Sigfox_Silica Documentation

Release 0

Silica

Apr 04, 2017

Contents

Version 1.11C

Date 6 Jul 2016

Reference name BAEVTSS003

CHAPTER 1

Sigfox's Long Range Low Power End Node solution


3.6V
75592
TELECOM
DESIGN

CHAPTER 2

INTRODUCTION

Telecom Design's **TD1207** devices are high performance, low current SIGFOXTM gateways. The combination of a powerful radio transceiver and a state-of-the-art ARM Cortex M3 baseband processor achieves extremely high performance while maintaining ultra-low active and standby current consumption. The **TD1207** device offers an outstanding RF sensitivity of – 126 dBm while providing an exceptional output power of up to +14 dBm with unmatched TX efficiency. The **TD1207** device versatility provides the gateway function from a local Narrow Band ISM network to the long-distance Ultra Narrow Band SIGFOXTM network with no additional cost. The broad range of analog and digital interfaces available in the **TD1207** module allows any application to interconnect easily with the SIGFOXTM network. The LVTTL low- energy UART, along with the numerous GPIOs can control any kind of external sensors or activators. Featuring an AES encryption engine and a DMA controller, the powerful 32-bit ARM Cortex-M3 baseband processor can implement highly complex and secure protocols in an efficient environmental and very low consumption way.

Features:

- SIGFOX READY
- Frequency range = ISM 868 MHz
- Receive sensitivity =-126 dBm
- **Modulation:**
  - (G)FSK, 4(G)FSK, GMSK,
  - OOK
- Max output power: +14 dBm
- **Low active radio power consumption:**
  - 13/16 mA RX,
  - 37 mA TX @ +10 dBm
- Power supply = 2.3 to 3.3 V

Developement tools

Firmware developed using: NXP Kinetis Design Studio. For installation and configuration of the project, follow instruction inside *Developing guide*

Document references

The board reference documentation is available on the architech-board website.

Contents:

Developing guide

This guide will provide instructions to install the development environment needed to compile and debug the demo firmware of the Sensor Node Sigfox. The development system is multiplatform, it supports Windows and Linux. This guide is written using Windows. The main steps are:

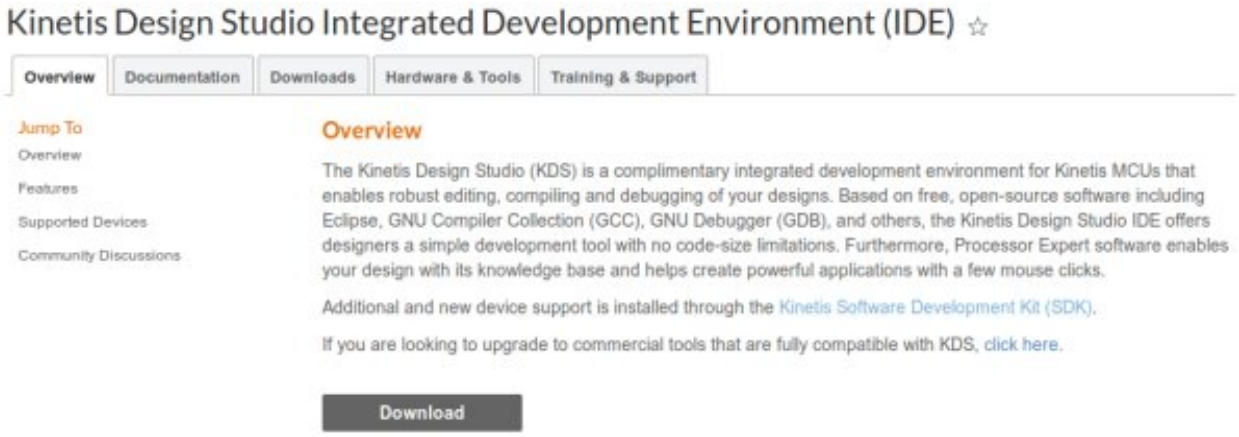
- Install Kinetis Design Studio 3.2.0, upgrade it and install KSDK 1.3.0
- Import build & debug the source project

Hardware required:

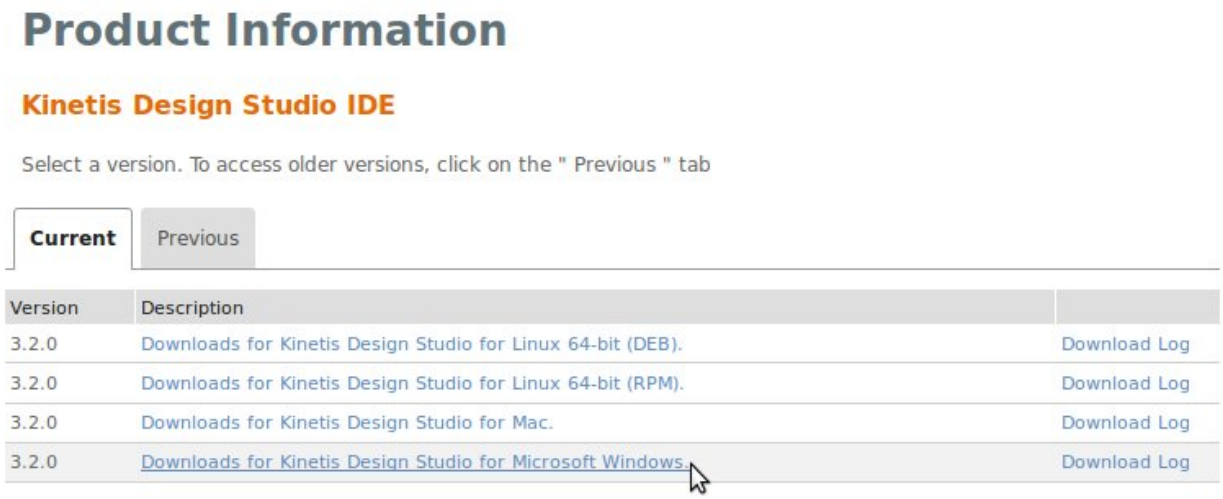
- Sensor Node Sigfox with battery or Mini-USB cable
- if you don't use Mini-USB cable Segger it is required a J-Link (Segger website) with SWD debug interface.
- PC with Windows or Linux

Install & Update Kinetis Design Studio

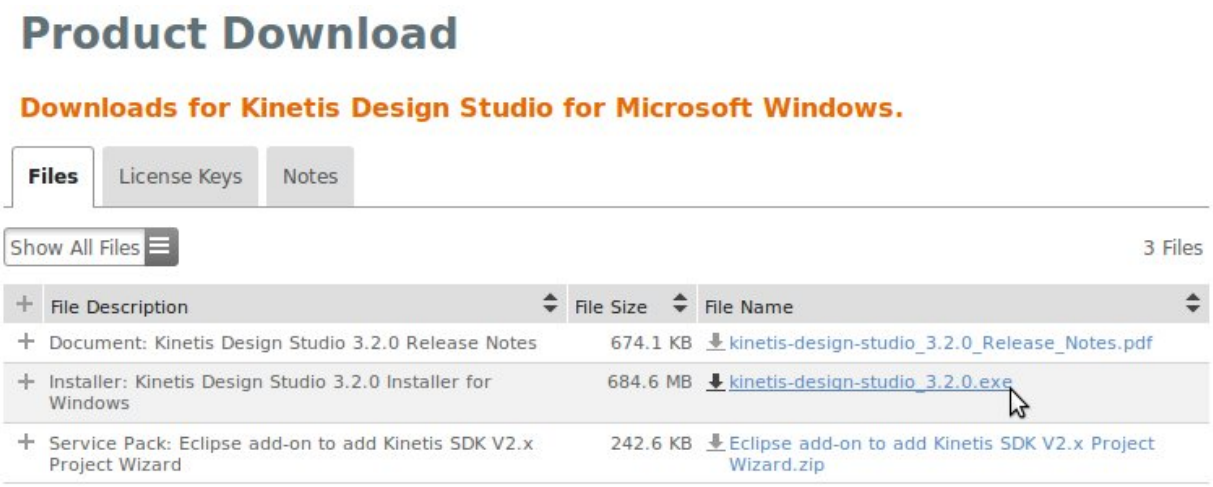
First up, register at the NXP website registration form then you can download the IDE from this page. Press on **Download** button.



We used Kinetis Design Studio **3.2.0**, press on **Downloads for Kinetis Design Studio for Microsoft Windows**.

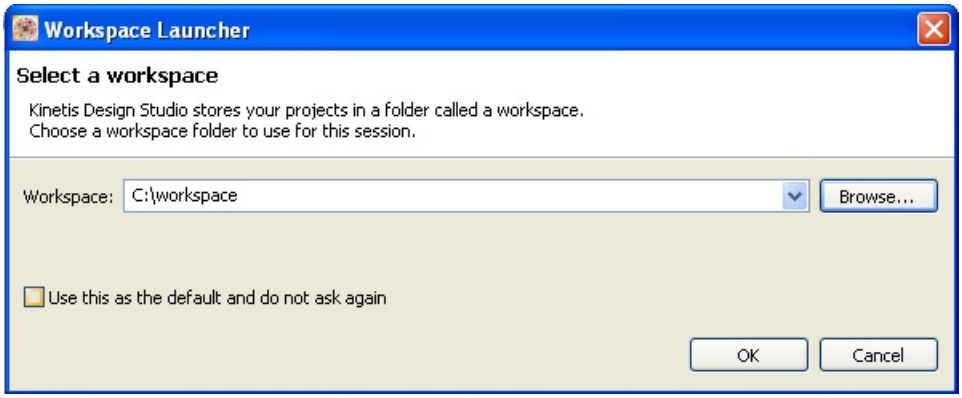


Agree the terms and download the file **Installer: Kinetis Design Studio 3.2.0 Installer for Windows**.

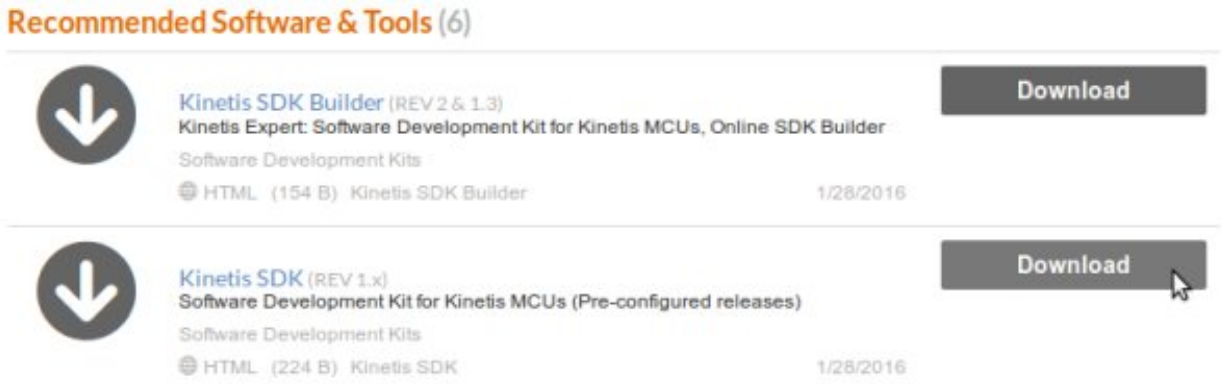


Next, run the downloaded file **kinetis-design-studio_3.2.0.exe** following all the default options.

Launch KDS and select a directory for the workspace. Our project will be imported in this folder. In this guide we used this path:

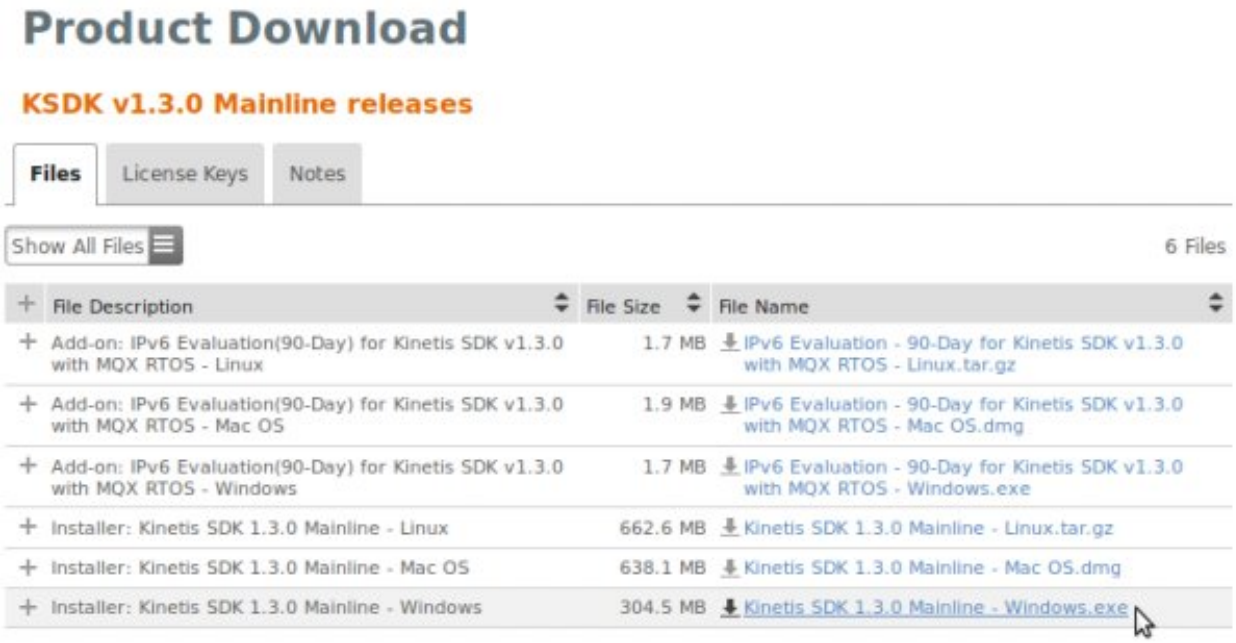


At this point, if you want to modify the project using **processor expert** it is required to install also the package **KSDK 1.3.0** in order to be compatible with the project. You can find this package on KINETIS-SDK page. Select Download button from **Kinetis SDK**.



Select **KSDK v1.3.0 Mainline releases**

Agree the terms and then download **Kinetis SDK 1.3.0 Mainline - Windows.exe**



Install it following all the default options, it will be installed into **C:\Freescale\KSDK_1.3.0**

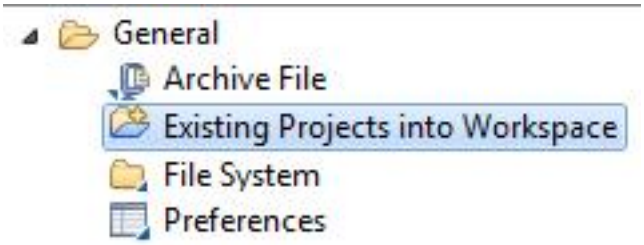
Follow these steps:

1. Run KDS 3.2.0
2. Select Help -> Install New Software
3. Click on **Add...** button
4. Then click on **Archive** button
5. Now select from **C:\Freescale\KSDK_1.3.0\tools\eclipse_update** the file **KSDK_1.3.0_Eclipse_Update**
6. Select the package **KSDK 1.3.0 Eclipse Update**
7. Continue with the wizard. Accept the license agreement during the installation process.
8. Restart KDS

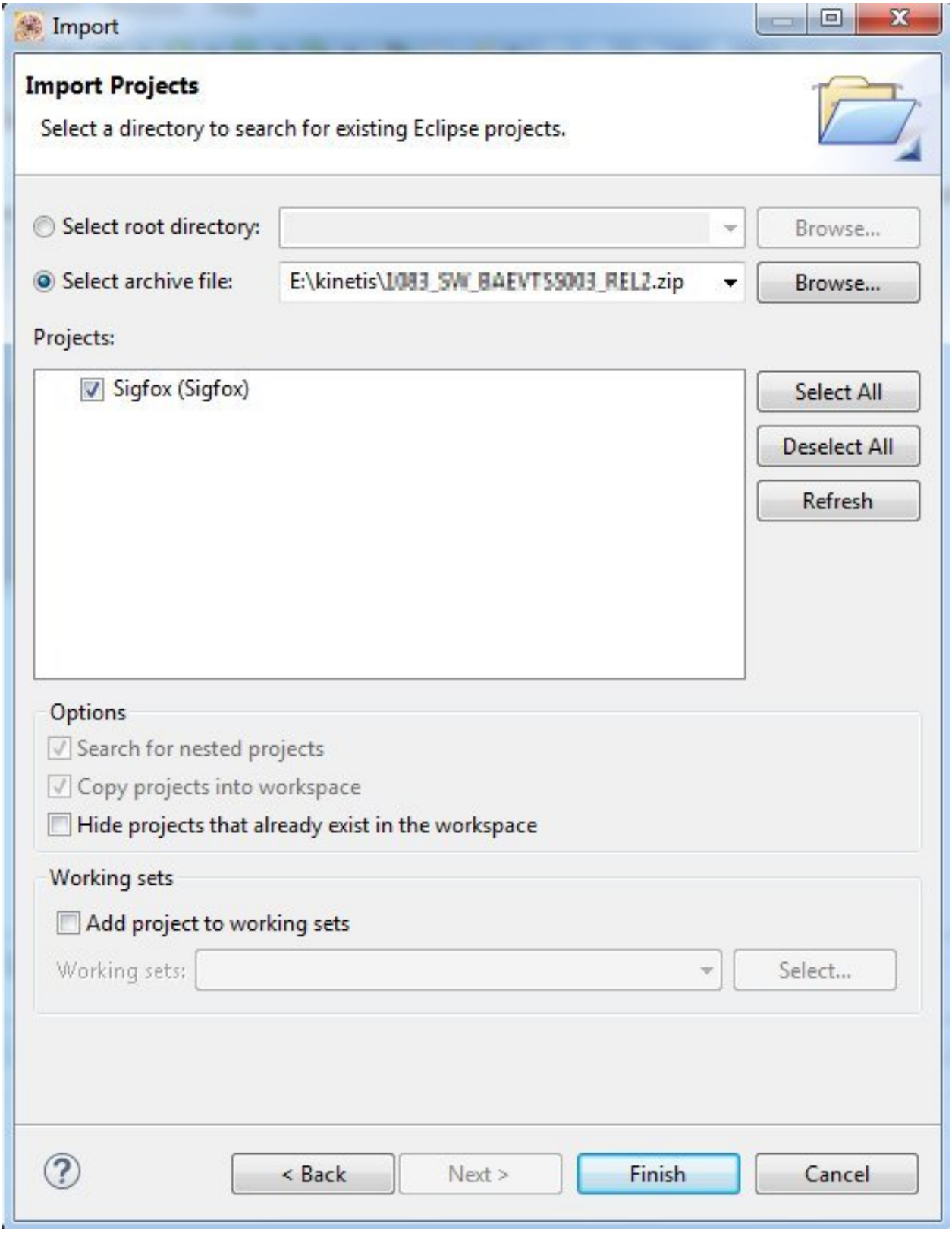
Now you are ready to import the project in your KDS.

Import Project

Download the project 1083_SW_BAEVTSS003_REL2.zip. Then go to **File->Import** and select **Existing Projects into Workspace**.



Browse to the zip file containing the project and select the project.



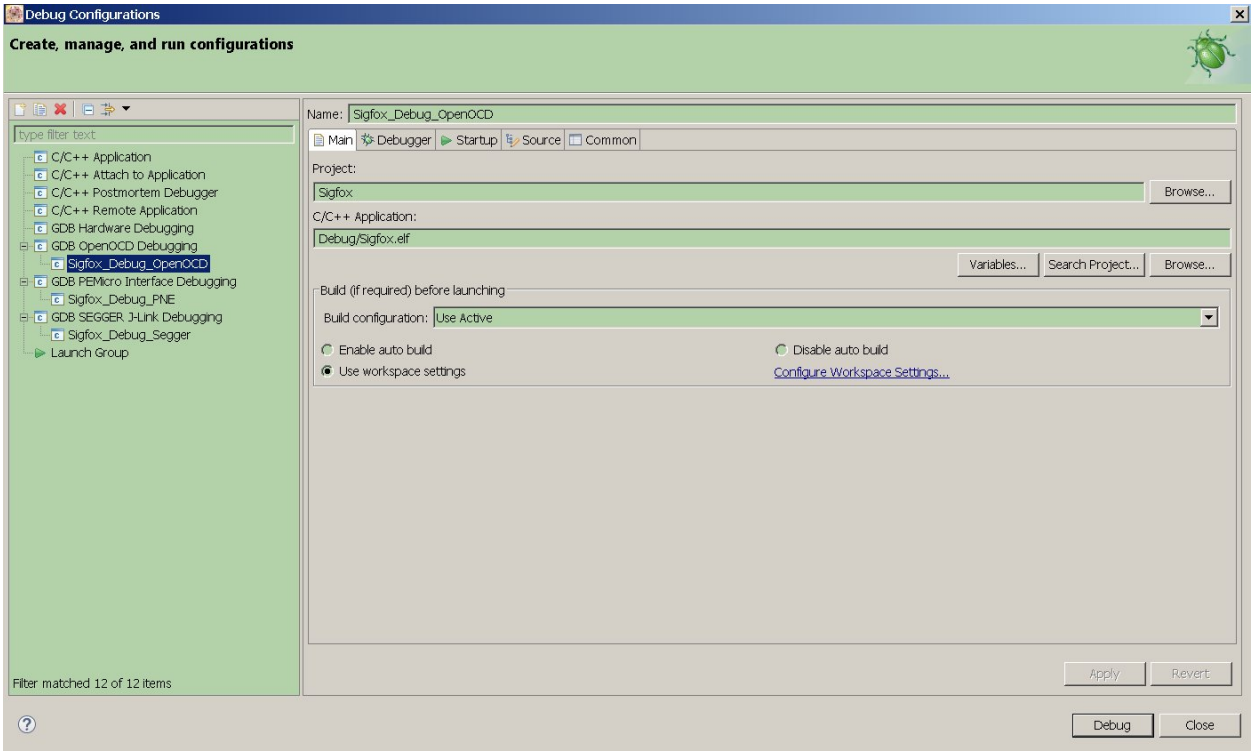
Press on **Finish**. Now you are ready to build and debug it. There are two ways to debug the board, the easiest is debug it via USB.

Build & Debug via USB

With the Mini-USB connector you are able to use **OpenOCD** inteface. In order to use it download and install the drivers for windows from mbed website,

After the installation, connect the sigfox-node board to the PC via **CN2** connector and turn on the board switching the **SW1**. Windows will recognize it.

Go to **Project->Build All**, to compile the entire project. Always on the KDS click on **Run->Debug Confiuration->Sigfox_Debug_OpenOCD**.



Clicking on **Debug** button the debug will start entering on the first line code of the **main()** function. During the debug session the sleeping mode doesn't work.

> **Warning:** Due some problem with KDS, sometimes you have to launch the debug twice before the KDS will run the debug correctly.

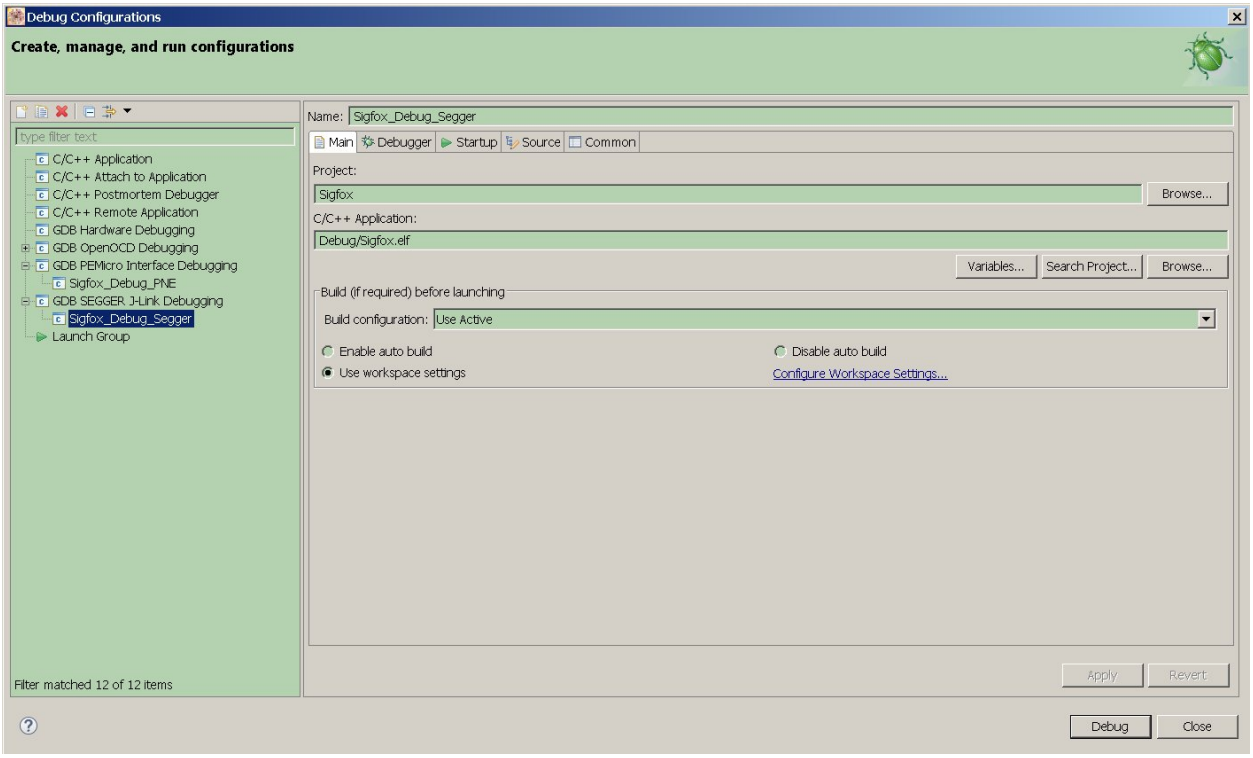
Build & Debug via SWD

Go to **Project->Build All**, to compile the entire project. In order to debug it connect the J-Link to the connector **CN4**.

Note: Connector **CN4** is not fitted on the sigfox-node board. It is a connector **SMD MALE STRIP 2x5 P1.27mm**. You have to fit it in order to debug the board via SWD.

The used debug interface is **SWD**. Then turn on the board switching the **SW1**.

Always on the KDS click on **Run->Debug Confiuration->GDB Segger J-Link Debug**.



Clicking on **Debug** button the debug will start entering on the first line code of the **main()** function. During the debug session the sleeping mode doesn't work.

Processor Expert

The **Kinetis software development kit (KSDK)** is an extensive suite of robust hardware interface and hardware abstraction layers, peripheral drivers, RTOS abstractions, stacks, and middleware designed to simplify and accelerate application development on Freescale Kinetis MCUs. The addition of **Processor Expert** technology for software and board configuration provides unmatched ease of use and flexibility. Included in the Kinetis SDK is full source code under a permissive open-source license for all hardware abstraction and peripheral driver software. Mainline releases include support for a collection of Kinetis MCUs, whereas standalone releases offer support for one or a few additional Kinetis MCUs only. For details read KSDK page.

Hardware Guide

The board is provided with:

- NXP MKL26Z microprocessor
- NXP MK20DX128VFM5 microprocessor
- TD1207 Module Sigfox
- Light Sensor
- Reset button
- Wake-up button
- NXP FXOS8700CQR1 Accelerometer and Magnetometer sensor
- LEDs which one is used directly by KL26Z

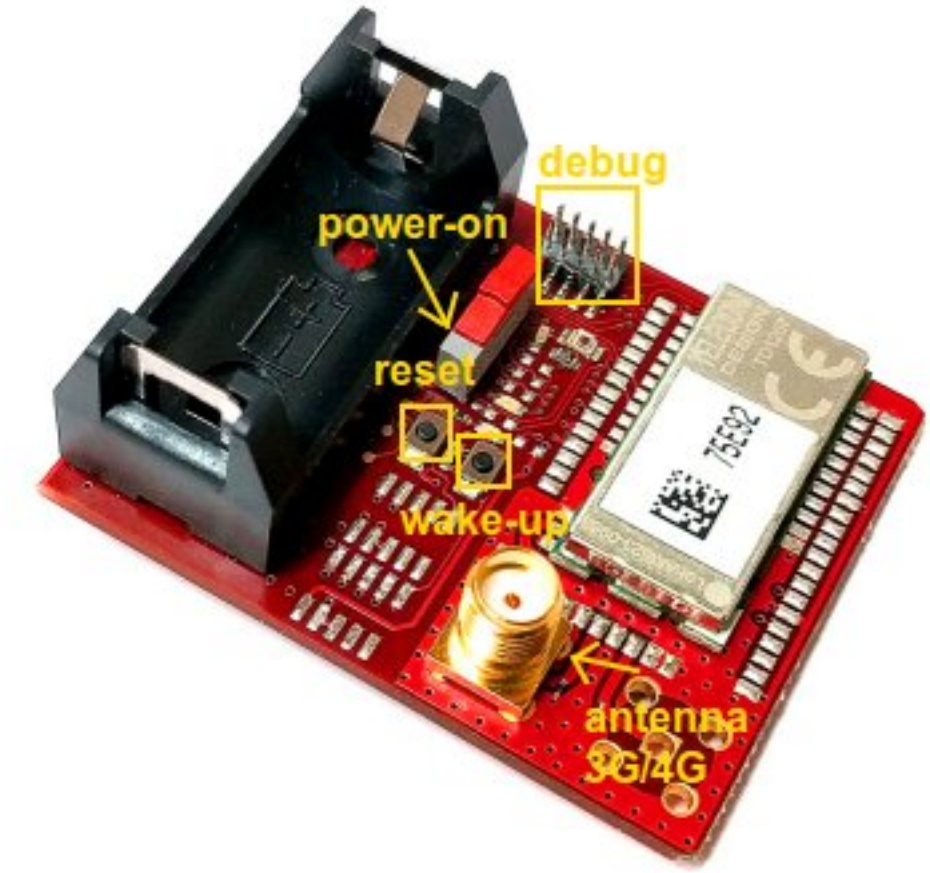
The TD1207 module provides a protocol connectivity using a simple UART interface. The NXP **MKLS26Z** is connected to the TD1207 module using the configuration 9600 8N1 without using RTS, CTS lines.

The Light Sensor is read from the ADC converter pheriperal of the MKL26Z.

The Accelerometer sensor is read from I2C interface.

The microcontroller uses the deep sleep mode **VLPS**, it is waken up by LPTimer every 30 seconds or by pin interrupt connected to the **S2** button.

The board



Commands used

The comand to send data in the Sigfox network is:

- **AT$SS=18 AA BB CC**: used to send the frame *18AABBCC*

Datasheet and more

Please refer to architechboards website.

Index