UM1792
User manual

STLUX385A development environment

Introduction

This user manual provides complete information for SW developers about the complete STLUX385A development environment flow.

The STLUX385A development environment is a suite of tools that help developing applications guiding the user through the whole prototyping process, from the initial idea to the onboard proof of concept. It also helps beginners facing to STLUX385A technology to get familiar with it and start developing applications as soon as possible.

The STLUX385A is an STMicroelectronics® digital device tailored for lighting applications. The heart of the STLUX™ is the SMED ("State Machine Event Driven") technology which allows the device to operate several independently configurable PWM clocks with up to 1.3 ns resolution. A SMED is a powerful autonomous state machine which is programmed to react to both external and internal events and may evolve without any software intervention. The information provided by this document will help you to understand how to install the whole STLUX development environment, how to run the tools and how to develop, run and test your applications.

The STLUX385A SMEDs are configured and programmed via the STLUX internal low power microcontroller (STM8). This manual describes the whole design flow to easily use the STLUX385A technology.

Reference documents

- For hardware information on the STLUX385A controller and product specific SMED configuration, please refer to the STLUX385A product datasheet.
- For information on programming, erasing and protection of the internal Flash memory please refer to the programming manual "How to program STM8S and STM8A Flash program memory and data EEPROM" (PM0051).
- For information about the debug and SWIM (single-wire interface module) refer to the STM8 SWIM communication protocol and debug module user manual (UM0470).
- For information on the STM8 core and assembler instruction please refer to the STM8 CPU programming manual (PM0044).
- For information on the SMED configurator please refer to the "STLUX™ SMED configurator 1.0" user manual (UM1760).
- For information on the STLUX385A peripheral library please refer to the STLUX™ peripheral library user manual (UM1753).
- For information on the STLUX385A examples kit please refer to the STLUX™ examples kit user manual (UM1763).

Contents

List of figures

1 Acronyms

A list of acronyms used in this document:

Table 1. List of acronyms

Acronym	Description
ACU	Analog comparator unit
ADC	Analog-to-digital converter
ATM	Auxiliary timer
AWU	Auto-wakeup unit
BL	Bootloader - used to load the user program without the emulator
CCO	Configurable clock output
CKC	Clock control unit
CKM	Clock master
CPU	Central processing unit
CSS	Clock security system
DAC	Digital-to-analog converter
DALI	Digital addressable lighting interface
ECC	Error Correction Code
FSM	Finite state machine
FW	Firmware loaded and running on the CPU
GPIO	General purpose input/output
HSE	High-speed external crystal - ceramic resonator
HSI	High-speed internal RC oscillator
I2C	Inter-integrated circuit interface
IAP	In-application programming
ICP	In-circuit programming
ITC	Interrupt controller
IWDG	Independent watchdog
LSI	Low-speed internal RC oscillator
MCU	Microprocessor central unit
MSC	Miscellaneous
PM	Power management
RFU	Reserved for future use
ROP	Read-out protection
RST	Reset control unit
RTC	Real-time clock

Table 1. List of acronyms (continued)

Acronym	Description
SMED	State machine event driven
STMR	System timer
SW	Software, is the firmware loaded and running on the CPU (synonymous of FW)
SWI	Clock switch interrupt
SWIM	Single wire interface module
UART	Universal asynchronous receiver transmitter
WWDG	Window watchdog

2 Development environment components

The STLUX385A development environment is composed of various tools aimed to help the user to develop his application aiding him during the whole design flow, from the idea formal specification to the onboard testing and prototyping as a proof of a concept.

This document describes how to set up all the development environment tools which are the SMED configurator, the STLUX385A peripherals library, the STLUX385A examples kit and the chosen software development Workbench® (IAR/Raisonance). It also explains how to set up a test bench connecting your PC together with the STEVAL-ILL068V1 evaluation board.

3 Design flow: from the idea to the prototype

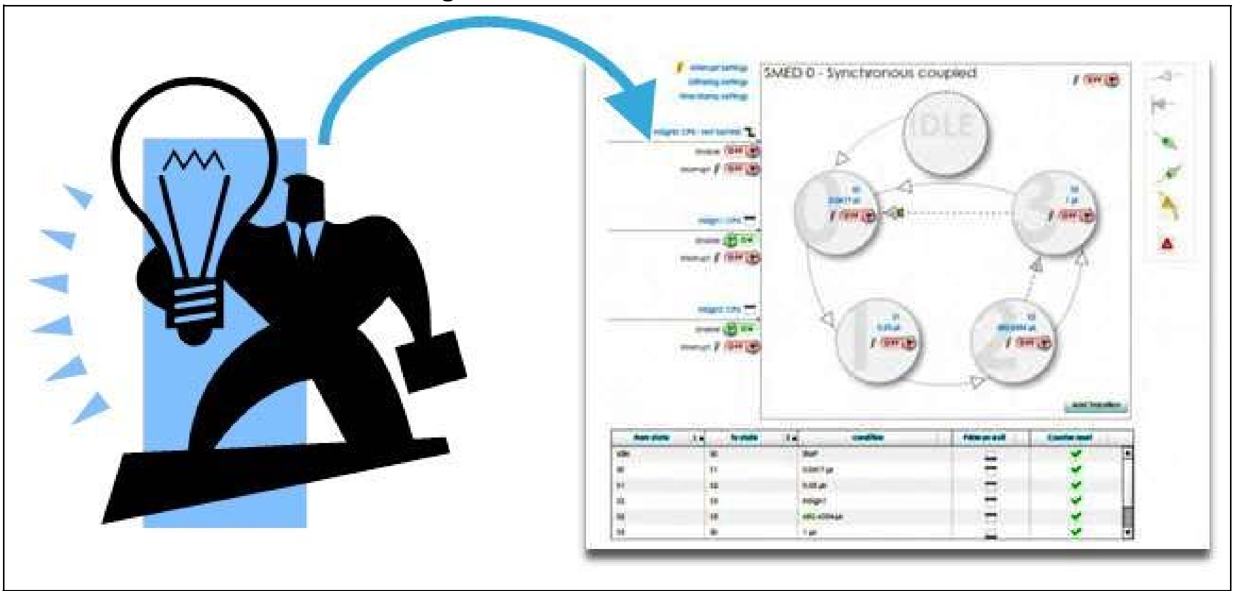
The aim of this paragraph is to give you a brief overview of the STLUX development environment design flow for fast prototyping. A more detailed description of the step-by-step operations for the whole flow will be given in the following paragraphs.

3.1 Shaping your idea

Expressing your idea into a real application can really be tough sometimes because one is focused on the main behavior of the system and doesn't want to pay attention to the implementation details and constraints. In order to help the user specifying the behavior of the system in terms of the FSM, the STLUX385A development environment provides the SMED configurator which is a powerful tool for guiding the SMEDs configuration. The SMED configurator user-friendly graphic environment allows the user to draw his FSM, setting inputs, clock frequencies and outputs. It also verifies the FSM correctness, checking logical conditions.

Once the application behavior is specified into a specific SMEDs configuration, the SMED configurator allows to save it into a proprietary format project file *.prj which allows to easily share, retrieve and modify it later. After saving the configuration project it is possible to automatically generate an ANSI C configuration file which defines the implementation of the void standard function SMED_Init() that can be easily integrated in your application firmware code.

Figure 1. SMED 0 state machine

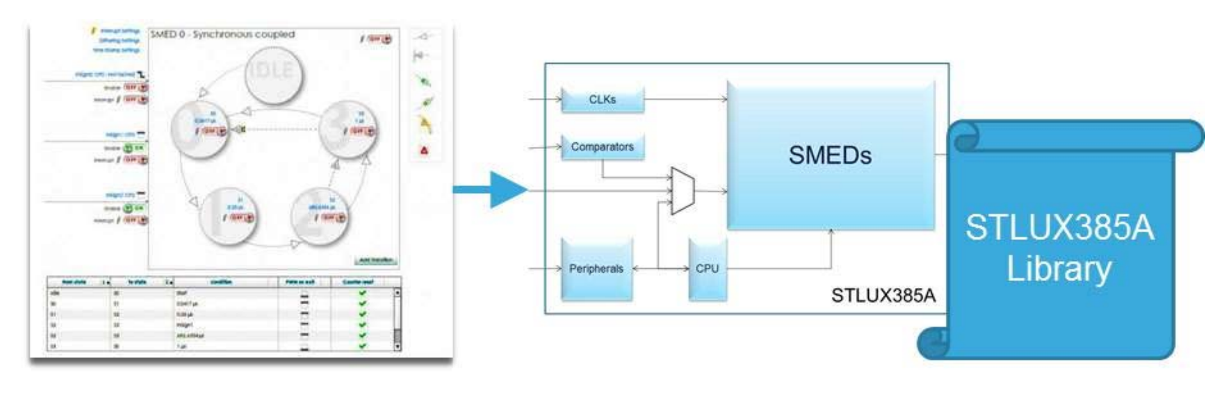


3.2 Firmware design

When the application behavior has been specified in terms of SMEDs configuration, you need to set the main frame of your application initializing the clocks and the peripherals you need to let the SMEDs be fed with the desired inputs now.

In order to ease this task, an STLUX385A peripherals library is deployed as a part of the development environment. The library includes several specific APIs taking care of all the implementation details to handle peripherals while designing your application code.

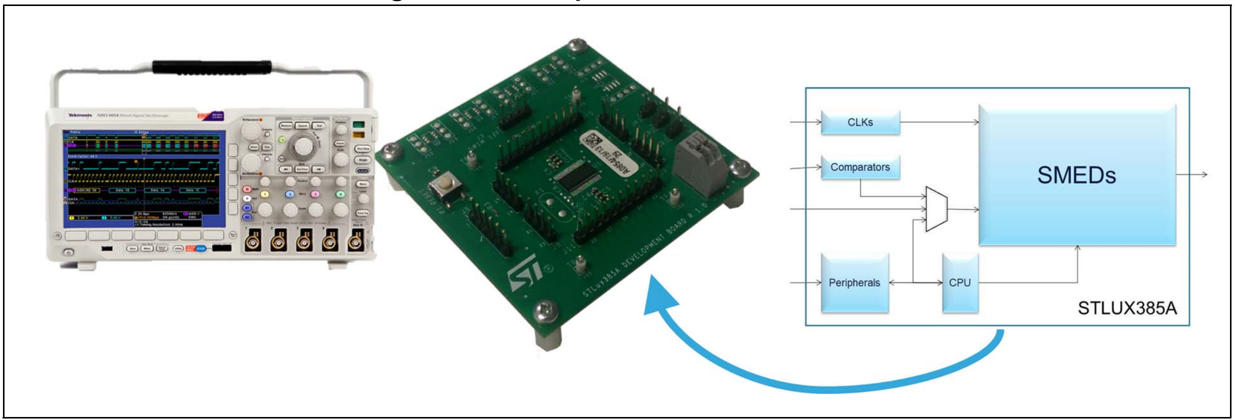
Figure 2. From FSM to FW development



3.3 Fast prototyping

The last step needed to demonstrate your concept is compile your code with an embedded environment supporting the STLUX385A (you can chose IAR or Raisonance), generate a bit file and download it into a STEVAL-ILL068V1 evaluation board. Then you will be able to run your application and to test it using testing and measurement equipment.

Figure 3. Code upload and test onboard

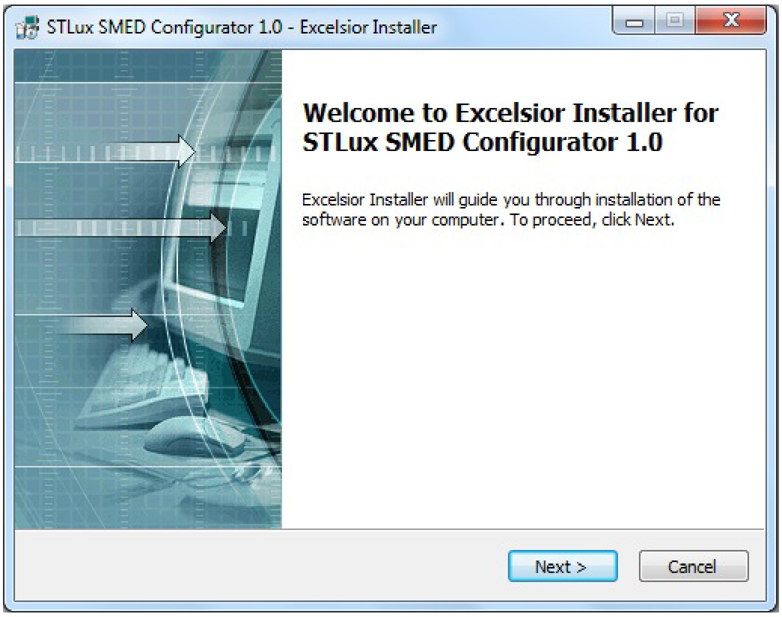


4 Step1: the SMED configurator

The first thing for building your application is getting the SMED configurator. You can download it for free from the STLUX web page: www.st.com/stlux/ at the “Software” section.

Once you downloaded the SMED configurator installation package, simply open it and follow the guided installation procedure that will setup the SMED configurator on your PC.

Figure 4. SMED configurator installer

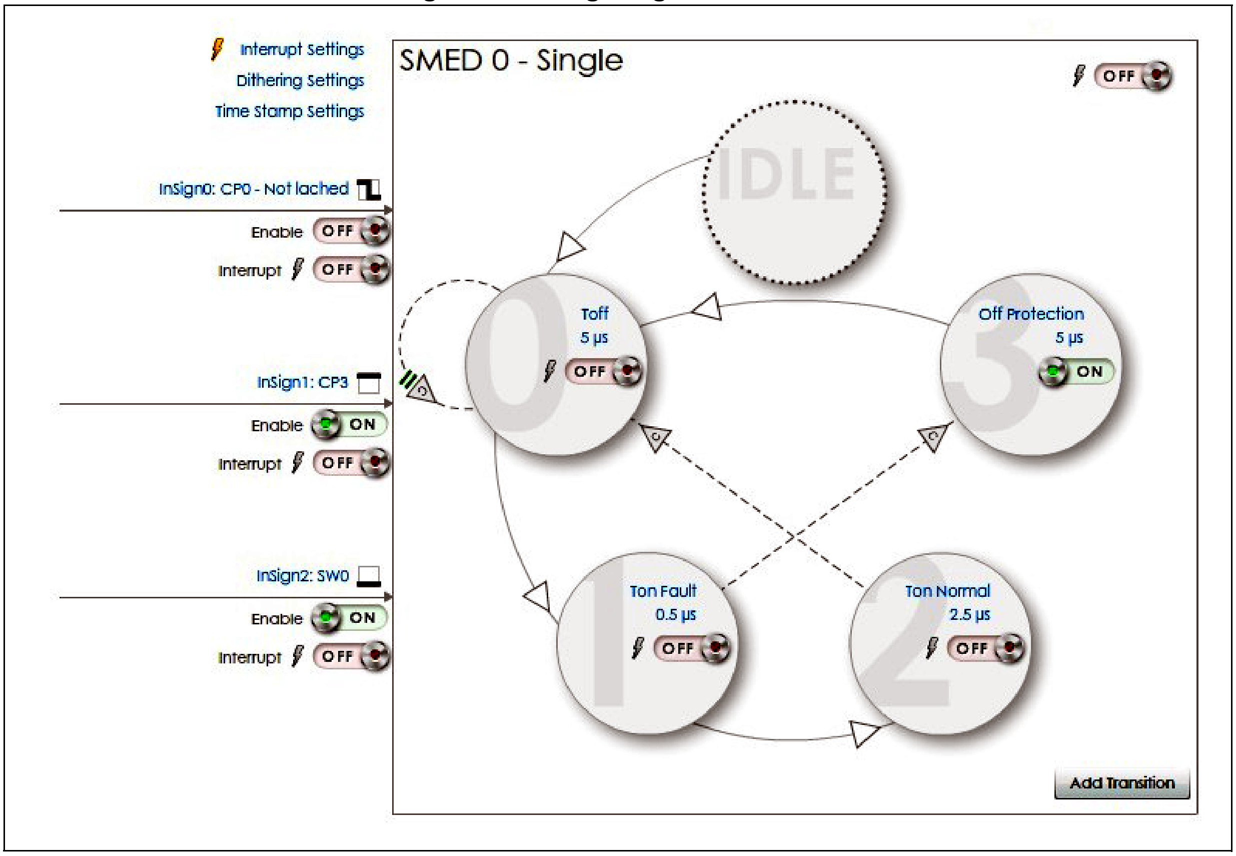


At the end of the installation process you will be able to open the SMED configurator simply choosing the SMED configurator from your listed programs or by clicking on the icon you will find on your desktop.

Figure 5. SMED configurator desktop icon

Now you will be able to start designing your application FSM by setting the SMED clocks, enabling inputs and outputs for the selected SMEDs, and finally determining the SMEDs behavior by specifying the respective FSM as can be seen in *Figure 6*.

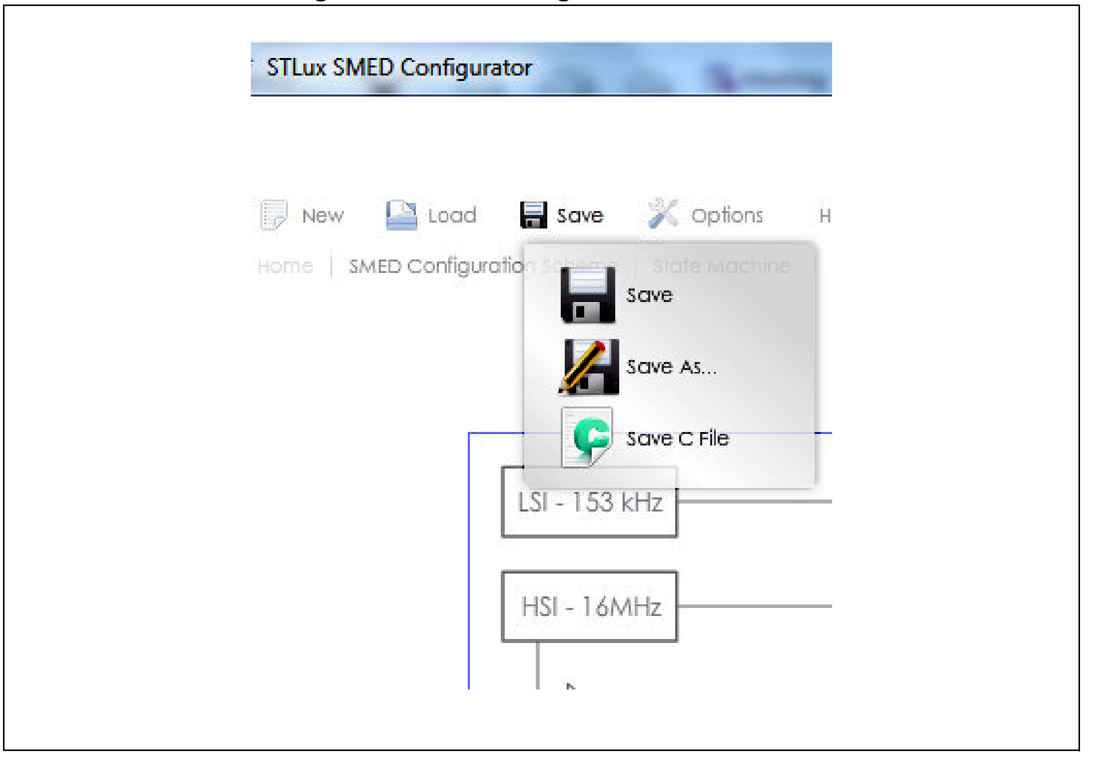
Figure 6. Configuring a SMED FSM



The SMED configurator helps you checking the logical consistency of your FSM and highlights you possible errors, making easier to correctly define your application behavior.

Once you are satisfied with your work, you can save your SMED configurator project in the proprietary format *.prj file to easily retrieve and modify your configuration later. It is possible also to automatically generate the *.c file containing the C code to be embedded in your application. The SMED configurator by default generates the implementation of the SMED_Init() virtual function defined inside the stlux_smed.h peripheral library header file.

Figure 7. SMED configurator save menu



5 Step 2: getting ready with tools

To be able to develop your application you need to get the other useful tools supporting the STLUX385A now. First of all you need to get the STLUX385A peripherals library which is provided by ST. The STLUX also integrates a STM8 core which can be programmed through an embedded software development environment. The ones supporting the STLUX385A are the IAR embedded environment or Raisonance embedded environment and you can chose one of them to develop your application.

5.1 STLUX385A peripheral library

The STLUX385A peripherals library is available for download at www.st.com/stlux and it is useful for handling the STLUX385A. In the library it is possible to find useful APIs making easier to configure peripherals like clocks, timers, I/O ports, SMEDs and comparators while developing your application. For more information about the library, please check out the STLUX385A peripherals library user manual UM1753.

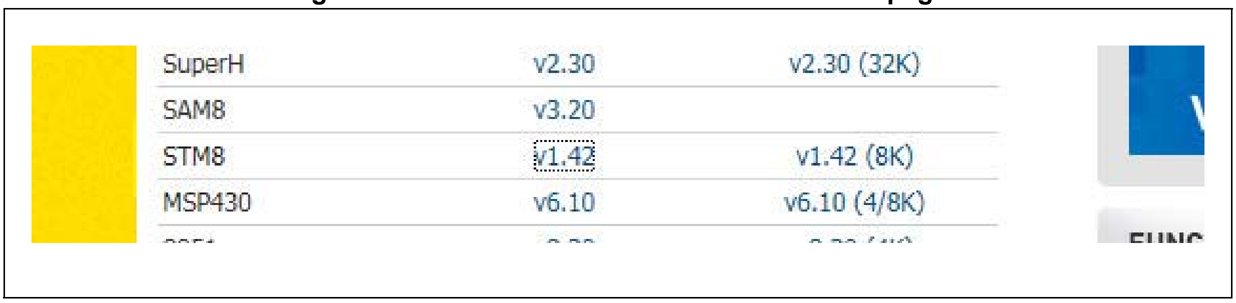
5.2 STLUX385A examples kit

The STLUX385A examples kit is available for download at www.st.com/stlux and it is useful for getting familiar with the STLUX385A and its functionalities before starting to develop your application. For more information about the examples kit, please check out the STLUX385A examples kit user manual UM1763.

5.3 IAR embedded environment

The IAR Embedded Workbench® supports the STLUX385A device and you can easily download it from the the IAR website www.iar.com/Service-Center/Downloads/.

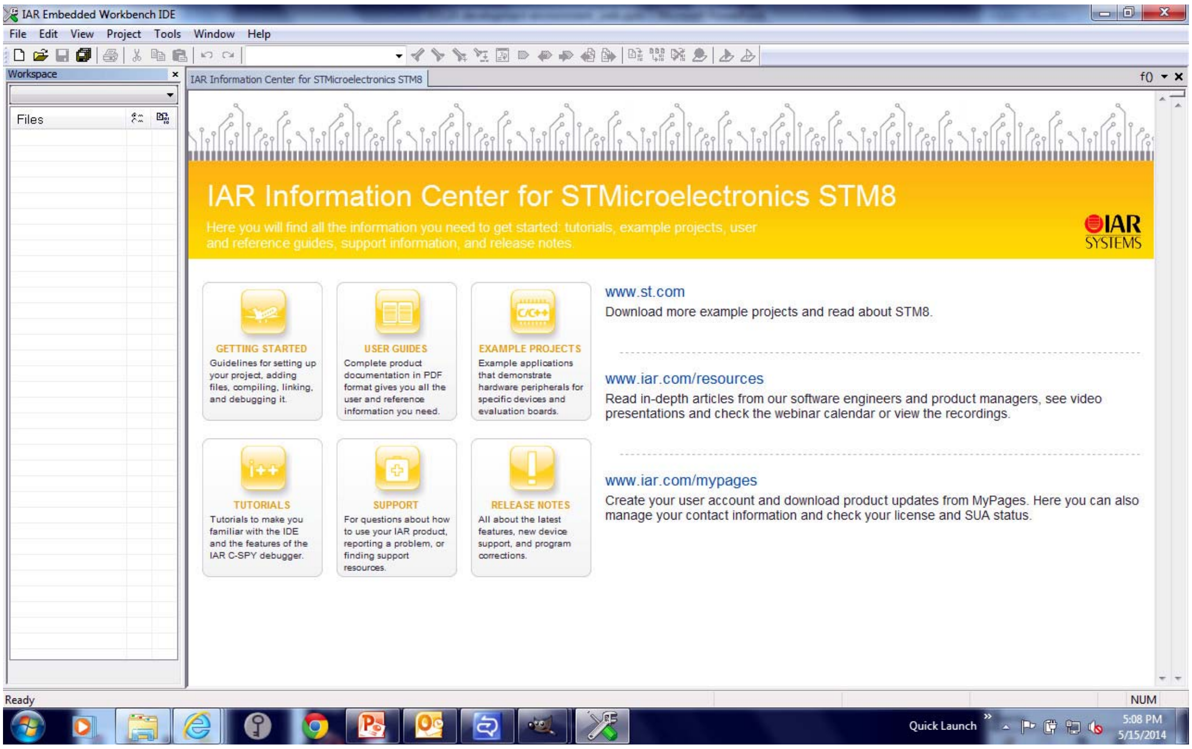
Figure 8. IAR Embedded Workbench download page



Once you downloaded the installation files, simply follow the step-by-step guided installation procedure to make the IAR Workbench properly run on your PC. IAR offers a free time-limited (30 days) license or a size-limited (8 KB code) license. You can update your free license to get a full license later.

Now you can run the IAR Embedded Workbench and start looking at the examples or developing your application.

Figure 9. IAR Embedded Workbench starting page



Once you designed your application code, compiled it and generated the bit file, you can download it into the STEVAL-ILL068V1 evaluation board to debug and verify it.

To do so, the IAR software development Workbench supports the ST_LINK/V2 in-circuit debugger/programmer interface. The ST_LINK/V2 interface can be connected to the STLUX evaluation board through the SWIM ("Single Wire Interface Module") interface and directly to your PC via a USB port.

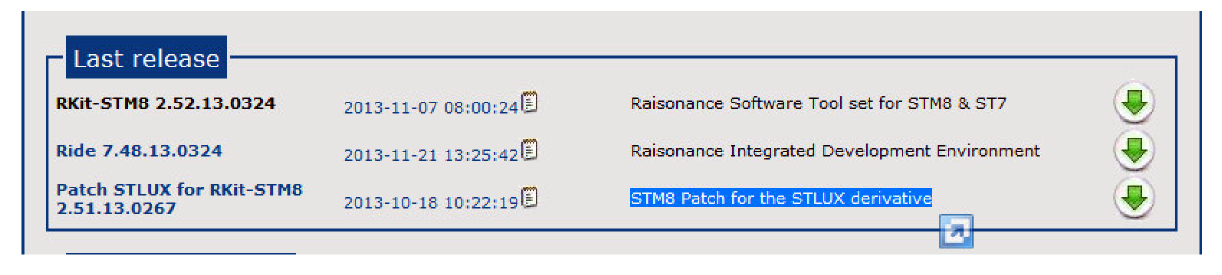
Figure 10. Connecting STEVAL-ILL068V1 to your PC via ST_LINK/V2

The ST_LINK/V2 is available online from most of the suppliers. You can find more information on the ST_LINK/V2:
http://support-raisonance.com/extranet/tools/index.php?param=tool__id__11.

5.4 Raisonance embedded environment

Raisonance embedded environment supports the STLUX385A and you can download Raisonance Embedded Development Tools for the STM8 from the Raisonance website: http://support-raisonance.com/extranet/tools/index.php?param=tool__id__11.

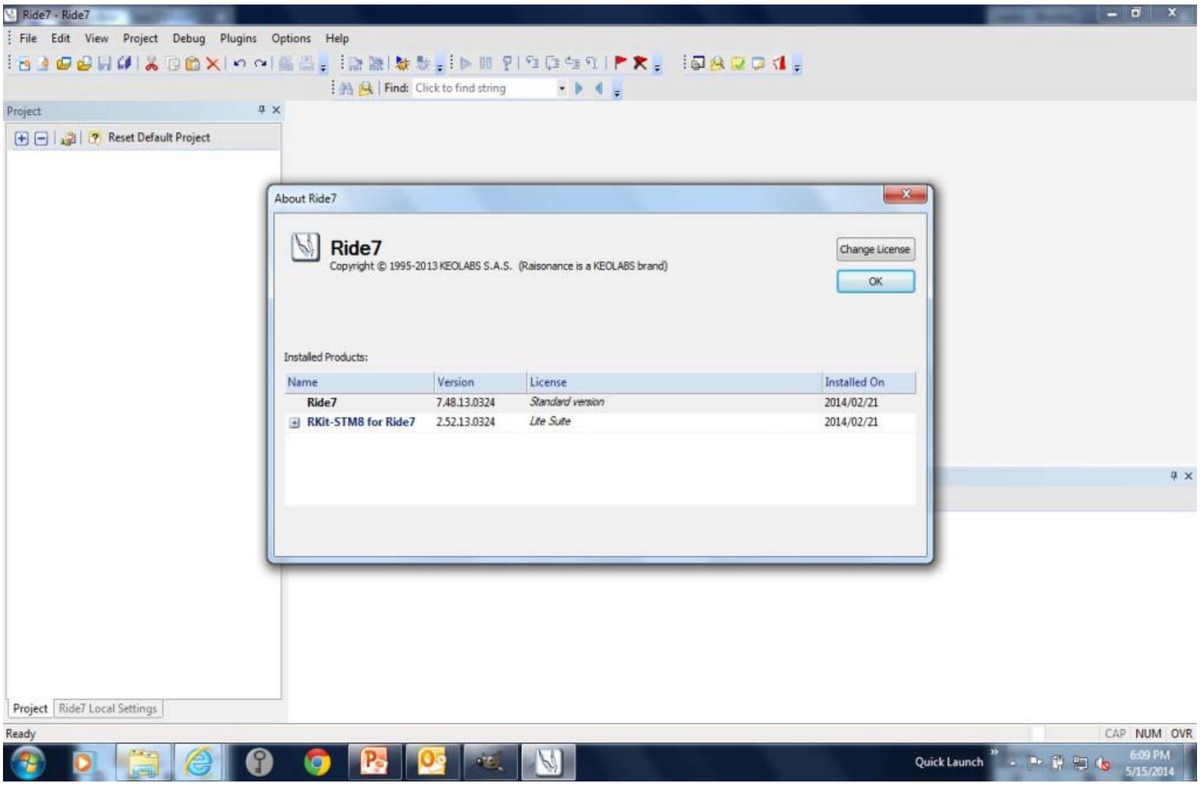
Figure 11. Raisonance Embedded Development Tools download page



As shown in *Figure 11*, Raisonance requires to install first of all Ride - Raisonance Integrated Development Environment. Then you need to install the RKit-STM8 - Raisonance software toolset for the STM8 and ST7 which enables the Ride to support the STM8. Last but not least, you need to install the Patch STLUX - Raisonance Patch to extend support to the STLUX385A. Installing each component is easy through a step-by-step guided installation procedure. Raisonance offers a 30 day free license for evaluation and after this time a size-limited (32 KB 2 KB code) license. You can update your free license for a full one later.

You can find more information on the Raisonance Embedded Workbench for the STM8 on the Raisonance website www.raisonance.com/stlux.html. Then click on the icon on your desktop and launch the Ride.

Figure 12. Ride development environment starting page



Once you designed your application code, compiled it and generated the bit file, you can download it into the STEVAL-ILL068V1 evaluation board to debug and verify it.

To do so, Raisonance software development environment supports the Raisonance RLink microcontroller interface. The RLink interface can be connected to the STLUX evaluation board through the SWIM ("Single Wire Interface Module") interface as explained in *Section : RLink non-standard connection for STEVAL-ILL068V1*. Then directly connect it to your PC via a USB port.

Figure 13. Connecting STEVAL-ILL068V1 to your PC via Raisonance RLink

The Raisonance RLink is available online online from the Raisonance website www.raisonance.com/rlink.html.

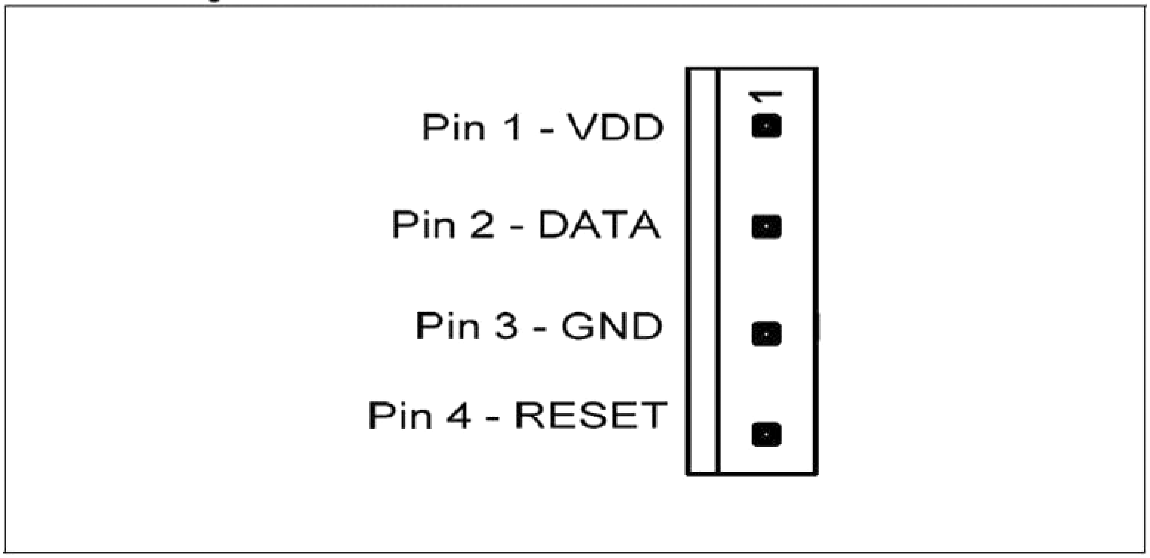
RLink non-standard connection for STEVAL-ILL068V1

Figure 14. RLink-SWIM flat ribbon connection scheme

SWIM flat ribbon connections

Pin no.	Name	Function	Target connection
1	VDD	Target VCC	MCU VCC
2	DATA	SWIM	MCU SWIM pin
3	GND	GROUND	GND
4	RESET	RESET	MCU RESET pin

Target SWIM connector



Since the STEVAL-ILL068V1 currently doesn't support the standard RLink flat connector, actually you need to implement a non-standard flat ribbon connection according to the scheme shown in *Figure 14*.The connection can be implemented using a maximum 10 mm long four-wire cable soldered to the RLink and connected to the four-pin connector (female 2.54 mm). For more information please refer to RLink documentation.

6 Step 3: STEVAL-ILL068V1 evaluation board

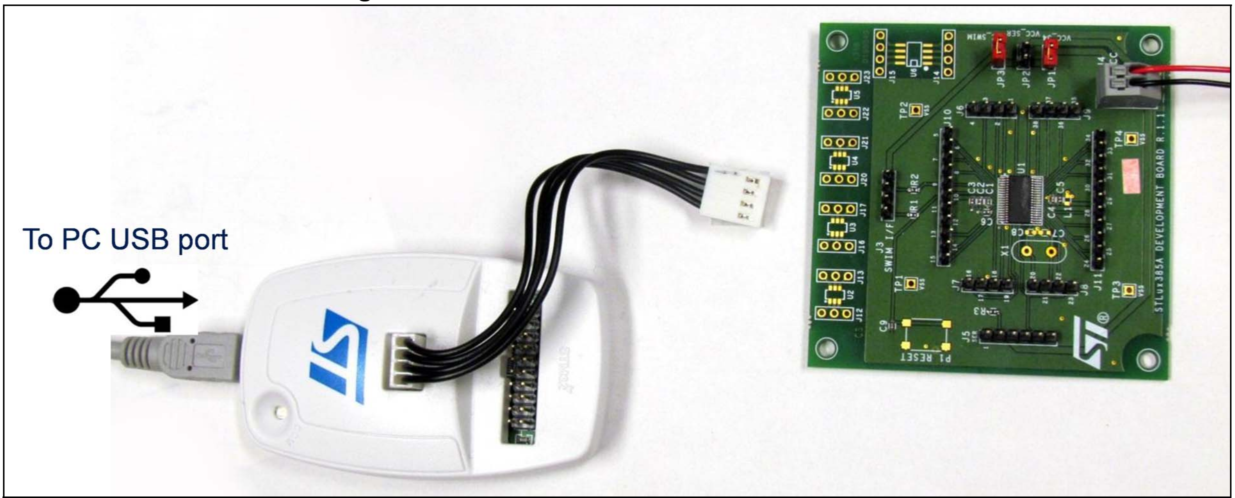
6.1 Regular power supply

First of all before getting a STEVAL-ILL068V1 evaluation board be sure to respect the minimum system requirement to operate with it: get a STEVAL-ILL068V1 evaluation board.

System requirements:

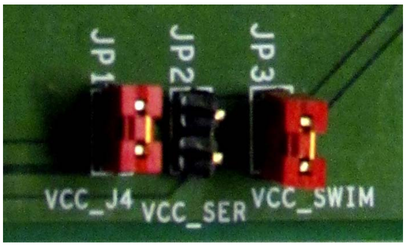
- Microsoft Windows® PC (XP, Vista, 7),
- USB 2.0
- USB A to mini-B cable
- External power supply (from 3 V to 5.5 V and 50 mA minimum).

Figure 15. How to connect STEVAL-ILL068V1



Before connecting the board to the regular power supply using the J4 power connection, be sure the jumpers JP1, JP2 and JP3 are connected as shown in *Figure 16*.

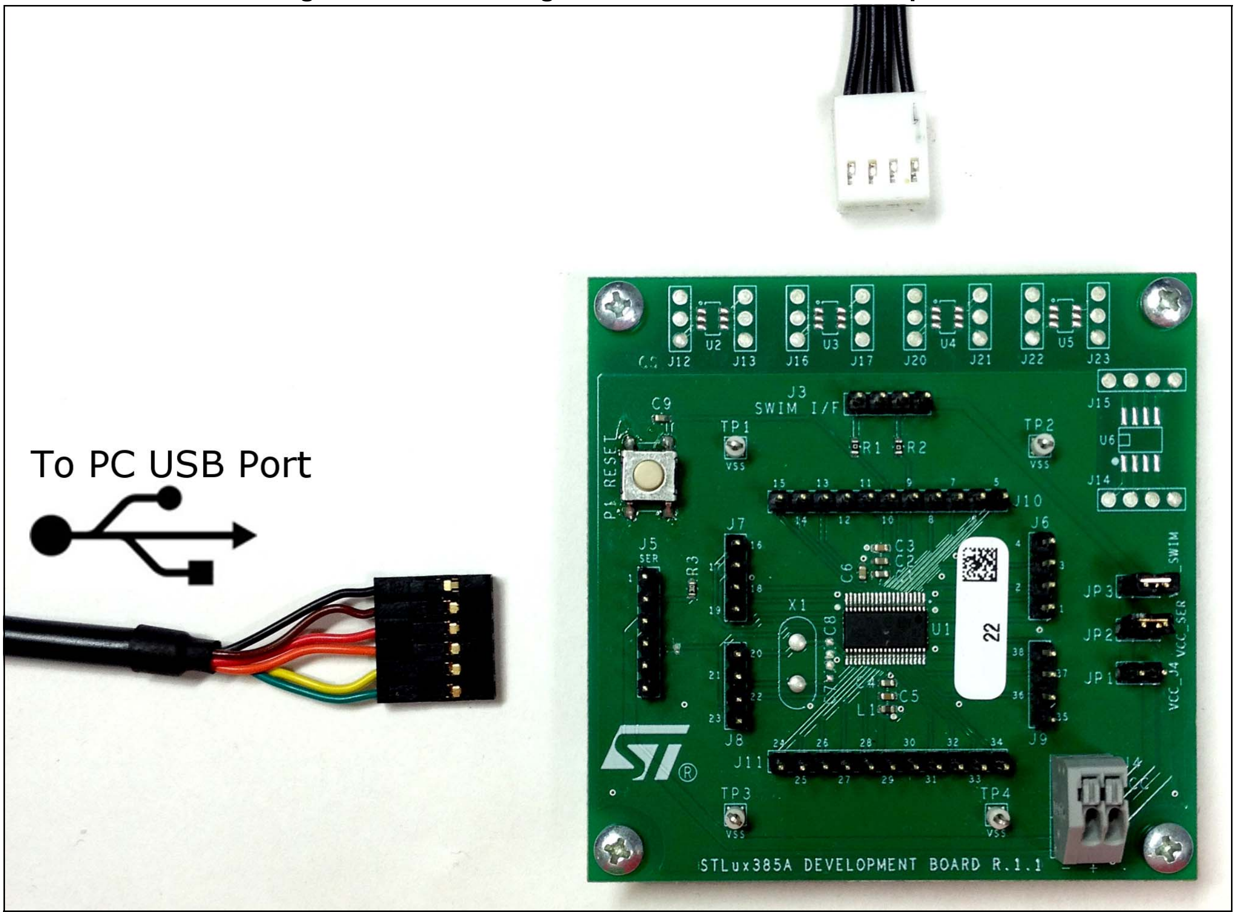
Figure 16. STEVAL-ILL068V1 jumpers configuration for regular supply



Now you can connect the evaluation board and power it up. Then you need to connect the board to your PC using a ST-LINK/V2 or an RLink according to your preferred Embedded development environment (IAR/Raisonance) as explained in *Section 5.3 on page 12* and *Section 5.4*.

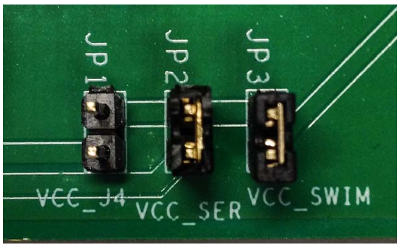
6.2 Serial port power supply

Figure 17. Connecting STEVAL-ILL068V1 to serial port



Alternatively you can chose to supply the board by connecting it to a PC via the serial interface but previously you must be sure to properly configure jumpers JP1, JP2 and JP3 as shown in *Figure 18*.

Figure 18. STEVAL-ILL068V1 jumpers configuration for serial port supply



Please note that the TTL-232R-3V3 USB cable shown on the left side of *Figure 17* is not included within the STEVAL_ILL068V1 evaluation board package and can be purchased directly from FTDI or from any online standard supplier. In this case, when the STEVAL-ILL068V1 power is acquired using the serial cable, the input voltage is fixed to 5.0 V with 50 mA maximum.

6.3 SWIM power supply

In case you adopted the Raisonance embedded environment together with the RLink microcontroller interface connecting the STEVAL-ILL068V1 to your PC, you can alternatively supply the board via the SWIM (“Single Wire Interface Module”) interface which provides a fixed 5.0 V input voltage.

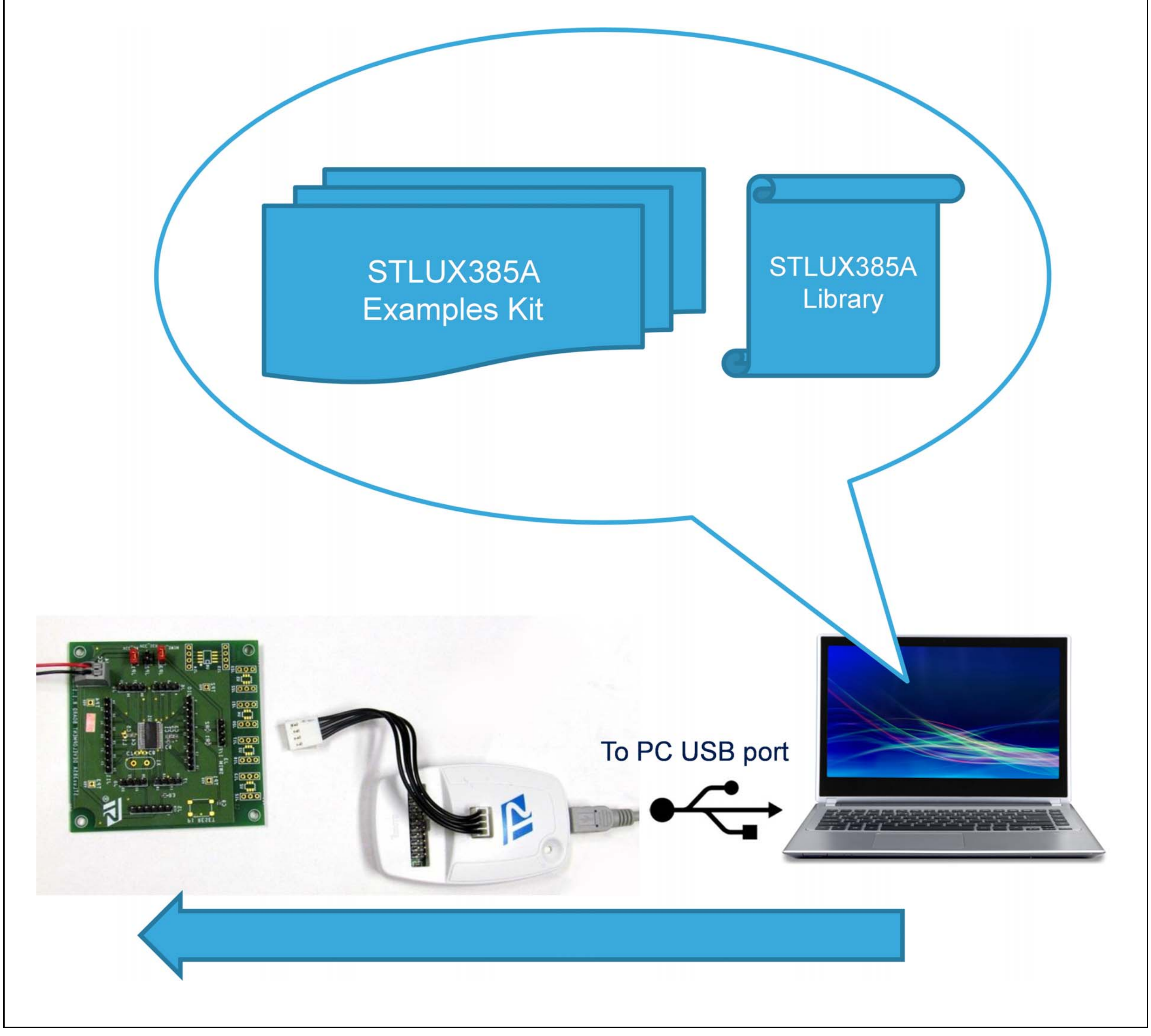
7 Step 4: getting into STLUX385A

By this time you're basically ready to start working with STLUX385A development environment. Now you can open IAR/Raisonance software development environment and start checking out the examples from the STLUX385A examples kit to quickly get hands on with the STLUX385A programming. Playing with the provided examples makes faster to learn how to use the API provided by the STLUX385A peripheral library giving easy access to STLUX385A SMEDs and peripherals.

Compile STLUX385A examples, download them to your STEVAL-ILL068V1 evaluation board using the STLink/RLink device to get an idea of the whole design flow.

Now you're ready to do the same with your firmware, compiling, downloading it on board and running it to verify your idea!

Figure 19. STLUX385A design flow



8 Revision history

Table 2. Document revision history

Date	Revision	Changes
01-Jul-2014	1	Initial release.