0x0451	0x9100	00.00	(N/A)		

Notes

To determine the hardware IDs of a connected camera, open the Device Manager, right-click on the desired camera device entry under "Imaging Devices" and choose "Properties"; select then the "Details" tab and choose "Hardware Ids" in the "Property" drop-down list. This will show values of the form:

USB\VID_xxxx&PID_xxxx&REV_xxxx&MI_00

The actual USB Vendor ID, Product ID and Revision of the camera can then be extracted from those values.

2.2 Platform compatibility

A host with USB 2.0 or 3.0 is required.

The host CPU requires at least SSE2 support. The minimum requirement CPU is thus Intel Pentium 4 or AMD Opteron/Athlon 64. The installer performs first a sanity check and will refuse to install the software if the target platform does not comply with these requirements.

The following operating systems are supported:

	32-bit	64-bit
Windows 7	✓	✓
Windows 8 / 8.1	✓	✓
Windows 10	✓	✓

Notes

The DepthSense® SDK is also freely available on Ubuntu 14.04 (x86, ARM) and Android 4.4 KitKat; for more information please contact our sales organization at sales@softkinetic.com.

Minimum PC requirements:

- Dual Core (Intel Core 2 Duo / AMD Athlon Duo / or equivalent)
- 2 GHz, 2 GB RAM

2.3 Tested compilers

The Windows versions are built with Visual Studio 2013 Update 5 and are known to be binary compatible with applications built with the following compilers:

- Microsoft Visual Studio 2013 REL
- Microsoft Visual Studio 2013 Update 4
- Microsoft Visual Studio 2013 Update 5

Notes

The Linux x86 packages are built with gcc 4.8.2 on Ubuntu 14.04.

The Linux ARM packages are built with gcc 4.7.3 for Ubuntu 14.04, armv7l hard-float ABI.

At this time no cross-platform validation is done with these packages.

2.4 Who should update?

Updating is recommended for use with the latest revisions of the following camera models:

- DS311 (revisions 07.08 and 07.09)
- DS325 and VF0780 (revisions 21.00 to 21.07)
- DS325B (revision 00.03)
- DS326 (revision 00.03)
- DS327 (revision 21.07)
- DS536A and DS536B (revisions 00.05, 00.06, 10.05 and 10.06)
- TICDK (revision 00.00)

2.5 Documentation

The DepthSense® SDK documentation can be found in the `doc` subfolder of the install path:

- The DepthSense® SDK 1.9.0 API Reference Manual (HTML)
- The Camera Model Documentation (PDF)
- The DepthSense® Viewer User Guide (PDF)
- The SK Filters User Guide (PDF)

2.6 Windows installation

On a 32-bit system, run the `DepthSenseSDK-1.9.0-X-win32-VS2013-installer.exe` executable file to start the installation (where `X` stands for the package build number).

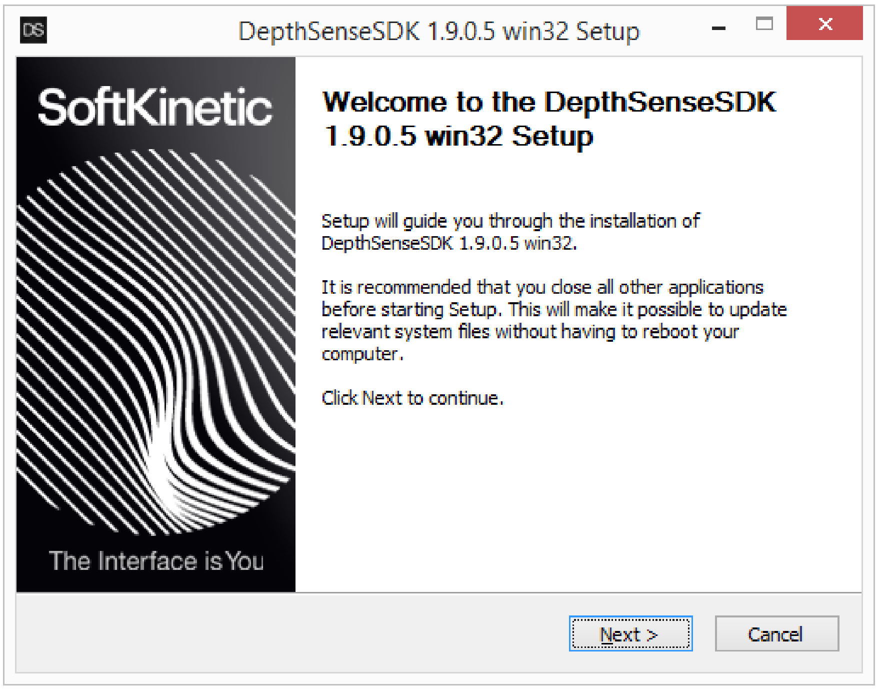
On a 64-bit system, run the `DepthSenseSDK-1.9.0-X-win64-VS2013-installer.exe` executable file.

The installer will propose first to uninstall any version of the same architecture already present on the system. See the "*Uninstallation*" section below for more details.

On a 64-bit Windows operating system (x86_64 processor architecture), both packages can be installed side by side; on such systems with both packages installed, running 32-bit and 64-bit client applications together at the same time is possible.

However, the installer will refuse to run if another DepthSense® SDK flavor of a non-matching build (*i.e.* whose package build number differs from `X`) is already installed on the system. In such a case, the existing packages must be uninstalled manually first. See the next sections for more details.

When running the installer, once all previous versions of the DepthSense® SDK have been removed from your computer, a welcome screen pops up:



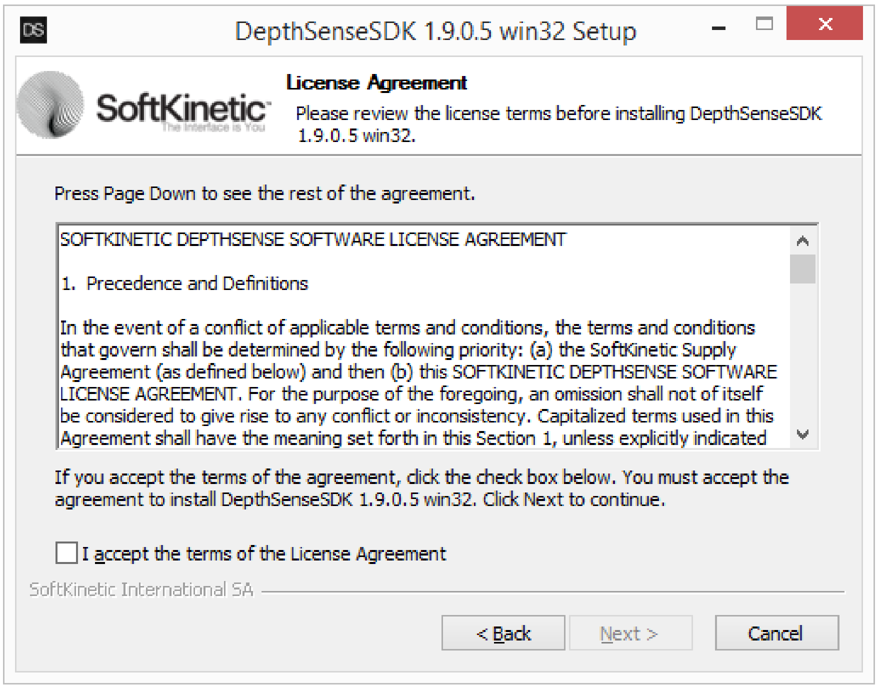
Welcome to the DepthSenseSDK 1.9.0.5 win32 Setup

Setup will guide you through the installation of DepthSenseSDK 1.9.0.5 win32.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

After clicking the "*Next*" button, the installer will display the End-User License Agreement:



License Agreement

Please review the license terms before installing DepthSenseSDK 1.9.0.5 win32.

Press Page Down to see the rest of the agreement.

SOFTKINETIC DEPTHSENSE SOFTWARE LICENSE AGREEMENT

1. Precedence and Definitions

In the event of a conflict of applicable terms and conditions, the terms and conditions that govern shall be determined by the following priority: (a) the SoftKinetic Supply Agreement (as defined below) and then (b) this SOFTKINETIC DEPTHSENSE SOFTWARE LICENSE AGREEMENT. For the purpose of the foregoing, an omission shall not of itself be considered to give rise to any conflict or inconsistency. Capitalized terms used in this Agreement shall have the meaning set forth in this Section 1, unless explicitly indicated

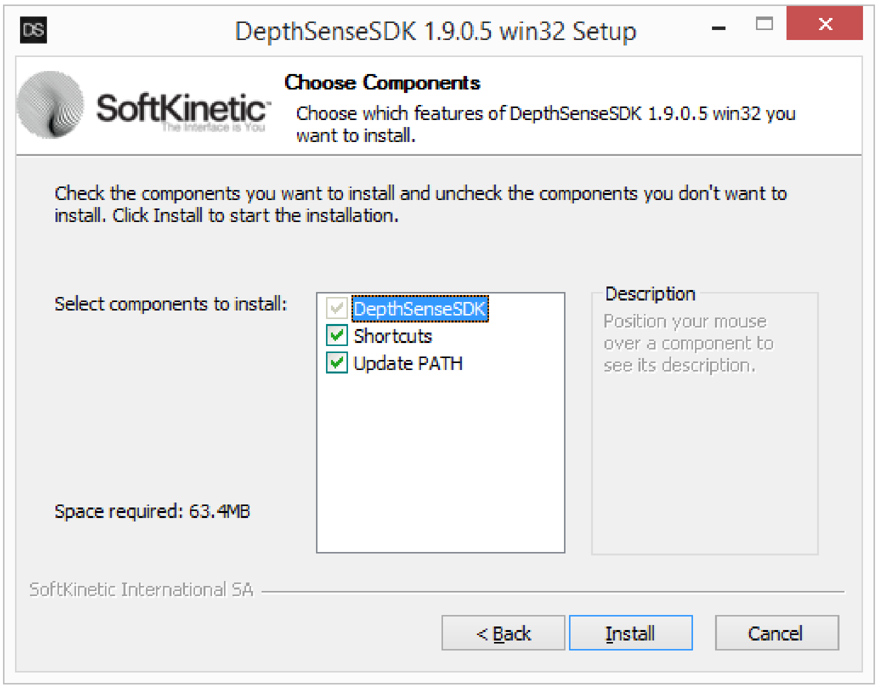
If you accept the terms of the agreement, click the check box below. You must accept the agreement to install DepthSenseSDK 1.9.0.5 win32. Click Next to continue.

□ I accept the terms of the License Agreement

SoftKinetic International SA

Read the License Agreement carefully; to accept it, select the checkbox and then click the "*Next*" button.

The next screen proposes a few install choices, like create menu shortcuts or append the DepthSense® SDK `bin` folder to the `%PATH%` environment variable:



DepthSenseSDK 1.9.0.5 win32 Setup

Choose Components

Choose which features of DepthSenseSDK 1.9.0.5 win32 you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Install to start the installation.

Select components to install:

☑ DepthSenseSDK
☑ Shortcuts
☑ Update PATH

Description

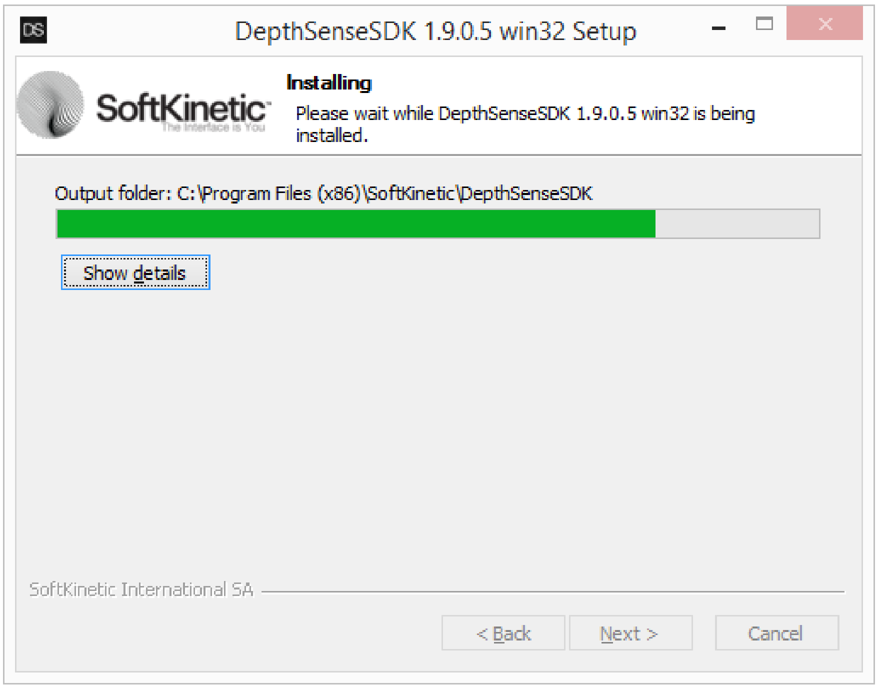
Position your mouse over a component to see its description.

Space required: 63.4MB

SoftKinetic International SA

The latter option, which is selected by default, will allow end-user applications linked to the DepthSense® SDK libraries to run from any location in the file system.

Once the desired options are set, click the "*Install*" button to launch the actual installation, and wait for the files to be copied to the output folder:



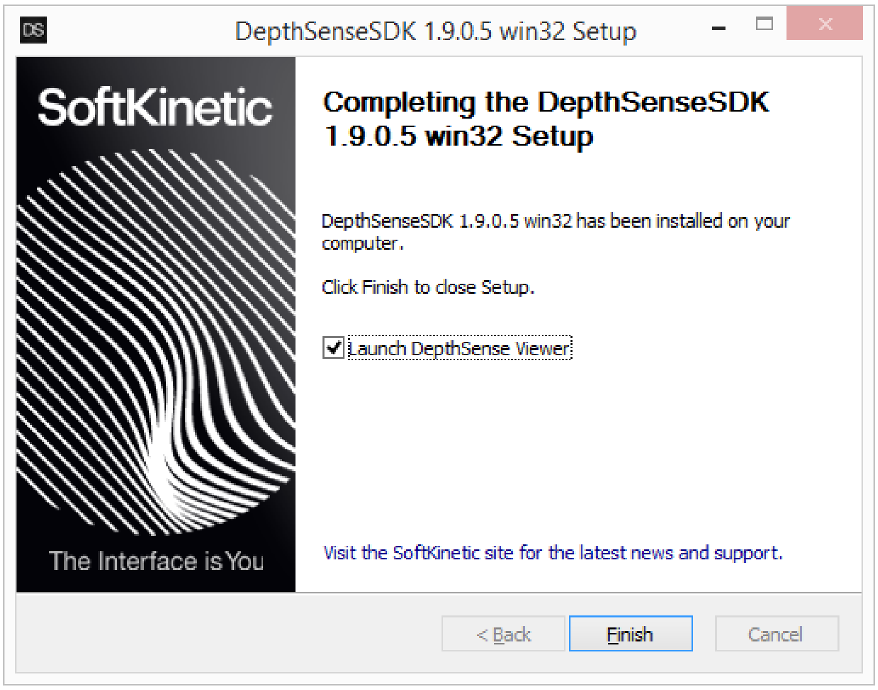
DepthSenseSDK 1.9.0.5 win32 Setup

Installing

Please wait while DepthSenseSDK 1.9.0.5 win32 is being installed.

Output folder: C:\Program Files (x86)\SoftKinetic\DepthSenseSDK

Once the installation is complete, the installer displays a last screen with a checkbox, which if kept selected, will start the DepthSense® Viewer application after clicking the "*Finish*" button:



DepthSenseSDK 1.9.0.5 win32 Setup

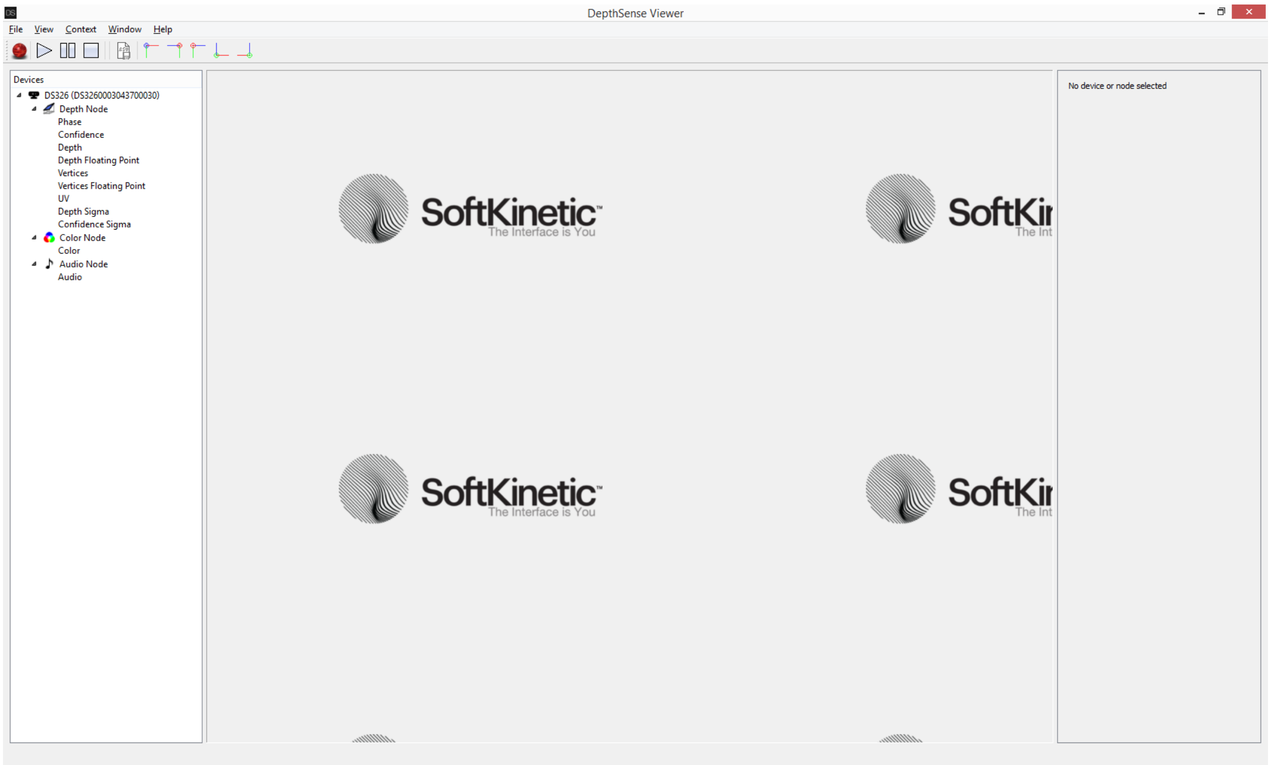
Completing the DepthSenseSDK 1.9.0.5 win32 Setup

DepthSenseSDK 1.9.0.5 win32 has been installed on your computer.

Click Finish to close Setup.

☑ Launch DepthSense Viewer

Visit the SoftKinetic site for the latest news and support.



Refer to the DepthSense® Viewer User Guide in `doc\pdf` folder of the DepthSense® SDK installation for more information about this application.

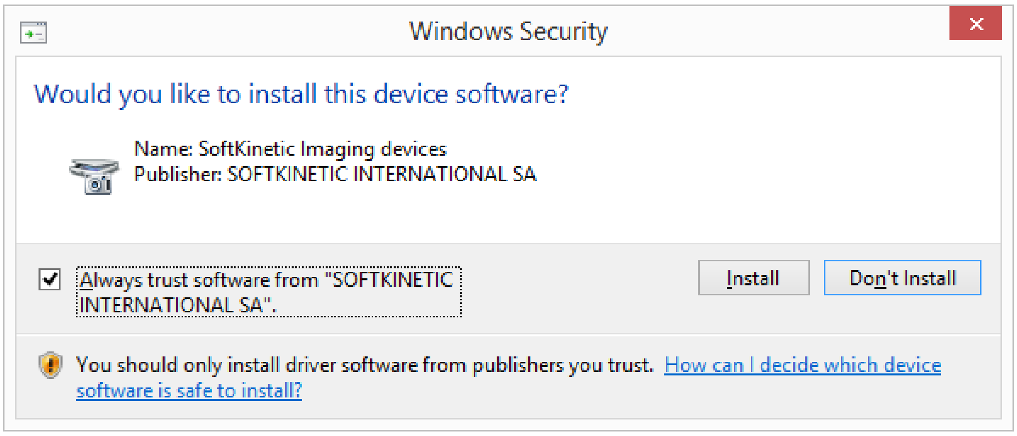
2.7 Cypress FX2 drivers installation for the DS536

The DS536 camera models and the TICDK require an additional custom installation of Cypress FX2 kernel drivers for proper operation. These modified drivers are bundled in the DepthSense® SDK package, and are installed automatically.

In some situations though, it may be necessary to perform a manual installation of the Cypress FX2 drivers after the DepthSense® SDK installer has been run.

To install these drivers manually using an Administrator account, open the `drivers\SoftKinetic_DS536` folder of the DepthSense® SDK installation, open the subfolder corresponding to the target system architecture (either `x86` or `x64`), right-click on the `SoftKinetic_DS536.inf` file and select "*Install*".

This will open a User Account Control prompt asking for confirmation to make changes to the computer. Click "*Yes*"; the following dialog box will then pop up (assuming this is the first time the DS536 drivers are installed on the system):



Windows Security

Would you like to install this device software?

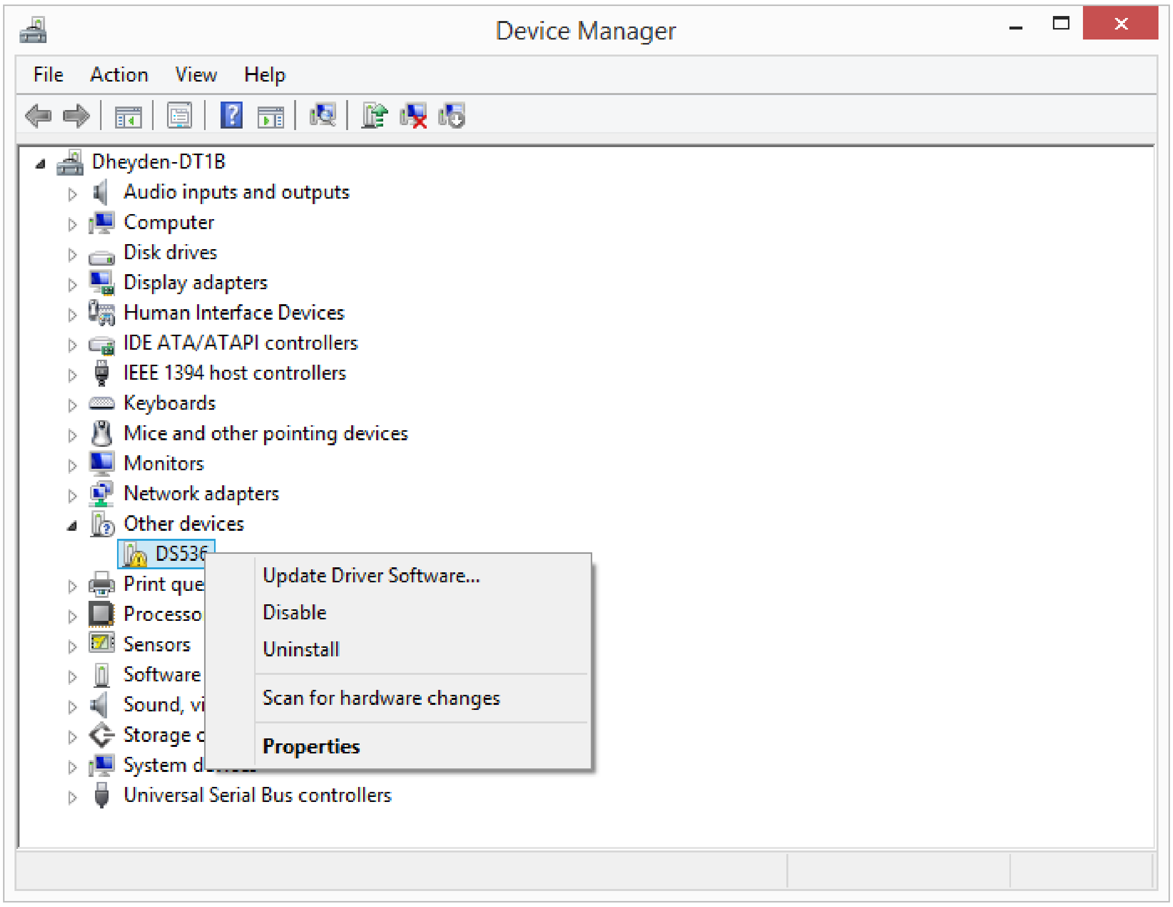
Name: SoftKinetic Imaging devices
Publisher: SOFTKINETIC INTERNATIONAL SA

☑ Always trust software from "SOFTKINETIC INTERNATIONAL SA".

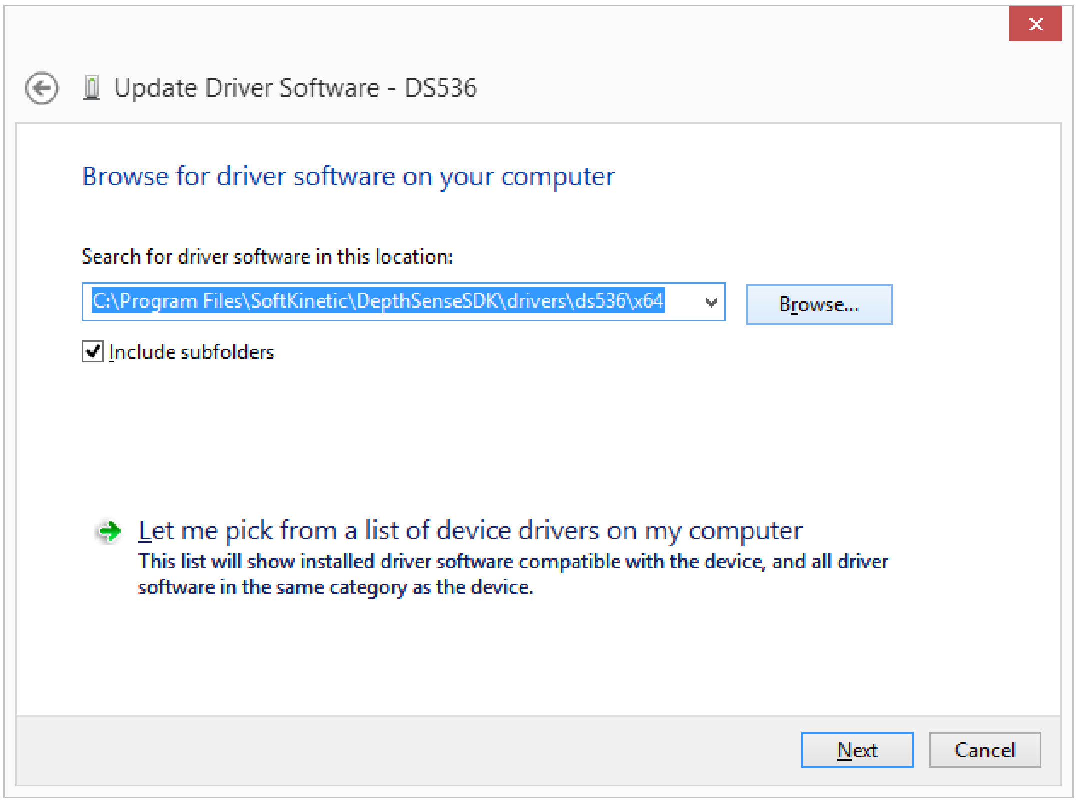
You should only install driver software from publishers you trust. How can I decide which device software is safe to install?

Click the "*Install*" button to start the installation of the drivers. If the "*Always trust software from SOFTKINETIC INTERNATIONAL SA*" box is left selected, you will not be prompted again the next time SoftKinetic drivers are installed on the same system.

Alternatively, this kernel driver installation can be performed by first connecting a DS536 module to the host system, and then by using the Device Manager. Locate the DS536 device in the device list under "*Other devices*", then right-click the device entry and select "*Update Driver Software*" in the context menu:



Choose the "*Browse my computer for driver software*" option in the next dialog, click the "*Browse…*" button and select the folder containing the `INF` file for the target system architecture:



Update Driver Software - DS536

Browse for driver software on your computer

Search for driver software in this location:

C:\Program Files\SoftKinetic\DepthSenseSDK\drivers\ds536\x64

☑ Include subfolders

Let me pick from a list of device drivers on my computer

This list will show installed driver software compatible with the device, and all driver software in the same category as the device.

Click then the "*Next*" button to continue the installation like described above.

<u>*Notes*</u>

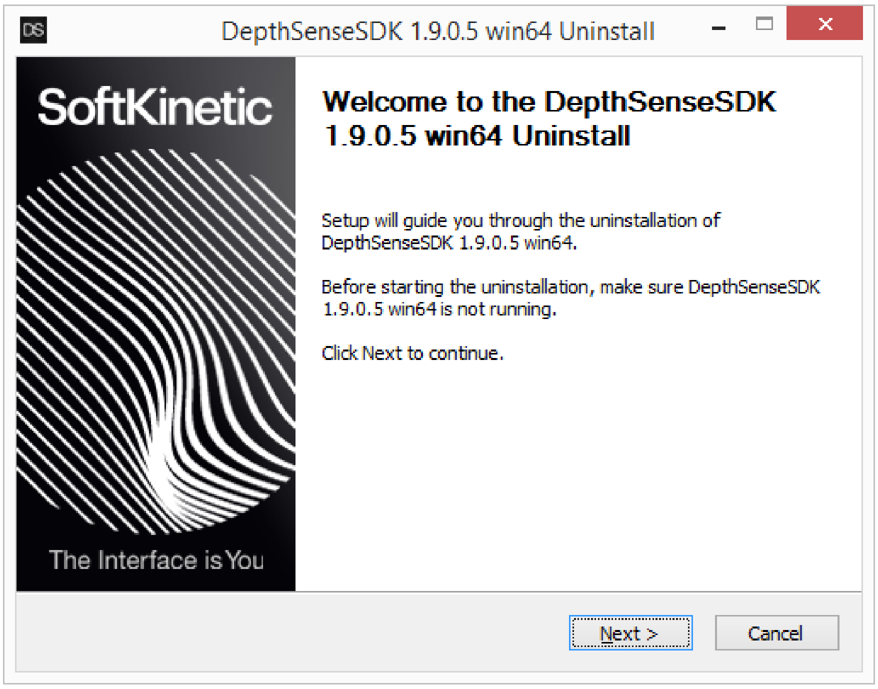
Using the exact same version and build number of the DepthSense® SDK for all co-installed flavors (32 and 64 bits) is highly recommended. The Windows installer may refuse to run on a system where a potentially incompatible package is already installed. In order to avoid issues at install and/or at run time, open the "Programs and Features" section of the Control Panel and if needed, uninstall all other flavors of different versions manually before running this installer.

By default, the PATH environment variable is updated by the DepthSense® SDK installer (unless the installer detects that the PATH variable is too long and would be truncated). This can be disabled at install time by unchecking the "Update PATH" checkbox. For proper operation however, the user must always make sure that the PATH variable never contains any conflicting directory entry pointing to another version of the DepthSense® SDK libraries.

2.8 Uninstallation

To uninstall, navigate to `SoftKinetic > DepthSenseSDK 1.9.0-X winXX > Uninstall DepthSenseSDK 1.9.0-X` in the Windows Start Menu or open the "*Programs and Features*" section of the Control Panel, right click on the "*DepthSenseSDK*" entry in the list of installed programs and select "*Uninstall*".

The following welcome screen pops up:



DepthSenseSDK 1.9.0.5 win64 Uninstall

Welcome to the DepthSenseSDK 1.9.0.5 win64 Uninstall

Setup will guide you through the uninstallation of DepthSenseSDK 1.9.0.5 win64.

Before starting the uninstallation, make sure DepthSenseSDK 1.9.0.5 win64 is not running.

Click Next to continue.

Click the "*Next*" button and merely follow the instructions.

<u>*Notes*</u>

If the uninstall is performed while one of the `DepthSenseSDK` *folders is open in the Windows Explorer, this folder will not be deleted (although all of its contents will). If any of the DepthSense® SDK executable files is still in use at uninstall time (for instance, if the DepthSense® Server and/or the DepthSense® Viewer are running), then the* `DepthSenseSDK` *folder and its contents will be moved to a temporary location, which will be deleted subsequently on the next reboot (which the user is informed of if the uninstall is run interactively).*

2.9 Applications

2.9.1 DEPTHSENSE® SERVER

The DepthSense® Server application is launched automatically when the first client application initializes the DepthSense® library in client/server mode. It serves as a common interface between the camera hardware and multiple clients running simultaneously. Each client can see the stream data from any registered device node, but only one at a time has full control over a given node selected for streaming and thus is allowed to change its specific settings or properties.

The DepthSense® Server application remains active as long as at least one client application is connected. Once the last client exits, the server exits in its turn.

2.9.2 DEPTHSENSE® VIEWER

The DepthSense® Viewer is a graphical application which can display the various image maps and signal data of all available device nodes connected to the host system – depending on the camera model and hardware capabilities, any combination of depth, color, confidence, vertices, UV map and/or audio signals captured by the microphones.

Several instances of the DepthSense® Viewer application can be launched simultaneously, but only one instance at a time can be granted full control over a given node.

The application provides a tree view of all camera devices connected to the host and for each device, shows all of its available nodes. Each node can be selected individually in the list; the application will then display GUI controls for all properties available for the selected node. If the application has been granted full control over the node, all these properties can be modified (including the ones shared between all clients); otherwise only the local, per-client properties will be editable.

Invalid depth node pixels are displayed in specific colors, depending on the map type; for instance in the phase map view, low confidence pixels are in dark gray while saturated pixels are in light gray.

Notes

The DepthSense® Viewer application is currently not available in the ARM and Android packages.

2.9.3 DEPTHSENSE® BENCHMARK

The DepthSense® Benchmark application is a general-purpose command-line tool which can be used to display information about the camera device connected to the host and provide various statistics on the depth, color and audio streaming data.

The tool accepts a number of command-line options which for instance allow the user to select specific camera nodes, choose the nodes configuration, export statistics data to the specified CSV files, record the audio input in a WAV file or display the color map in ASCII form.

<u>Notes</u>

A number of command-line options (such as the `--depth-export` or the `--audio-record` options) can be used to store data in external files. These options take a pathname as argument. The user must have write access to the specified files otherwise the application will exit with a non-zero status.

2.10 Sample code

The DepthSense® SDK package provides a simple console demo application in source form, along with a basic Visual Studio 2008 SP1 solution which allows users to build the sample application directly from source. These files can be found in the `samples\ConsoleDemo` subfolder of the install path.

Building the sample project from its original location, however, will generally require administrator rights. This can be avoided if the `samples` folder is first copied to some other location the user has write access to. In that case though, the Visual Studio project properties will have to be adapted accordingly as the `include` and `library` paths they refer to are relative.

<u>Notes</u>

For seamless interoperability of this sample code with the DepthSense® SDK installation, it is recommended to convert the provided Visual Studio solution to Visual Studio 2013. This can be done easily by opening the solution in Visual Studio 2013 and following the instructions.

2.11 TICDK support

This release supports basic operation of the TICDK Camera Development Kit platform. The primary purpose of the TICDK platform is to provide a reference design for future cameras built around the TI 3D ToF sensor chipset (which relies on the DepthSense® sensor technology).

2.12 ChangeLog

- Version 1.9.0
  - [API] Disabled accelerometer support completely on all DS536 variants as it is not calibrated in current devices
  - [API] Set the range of the `ConfidenceThreshold` depth node property on all DS536 variants to [0 – 32767] instead of [0 – 65535]
  - [API] Fixed filters ranges in the API and improved the related documentation
  - Added support for DS536A and DS536B USB revisions 00.06 and 10.06

  - Experimental support for DS536A USB revision 15.00 with high power mode and no current limiter
  - Further improvements to the DS536 calibration support
  - Support auto-switch between Sensitive and Robust modes on DS536A and DS536B USB revisions 00.06 and 10.06
  - [BUGFIX] Fixed wrong version of libudev dependency expected by ARM Linux packages
  - [BUGFIX] Fixed Context::create() failing on Linux when called repeatedly
  - [BUGFIX] Made the `EnableUvMap` depth node property R/O on the DS536 and removed the corresponding window from the Viewer
  - [BUGFIX] Fixed calibration data not applied correctly on the DS326
  - [BUGFIX] Fixed APIs crashing on ARM when throwing an exception
  - [BUGFIX] Fixed Viewer not running on Windows 32 bits due to missing MSVCP120.dll
  - [BUGFIX] Fixed Export ROM Contents failing on DS536 in some situations
  - [BUGFIX] Fixed loss of depth calibration when changing to/from auto-switch modes on the DS536
  - [BUGFIX] Fixed various minor packaging and installer issues on Windows

- Version 1.8.2
  - [BUGFIX] Fixed wrong digital signature on updated custom DS536 Cypress FX2 drivers on Windows

- Version 1.8.1
  - [ABI] Switched to Visual Studio 2013 on Windows
  - Updated DS536 mode settings

- Version 1.8.0
  - Support all DepthSense® camera models in a single unified DSSDK release
  - Support Export ROM Contents feature in Benchmark and Viewer
  - Preliminary support of USB 3.0 backend
  - [BUGFIX] Fixed crash in Server when setting `HighSensitivityMode` to 2

- Version 1.7.4
  - Added preliminary support for the DS536A and the DS536B
  - Made the stop sequence on all DS536 variants more robust
  - Improved mode and mode calibration support for all DS536 variants
  - Display the camera PRV number in the Viewer
  - Added support for the alternating laser illumination mode on the DS326
  - [BUGFIX] Fixed DS536 firmware files not available on ARM Linux and Android
  - [BUGFIX] Fixed crash in Viewer when enabling or disabling the `DepthMap3Planes` depth node property
  - [BUGFIX] Fixed various issues in the `Context::create()` implementation
  - [BUGFIX] Fixed crash of the Server when camera is disconnected
  - [BUGFIX] Fixed the latest DS536 revision (2113:5360) not recognized on Android
  - [BUGFIX] Fixed DS536 camera initialization failures resulting in Server lockup
  - [BUGFIX] Fixed Server lockup when unplugging a streaming DS536 on Linux
  - [BUGFIX] Fixed streaming startup failure when re-registering the DS536 depth node

- Version 1.7.3
  - Improved ROM usage of the DS536 to allow calibration of additional modes
  - Improved performance and accuracy of depth processing pipeline on all platforms
  - Further improvements to the calibration support on the DS536
  - Made the DS536 firmware upload more robust
  - Bundle digitally signed custom Cypress FX2 drivers for the DS536 in the Windows packages
  - [API] Enabled support of SK Filters and High Sensitivity Mode on the DS326
  - Updated copyright data in distributed files
  - [BUGFIX] Removed invalid calibration data files from package

  - [BUGFIX] Fixed invalid mode settings on the DS536

- Version 1.7.2
  - Support additional high-power modes on the DS536
  - [BUGFIX] Fixed DS536 firmware upload failure on ARM Linux and Android

- Version 1.7.1
  - [API] Defined proper mode names for the DS536
  - Improved performance of ROM data transfers on the DS536
  - Support additional modes and illumination types with the DS536
  - Improved calibration support on the DS536
  - [BUGFIX] Fixed 50 fps DS536 mode streaming at 65 fps

- Version 1.7.0
  - [API] Added preliminary SK Filters support
  - [API] Added High Sensitivity Mode support
  - Added support for the DS536 revision 4 with both short and long-range mode sets
  - Improved support of calibration data storage in the DS536
  - Added support for standard deviation and average on confidence in the Viewer
  - [BUGFIX] Fixed the PATH update taking a very long time during install on Windows
  - [BUGFIX] Fixed Viewer displaying wrong standard deviation values
  - [BUGFIX] Fixed memory leak in Viewer when streaming is active
  - [BUGFIX] Fixed inconsistent behavior of the `HighSensitivityMode` depth node property with the DS536 revision 4
  - [BUGFIX] Fixed long delays when changing DS536 camera mode in the Viewer

- Version 1.5.3
  - [API] Added getter to the API to know which laser was used during frame capture on the DS326

- Version 1.5.2
  - Support MAT20 revision of the DS326
  - Support MAT20 revision of the DS325B

- Version 1.5.1
  - [API] Added API to control the LEDs on the DS326
  - [API] Added support and API for user button events on the DS326
  - Added support for hardware RGB+Z synchronization on the DS326
  - Support storage of enhanced depth calibration data in cameras
  - Support MAT10 revision of the DS326

- Version 1.5.0
  - Added preliminary support for the DS326 camera model
  - Added preliminary support for the DS536 camera model
  - [BUGFIX] Fixed failure to start streaming with older DS325 cameras (USB revisions < 21.00)

2.13 Known bugs

- The DepthSense® Server may crash if a `Context` instance is destroyed (goes out of scope) immediately after creation and a call to the `Context::getDevices()` API.

- If the last context (client connection) is released before the first device has been completely detected, the server may not exit as it normally should. A later connection from another client instance will have the server resume its normal operation, though.

- On Linux, the audio streaming sometimes fails to start with a "*Device busy*" error. In addition, changing the audio node configuration while streaming is active may in some circumstances take a considerable amount of time.

- The reported frame drop count for the color node may be higher than the actual number of lost color images (false positives).

- Minor memory leaks can be observed in the DepthSense® Viewer when disconnecting and reconnecting the camera a high number of times.

- The DepthSense® Server may crash if a client application exits during the processing of a Device Removed or of a Node Removed event.

2.14 Known limitations

- The Windows 7 USB 3.0 core driver imposes stricter bandwidth utilization rules on USB 2.0 devices than the USB 2.0 driver. As a result, dual depth/color streaming on USB 3.0 ports with a DS325 camera fails for all frame rates under the following configurations:
    - For camera revisions prior to 20.28, depth in QVGA resolution and color in WXGA-H (HD) resolution with color compression enabled;
    - Depth in QVGA resolution and color in VGA resolution with color compression disabled.
- Both the Linux USB 2.0 and USB 3.0 drivers impose the same bandwidth usage restrictions as the Windows USB 3.0 driver. So in the Linux case, dual depth/color streaming fails in the same way as described above, regardless of the USB port type.
- Calling the `Context::create()` DepthSense® SDK API from the constructor of a global object is not supported and will most likely cause the application to crash.
- Client applications built with Visual Studio 2010 are not fully binary compatible with the VS2013 DepthSense® libraries, which may result in unpredictable behavior in some situations (specifically when calling `DepthSense::Exception::getMessage()`).
- When streaming both depth and color to multiple clients, frame drops can be observed on the depth node if one or more client has not enabled the color stream.
- The audio streaming may occasionally fail to restart on audio configuration change.
- On some systems running the FastAccess Facial Recognition software, it is not possible to share the video streams between the application and DepthSense® SDK clients.
- Despite the fact that from the 21.00 USB revisions of the DS325 camera onwards, the hardware RGBZ synchronization is now automatically turned on for any color/depth configuration combination which supports it, proper streams matching is not available on Windows due to a lack of support of UVC timestamps by the operating system.
- The serial number of a newly connected DS32x camera can only be known once the depth node is fully detected and initialized.
- Objects close to the camera may appear in double in the UV map due to a parallax effect.
- If the confidence threshold is set to the minimum value, and under certain lighting conditions, moiré artifacts can be seen in the vertices images displayed in the DepthSense® Viewer.

- After installation, application icons or start menu shortcuts may not be set up properly until the next reboot.