# Introduction

The bachelor’s Thesis is realised in collaboration with HES-SO Valais/Wallis, Nordic Semiconductor and NTNU. All the project is developed at Nordic Semiconductor in Trondheim, Norway.

Nordic Semiconductor is specialized in the development of SoC and provide a large range of tool to develop application using Bluetooth. Therefore, The Bluetooth Low Energy, BLE, is the main element of this project.

The main idea of this project is to measure the behaviour of Zephyr RTOS with nRF5x series SoC from Nordic Semiconductor. Zephyr is a recent RTOS developed for IoT application and provides a Bluetooth API.

Generally, the Nordic’s customers use the SoC with Bare Metal system and using libraries, SoftDevices and Software development kit, developed by Nordic Semiconductor. But, Nordic Semiconductor is interested by the possibility of using a RTOS with its products.

However, it is important for Nordic Semiconductor that the requirements to use Zephyr RTOS and the Bluetooth Low Energy with high traffic are respected and provide an advantage compared to a Bare Metal system.

# Why Zephyr RTOS?

The Zephyr™ Project of the Linux Foundation is a scalable real-time operating system (RTOS) supporting multiple hardware architectures, optimized for resource constrained devices, and built with security in mind. It is designed from the ground up to be a modern, scalable, secure and responsive RTOS with a clear focus on IoT protocols and foundations.



Figure 2: Zephyr RTOS logo

## Philosophy

Zephyr is Apache 2.0 licensed and has a fully open development model. The terms and conditions of this license are mainly:

* all copies, modified or unmodified, are accompanied by a copy of the licence
* all modifications are clearly marked as being the work of the modifier
* all notices of copyright, trademark and patent rights are reproduced accurately in distributed copies
* the licensee does not use any trademarks that belong to the licensor

Hence, it is possible to anyone to access to the source code and to modify it as he wants regards to the license’s conditions.

In addition of the Apache license, Zephyr Project source is maintained on a public GitHub Repository. GitHub allow anyone to be a contributor of the project by notifying bugs or by proposing improvements.

To contribute to the project, a programmer pulls a request for a bug or improvement that will be analysed by one or several main developers of the project. If the request is approved, the code of the request is merge with the source code of the project.

## Benefits for Nordic Semiconductor

Nordic is currently member of the Zephyr RTOS. Therefore, it is already possible to use Zephyr RTOS with nRF5x series SoC. However, it is already possible to use different RTOS as FreeRTOS or Keil with Nordic’s produces. So, why is it interesting for Nordic Semiconductor?

At first, Zephyr RTOS provides all the tools to develop a IoT application with Bluetooth, Networking, I/O drivers API.

Secondly, Zephyr RTOS supports multiple hardware architectures and it allows to easily port an application on new nRF5x series release without changing the Nordic’s SoftDevices and SDK version.

Then, Zephyr RTOS is an Open Source Project. Hence, it is possible to adapt easily the RTOS and to add tools for a specific project and to contribute to the project.

The last reason concerns the memory. The nRF5x series allows to use complex RTOS that require more RAM and ROM memory as Zephyr METTRE DES CHIFFRES. More, Zephyr is a fusion between the SDK and the SoftDevices and therefore saves more ROM and RAM memory.

## My Contribution

As Zephyr is a recent project, some tools, useful for my project, were not already provided by Zephyr. Therefore, I contributed to the Zephyr project during my Bachelor’s Thesis. My contribution is:

* Fix a bug on the nRF5x SPI driver configuration
* Write the source code of the nRF5x GPIO Port 1 driver
* Write the source code of the nRF5x SPI2 driver

## Sources

<https://www.zephyrproject.org/>

<https://nexus.zephyrproject.org/content/sites/site/org.zephyrproject.zephyr/dev/api/api.html>

<https://devzone.nordicsemi.com/blogs/1059/nrf5x-support-within-the-zephyr-project-rtos/>

<https://www.apache.org/licenses/LICENSE-2.0>

<http://oss-watch.ac.uk/resources/apache2>

# Specifications

The Bachelor thesis of Nathan has as goal to find behavioral differences and

performance limitations of the NRF52840 Bluetooth Low Energy chip under different

conditions.

Two systems that are implemented differently but that are doing the same task are

compared against each other. Interrupt latency time as well as the power consumption

are the indicators of principal interest and must be measured and compared.

The difference of the systems consists of the fact that one is implemented using the

Zephir RTOS and the other is implemented without using an RTOS (bare metal).

The hardware setup, the software engineering of both systems as well as the

measurements must be documented precisely. A final report must resume the

technical elements as well as compare the two systems.

The following block diagram shows the systems of interest:

The behaviour of a Nordic’s nRF5x series SoC with Zephyr RTOS is tested under different conditions and the following performances are measured:

* **Interrupt latency**
* **Bluetooth Stack propagation**
* **Bluetooth Request/Response**
* **Power consumption**

Those performances are measured when running the BLE controller with high traffic.

To determine if using Zephyr is an advantage, the performances are measured on a Bare Metal system that executes the same tasks.

To create test conditions, a peripheral BLE acquires data from different devices:

* **A/D converter**
* **Accelerometer**
* **Interrupt generator**

The different devices allow to test the processor in real circumstances and to generate interruptions to stress the system.

Then a central BLE are notified of the data acquired via BLE.

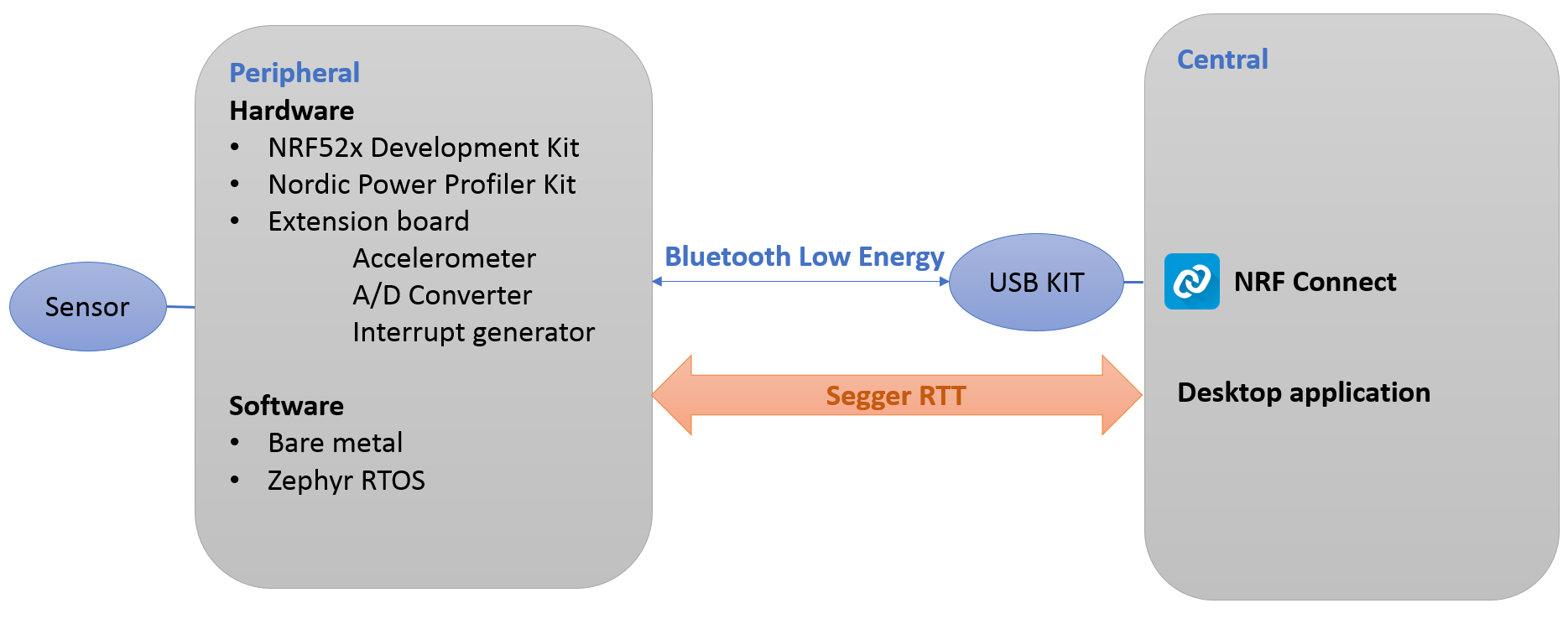


Figure 1: Systems representation

## Provided materials

To realise the Bachelor’s Thesis, some materials are provided:

* **Nordic nRF52840 Development Kit** to implement the BLE peripheral
* **Nordic Power Profiler Kit** to measure the power consumption

# Bluetooth Low Energy

## Introduction

## Requirement

## Test cases

## Type of Measurements

# Hardware

The Hardware is separate in six parts:

* **nRF52840 SoC**
* **Micro USB-B** to communicate with a PC and programme the chip
* **Power supply** provided by the Micro USB-B
* **Nordic Power Profiler** **Kit (PPK)** to measure the power consumption of the chip only
* **Connector interface** to connect the extension board, Power Profiler Kit and nRF52840 DK
* **Extension Board** to provide data to the chip

The extension board is the only part that is developed for this project. The Power supply 3V, the Micro USB-B and the nRF52840 SoC are on the nRF52840 Development Kit.

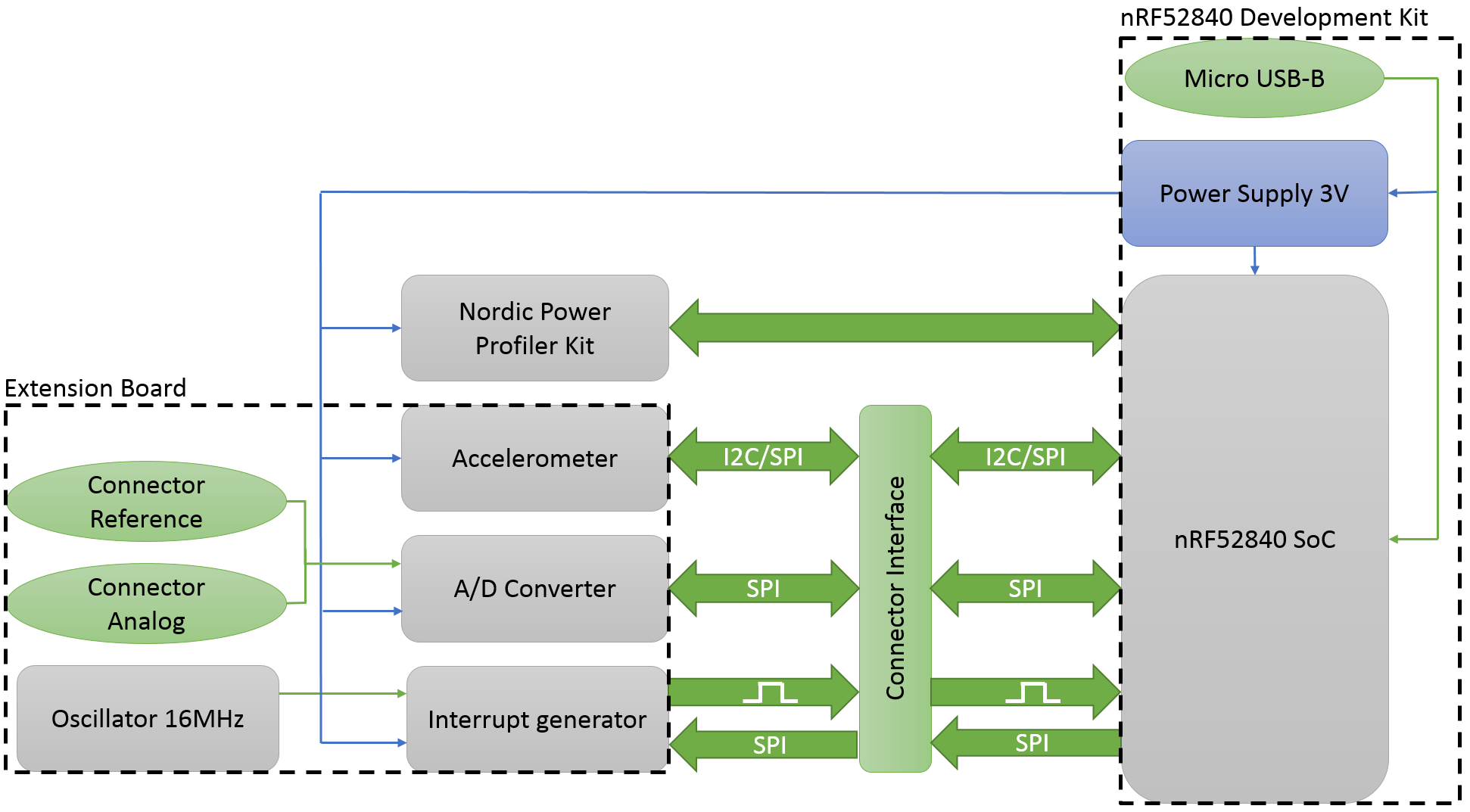


Figure 3: Hardware schema block

## nRF52840 SoC

… Why this chip

Describe some features

## Micro USB-B

The connector Micro USB-B is used for different purposes:

* **To programme** the nRF52840 Chip
* **To provide 5V power supply** to the system
* **To Debug** the nRF52840 Chip
* **To transfer the data measured** with SEGGER Real Time Transfer

The connection SEGGER Real Time Transfer (RTT) is used by the Nordic Power Profiler Kit to transfer the data to a PC. This connection can be used to transfer other data from nRF52840 chip as results of performance measured.

## Power Supply

The power supply transforms the power supply of 5V, provided by the Micro USB-B, to 3V to supply all the system. To transforms the power supply, there is a fixed 3V buck regulator and one voltage follower regulator on the nRF52840 DK.

Due to the low consumption of the system, **estimation et courant fournit par PC**, the power supply of the nRF52840 DK is far enough to provide power to all the system.

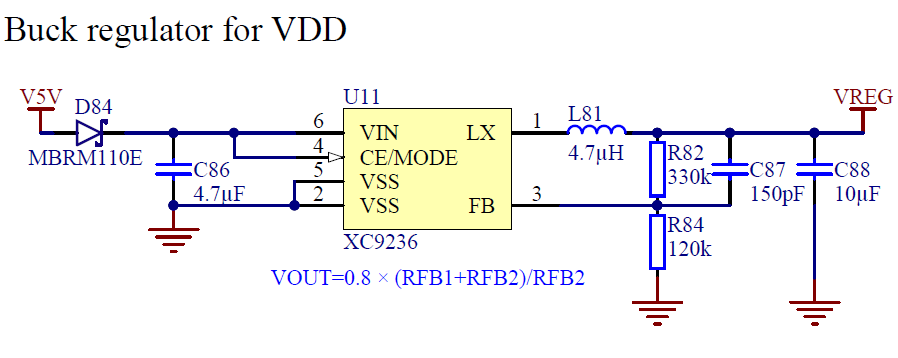


Figure 4: Regulator buck 3V schematic

## Connector Interface

The connector interface is defined by the nRF52840 development kit and is almost the same for all the Nordic’s DK.

The connector allows:

* **To supply the extension board and the Power Profiler Kit**
* **To measure the power consumption** of a nRF5x SoC
* **To access to the GPIOs** of a nRF5x SoC

The extension board and the Power Profiler Kit are plugged on the nRF52840 DK with the connector interface that allow an easy connection of the different part of the system.

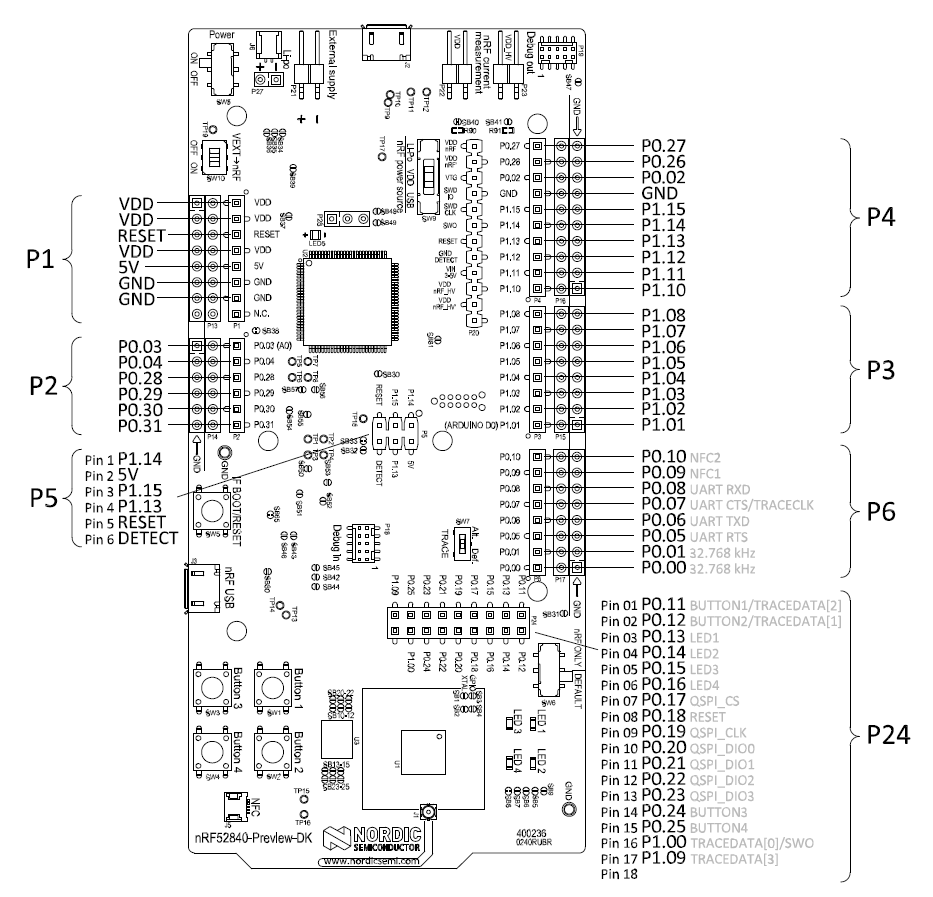


Figure 5: nRF52840 Preview DK board connectors

## Nordic Power Profiler Kit

The Nordic Power Profiler Kit is an easy use tool for the measurement and power consumption optimization of embedded solutions. It provides the following features:

* **1µA-70mA current measurement range**
* **0.2µA measurement resolution**
* **77kHz sampling rate**
* **Desktop application** in python allowing customization

The Nordic Power Profiler Kit is the best solution because it allows to measure only the power consumption of the nRF5x chip.

The Desktop application communicate with the PPK using the SEGGER Real Time Transfer of the nRF5x Chip. This connection can be used to transmit other measurements via RTT. The Desktop application can be modified to display those measurements.

## Extension Board

The extension board is the single part that the hardware is developed. It is connected to the nRF52840 DK using the interface connector.

The purpose of the extension board is to use the nRF52840 Chip under real conditions of use. To do that, the extension board provide the different elements.

* **A/D converter** that can be connected to a generator function or an external sensor.
* **Accelerometer**
* **Interrupt generator**

All components as some general criterion to ease the order:

* **Not too expensive (max 5CHF/43NOK)**
* **Package easy to solder**
* **Same provider**

### A/D Converter

The A/D Converter provides a large quantity of data that the chip must be able to deal with no loss. As the large quantity of data to stress the chip, the way to get the data must stressful as well.

The component used is the Delta-Sigma ADC **MAX11200** that provides the following features:

* **24 bits Resolution**
* **2.7 to 3.6V power supply current**
* **Reference Voltage**
* **SPI (SCL max 5MHz)** to calibrate and get the data
* **Ultra-low-power** with power-down mode

An ADC communicating with SPI is chosen due to the requirement to communicate fast.

The Analog input can be provided by a function generator (Connector BNC 50Ω) or by an external analogue sensor (Pin 2x1). Two resistances can be soldered to use the ADC in current loop system.

The Reference voltage can be the power supply voltage or an external reference if the external sensor has specific requirement.

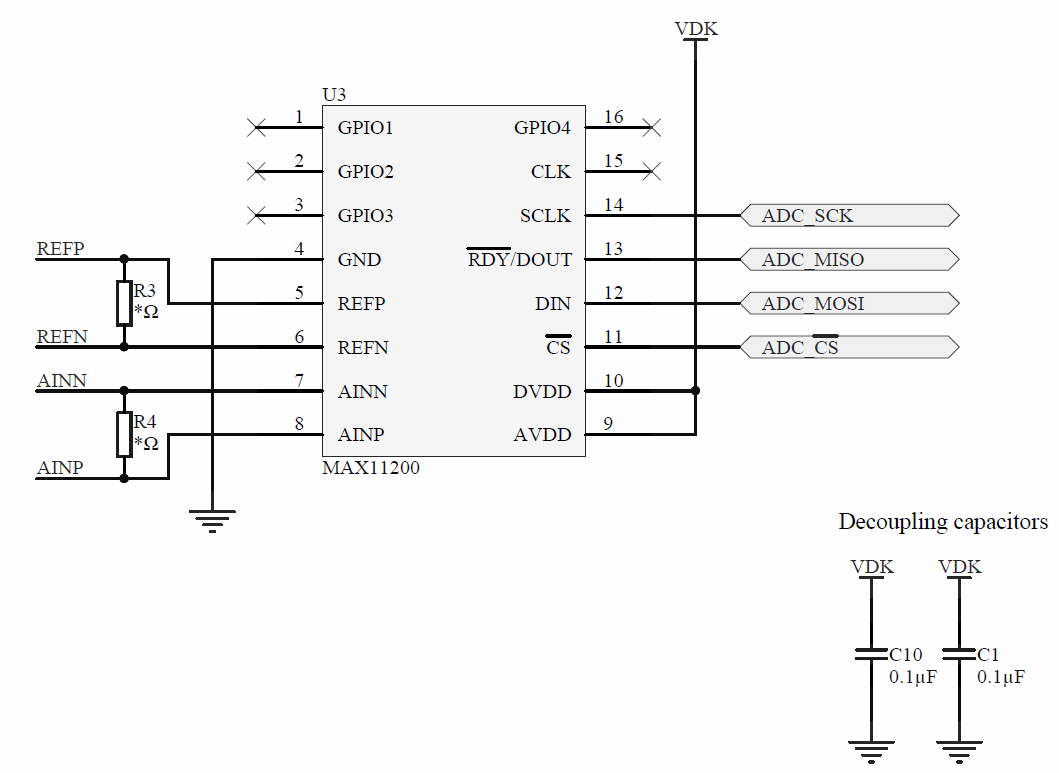


Figure 6:MAX11200 schematic

#### *Throughput*

Data Rate:

Continuous Conversion SCYCLE=0

60 120 240 480 [sps] LINEF=0

50 100 200 400 [sps] LINEF=1

Single-Cycle Conversion SCYCLE=0

1 2.5 5 10 15 30 60 120 [sps] LINEF=0

0.83 2.08 4.17 8.33 12.5 25 50 100 [sps] LINEF=1

Resolution: 24Bits

Data Frame: 32Bits (8bits addr + 24bits data)

Time between new data:

min = 2.08ms(1/480),

max = 20ms(1/50)

Time data frame transmition:

64us (1/5M \* 32)

### Accelerometer

As the A/D Converter, the accelerometer provides a large quantity of data that the chip must be able to deal with no loss.

The component used is the accelerometer **LIS3DH** that provides the following features:

* **16 bits Resolution**
* **3-axis**
* **±2g/±4g/±8g/±16g**
* **1.7 to 3.6V power supply Current**
* **FIFO 32-level 6 bytes**
* **I2C (SCL max 400kHz)/SPI (SCL max 10MHz)** to get the data
* **2 Interrupt pins** to notify when new data are available
* **Ultra-low-power** with automatic power-down mode

An accelerometer communicating with I2C is chosen to use different features of the chip. However, the SPI can be used as well.

The LIS3DH provide a FIFO to store data. This FIFO can be read at one time with a frame of 192 bytes, 2 bytes per axis. This frame provides a good test to stress the chip.

Other features of the LIS3DH are the Click and Free-fall detection that generate interruption. Those features are interesting because they can simulate real interruption from a sensor.

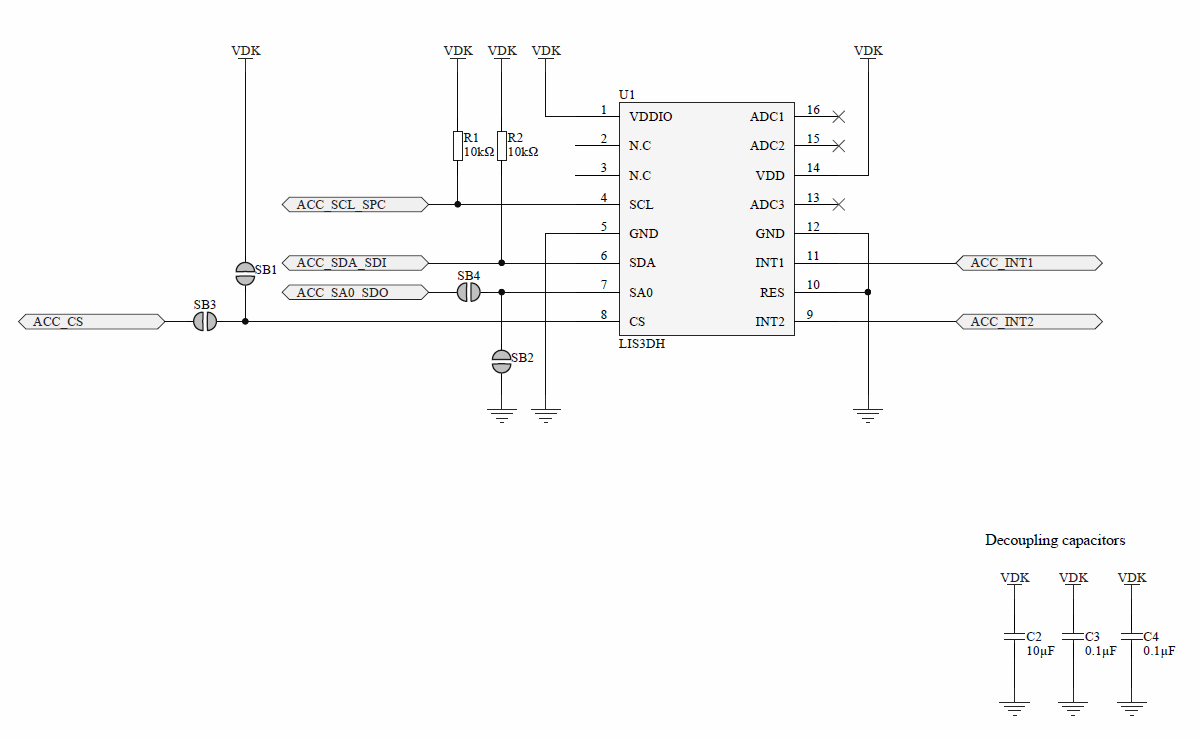


Figure 7:LIS3DH Schematic

#### *Throughput*

I2C CLK: 400kHz

Data Rate:

HR

1 10 25 50 100 200 400 1.344k [Hz]

Normal

1 10 25 50 100 200 400 1.344k [Hz]

Low-power

1 10 25 50 100 200 400 1.6k 5.376k [Hz]

Resolution:

HR 12Bits

Normal 10Bits

Low-power 8Bits

Data Frame: 16bits (8bits device addr + 8bits register addr)

+ 192Bytes data (32\*Axis-X LSBytes + 32\*Axis-X MSBytes

+ 32\*Axis-Y LSBytes + 32\*Axis-Y MSBytes

+ 32\*Axis-Z LSBytes + 32\*Axis-Z MSBytes)

Time between new data:

min = 1s(1/1),

max = 0,186ms(1/5376) or max = 0,744ms(1/1344)

Time data frame transmition:

min = 80us (1/400k \* (16 + 16bits (MSBytes and LSBytes of one axis)))

max = 3.88ms (1/400k \* (16 + 192Bytes))

### Interrupt generator

The interrupt generator generates pulse that create interruptions in the programme. The period of interruptions can be easily changed to modify the test conditions.

The component used is the Programmable Waveform Generator **AD9837** that provides the following features:

* **16MHz Clock**
* **28 bits (0.06Hz)** **Resolutions**
* **2.3V to 5.5V power supply current**
* **3 Wires SPI** to programme the waveform type and frequency
* **Low power** with power-down option

A 28Bits Register is used and programmable via SPI to calculate the frequency. The formula below defines the frequency:

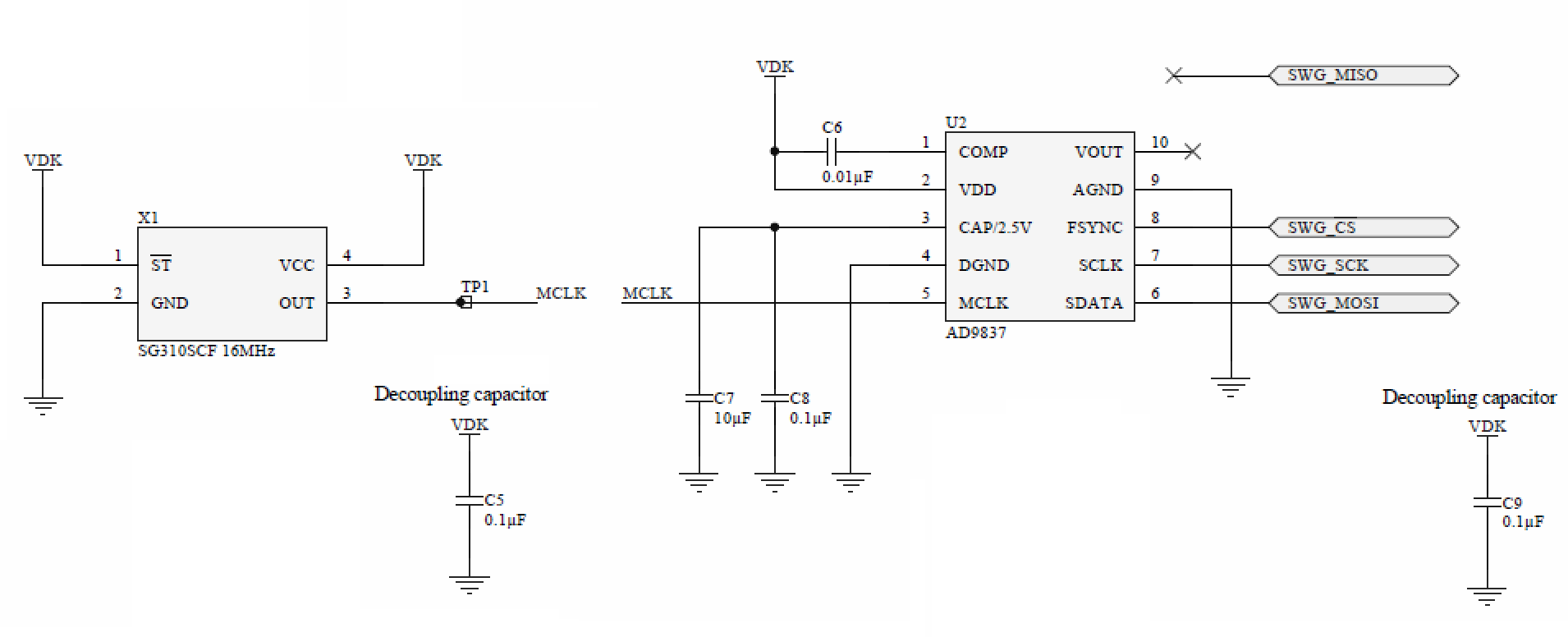


Figure 8: AD9837 schematics

## Source

# Software

## Architecture

### Abstract Layer

### Drivers

### Application

## BareMetal Abstract Layer

## Zephyr Abstract Layer

## Peripheral

## Central

## Measurements Code

### General

### Zephyr

### BareMetal

# Measurements

## Equipment

## Results

## Comparisons

# Conclusion