# G4XS_KL25Z_XSG_Measurement

Processor Expert Example Project

**Date:** *21. 11. 2016*

***Revision:*** 2.0

## Requirements

Kinetis Design Studio 3.2.0 or newer

FRDM-KL25Z MCU board

FRDM-32XSG-EVB Gen4 eXtreme Switch board

channel load

USB cable

## Overview

The purpose of this example project is to show how to configure feedback monitoring and how to get measured values. In this example monitored output is constantly changed and measured after synchronization occurs.

## Setting up hardware

Target platform for this example is FRDM-KL25Z MCU board with FRDM-32XSG-EVB eSwitch board.

FRDM eSwitch board has to be properly configured in such way that CSB0 pin is correctly routed to PTD0 on MCU. This can be achieved by switching number 0 on SW2 switch on the eSwitch board to ON position. The rest of CSB routes should remain in OFF position.

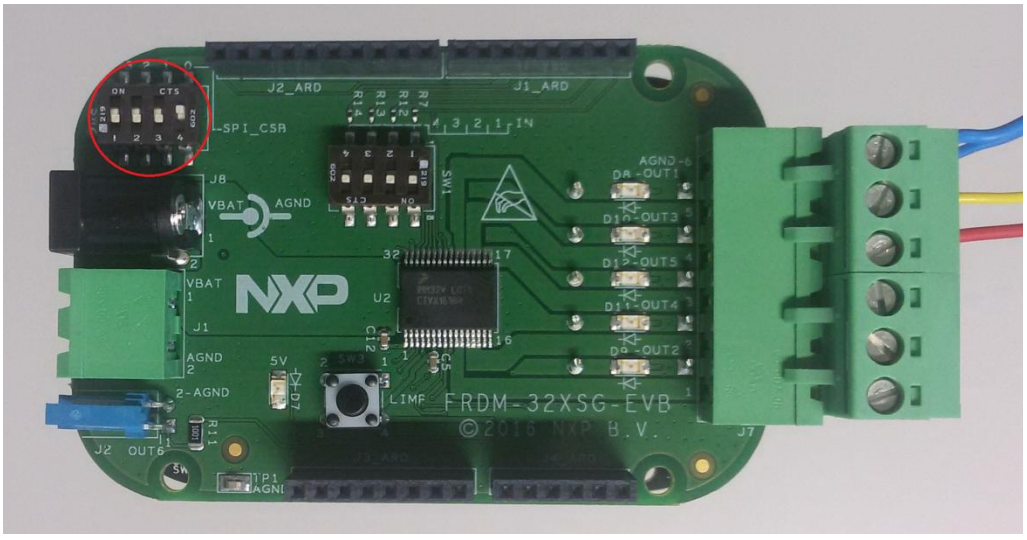


**Figure 1 CSB Route Selection**

In order to get the project successfully running you need to connect some load to channel 1 on your FRDM eSwitch board.

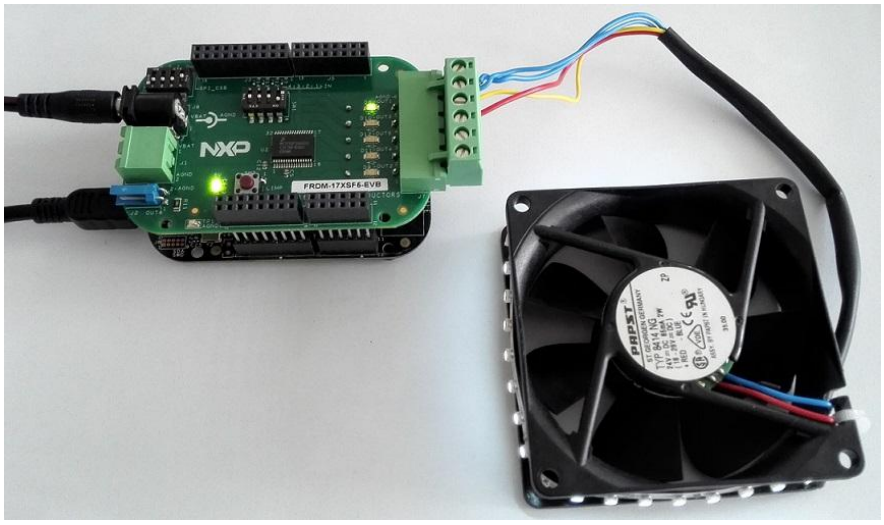


**Figure 2 Channel Load**

Set up connection between PC and MCU board. Connect USB cable to OpenSDA port on MCU side.

**Figure 3 OpenSDA PC-MCU Debug Connection**

## Setting up software

Make sure you have installed KDS 3.2.0 or newer which comes with Processor Expert support.

Then components must be imported to the Processor Expert Component Library.

- **SPI_Device** component

- **Gen4eXtremeSwitch** component

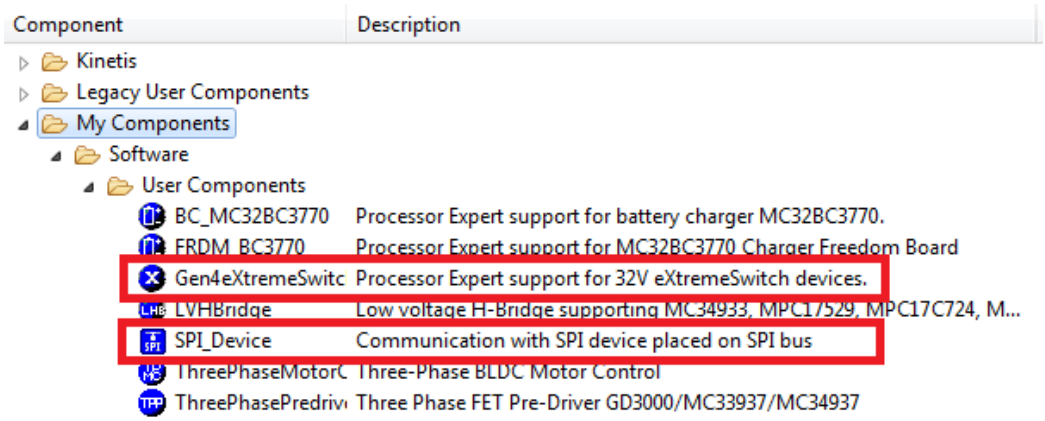


**Figure 4 Processor Expert Component Library**

Check that your OpenSDA connection has properly installed driver.

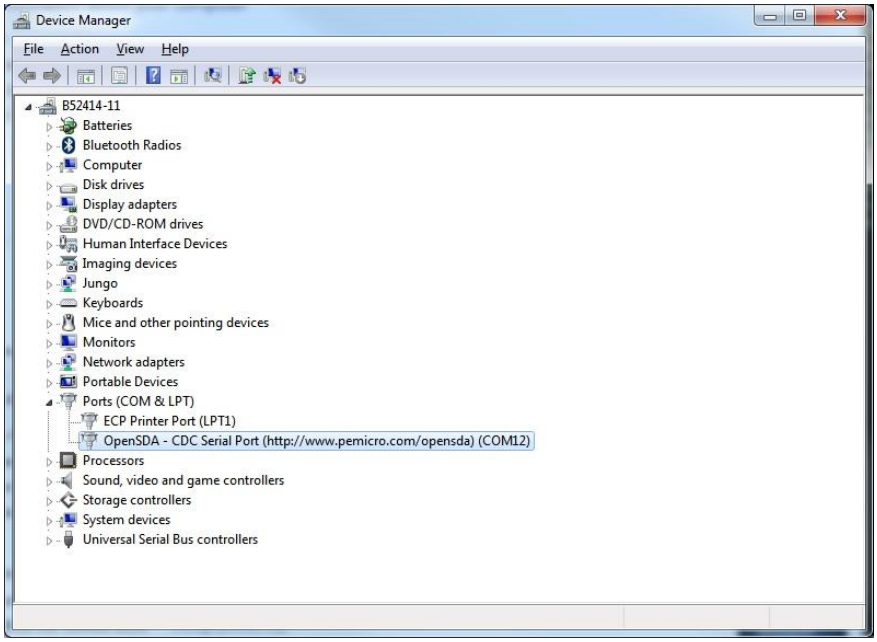


**Figure 5 OpenSDA Virtual Port**

## Description

The example is preconfigured in the following way. ESwitch board is configured in **Gen4eXtremeSwitch** component, see Figure 7. SPI settings are configured separately in **SPIMaster_LDD** component, see Figure 8 and in inherited **CSpin1** component, see Figure 9. All components can be accessed in component tree, see Figure 6.

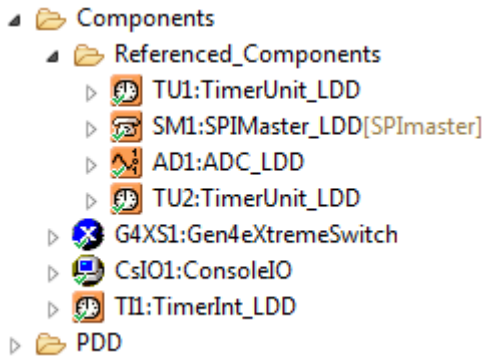


**Figure 6 Component Tree**

CSNS Function is set to Channel 1 current measurement, but will be changed in the code. OCLO Threshold is set to Low and Advanced Sensing Mode is enabled for more accurate measurement of low currents (25 % of Full Scale Range = Maximum Current), for higher currents set these properties to High and Disabled (100% FSR). Pay attention to Watchdog Timeout value, make sure that if you modify project, you still hold the period needed to feed watchdog.

| | | |
|---|---|---|
| Component Name | G4XS1 | |
| SPI Configuration | Parallel SPI | |
| **Global Configuration** | | |
| RSTB Pin | PTC12/TPM_CLKIN0 | |
| External Clock Frequency | 50000 D | |
| CLK Pin | PTC3/LLWU_P7/UART1_RX/TPM0_... | PTC3/LLWU_P7/UART1_RX/TPM0_... |
| Watchdog Timeout | 128 ms | |
| **Direct Input Control** | Disabled | |
| **Current, Voltage and Tempe** | Enabled | |
| ADC Conversion Time | 4.169068 µs | 4.169 µs |
| CSNS Function | Channel 1 | |
| CSNS Pin | ADC0_SE12/TSI0_CH7/PTB2/I2C0_... | |
| **CSNS Synchronization** | Trigger 1/2 | |
| CSNS SYNC Pin | PTD4/LLWU_P14/SPI1_PCS0/UART... | |
| **Devices** | 1 | |
| **Device1** | | |
| Device Model | MC17XSF500 | |
| SOA Mode | Single read | |
| Overtemperature Warning | 115 °C | |
| HID Selection | Disabled | |
| OCHI Type | Default | |
| Global PWM Duty Cycle | 0 D | |
| **Channels** | 6 | |
| **Output1** | Enabled | Corresponds to channel with index... |
| **PWM Output Cont** | | |
| Global PWM | Disabled | |
| Channel Duty C | 0 D | |
| Phase Selectior | 0° | |
| Pulse Skipping | Disabled | |
| Slew Rate Presc | 1 | |
| Output Initial S | Off | |
| **Direct Input Contr** | Disabled | |
| **Open Load** | | |
| Open Load LED | Disabled | |
| OLON Deglitch | 64 us | |
| **Overcurrent** | | |
| OCLO Threshol | Low | |
| Advanced Curr | Enabled | |
| Short OCHI | Disabled | |
| No OCHI | Disabled | |
| **Output2** | Disabled | Channel disabled. |
| **Output3** | Disabled | Channel disabled. |
| **Output4** | Disabled | Channel disabled. |
| **Output5** | Disabled | Channel disabled. |
| **Output6** | Disabled | Channel disabled. |
| Auto Initialization | yes | |

**Figure 7 G4XS Component Settings**

SPI setting involves pin selection (MISO, MOSI, CLK and CSB). The first three pins are configured in SPIMaster_LDD, see Figure 8 and the last one is configured in CSPin component, see Figure 9, which is inherited by SPI_Device component.

| | |
|---|---|
| Device | SPI0 |
| **Interrupt service/event** | Enabled |
| **Settings** | |
| **Input pin** | Enabled |
| Pin | PTD3/SPI0_MISO/UART2_TX/TPM0_CH3/SPI0_MOSI |
| **Output pin** | Enabled |
| Pin | PTD2/SPI0_MOSI/UART2_RX/TPM0_CH2/SPI0_MISO |
| **Clock pin** | |
| Pin | ADC0_SE5b/PTD1/SPI0_SCK/TPM0_CH1 |
| Chip select list | 0 |
| **Attribute set list** | 1 |
| **Attribute set 0** | |
| Width | 8 bits |
| MSB first | yes |
| Clock polarity | Low |
| Clock phase | Change on leading edge |
| Parity | None |
| Chip select toggling | no |
| Clock rate index | 0 |
| Clock rate | 1.497966 MHz |
| **Initialization** | |
| Auto initialization | yes |

**Figure 8 SPI Master Component Settings**

| | |
|---|---|
| Pin for I/O | PTD0/SPI0_PCS0/TPM0_CH0 |
| Direction | Output |
| **Initialization** | |
| Init. direction | Output |
| Init. value | 1 |
| Auto initialization | yes |

**Figure 9 Chip Select Pin Settings**

This demo project uses output 1 which is enabled in the Gen4eXtremeSwitch component. It is important to mention that index of this channel is 0. Please see the Detail column in Component Inspector to know indexes of available channels.

In main function *SetPWMduty* method is called to set 88% duty cycle for output 1. After this method a *FeedWatchdog* method is periodically called in an infinite loop. Note that it is necessary to periodically feed the watchdog otherwise the device would get in fail-safe mode. In the infinite loop also the *IsInterrupt* flag is checked which is set by timer every 2 seconds (interrupt service routines are located in *Events.c* module). If the flag equals *TRUE*, measurement is started and the result is sent to console. Also with every timer interrupt the measured quantity is changed. At first the current on channel 1 is measured, then the supply voltage and after that the die temperature. These values are sent to console in this order.

Two methods for measurement setting and reading data are provided by Gen4eXtremeSwitch component. The first one is *ConfigureMonitoring* which selects device, sets a quantity to be

measured and a type of trigger. Method *GetSenseValue* starts ADC measurement and returns measured data. If the device is set for current measurement it is important to set some of available triggers to get precise data. The trigger sets an interrupt flag if the measurement is synchronized with PWM signal. This flag is checked in an infinite loop in *StartMeasurement* method and if it equals *TRUE*, measurement is started. The method *GetSenseValue* returns after the measurement has finished.

## Import the example project

Pre-requisites such as needed components are assumed to be imported already. If that is correct, you can approach to importing of the example project.

1. In KDS click on the File / Import.
2. Choose General / Existing Projects into Workspace.
3. Click Browse to select root directory with your downloaded example projects.
4. Select project named G4XS_KL25Z_XSG_Measurement and click Finish to complete process.
5. Now the example project should be copied to your workspace and ready to run.

## Build and Debug

In order to build and run the project you need to generate code at first. Then you can build the project usual way. If the build is successful, debug and run the project.