# AS7341 ChipLib RPC

Documentation
revision v0.10.0

Generated by Doxygen

# Contents

# 1 AS7341 ChipLib RPC

## 1.1 Introduction

This document provides an overview on how the spectral sensor AS7341 can be controlled via the RPC-Library (Remote Procedure Control Library). With the help of this library (which will be provided as ANSI-C-DLL) the ChipLib on the remote target can be accessed like you call the ChipLib-functions directly. This is very helpful for the first steps with ChipLib. Feasibility studies and evaluations can be implemented on host-PC. After successful work and test, the application source code can be transferred to the target hardware. In this case, the application doesn't use the RPC anymore. But with the same interface functions, the ChipLib can be called directly and the effort to move the application to the target will be lower.

Key features:

- Same interface like ChipLib
- Runs on Windows and Linux Hosts
- Easy use case evaluation without playing with custom hardware and firmware development
- Execution of ChipLib is much faster than running ChipLib directly on host PC with I2C-bridge like FTDI-cable

## 1.2 ChipLib-Library

ChipLib is a library which handles the communication to our color and spectral sensors. The interface to this library is standardized. For compatiblity reasons, the ChipLib RPC has the same interface.

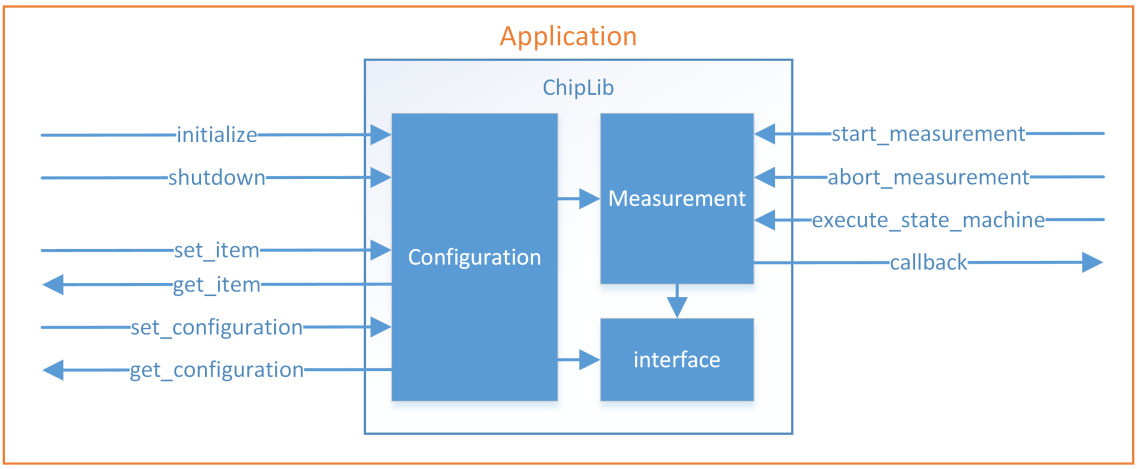


**Figure 1 ChipLib API**

For more detailed information, please have a look into the corresponding ChipLib documentation.

## 1.3 Overview

This chapter describes the software-stack of the ChipLib-RPC.

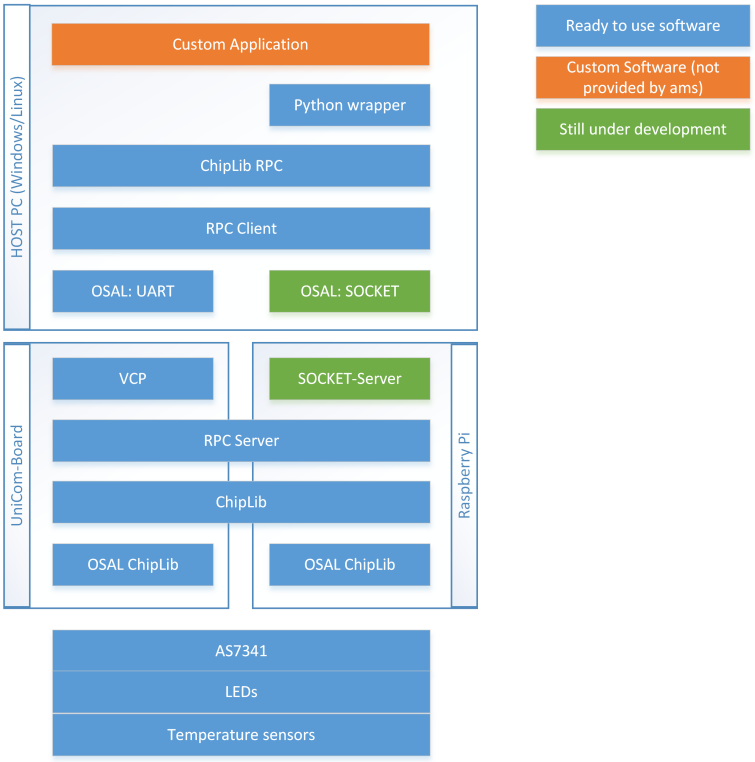


**Figure 2 Software-Structure**

- OSAL ChipLib: This abstraction layer provides the basis to move the current test environment to new target platform after finished evaluation. The detailed description of this layer is shown in the documentation of the ChipLib.
- ChipLib: That's the heart of the sensor control. The configuration and measurement process is included here. Please see chapter ChipLib-Library.
- RPC-Server: The RPC-server decodes the bytes stream of the hardware interface to dedicated function calls of the ChipLib.
- VCP: Virtual Com Port class of USB. This handles the messages from host device via USB and sends answers back.
- SOCKET-Server: The same functionality like VCP but the hardware interface is ethernet and as protocol UDP shall be used. (Currently planned but not finished yet)
- OSAL UART: Abstraction layer for the the RPC client. OS-specific periphery and interfaces like UART, threads and events will be handled here.
- OSAL SOCKET: The same like OSAL UART, but the interface is a client socket.
- RPC Client: Generic RPC client which converts function calls to byte streams and send this to the remote device.

- ChipLib RPC: This library has the same interface like the ChipLib and provides access to the real ChipLib on the remote device.
- Python wrapper: If you don't want to use the C-DLL directly you can use the Python-wrapper to access the DLL.

## 1.4 ChipLib-RPC

### 1.4.1 Interface-Initialization

The native ChipLib and the RPC-derivate have the same interface. Therefore the same header-file can be used. For purposes of hardware-detection, the initialize-function must be called with one different parameter. The parameter "p_interface_descr" will not be transferred to ChipLib-OSAL. Instead it will be used to initialize the OSAL of the RPC-client. To support UART and SOCKET, two different prefixes are supported:

- COM: Write to the paramter "p_interface_descr" <COM:COM22> initializes the UART-OSAL with COM-port 22 on Windows PCs. <COM:ttyUSB0> or <COM:ttyS0> initializes the COM-port on Linux-based operating systems.
- UDP: Write to the same parameter <UDP:192.168.0.2:55>, the SOCKET-OSAL will be used instead of the UART-OSAL. The first four values are the IP-Address. After colon follows the port number. (Will be supported in future)

### 1.4.2 ChipLib statemachine

The ChipLib-function "execute_state_machine" will be executed for performance reasons directly on the target. It runs all the time, independent of the state machine state. Therefore it is not necessary to call this function cyclic on the host machine. Of course, the state machine state can be retrived with the function "execute_state_machine". It can be used to get information about the current measurement and if it is finished.

## 1.5 Acronyms and Abbreviations

### 1.5.1 Acronyms

ANSI = American National Standards Institute
API = Application Programming Interface
DLL = Dynamic Link Library
IP = Internet Protocol
I2C = Inter-Integrated Circuit
LED = Light-Emitting Diode
OSAL = Operating System Abstraction Layer
RPC = Remote Procedure Control
UART = Universal Asynchronous Receiver Transmitter
UDP = User Datagram Protocol
VCP = Virtual Com Port

### 1.5.2 Abbreviations

ChipLib = Chip Library for higher level communication with the sensor
FTDI = Future Technology Devices International Ltd.
Item = A property or a setting inside the ChipLib or the sensor

## 1.6 List of References

[1] AS7341 ChipLib Documentation, ams AG, as7341_chiplib_docu_vX.Y.Z.pdf