



Quick Start Guide

Document Number

AS7352 SDK

Quick Start Guide

Source Development Kit

v1-00 • 2022-May-24

Content Guide

1	General	3	6	Building the Sample Code	10
2	System Requirement.....	4	7	Customizing OSAL and Example Code.....	12
2.1	AS7343 or AS7352 Evaluation Kit.....	4			
2.2	SDK Software and Tools	5	7.1	Including the Chip Library in a Custom Application	12
3	Operating System Abstraction Layer - OSAL.....	6			
4	Sample Codes.....	7			
5	Executing the Sample Code in Windows.....	8			

1 General

The AS7352 Software Development Kit (SDK) is a collection of software components written in C programming language and [Python](#), which allow you to quickly prototype solutions using a spectral sensor mounted on the Evaluation Kits (EVK).

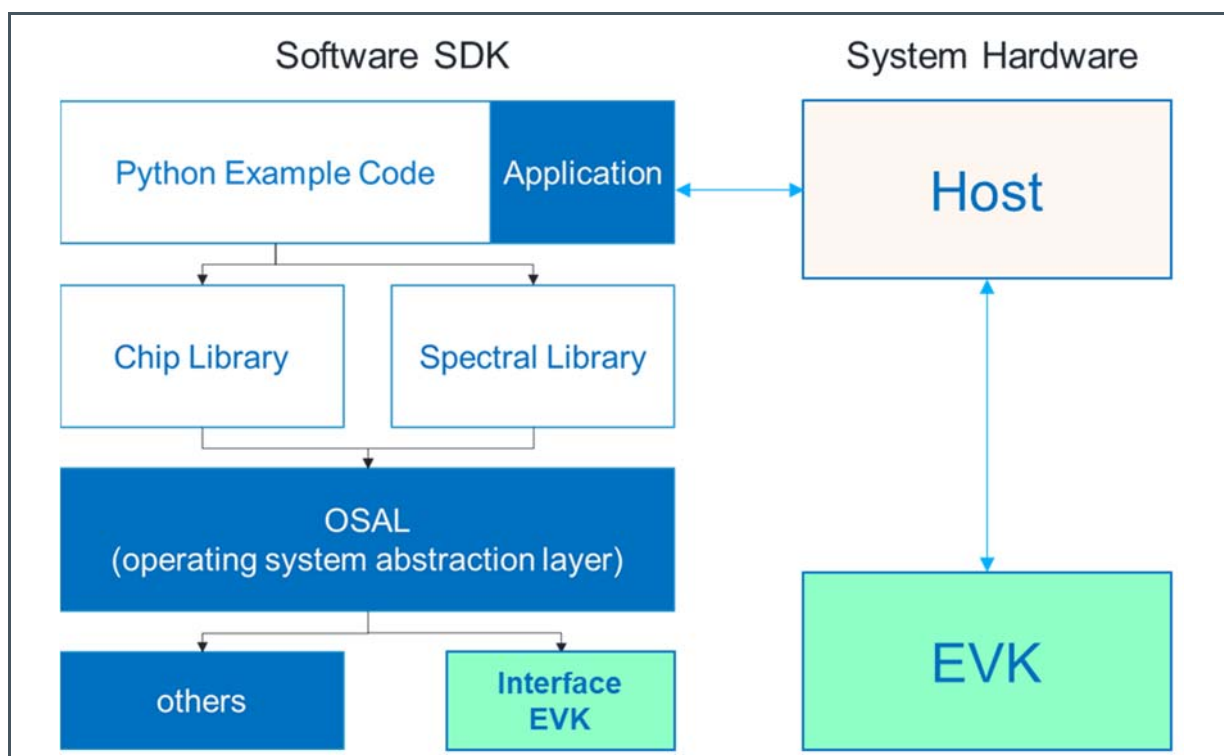
This document provides an overview of the software components and explains their usage.



Information

The AS7343 and AS7352 are compatible sensor chips in pins, functions, and parameters. If either of them is mentioned in files or descriptions, it also applies to the other.

Figure 1:
Structure of the SDK Source Development Kit



The SDK is an example of programming and controlling the sensor via ChipLib. It can be extended by a customer and application-specific software and can be based on application-specific software.

A hardware extension of the SDK would be also possible by adapting the operating system abstraction layer (OSAL). The OSAL and SDK are prepared to work as a plug-and-play system.

2 System Requirement

A compatible AS7343 or AS7352 EVK Board is required to use the SDK. Running the provided sample code requires a 64-bit Windows computer, which is connected to the board using a USB cable - depending on the used sensor board interface.



Information

Two interfaces (FTDI and ams Osram Unicom Interface Board) are prepared for connecting to the specific sensor board. The SDK supports both interfaces.

If an FTDI-based EVK board is used, the installation of separate drivers and libraries is required before the sample code can be run. Download and run the FTDI D2XX Direct Drivers installer to install the dependencies. Other system components are possible but require adaptations that are described later in this document.

Two different EVKs (ALS and Reflection Mode) exist - which are based on an identical sensor board but equipped with different adapters. Both versions of sensor boards can be used as hardware components in common with the SDK.

With the libraries and SDK, a concept is implemented that allows the sensor to be used with the libraries in any hardware. The user only needs to adapt the OSAL and the source code to their target system and program their application. The libraries can be used everywhere identically and already contain all functions of the sensor for control and many functions for correction and application.

2.1 AS7343 or AS7352 Evaluation Kit

The Evaluation Kits for the AS7343 or AS7352 are described in their specified user manual. All the technical details for the sensor chips can be found in the datasheet. It is recommended to study both documents before using the SDK because in this description only specific details of the SDK are described. The SDK includes hardware and software parts. The hardware is identical to the used components in the EVKs and consists of a sensor board and I²C interface connected via USB to the PC.

Figure 2:
Hardware solution of the SDK with a Unicom interface to the PC



The SDK software tools are source codes and libraries in Python and C-Code that a customer can modify. The sensor connection, ADC setup, and sensor readings are included in a code example as a plug-and-play solution and a first test. In addition to the codes and libraries, the SDK also contains the instructions, tools, and environment for compiling the scripts with the sensor libraries.

2.2 SDK Software and Tools

The SDK includes the following components in the single subfolders:

- **bin:** This directory contains a collection of tools that are necessary to convert and link the codes and libraries together into an executable file (see chapter 4).
- **build:** This represents the `build` directory for the project results after compiling the source code together with the libraries (see chapter 6).
- **chiplib:** The `Chip Library` implements a driver for the sensor devices. It handles communication with the device and is used to configure the device and perform measurements. This software component is used in general as a software library and a black box. The functional and interface description is part of the documents in specific directories. The source code can be provided.
- **docu:** This directory contains this manual and other documents.
- **sample_code:** The current directory contains sample codes using the software components provided in the SDK. All the examples use the sensor-specific `Chip Library` to acquire raw measurement data from the sensor. The example named `"as7352_sample_code_chiplib.c"` prints the raw data directly. The sample code programs are intended to be executed directly on a Windows computer. All the examples are provided as source codes and executables (see the directory: `"...\\sample_code\\bin\\x86-64\\windows\\"`). When executing the sample code, all the software components run on the computer. The microcontrollers on the interface boards are only used as bridges for I²C communication and to signal interrupts from the sensor.
- **utilities:** This directory contains a helper code that is used by the other software components, such as error code definitions. It is provided as a source code.

Other project-oriented files (e.g. make and help files) are in the main directory.

3 Operating System Abstraction Layer - OSAL

The software components of the SDK are designed for use on different platforms. Since the interface to access I²C buses, GPIO pins, and timer ticks differ per platform, each software component using such resources contains an Operating System Abstraction Layer (OSAL) - which contains the platform-specific code to access these resources. When porting a software component to a custom platform, only the corresponding OSAL needs to be adopted. All other source codes of the software component can be left unmodified.

For the sample code provided, the SDK contains an OSAL library, which implements the OSAL for the Chip Library. The OSAL library, which is only available as a binary, uses the board as a bridge for I²C communication and signal interrupts from the sensor. It is located in the `chiplib\bin\x86-64\windows` directory.

The OSAL description and functional dependencies can be found in the ChipLib document: “as7352_chiplib_doc_v1.1.2.0.pdf”.

An adaptation of the OSAL when modifying a specific custom hardware, is shown in chapter 7.

4 Sample Codes

Within the SDK are pre-designed sample codes in the `sample_code` directory. Each of the examples contained in the SDK is provided as pre-built executables (.exe). They can be found in the “`sample_code\bin\x86-64\windows`” directory.

Figure 3:
Directory: `sample_code\bin\x86-64\windows`

```
18.05.2022 08:22 <DIR> .
18.05.2022 08:22 <DIR> ..
11.05.2022 17:15      403.474 as7352_chiplib_ftdi.dll
11.05.2022 17:15      419.718 as7352_chiplib_unicom.dll
11.05.2022 17:15      360.019 as7352_sample_code_chiplib_ftdi.exe
11.05.2022 17:15      360.023 as7352_sample_code_chiplib_unicom.exe
      4 File(s)      1.547.330 bytes
      2 Dir(s) 248.128.430.080 bytes free
```

Each sample demonstrates the data and library including the definitions, global and local library functions, the sensor configurations and initializations, as well as sensor measurements and alternative post-processing.

- (A) The “`as7352_sample_code_chiplib_ftdi.exe`” shows how the AS73xx Chip Library can be used in common with the FTDI interface.
- (B) The “`as7352_sample_code_chiplib_unicom.exe`” shows how the AS73xx Chip Library can be used in common with the Unicom interface.

Necessary libraries for the interfaces are stored as DLLs in the directory. They are required for the compiling process and should be not removed or changed.

5 Executing the Sample Code in Windows

Figure 4 shows an example of a call with the parameters and results on the screen using a sensor board with an FTDI interface.

CALL Example (A) > `as7352_sample_code_chiplib_ftdi.exe -interface COM:COM47`

Figure 4:
Sample Call and results

```
J:\AMSIAMS products\AS7343 + AS7352 Skywalker\02c_firmware\SDK\sample_code\bin\x86-64\windows>as7352_sample_code_chiplib_ftdi.exe -interface COM:COM214
Press Ctrl-C to stop the measurement.
ADC Values: 0, 28, 66, 58, 203, 165, 4193, 14, 37, 51, 65, 166, 4192, 9, 89, 83, 18, 166, 4192; Gain: 256x; Integration Time: 182000 us
ADC Values: 0, 232, 559, 487, 1664, 1354, 35197, 118, 313, 421, 527, 1357, 35166, 79, 757, 689, 147, 1355, 35174; Gain: 2048x; Integration Time: 182000 us
ADC Values: 0, 234, 559, 484, 1657, 1352, 35178, 118, 313, 421, 528, 1351, 35141, 78, 756, 691, 146, 1352, 35171; Gain: 2048x; Integration Time: 182000 us
ADC Values: 0, 232, 559, 489, 1663, 1353, 35132, 117, 308, 418, 526, 1336, 35070, 76, 756, 694, 144, 1333, 35046; Gain: 2048x; Integration Time: 182000 us
ADC Values: 0, 232, 559, 484, 1665, 1337, 35194, 118, 310, 422, 526, 1339, 35176, 77, 756, 694, 144, 1335, 35121; Gain: 2048x; Integration Time: 182000 us
```

The pre-build executables are located in the “`sample_code\bin\x86-64\windows`” directory.

To run the pre-built examples, perform the following steps:

1. Connect the sensor board with the interface (FTDI or Unicom) to the PC via USB.
2. Identify the serial port (also known as COM port) assigned to the board. This information is located in the Device Manager. The board is displayed as “USB Serial Device (COMx)” (with x being an integer) in the “Ports (COM & LPT)” section of the Device Manager. The text within the parentheses represents the serial port assigned to the board, for example, “COM47”¹.



Information

The Device Manager can be opened by pressing the “Window key+R” on the keyboard. In the newly opened window, type “devmgmt.msc” and click OK.

If there are multiple USB Serial Device (COMx) entries in the Device Manager, you can inspect each entry to identify the entry belonging to the EVK board. To do so, double-click the entry, navigate to the “Details” tab in the newly opened window, and select “Bus reported device description” from the dropdown menu. If the selected entry belongs to the EVK board, the text “AS7352 EVK” or similar is shown below the drop-down menu.

3. Open a Command Prompt and navigate to the “`sample_code\bin\x86-64\windows`” directory within the SDK.

¹ COM47 is only an example and must be adapted.



Information

The Command Prompt can be opened by navigating to the extracted “sample_code\bin\x86-64\windows” directory in Explorer. Then, type “cmd” in the address bar of Explorer and press Enter. This opens a Command Prompt that has already navigated automatically to the required directory.

4. In the Command Prompt, insert the file name of the sample code executable you want to execute, followed by the arguments required by the program. Ensure to insert a space character before each argument. Afterward, press Enter to run the sample code. All programs require the serial port of the board to be provided via an argument that uses the format `--interface COM:COMx`.
Example: Assuming the “*as7352_sample_code_chiplib_ftdi.exe*” is to be executed and the serial port COM47 has been assigned to the board,
`as7352_sample_code_chiplib_ftdi.exe --interface COM:COM47` needs to be entered in the Command Prompt.
5. Afterward, the output of the sample code is printed to the Command Prompt.
6. Lastly, press the keys Ctrl-C to stop the sample code.

6 Building the Sample Code

The AS7352 Software Development Kit contains a Make file to build the sample code executables from the provided source code and libraries. A GCC-based build environment suitable for building Windows binaries needs to be installed on your computer.

There are multiple ways to install GCC on Windows systems. This section briefly explains how it can be installed using the [Mingw-w64 project](#) and how this GCC installation can be used to build the sample code. However, any other suitable GCC installation can alternatively be used to build the sample code.

1. To install GCC via Mingw-w64, download the [Mingw-w64 installer](#).
2. Execute the installer. When prompted select the following settings:
 - Version: 8.1.0
 - settings: Version: 8.1.0
 - Architecture: x86_64
 - Threads: posix
 - Exception: seh
 - Build revision: 0
3. After completing the installation, add the subdirectory “*mingw64/bin*” of your Mingw-w64 installation to the “PATH” environmental variable of your system or user.

Example: If you select “*C:\Program Files\mingw-w64\x86_64-8.1.0-posix-seh-rt_v6-rev0*” as destination folder during installation, the directory “*C:\Program Files\mingw-w64\x86_64-8.1.0-posix-seh-rt_v6-rev0\mingw64\bin*” needs to be added to the PATH environmental variable.



Information

Perform the following steps to add a PATH environmental variable:

1. Open the start menu, enter “Edit environment variables for your account” in the search bar and select the same result.
 2. In the upper part of the window (“User variables for x”), double-click on the “Path” entry.
 3. Then, add the installation path using the “New” button.
 4. Afterward, close all the windows with the OK button.
 5. Finally, close the command prompt and reopen it again.
-

6. To build the sample code, navigate to the root directory of the extracted SDK in a Command Prompt and run the command `"mingw32-make CC=gcc all"`.
After building is completed successfully, the executables can be found in the newly created "build" directory of the SDK. The file names of all executables built from the source code include the suffix `"_source"`.
Example: `"mingw32-make CC=gcc AS7352_sample_code_chiplib_ftdi"` calls the compiler to make an executable file using the C++ compiler.
The file names of the executables built from source code are:
`as7352_sample_code_chiplib_ftdi_source.exe` or
`as7352_sample_code_chiplib_unicom_source.exe`.

7 Customizing OSAL and Example Code

When porting the AS7352 Chip Library to a customer-specific platform (hardware and/or software), the corresponding OSAL needs to be implemented for the given platform.

For the AS7352 Chip Library, a template named `“AS7352_osal_chiplib_template.c”` is provided in the `“chiplib\src”` directory of the SDK. The OSAL template file can be used as a starting point when implementing a custom OSAL. All the places where the insertion of platform-specific code is required are marked with `“TODO”`.

7.1 Including the Chip Library in a Custom Application

To include the AS7352 Chip Library with the customized OSAL in an application, the OSAL implementation and the following files of the SDK need to be copied to the target project:

- All contents of `“chiplib\src”`.
- All contents of `“chiplib\inc”`, including the subdirectory `“chiplib”`.
- `“error_codes.h”` of `“utilities\inc”`

When compiling the source files, set the ‘compiler’s include search paths’ to include the path of the directory where the contents of `“chiplib\inc”` were copied. The included search paths must also contain the path of the directory containing `“error_codes.h”`. Additionally, the compiler must be used to define `“NUM_SUPPORTED_DEVICES”` to the number of used AS7352 Chip Library instances.

Example: Assuming that all the contents of the SDK ZIP archive have been extracted, a single AS7352 Chip Library instance is required, and the `chiplib` is the current working directory, the AS7352 Chip Library can be built from the source as a shared library by invoking:

```
gcc src/*.c -Iinc -I../utilities/inc -DNUM_SUPPORTED_DEVICES=1 -shared -  
fpic -o AS7352_chiplib_source.dll.
```

Headquarters

ams AG
Tobelbader Strasse 30
8141 Premstaetten
Austria, Europe
Tel: +43 (0) 3136 500 0

Please visit our website at www.ams.com

Buy our products or get free samples online at www.ams.com/Products

Technical Support is available at www.ams.com/Technical-Support

Provide feedback about this document at www.ams.com/Document-Feedback

For sales offices, distributors and representatives go to www.ams.com/Contact

For further information and requests, e-mail us at ams_sales@ams.com

Copyright ams AG. Trademarks Registered. All rights reserved. The material herein may not be reproduced, adapted, merged, translated, stored, or used without the prior written consent of the copyright owner.

ams AG shall not be liable to recipient or any third party for any damages, including but not limited to personal injury, property damage, loss of profits, loss of use, interruption of business or indirect, special, incidental or consequential damages, of any kind, in connection with or arising out of the furnishing, performance or use of the technical data herein. No obligation or liability to recipient or any third party shall arise or flow out of ams AG rendering of technical or other services.