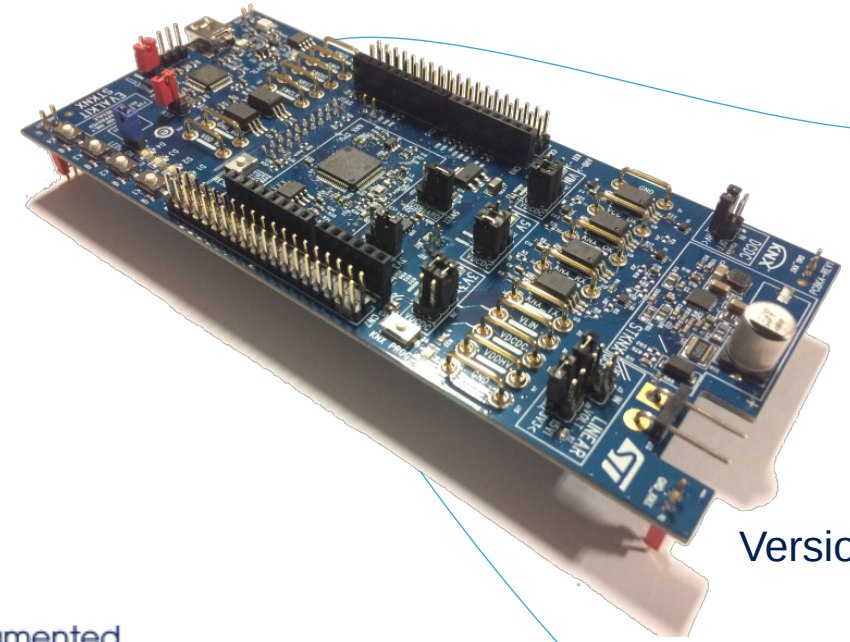
Quick Start Guide

STKNX evaluation board (EVALKITSTKNX)



Version 1.4 (February 9, 2018)

life.augmented

Overview

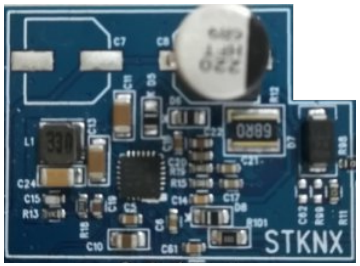
1. EVALKITSTKNX Development Platform overview
2. Software overview and setup of the development environment
3. Setup example

Overview

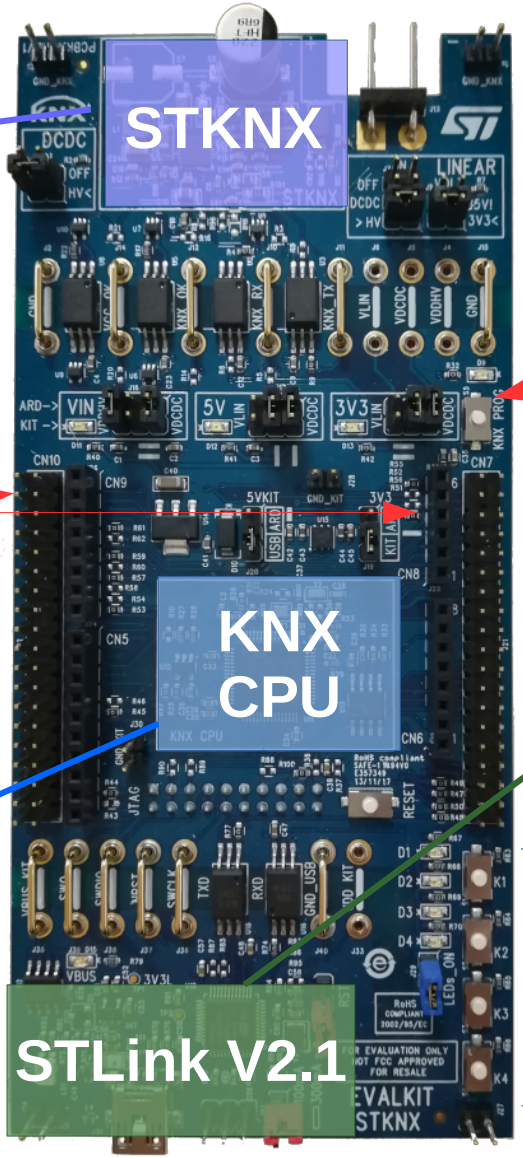
1. EVALKITSTKNX Development Platform overview
2. Software overview and setup of the development environment
3. Setup example

EVALKITSTKNX Development Platform overview

Board description



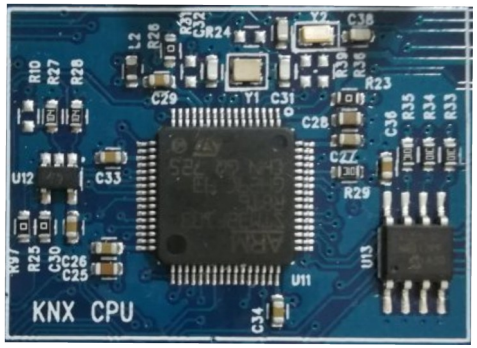
34 mm x 24 mm



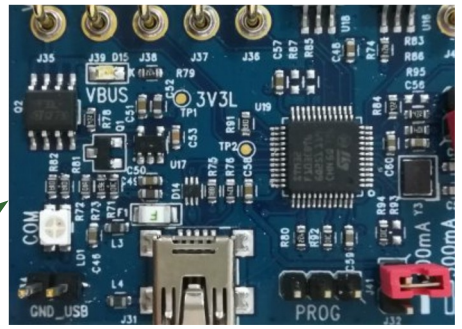
70 mm x 155 mm

KNX Prog key & LED

UNO &
Morpho connectivity



33 mm x 24 mm



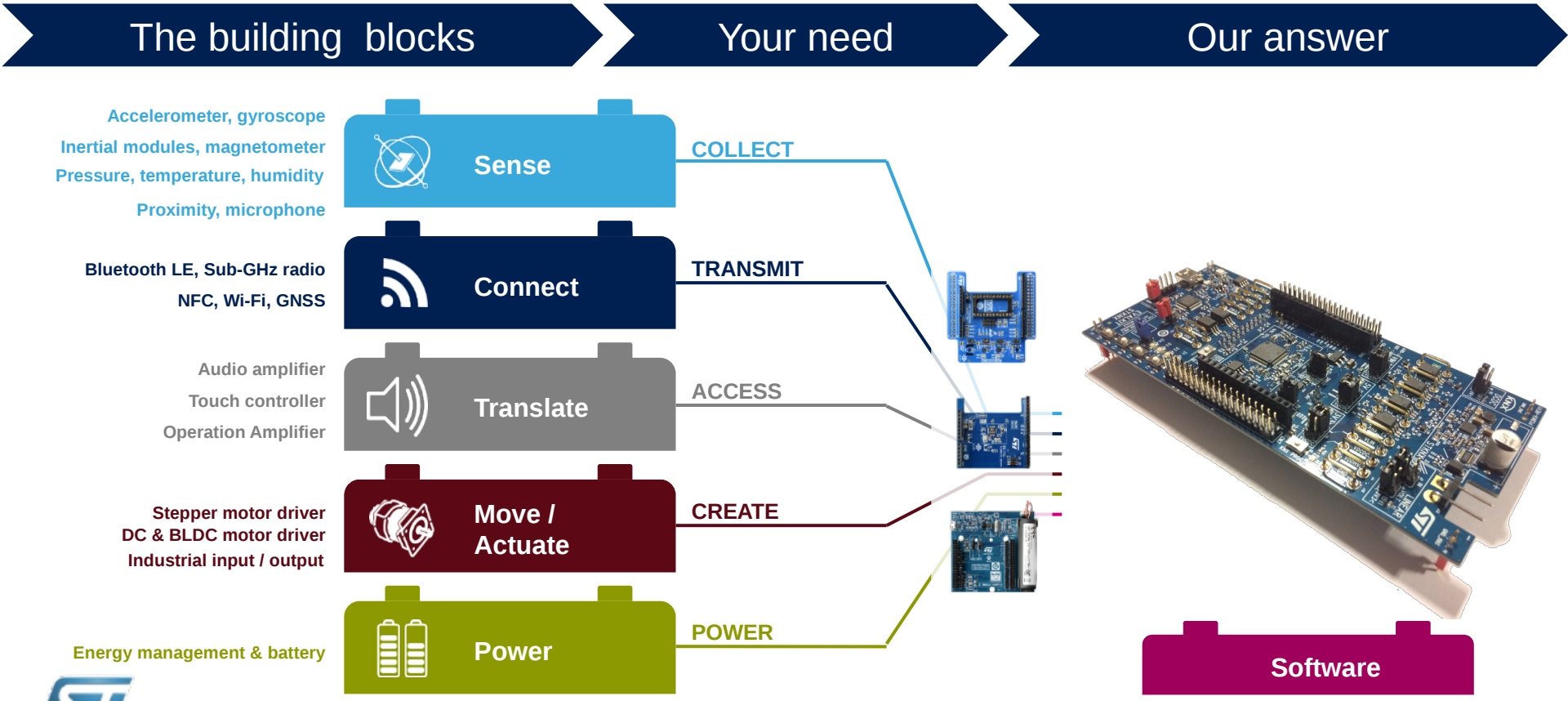
4 user LED & buttons

life.augmented

EVALKITSTKNX Development Platform overview

Development platform spirit

- **EVALKITSTKNX** has been developed in the spirit of the STM32 Nucleo boards.
- Expansion boards with additional functionality can be **plugged directly on top of the Eval Kit** development board or stacked on another expansion board.



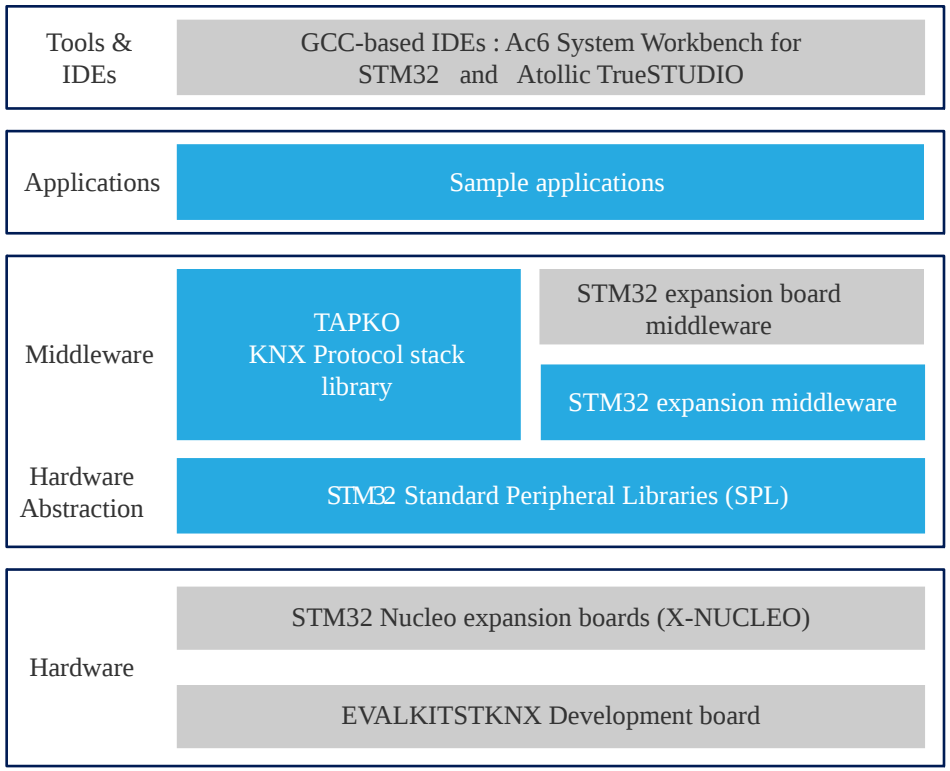
www.st.com/x-nucleo

Overview

1. EVALKITSTKNX Development Platform overview
2. Software overview and setup of the development environment
3. Setup example

Software Overview and Setup
Software components

- **EVALKITSTKNX** comes with a complete software package. A sample application is provided.
- Access to hardware features is simplified with the utilisation of the **STM32 Standard Peripheral Libraries** (SPL)
- A demonstration version of the **TAPKO's KNX Protocol stack** is provided as a binary file.
- The software package is compatible with two free GCC-based IDEs : **Ac6 System Workbench for STM32 and Atollic TrueSTUDIO**. Versions of thes two IDE exist for Linux and Windows OS.

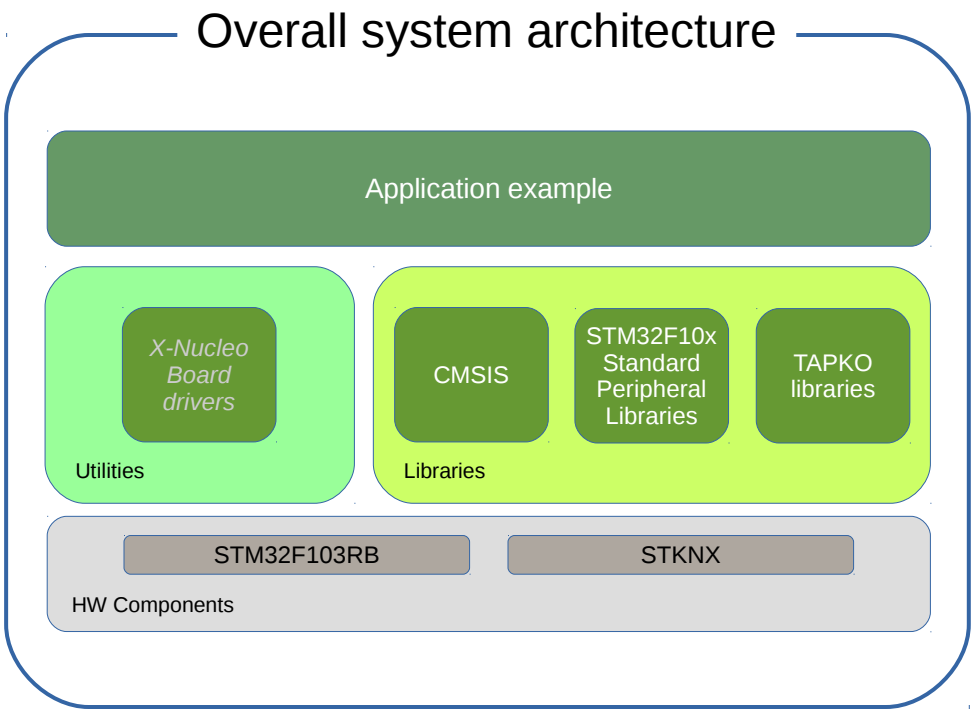


Software Overview and Setup

Software overview

STM32F103RB-STKNX software description

- This software, running on the STM32F103, demonstrate the STKNX capabilities.
- It is built on top of the STM32 Standard Peripheral Libraries (SPL) that eases access to STM32 features.
- It uses a demonstration version of the TAPKO KNX protocol stack.
- Example to demonstrate actuator and sensor.



Software Overview and Setup
Software prerequisites

- A Linux computer or a Windows computer with one of the supported development toolchains:
  - Atollic: TrueSTUDIO (Link)
  - AC6 System Workbench for STM32 : SW4STM32 (Link)

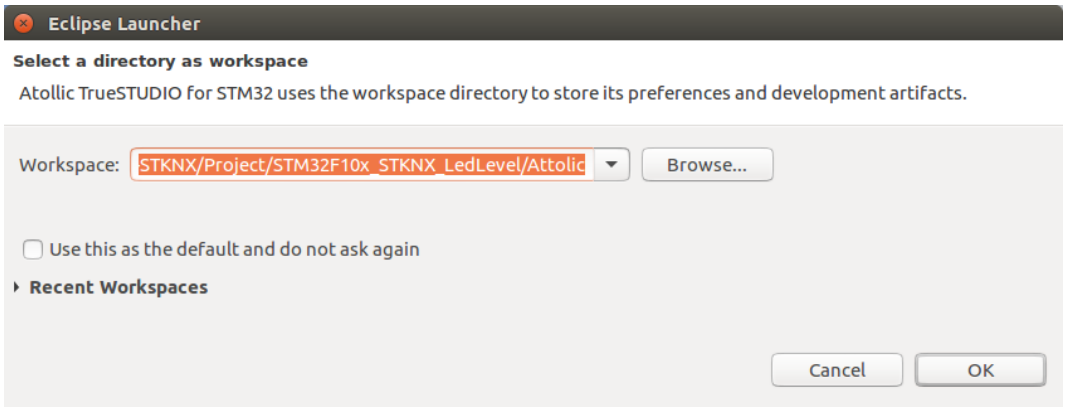
  or a Mac OSX computer with AC6 System Workbench for STM32
- EVALKITSTKNX firmware example
- ST-LINK/V2-1 USB driver (Link)
- ST-LINK/V2-1 firmware upgrade (Link)
- KNX ETS5 (Engineering Tool Software) Application (Link)

Software Overview and Setup

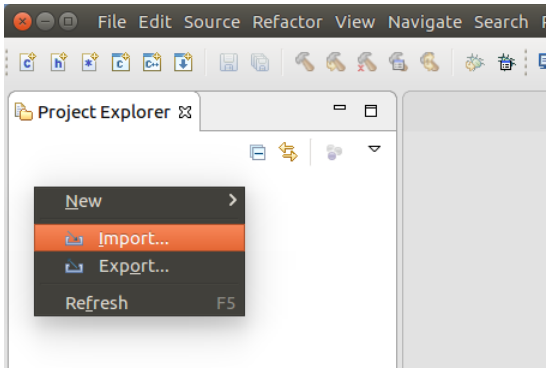
Install and build firmware with Atollic TrueSTUDIO

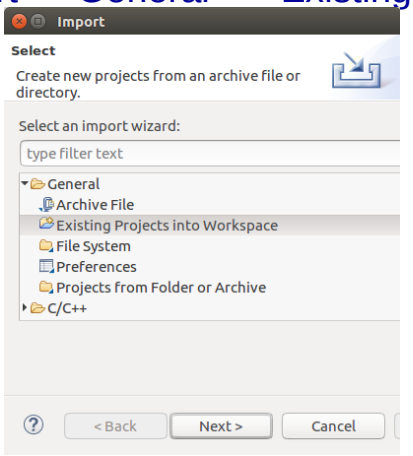
1. Download the STM32F103RB-STKNX package and extract it in your file system.

   Open Atollic TrueSTUDIO and when requested to select a directory as workspace, browse to : `<your path>/STM32F103RB-STKNX/Project/STM32F10x_STKNX_LedLevel/Attolic`



2. In the **ProjectExplorer** panel, right click and select Import → General → “Existing Projects into Workspace”

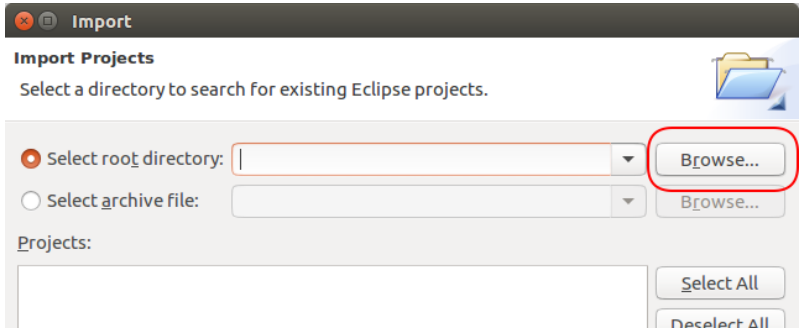


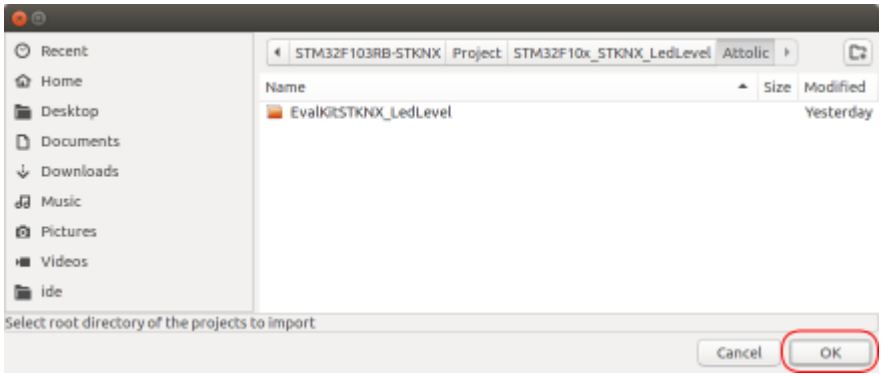


Software Overview and Setup

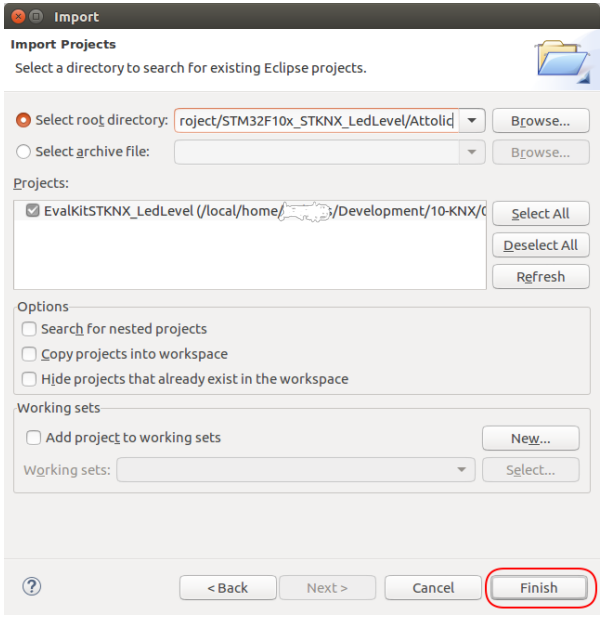
Install and build firmware with Atollic TrueSTUDIO

3 In the **Import** windows, click on Browse and on OK in the next window.





4 The project to import has been automatically selected. You can now click on Finish. The project is imported.



life.augmented

Software Overview and Setup

Install and build firmware with Atollic TrueSTUDIO

5 Select the project name in the **Project Explorer**, and click on Clean icon  or select Clean Project in the contextual menu.

6 Click on build icon  or select Clean Project in the contextual menu.

Problems Tasks Console Properties

CDT Build Console [EvalKitSTKNX_LedLevel]

```
arm-atollic-eabi-gcc -c ../../../../../Libraries/STM32F10x_StdPeriph_Driver/src/stm32f10x_rtc.c -mthumb -mcpu=cortex
arm-atollic-eabi-gcc -c ../../../../../Libraries/STM32F10x_StdPeriph_Driver/src/stm32f10x_spi.c -mthumb -mcpu=cortex
arm-atollic-eabi-gcc -c ../../../../../Libraries/Tapko/dev_mod/07B0/dat_07B0.c -mthumb -mcpu=cortex-m3 -std=gnu11 -D
arm-atollic-eabi-gcc -c ../../../../../Libraries/STM32F10x_StdPeriph_Driver/src/stm32f10x_pwr.c -mthumb -mcpu=cortex
arm-atollic-eabi-gcc -c ../../../../../Libraries/Tapko/dev_mod/07B0/app_07B0.c -mthumb -mcpu=cortex-m3 -std=gnu11 -D
arm-atollic-eabi-gcc -c ../../../../Src/demo_dimmer.c -mthumb -mcpu=cortex-m3 -std=gnu11 -DSTM32 -DSTM32F1 -DSTM32F103R
arm-atollic-eabi-gcc -c ../../../../../Libraries/STM32F10x_StdPeriph_Driver/src/stm32f10x_flash.c -mthumb -mcpu=cort
arm-atollic-eabi-gcc -c ../../../../../Libraries/Tapko/dev_mod/07B0/knx07B0_if.c -mthumb -mcpu=cortex-m3 -std=gnu11
arm-atollic-eabi-gcc -c ../../../../../Libraries/CMSIS/CM3/CoreSupport/core_cm3.c -mthumb -mcpu=cortex-m3 -std=gnu11
arm-atollic-eabi-gcc -c ../../../../../Libraries/STM32F10x_StdPeriph_Driver/src/stm32f10x_iwdg.c -mthumb -mcpu=corte
arm-atollic-eabi-gcc -c ../../../../../Libraries/STM32F10x_StdPeriph_Driver/src/stm32f10x_rcc.c -mthumb -mcpu=cortex
arm-atollic-eabi-gcc -c ../../../../../Libraries/STM32F10x_StdPeriph_Driver/src/stm32f10x_exti.c -mthumb -mcpu=corte
arm-atollic-eabi-gcc -c ../../../../../Libraries/STM32F10x_StdPeriph_Driver/src/stm32f10x_usart.c -mthumb -mcpu=cort
arm-atollic-eabi-gcc -c -mthumb -mcpu=cortex-m3 -g -Wa,--warn -x assembler-with-cpp -o startup/startup_stm32f10x_md.
arm-atollic-eabi-gcc -o EvalKitSTKNX_LedLevel.elf Examples/Src/app_data.o Examples/Src/demo_dimmer.o Examples/Src/ma
/opt/Atollic_TrueSTUDIO_for_STM32_x86_64_9.0.0/ide/jre/bin/java -jar /opt/Atollic_TrueSTUDIO_for_STM32_x86_64_9.0.0/
Generate build reports...
Print size information
   text    data     bss     dec     hex filename
  65100    1750    2432   69282   10ea2 EvalKitSTKNX_LedLevel.elf
Print size information done
Generate listing file
Output sent to: EvalKitSTKNX_LedLevel.list
Generate listing file done
Generate build reports done
arm-atollic-eabi-objcopy -O binary EvalKitSTKNX_LedLevel.elf EvalKitSTKNX_LedLevel.bin
 arm-atollic-eabi-size EvalKitSTKNX_LedLevel.elf

17:25:31 Build Finished (took 18s.320ms)
```

life.augmented

Software Overview and Setup

Install and build firmware with Atollic TrueSTUDIO

7 Once build is finished, you can start debugging the project : click on Clean icon  or select Debug As → Embedded C/C++ Application in the contextual menu.

Or you can drag and drop the binary (*EvalKitSTKNX_LedLevel.bin*), available in **Debug,** to the virtual drive that is mounted when you connect the EVALKITSTKNX board to your computer. Name of this virtual drive is STKNX.

Overview

1 EVALKITSTKNX Development Platform overview

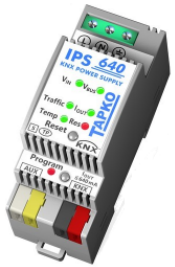
2 Software overview and setup of the development environment

3 Setup example

Setup & demo examples
Hardware prerequisites

- One EvalKitSTKNX board
- One KNX sensor : Apricum TAI-KNX 4

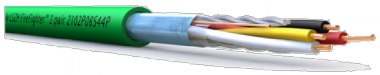
- One KNX Power Supply



- One KNX TP interface (USB-KNX interface)

- A type A to type B USB cable

- KNX cable



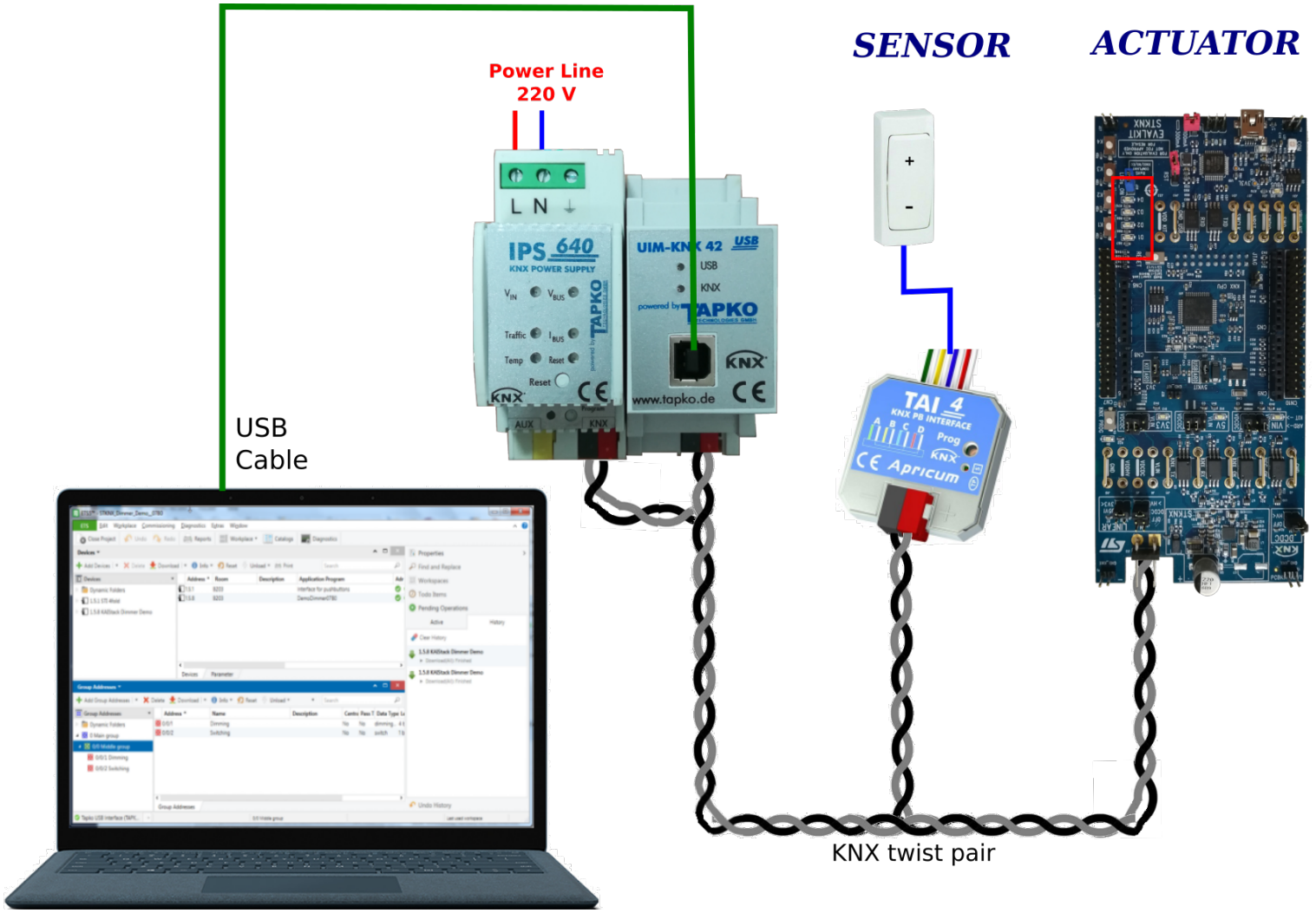
(*) Power stage supply voltage from 8 V to 50 V

life.augmented

Setup & demo examples

Setup a simple KNX network

1 Setup the KNX network according as described by the picture

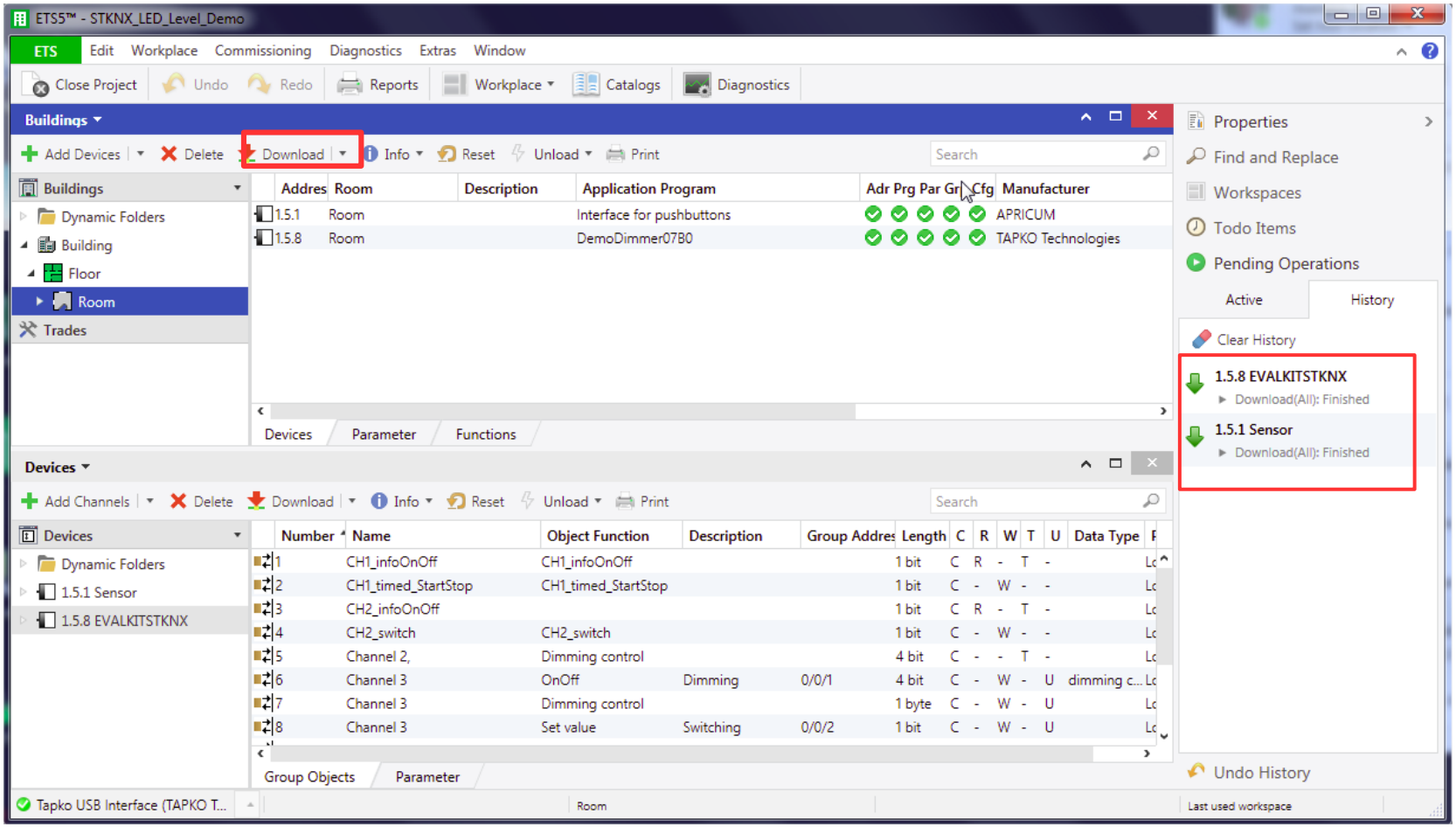


Setup & demo examples

Setup a simple KNX network

2 In the ETS5 application, import the project STKNX_LED_Level_Demo.knxproj available in STM32F103RB-STKNX/Project/STM32F10x_STKNX_LedLevel/ETS5_ProjectFile

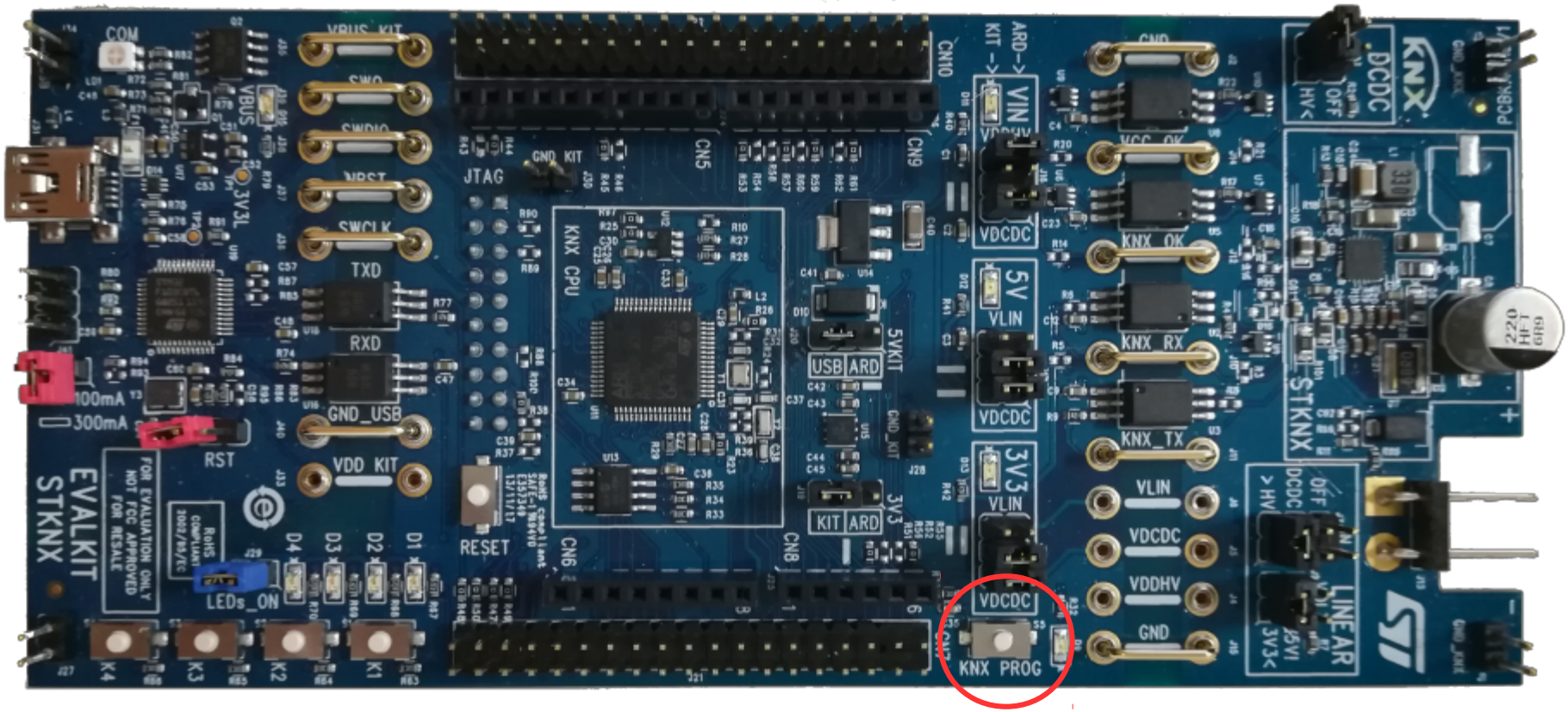
3 Download the full configuration in both STI and EVALKITSTKNX devices



Setup & demo examples

Setup a simple KNX network

4 When asked in the ETS5 application, press the “KNX PROG” button of the EVALKITSTKNX board.

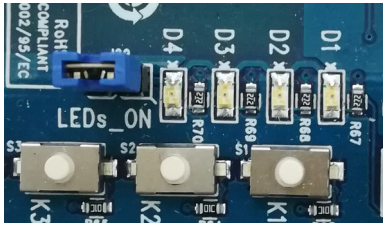


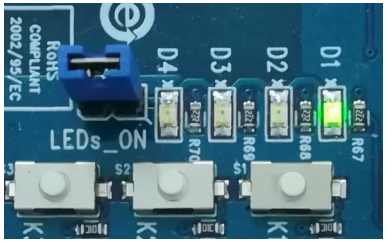
Setup & demo examples

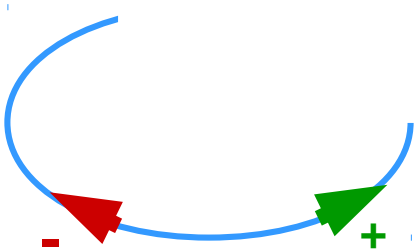
Setup a simple KNX network

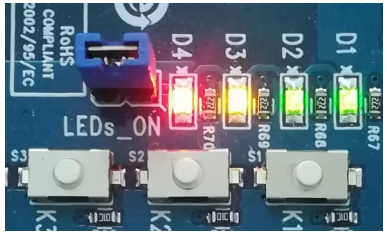
5 Tests :

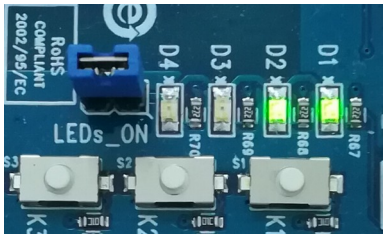
- ON button of the sensor is used to increase the number of led switched on
- OFF button of the sensor is used to decrease the number of led switched on

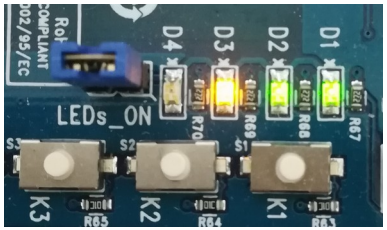




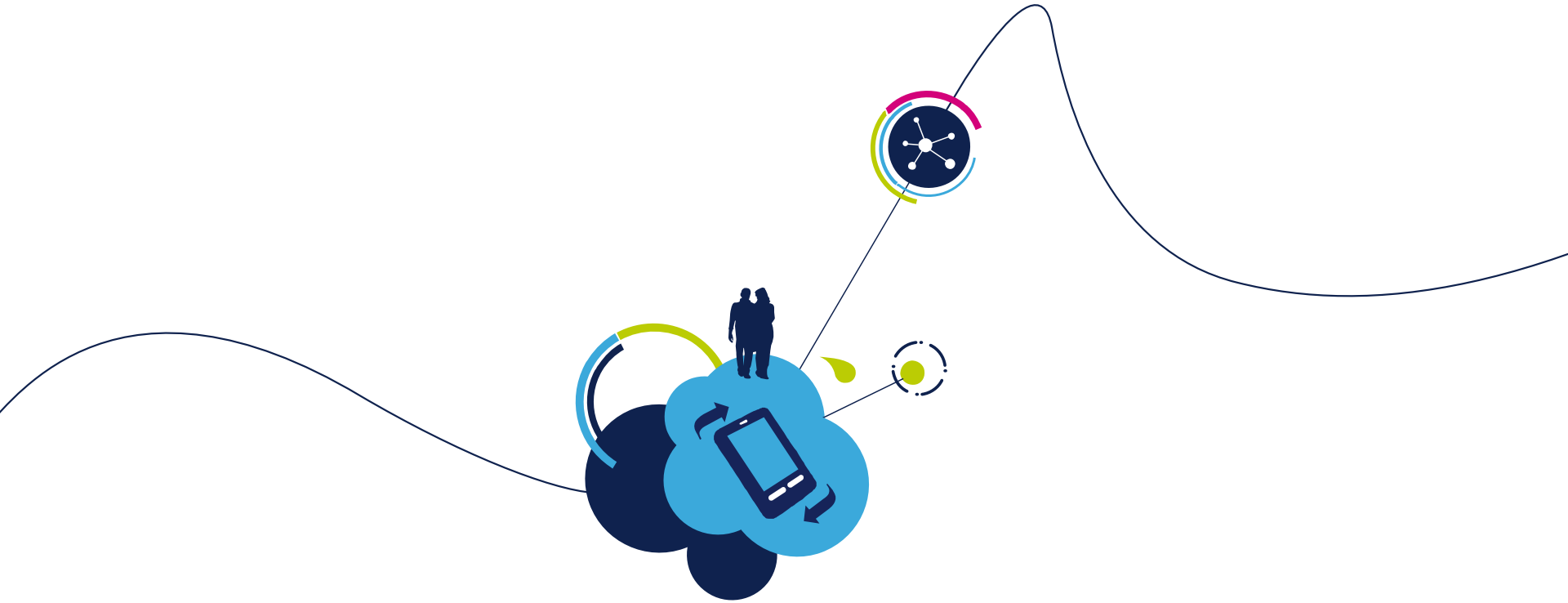








life.augmented



www.st.com