Design of a low cost vibration monitoring system - Appendix I: **Technology prototype design and implementation**

By
Morten Opprud Jakobsen, AU Herning
June 10, 2016

1. Introduction

This project includes the technical activities for design and implementation of the vibration monitoring prototype.
The technical report is structured as follows:
Chapter 3: Summarizes the overall requirements
Chapter 4: Contains the design of the edge-node, hardware and software for digitizing and collecting vibration signals, and allowing network connectivity.
Chapter 5: Describes the design and details of the data analysis, visualization and storage
Chapter 6: Looks at the architecture for a typical cloud providers "stack"

2. Table of contents

3. Design requirements

A number of functional design requirements are discussed and summarized in the main report chapters 8 and 9. The relevant technical requirements are derived in each of the technical chapters. The technical detail requirements are used as guideline for the design, and will be used for verification of each of the functional blocks

3.1. Calculation of characteristic bearing frequencies

The essential fundamental frequencies are as follows :

fundamental bearing frequencies, for stationary outer-ring applications		
f_r	=	*inner ring rotational frequency*
$f_{c/o}$	=	*Fundamental cage frequency, relative to outer ring* $f_{c/o} = f_r/2 * (1 - d/D\ Cos(a))$
$f_{c/i}$	=	*Fundamental cage frequency, relative to inner ring* $f_{c/i} = f_r/2 * (1 + d/D\ Cos(a))$
$f_{b/o}$	=	*Ball pass frequency, outer ring* $f_{b/0} = Z * f_{c/o}$
$f_{b/i}$	=	*Ball pass frequency, inner ring* $f_{b/0} = Z * f_{c/i}$
f_b	=	*Rolling element spin frequency* $f_b = D/2d * f_r * (1 - (d/D * cos(a))^2)$
D	=	*Pitch circle diameter (center diameter of rolling element path)*
d	=	*Diameter of roller elements*
Z	=	*Number of rolling elements*
a	=	*Contact angle*

Table 1 - characteristic frequencies for stationary outer ring bearing applications [13]

3.2. What to expect from practical systems

Based on the applications where bearings are fitted today, the minimum and maximum characteristic bearing frequencies are derived, and used via the calculations listed above.

Low speed application

Table 2 - Bearing from dairy centrifuge

Parameter	**Value**
Inner diameter	130 mm
Outer diameter	185 mm
Roller diameter	28,5 mm
No of rollers	12
Min RPM	50
Max RPM	75

High speed application

Table 3 - Bearing from spindle on wood or metal cutting machine

Parameter	**Value**
Inner diameter	85 mm
Outer diameter	105 mm
Roller diameter	9,0 mm
No of rollers	25
Min RPM	15.000
Max RPM	20.000

The contact angle in bearings is typically between 30-40 [deg], 35[deg] will be used.
The one extreme condition in the **low speed range**, results in inner- and outer- ball pass frequencies of ***4,2[Hz]*** and ***5,8[Hz]***
The other extreme condition in the **high speed range**, that results in inner- and outer- ball pass frequencies of ***571[Hz]*** and ***678[Hz]***
Thus the measuring system should be able to detect these frequencies, and the distance between them, as this may indicate beginning wear.

4. Edge-note prototype design

The following section contain chapters regarding design of a data acquisition device, *"the vibration monitor"*, with the intent of fulfilling requirements outlined for the project.

4.1. System architecture

The device design in this chapter revolves around the design of a device, the initial prototype, *PT1* capable of fulfilling the necessary requirements in the lower layers 1, 2 and 3

Layer	Functionality	
6:Act	Process and tools for taking action based in the insights	
5:Analyze	Capabilities for data to develop insight	
4:Store	Systems for storing and organizing data	
3:Connect	**Physical connection to pass data to the cloud / central system**	
2:Collect	**Edge hardware and software for gathering and formatting sensor data**	
1:Sense	**Sensors and endpoints digitizing physical world quantities into data**	

Table 4 : implemented IIoT "stack" layers in the data acquisition device

The design will be build as an embedded system, composed of a processor platform with, memory, ethernet and analog sampling capabilities. Furthermore the design includes an analog sensor interface, capable of reading the selected accelerometers. The design is to be modularised, into a number of functional blocks. The eases reuse, design and testability of the design.

The initial prototype of the vibration monitor will crafted from the system architecture outlined in figure ***1***

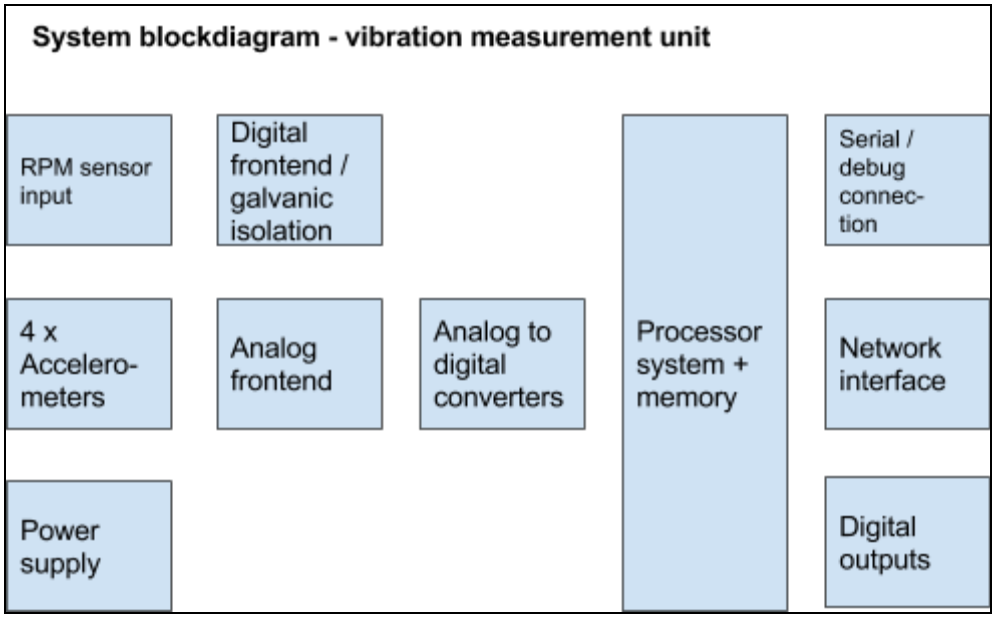


Figure 1 - vibration monitor architecture

To ease the development effort required to design and build a functional prototype, the prototype will be build around a processor-platform development-board.
A lot of development boards for experimenting with different processor architectures exists, often with good and extensive software libraries and development tools included. Utilizing a suited processor-development platform will save valuable design, build and verification time, so emphasis can be put on the actual application, the measurement of vibrations, and analysis.

4.1.1. Freescale FRDM K64F processor platform

The FRDM K64F platform from freescale semiconductors is one such platform, and will be used for building *PT1*. The K64F platform*[28]* has a number of particular features that deems it well suited for the requirements of the *PT1* device.

Dual 16 bit ADC's, the processor has 2 independent 16 bit analog to digital converters on board:

- Sample Rate up to 200 Ksamples / second in 16 bit mode / 800 Ksamples in 13 bit mode
- Hardware averaging
- On chip sample & hold circuitry
- DMA integration

ARM Cortex M4 architecture a proven 32 bit architecture, with compatibility towards a large range of development tools :

- Up to 120 MHz execution speed
- 32 bit core with DSP capabilities (suited for on-chip FFT)
- Floating point unit (also good for FFT and floating point number operations)

On chip ethernet Media Access Controller (eMAC) allows connection to ethernet via a physical transceiver chip (PHY) to a 10/100 Mbit Ethernet.

Also the K64F chip has a wealth of timers, serial interfaces, and an on-board hardware debugger, for integration with the available Freescale or 3rd. party software development tools.

A comprehensive set of software libraries (an SDK), for network communication, configuration of timers, ADC's and other relevant peripheral functions is available for free download and commercial usage.

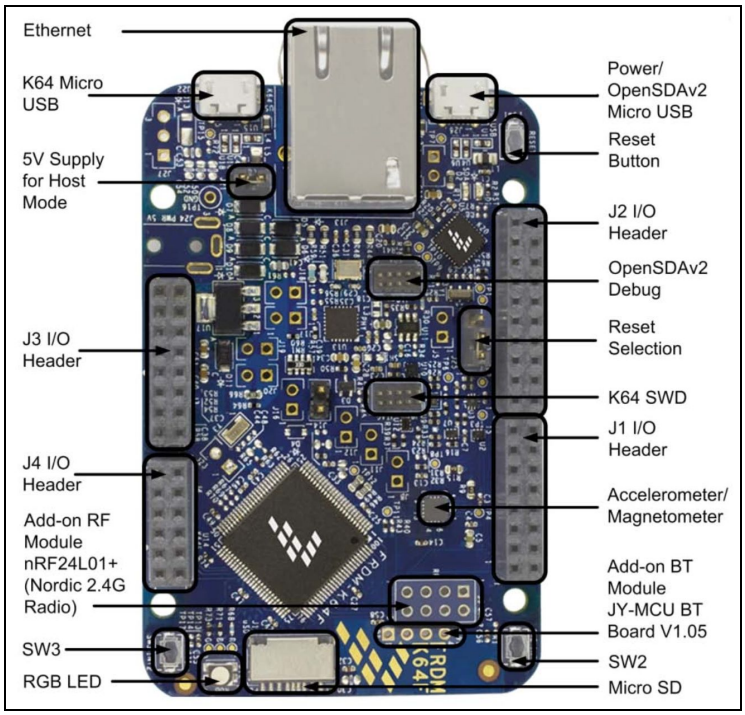


Figure 2 - FRDM K64F platform overview

The platform has additional sensors and features, inputs and outputs, that is not utilized in this intended application. The functional blocks for interfacing accelerometers, power supply, and digital I/O will be designed and build on an additional add-on circuit board, mounted on top of the FRDM K64 via J1, J2, J3 and J4 headers.

4.2. Accelerometer

The ACH-01 accelerometer, for vibration measurement is selected due to low cost, and compact size, compared to the commercial rugged stainless steel type accelerometers used in vibration measurements setups today.

To accommodate the electrical interface requirements for this accelerometer, a conditioning circuit, specifically an analog front end has to be designed, so the signal is conditioned and scaled to fit the data collection system.

Furthermore the accelerometer front-end has the purpose of minimizing unwanted signal distortion, from noise and disturbance sources. Electrical noise from frequency converters, as well as electric noise induced by magnetic disturbance from the rotating motors in the machines to observe can cause undesired contributions in the signal of interest, generated by the accelerometer.

4.2.1. Specifications

For the ACH-01 sensor following functional specifications are known [14]

4.2.1.1. Performance specifications

The sensor ACH-01 from measurement specialities is to be used and has:

1. a sensitivity of 9[mV/g] +/-2[mV/g]
2. a 3[dB] frequency range from 2[Hz] to between 10[kHz] and 20[kHz]
3. a dynamic range is 150[g] @-3dB
4. a typical linearity of 0.1[%], up to 1[%]
5. a resonant frequency of 35[KHz]

4.2.2. Functional description ACH-01

The accelerometer is build from a piezoelectric polymer film, on a ceramic substrate. Acceleration on the ceramic substrate will cause a slight deflection of the piezo material, which due to its piezoelectric nature will generate a voltage proportional to the applied acceleration.

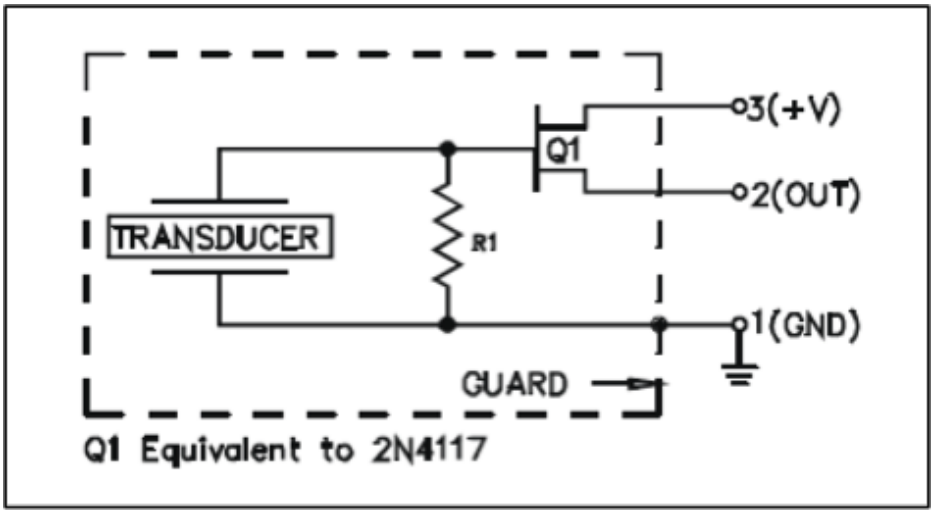


Figure 3 – ACH01 functional diagram

Due to the relatively high output impedance of the piezo crystal, a *common-drain amplifier* is build with R1 and Q1. A common drain amplifier has a voltage gain of 1, but has the benefit of working as an impedance transformer, allowing a lower output impedance, of the circuit, and from this, less sensitivity to impact from electrical noise.
For the common drain amplifier to function properly, it requires a biasing circuit from the output towards ground.
When properly biased, the ACH-01 can measure both positive and negative acceleration.

4.3. Accelerometer front end

The accelerometer front end interfaces and conditions the signal from the ACH-01 sensor, so it fits the requirements of the sampling system, that digitizes measurements.

4.3.1. Design requirements, initial design

4.3.1.1. Functional requirements

The analog front end should fulfill following requirements:

1. Ability to utilize the maximum dynamic range of the systems analog to digital converter
2. Minimize temperature impacts on measurements
3. Design to shield critical signal paths, and possible impact from electric and magnetic noise sources.
4. Should allow acces to measure vital parameters around the accelerometer

4.3.1.2. Performance requirements

1. Measurement range 1[Hz] … 20[KHz]
2. Temperature range -40…85[*C] or better
3. Linearity of 5% or better
4. Range +/- 10 [g]

4.3.2. Component selection

Since the sensitivity of the accelerometer is relatively low, 9[mV/g], compared to the typical input range of 0-3.3[V] of most analog to digital converters, additional amplification is required, to utilize the dynamic range of the converter.
The manufacturer of the accelerometer recommends using operational amplifiers similar to LF353 or better, for building conditioning and amplification circuitry [15].

4.3.3. Noise and disturbance

In an industrial environment, the presence of electric motors, heaters, actuators and electrical power installations will generate electric noise, capable of coupling into sensitive electronic circuits.
Many industrial installations are electrically shielded, by using shielded wires, and metal enclosures, since these minimise the impact, by providing a magnetic and electric shield, between the circuit, and the noise source. The accelerometer comes with a shielded cable, and the prototype will be designed with noise immunity in mind. Whether encapsulation in a metal enclosure is required, or a plastic enclosure will suffice, is to be determined by tests.

4.3.4. Dimensioning the accelerometers working point

The output stage of the ACH-01 is comprised from a 2N4117 compatible JFET transistor[16]. To minimize the temperature impact on the output stage, which may typically be rather significant, if using the transistor unbiased, it is beneficial to dimension a working point also known as *Q-point*, for the transistor, with respect to temperature deviation.

Looking at the 2N4117' s transfer characteristics we see that aiming for a Q-point at 35[uA] will leave us unaffected of temperature changes, in the entire working range of the device, from -55[*C] to 125[*C].
Since the 2N4117 has a typical *Vgs threshold voltage* of 1.1 V, thus marking the top graph gives a desired Q-point of 35[uA]

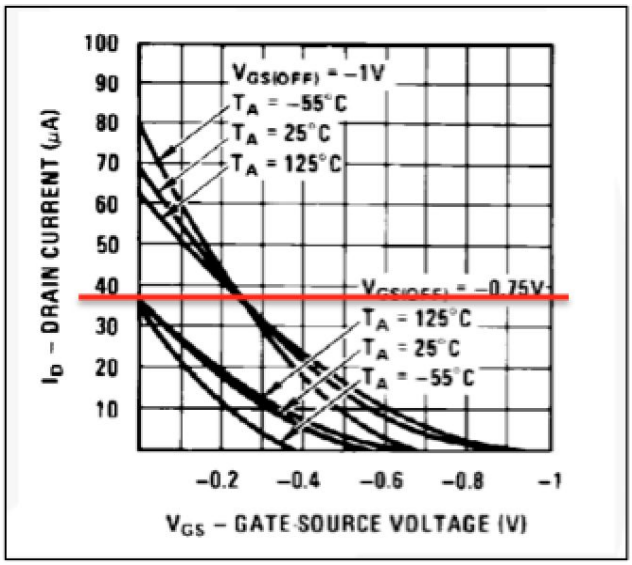


Figure 4 – selection of 2N4117's Q point

To get the accelerometers Q-point correctly biased, a current source, capable of sinking 35[uA] is required.
A simple current source can be designed as a *current-mirror* [17], composed of two identically bipolar transistors Q_1 and Q_2. Ideally these two transistors are matched, or a dual-type transistor is used, to ensure identical physical characteristics.
Furthermore the emitter resistors R_E must be identical.

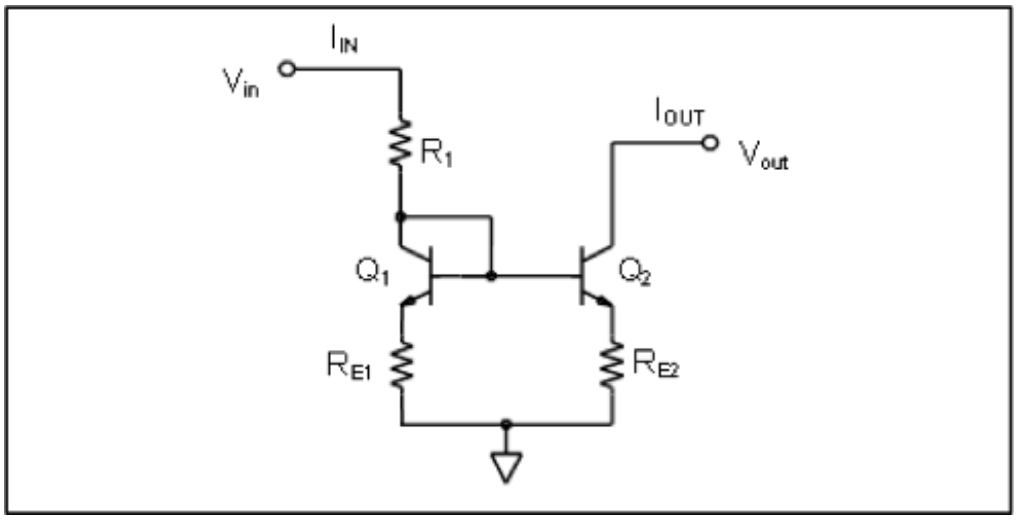
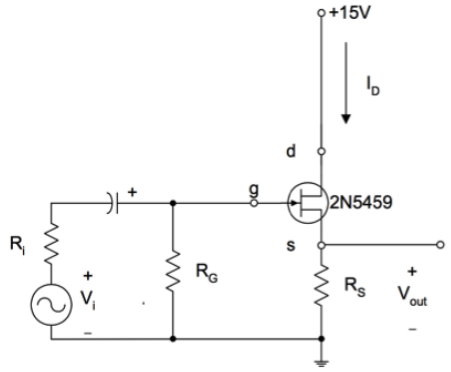


Figure 5 – a current mirror, where $I_{IN} = I_{OUT}$

The recommended system supply voltage, *VCC* is +5V, so the resistor in the current mirror is dimensioned to give a working point of the Vout voltage to be VCC/2, to allow maximum voltage swing from the accelerometer.

4.3.4.1. Determining the voltage gain of common-drain circuit.

The common drain circuit, similar to the output stage in the accelerometer, has a voltage gain described by the formula below [18].



$$A_v = \frac{g_m R_S}{1 + g_m R_S}$$

Figure 6 :Common drain circuit and associated voltage gain

g_m is a parameter describing the relation between V_{GS} and I_D expressed in [A/V^2]. Since g_m is between 70 and 210 for the 2N4117 JFET, the voltage gain A_V can be assumed to have the value 1.

4.3.5. Amplification and aliasing filter

The front end circuit is designed to deliver a *rail-to-rail* output signal, so amplification is to be so the output signal can be in the range 0..5[V], when measuring the maximum specified acceleration of +/-10[g].

Furthermore, the amplifier should limit the frequency amplitude above 20[KHz] to avoid amplifying possible signal, outside the ACH-01's measurement range, as well as provide an aliasing filter, to avoid sampling frequency components above $F_s/2$, according to the *Nyquist sampling theorem*.

A DC filter, to avoid DC bias of the amplifier, below 2[Hz] is also to be included in the amplifier.

The amplifier is build around the LF353, a dual operational amplifier (opamp). The circuits gain is set to 27.7, to get the desired 250[mV/g] resolution. The second opamp is used to generate an *artificial-ground,* allowing the accelerometer and amplifier to amplify both measured positive and negative accelerations force (both directions).

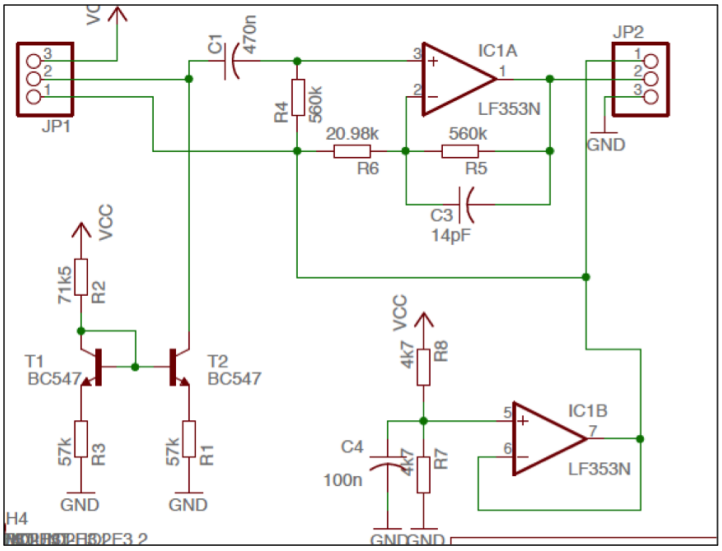


Figure 7 – amplifier and current source

Circuit description:

C1 and *R4* makes up a 1.6[Hz] 1 order high-pass filter, blocking out DC bias from the accelerometer

R5, *R6* and *IC1A* makes up a non-inverting amplifier, with a fixed gain of 27.7.

R5 and *C3* composes a 1 order low-pass filter, allowing signals below 20[KHz] to pass.

R1, R2, R3, T1 and *T2* is the current source, pulling 35[uA] from the ACH-01, to bias it in the desired Q-point.

4.3.6. Circuit implementation

The circuit is designed on a two sided prototype PCB, with all excessive copper area used as ground plane, to improve noise immunity.

The calculated components are rounded off to the nearest available resistor and capacitor sizes.

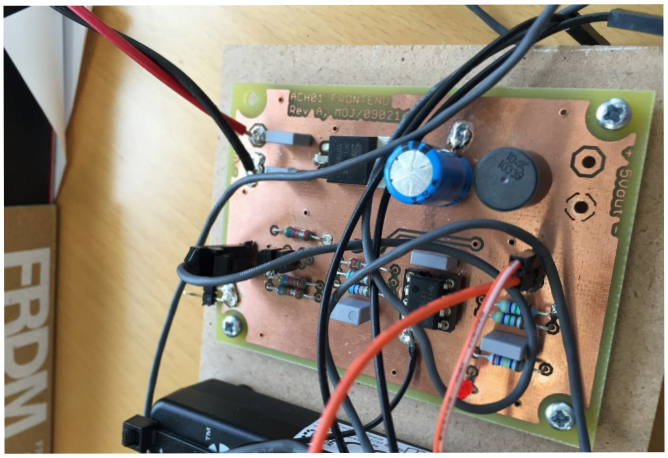


Figure 8 - accelerometer front end prototype circuit board

4.3.7. Circuit simulation

In order to verify the designed circuit, a set of circuit simulations are performed.

4.3.7.1. DC simulation and gain @ 1[KHz]

A circuit simulation is performed on the designed circuit to verify current bias, and amplification at 1[KHz] with 10[mV] input, corresponding to 1.11[g]. Measurements are made at the amplifiers input stage, as well as the output of the amplifier.

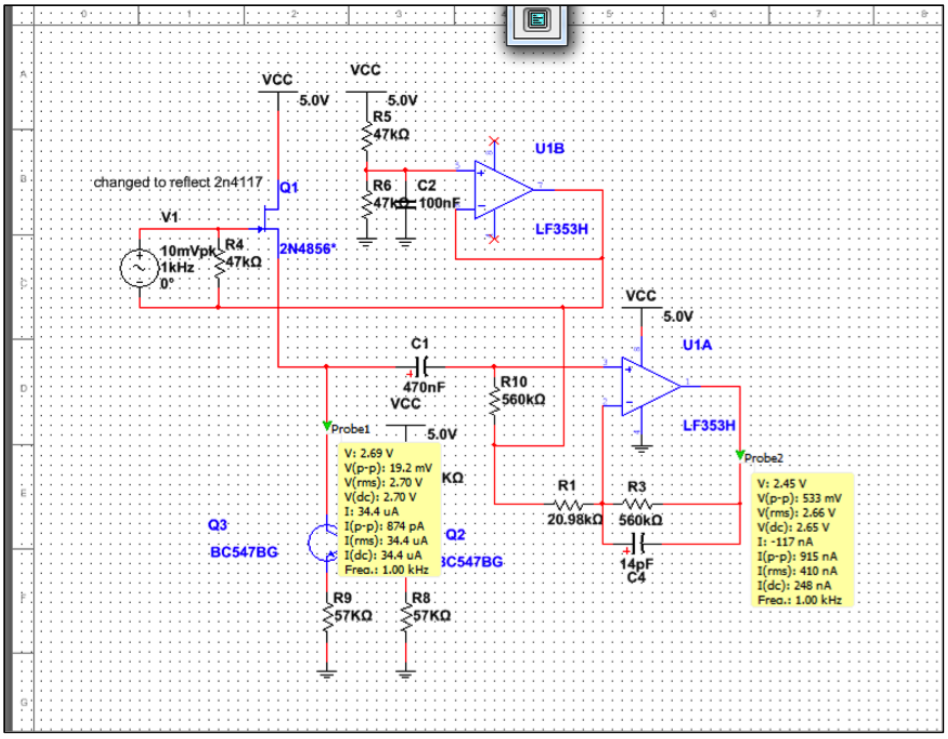


Figure 9 – DC simulation, showing the achieved DC bias for the input, amplitude from the accelerometer, and output voltage and amplitude for the output.

The gain at 1[KHz] is seen to be 533[mV]/19.2[mV] = 27,7 times.

4.3.7.2. AC sweep simulation

An AC sweep from 1[Hz] to 20[KHz] is performed, to verify the high and low pass filters limiting the signal range to be within the measurable range from the ACH01.

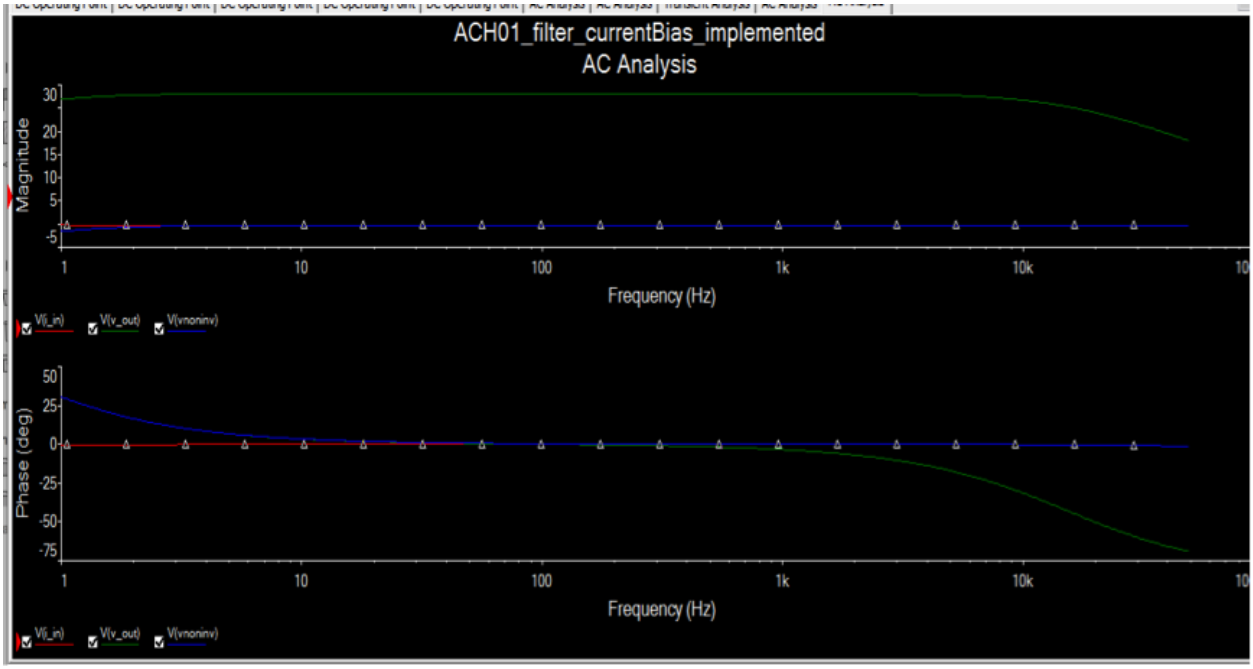


Figure 10 - circuit simulation, frequency sweep

4.3.8. Circuit verification

In order to verify the functionality of the implemented circuit a set of tests are performed.

4.3.8.1. AC sweep analysis

The build prototype amplifier is connected to a network analyzer to verify filter characteristics and the amplifier gain.

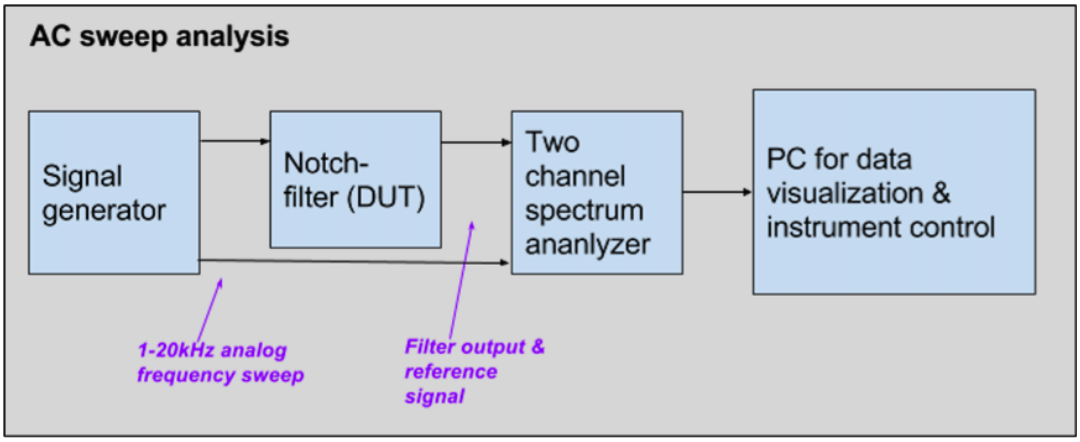


Figure 11 – AC sweep measurement setup

The connected signal generator is programmed to sweep from 1[Hz] to 20[KHz]. The spectrum analyzer records the inputs from signal generator and the filter response, and plots the result.

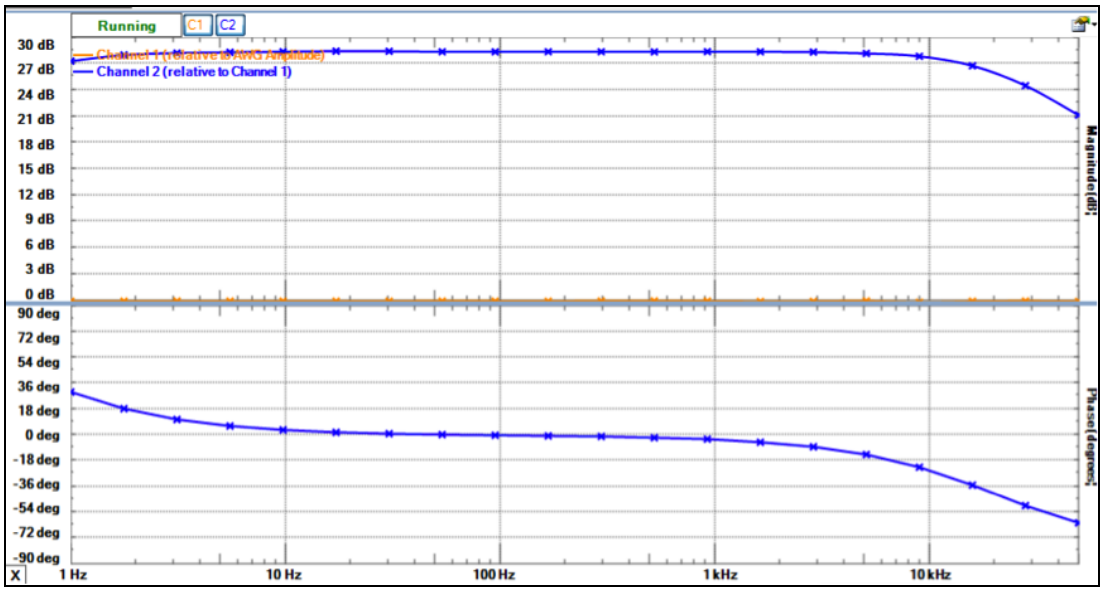


Figure 12 - circuit verification, shows that the 3dB bandwitdth is app. 1[Hz] - 18[KHz]

4.4. Frontend, version 2 design

Version 2 of the accelerometer frontend is built from the initial prototype design. Version 2 has 4 analog inputs and one digital tacho input, as well as 3 digital power outputs, for driving indicator lights. An on-board switch mode power supply is also implemented on the version 2 circuit board.
The designed board is intended to be designed as a "shield" capable of plugging into the freescale K64F board, resulting in a compact "sandwich" design, where signal conditioning is close to the processor, and the analog inputs on the microcontroller.

4.4.1. Design partitioning

Version 2 of the board will contain both low power analog, and relatively high power analog and digital circuits. When working with sensitive analog circuits, which is the case for the amplifiers and analog front end, one should respect that these are sensitive to noise, generated by high power, or high frequency switching parts of the design.
The functional blocks to implement on the PCB are shown on figure *13*.

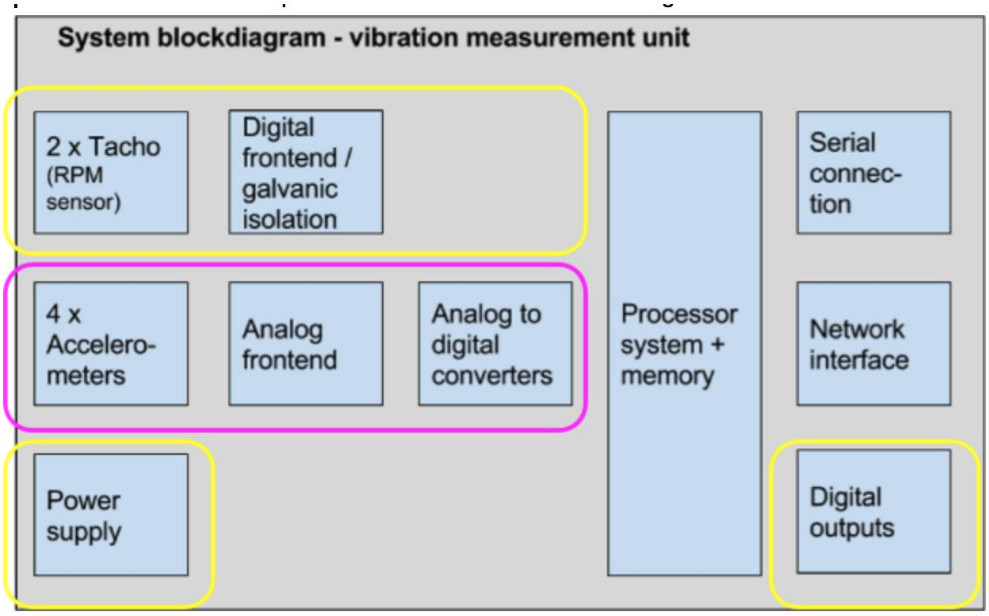


Figure 13 - low power analog domain - pink, and digital & power domain - yellow

The circuits are split into two domains, low-power analog, and digital & power, to minimize the impact of noise. Furthermore, all unused pcb area is left as a connected copper plane, effectively providing a "ground-plane" acting as a shield, and improving signal integrity.
Figure 14 shows the designed PCB, with the analog domain on the left side, separated from the power and digital part on the right.

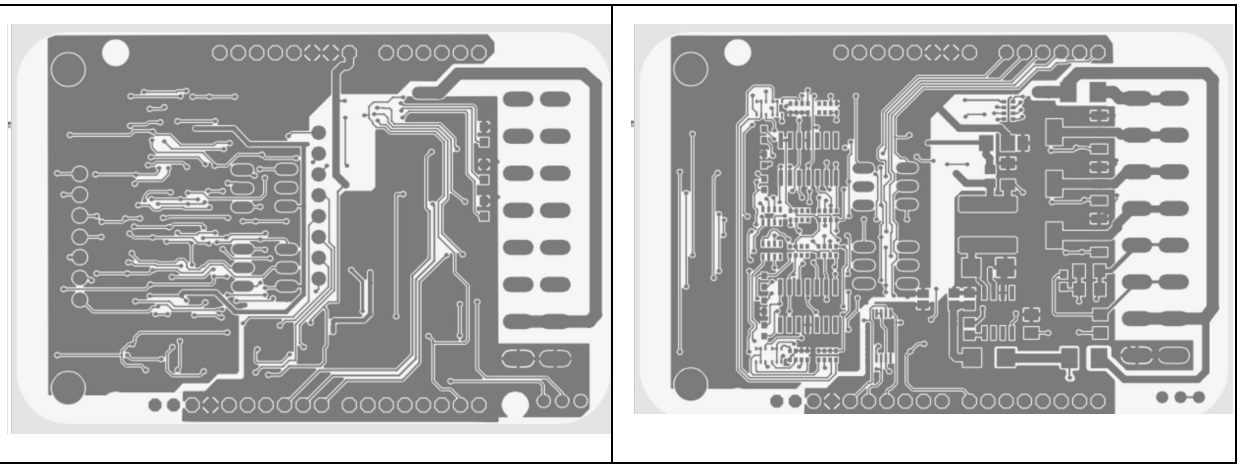


Figure 14: bottom copper (left) and top copper layers (right), illustrate the two signal domains

Additional filter components are included in the design, to minimize the impact and noise from the digital circuit.

4.4.2. Power supply design

The accelerometers, the sampling system, the processor and the network communication circuit runs from a 5V supply. In most industrial environments a robust, 24V DC supply is available.
A suited switched mode power supply is to be designed to generate the 5V power supply.
To derive design requirements for the power supply a power budget is estimated, from available data.

Functional block	**Estimated current consumption**	**Comment**
Processor	*50 [mA]*	Maximum consumption when executing code from internal Flash memory [19]
Ethernet	*100 [mA]*	100MBit operation consumption for a similar ethernet transceiver [20]
Analog frontend	*100 [mA]*	High estimate, based on current sourcing capabilities of operational amplifiers in the circuit
LED's & digital input	*50 [mA]*	6 low power LED's + one optocoupler
Sum	***300[mA]***	

Table 5, power-budget

The power supply will be derated 50%, to increase lifetime, by avoiding temperature increase when the converter is loaded. Furthermore the expected voltage of 24V can be expected to fluctuate several volts, as other varying load from other devices on a shared 24V supply rail may cause impact .

4.4.2.1. Design requirements

1. Output voltage 5VDC +/- 50mV
2. Output current 600 mA
3. Input voltage min 16V
4. Input voltage max 32V

4.4.2.2. Converter topology

Two types of voltage regulator exists, a simple linear-regulator, or a switching type that requires additional filtering components. The selection regulator type depends on the amount of power to convert, and the allowable loss.

Power loss translate into heat, which will increase the temperature of the entire design, and impact the lifetime, negatively. Contrarily selecting a switch-mode power supply type is more expensive and should only be selected if the power demands are significant.

Assuming a nominal voltage of 24V, and the nominal load of 300[mA] the power loss can be found as

$$P_{loss} = (V_{in} - V_{req}) * I_{load}$$

$$P_{loss\,nom} = (24 - 5) * 0.3 = 5,7\,[W]$$

Temperature increase is affected by whether the construction is encapsulated or is ventilated, so heat can be transferred away from the circuit board.
A quick estimate of temperature increase, when using a standard linear regulator (type TLV1117-50IDRJR), on a non-ventilated circuit board (PCB), can be derived from [21].
The resulting PCB temperature will reach a target temperature of up to 168°C, and the junction temperature of the regulator can reach 284°C, which is <u>not</u> acceptable.

The design will therefore be implemented using a simple switched mode regulator.
An integrated controller, type LM22675[22] from TI is used. An application calculator[23] is used for finding optimum filter components for the power supply.
The resulting design is capable of fulfilling the design requirements, and can operate from 11-34 V, and supply 1000mA, with a maximum PCB temperature increase of app 9°C at 30 ambient temperature. The converter has an efficiency between 82% and 91% depending on input voltage.

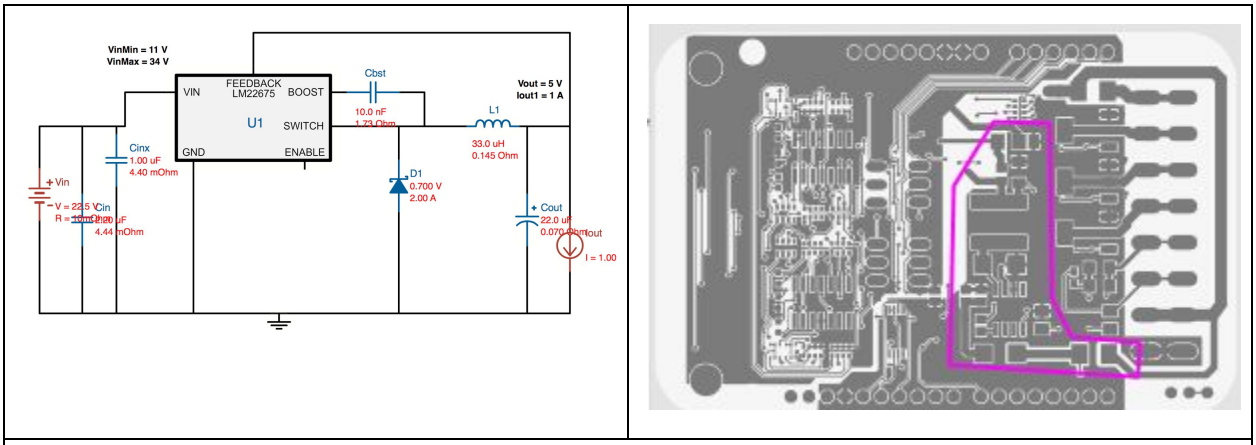


Figure 15 : power supply schematic (left), power supply PCB design (right, highlighted)

When designing the PCB is is essential to keep the length of the switching powersignals short, to avoid noise and disturbances in the analog circuits, as well as too long feedback loops. Figure ***15*** *(right)* highlights the power supply design on the PCB.

4.4.2.3. Circuit Verification

Following practical measurements are performed on the power supply

Test	condition	expected	measured	result
V no load	V_nom applied, no load connected	5.00V +/- 50mV	4.963 V	OK
V nom load	V_nom applied, 300mA load connected	5.00V +/- 50mV	4.962 V	OK
V max load	V_nom applied, 600mA load connected	5.00V +/- 50mV	4.961 V	OK
Temp increase max load	V_nom applied, 600mA load connected for 60 min, pcb enclosed.	5.00V +/- 50mV	tbd	tbd

Table 6 - power supply verification testing

The power-supplies deliver the expected supply voltages under the given test conditions. A long time test for heat development, and boundary conditions, regarding supply voltage should be performed, before additional revisions of the design is made.

4.4.3. Digital I/O design

To perform frequency analysis, and correlate the measured vibrations to the current rotational speed of the application, a rotation input is required. Furthermore to allow a simple local user interface, capable of notifying the operator of a machine, three digital outputs, able to drive signal lights are to be implemented.

In order to able to measure the rotational speed of the application, a general purpose digital input is designed. The input is build around a *dual-optocoupler*, allowing different types (active high or active low) and different voltage signals to trigger the input. Furthermore, the use of an optocoupler allows the input to be connected in parallel with existing digital-rotation sensors, that are typcally availible on machines.

The three digital outputs are to be capable of delivering 1.5 A, 24V, to be able to supply upto three signal lamps, for notifying operators. Furthermore, the outputs should be short-circuit protected, to avoid damaging the common 24V supply rail, if shorted by accident.

4.4.3.1. Design requirements

Digital inputs

1. Input trigger voltage 5-48V DC
2. Frequency range 0.2 - 1000 Hz (30-60.000 RPM)
3. Ability to filter out high frequency noise

Digital outputs

1. Capable of supplying 1.5A current
2. Output voltages up to 24V
3. Short circuit protected
4. Capable of driving inductive loads, e.g. lamps

4.4.3.2. Circuit design

The digital input's dual optocoupler[25] can detect and handle input currents from 1 to 60mA, and can transfer input signals up to 200 KHz.
A current limiting resistor pair is used to limit the maximum allowable forwards current, and together with C13, forms a low-pass filter, that can be tuned after a number of rotation sensors have been tested.

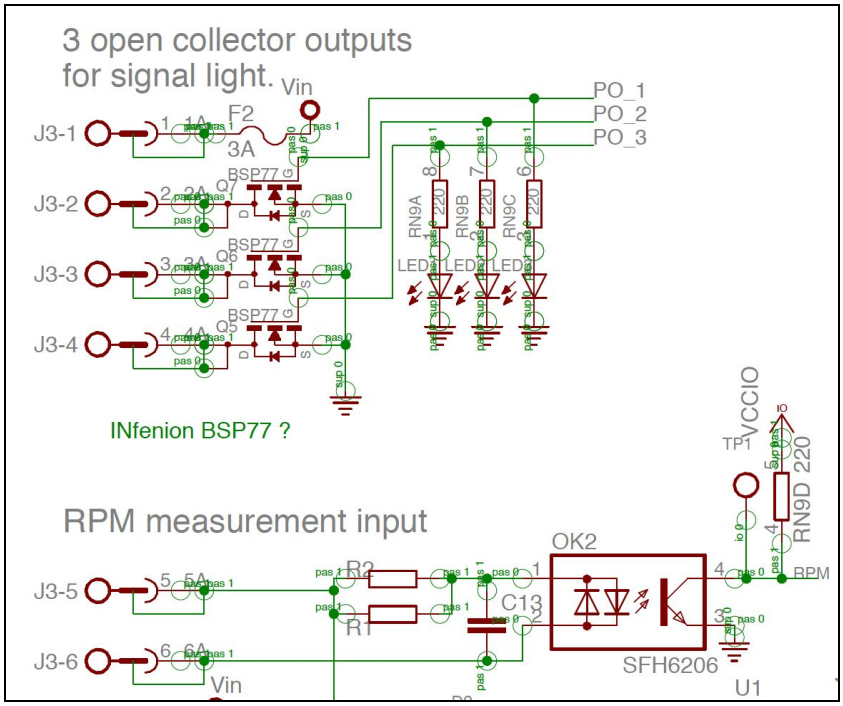


Figure 16 - schematic of digital input (RPM) and outputs

The three digital outputs are build around *Infineon BSP77 smart low-side Power Switches* [26] these devices features internal short circuit, and overload protection , and are capable of driving resistive, inductive and capacitive loads.
J3-1 provides 24V DC output, for supply voltage for signal lights, the pin is protected by a 3A auto resettable PTC fuse.

Three signal LED's are driven by the control signals for the digital outputs, allowing a quick visualization of the output states, if no signal light is connected, or as additional verification of output state.

4.4.4. Circuit verification

Following practical measurements are performed on the digital inputs

Test	condition	expected	measured	result
F_min V_min	Minimum voltage, Minimum frequency	3.3V / 0V 1-5uS rise/fall time	3.3/2.0V	Failed
F_max V_min	Minimum voltage, Maximum frequency	3.3V / 0V 1-5uS rise/fall time	3.3/2.0V	Failed
F_min V_max	Maximum voltage, Minimum frequency	3.3V / 0V 1-5uS rise/fall time	3.3/2.0V	Failed
F_max V_max	Maximum voltage, Maximum frequency	3.3V / 0V 1-5uS rise/fall time	3.3/2.0V	Failed

Table 7a- digital input tests

The optocoupler, *OK2* on the RPM input is not able to pull the voltage towards 0V, due to a too small pullup-resistor on 220[ohm]. This will be modified on the prototype PCB.

Following measurements are performed on the digital outputs:

Test	condition	expected	measured	result
On	Nominal voltage, resistive load connected	Vin across load	Vin - 200[mV]	OK, small voltage drop caused by fuse resistance
Off	Nominal voltage, resistive load connected	No current through load	0 [mA]	OK
Short	Output on, output shorted (Vin and Output 3 shorted)	Output turns off, can be retriggered	Output turns off when temperature increases, Current is limited to 6[A], due to power supply	OK

Table 7b- digital output tests

The digital outputs behave as expected. A more exhaustive test should be performed, to ensure an inductive and capacitive load can be driven properly. The outputs are intended for driving a signal lamp, and the signal lamps load characteristics should be included in further testing.

4.4.5. Mechanical considerations

The circuit-board is designed as a two layer design, to be mounted as a *shield*, with the processor platform. This gives a compact sandwich-like construction, and can connect to the processor platforms available IO interfaces for analog, digital and power signals.

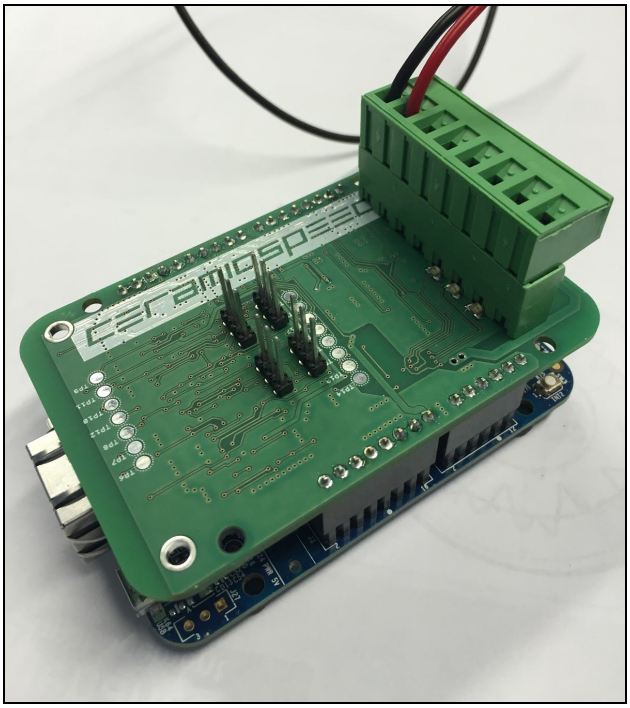


Figure 17 - sandwich construction of the two PCB's

4.4.6. Design for cost and manufacturing

The board is designed with ease of manufacturing, and low cost in mind.
The cost of a producing a circuit board is made up of several factors, listed below:

1. Raw circuit board (PCB) cost, depends on
    a. Number of ordered units
    b. Number of active signal layers to route signals, layers can be from 1 to 16
    c. PCB finish (text, solder plating, copper thickness, minimum geometry)
2. Component costs are affected by
    a. Order quantity, bigger purchase = lower price. Many low cost devices have a minimum order quantity (MOQ) of 1000 or 5000, so you cannot buy 5 or 10, unless you find a reseller, that charges additional cost for stocking and handling.
    b. Component tolerances, typically 1%,5%,10% version exists, more precise = more expensive

    c. Number of variants used, if you can reuse the same component type multiple times in your design, it is easier to meet the MOQ
3. Mounting costs, the process of fitting and soldering components on the board, should consider
    a. One or two component layers. Mounting components on two layers requires components on one layer to be glued, before soldering, otherwise they will fall off when being reflow soldered.
    b. Tooling cost, what tools and how many process steps are required to prepare the board for mounting
    c. How many different components are used. If using many different parts, a lot of components will have to be installed in the mounting machine.
    d. Use dual or quad. components with multiple instances of the same component in the same package, to reduce the number of mounting “pick-and-place” sequences.
4. Design for test
    a. Design the circuit board with testability in mind. Repair is easier when the product is not completed
    b. Use test pads, accessible from automated test equipment, this will be cheaper than manual connection and verification.

4.4.7. Board assembly

10 professional manufactured raw *FR4 circuit boards*, and a stencil for applying solder-paste is ordered. Two boards are assembled for circuit and components verification.

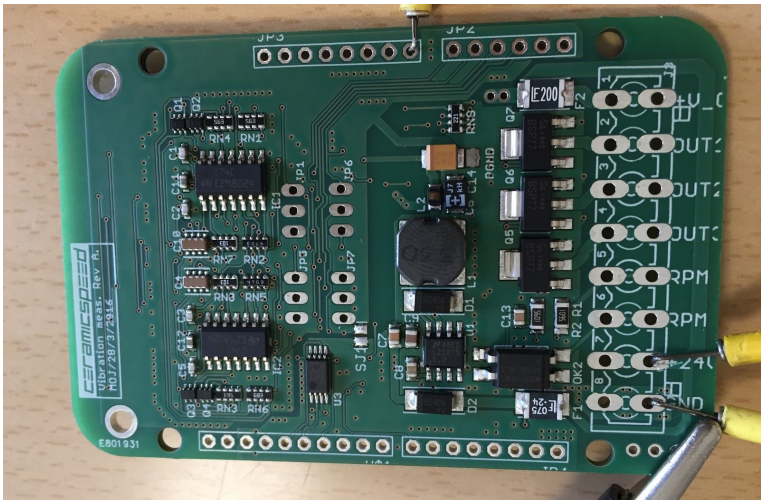


Figure 18 - assembled circuit board

4.4.8. Estimated costs

Appendix n reveals the schematics, components and detailed cost of the design based on sourcing low-volume components (1000 pce. prices), from a large components reseller. Therefore the resulting cost of should be seen as a rough estimate, for low volume production. The actual assembly cost of the PCB is not included in the estimate, since it requires establishing initial relations with a volume producer of OEM products, and is beyond the scope of this project.

Element	***Cost DKK, excl VAT***	***Comment***
4 channel Measurement front end	*96*	*Based on 1000 pce prices from farnell.com*
4 Accelerometers	*4x124*	*Based on inquiry from supplier, 1000-4999 pce.*
Enclosure, mounting screws	*50*	
K64F processor platform	*236*	*Based on single item price*
Sum	*878*	*Assembly cost missing*

Table 8 - current cost of the prototype edge-node

A target price of app 1000 DKK seems realistic, if producing some hundred devices.

4.5. Sampling system

The sampling system in the vibration monitoring systems has the responsibility for ensuring precise digitization of the measured acceleration signals from the analog front end.
The sampling system is composed into of the following functional blocks, each responsible for a part of the systems functionality.

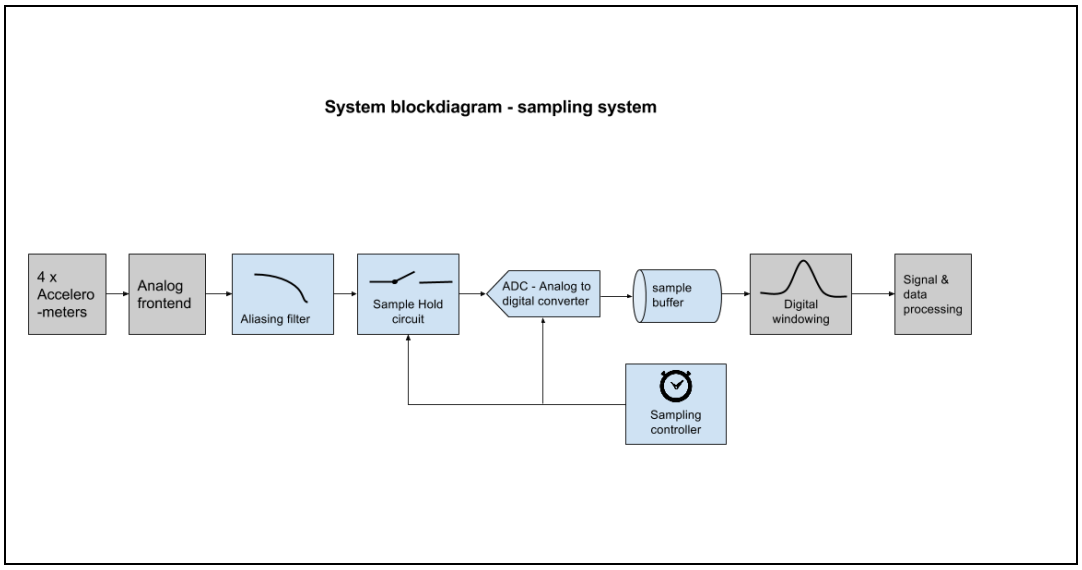


Figure 19 – sampling system blocks (blue) and interfaced functional blocks (dark grey)

4.5.1. Design requirements

The functionality of the sampling system will be based on following design requirements

4.5.1.1. System requirements

The sampling system must be:

1. able to measure analog inputs from 4 analog channels
2. each analog channel has a range of 0..5 [V]
3. able to measure signals with a frequency range of 1[Hz] to 10[KHz]

4.5.1.2. Non functional requirements

1. The sampling system should be able to run at full speed, causing as little as possible load to the remaining system, if possible, by utilizing hardware acceleration features in the processor.
2. The sampling system should be able to run deterministic, with a fixed sampling frequency, with as little as possible jitter, to ensure accurate sampling.

4.5.2. Sampling system design

The vibrations monitoring system is build around the *K64F processor* processor that has a build in *Analog To digital Converter – ADC [19]*

The selection of the K64F family processor family is partly affected by the ADC's specifications, in particular following key specifications is of interest

- *Linear successive approximation algorithm with up to 16-bit resolution*
- *Up to four pairs of differential and 24 single-ended external analog inputs*
- *Single or continuous conversion, that is, automatic return to idle after single conversion*
- *Configurable sample time and conversion speed/power*
- *Conversion complete/hardware average complete flag and interrupt*
- *Operation in low-power modes for lower noise*
- *Hardware average function*
- *Selectable voltage reference: external or alternate*
- *Self-Calibration mode*

The device has two independent ADC's, with up to 16 bits resolution, allow for high accuracy sampling. Furthermore the ADC can be configured to run with a fixed sample rate, above the desired sample rate and perform hardware averaging, that is the resulting sample will be an average of the previous 4,8,16 or 32 samples, obtained by the ADC.
The ADC's available conversion speeds are outlined in table 9:

Conversion mode	*Max. clock [MHz]*	*Conversion rate [samples/sec]*
≤13 bit	*18.0*	*818.330*
16 bit	*12.0*	*461.467*

Table 9 – ADC conversion rates

With two independent ADC's, it is possible to achieve double the sampling capability, as the table 9 outlines.
Since we are to measure 4 analog channels, the effective maximum sample rate is equal to approximately half the available conversion-rates, assuming that the selection of different channels is not consuming significant resources.

4.5.3. Analog signal conversion

In the vibration monitoring system, the initial design will try to utilize the ADC fully, i.e. collecting 16-bit samples at full speed, to verify the signal path, and capabilities of the system.

Based on the system specification we want to find the maximum reasonable sampling frequency for each of the 4 channels in the system. Each ADC has to sample from two channels.

At 16 bit mode we can sample 461.467[sps], or *samples per second*, thus app. 230[Ksps] per channel. The *Nyquist-Shannon[ref]* sampling theorem states that we have to sample any signal of interest at least twice the frequency of interest.

If we utilize the ADC's ability to do hardware averaging, electrical noise present in the signal will be minimized by the averaging.

resolution [bits]	**Sample rate pr. channel [Ksps]**	**HW avg. setting**	**F_{SAMPLE} [KHz]**	**F_{IN_max} [KHz]**
16	230	4	57.5	28.8
16	230	8	28.8	14.4
13	409	8	51	25.5

Table 10 - ADC sampling & resolution capabilities

4.5.3.1. Quantization and resolution

The process of digitizing an analog signal involves an analog to digital conversion, by which the analog signal of interest is quantized into a finite number of intervals. Digital representation of an analog signal will introduce an error, known as *quantization error*, this is the difference between the *actual* signal and its *digital quantized* representation.

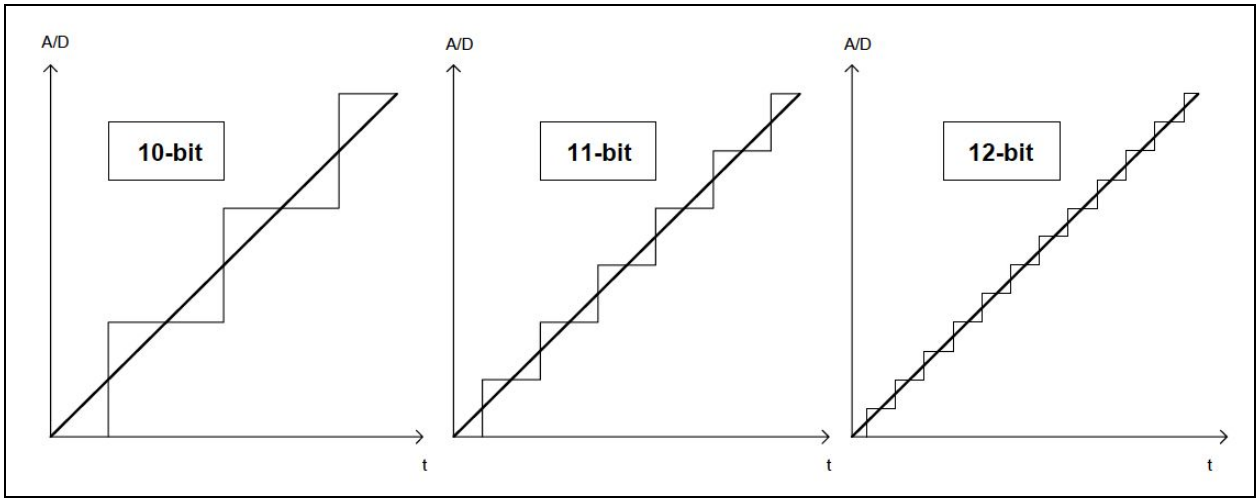


Figure 20 - increasing conversion resolution

Thus from figure 20, it can be seen that the higher resolution the less quantization error will appear. A higher resolution converter will typically have a longer *conversion time*.
The given resolution of an *N* bit analog to digital converter is described by the equation

$$resolution\ [V] = \frac{Vref[V]}{2^N - 1}$$

Vref is the reference voltage supplied to the converter, often identical to the systems supply voltage.

Thus the achievable resolution of a 16 bit converter, supplied by a 3.3V reference voltage is

$$3.3/((2^{16})-1) = 50{,}4[uV]$$

The resolution is also known as V_{LSB}, representing the voltage weight of one bit, in the digital system. Any analog to digital converter cannot detect changes below +/- 0.5 V_{LSB}.
In order to utilize the converters full dynamic range, and achieve best possible accuracy in signal representation, it is desired to use the full range of the converters range.
In the example above, that is ensuring that whatever is generating the signal of interest, will utilize the full range from 0 to 3.3[V]

4.5.3.2. Oversampling

Oversampling is basically reading "the same" value, multiple times and then using the average of the readings as result. When measuring real world analog signals there will be noise, distorting the signal. The noise signal will be a sum of different sources, all contributing a bit of "random" signals, like thermal noise, processor clock frequency, IO ports and data communication, as well as varying supply voltage. Oversampling is done to achieve a better digital representation of an analog signal. In order for oversampling to have any effect, the impact of the noise sources must exceed 0.5 LSB, in order for the ADC to measure it. That also means, the better resolution (higher N), the finer details can be measured, and the more impact will be caused by the noise.
Performing oversampling requires the sampling system to be fast enough to perform the required amount of samples per second, multiple times, and averaging the result.
Doubling the sampling frequency on a given signal, and averaging the signal will give a reduction in the *in-band* noise of 3dB, and an effective increase in the resolution by 0.5 bits.

The Freescale K64 processor has a build in hardware oversampling and averaging feature, capable of averaging 2/4/8 or 32 times, that will be utilized.

4.5.3.3. Aliasing

When sampling a signal of interest, one must obey the Nyquist criterion, stating that the sample-rate or Sampling frequency must be at least twice that of the maximum frequency of the signal of interest (the input signal). Failure to obey the Nyquist criterion, results in a phenomenon called aliasing, illustrated in figure ***21*** below. Aliasing results in false reading, when the signal is interpreted.

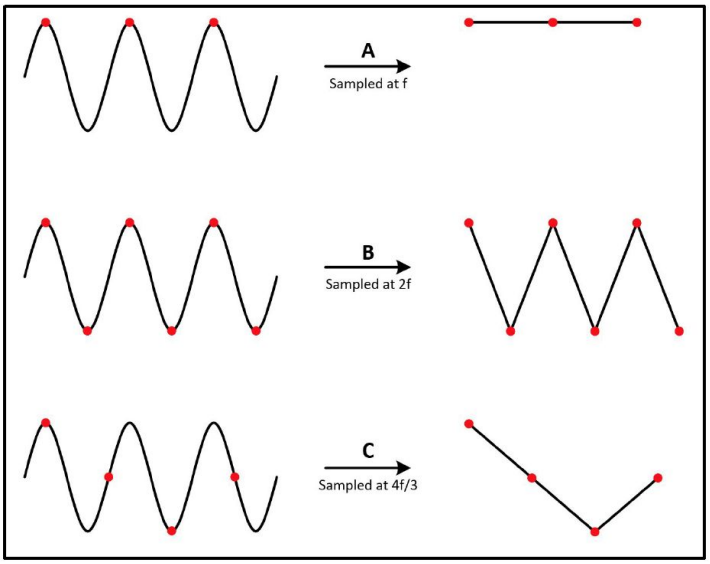


Figure 21 - aliasing when sampling a signal above $F_s/2$

By placing an antialiasing filter before the analog to digital converter, the aliasing phenomenon can be avoided. The accelerometer front end circuit has a build in aliasing filter.

4.5.3.4. Electrical interference

To ensure best possible signal quality, the analog and digital electrical signal paths in the system should be divided into analog and digital signal domains, as well as shielding especially the analog signals by using ground planes, is essential, to minimize impacts from noise sources. This is illustrated in chapter *9.4 Frontend, version 2 design*

4.5.3.5. ADC, sample hold and channel multiplexing

The ADC is the bridge between the analog and the digital domain, in the system.
The K64F's ADC has building Sample Hold circuit, as sell as a sampling sequencer, allowing precise deterministic sample-rates, without continuous interventions required by the system SW.
Furthermore the ADC has build-in channel multiplexing, allowing selection between up to 12 different channels per ADC.
In the vibration monitoring system, the sampling system is to be build from both the ADC's running in parallel, each responsible for sampling two analog channels.

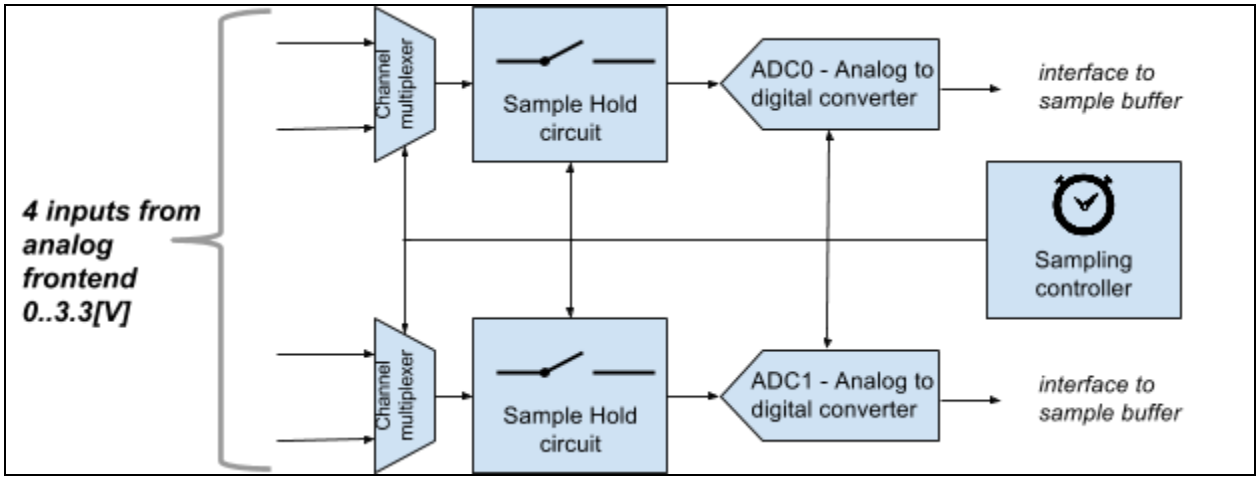


Figure 22 - intended sampling system

9.5.3.5.1 ADC Interface analysis

From the ADC's electrical datasheet following specifications are derived

Symbol	Description	Conditions	Min.	Typ.[1]	Max.	Unit	Notes
V_{DDA}	Supply voltage	Absolute	1.71	—	3.6	V	
ΔV_{DDA}	Supply voltage	Delta to V_{DD} ($V_{DD} - V_{DDA}$)	-100	0	+100	mV	2
ΔV_{SSA}	Ground voltage	Delta to V_{SS} ($V_{SS} - V_{SSA}$)	-100	0	+100	mV	2
V_{REFH}	ADC reference voltage high		1.13	V_{DDA}	V_{DDA}	V	
V_{REFL}	ADC reference voltage low		V_{SSA}	V_{SSA}	V_{SSA}	V	
V_{ADIN}	Input voltage		V_{REFL}	—	V_{REFH}	V	
C_{ADIN}	Input capacitance	• 16-bit mode • 8-bit / 10-bit / 12-bit modes	— —	8 4	10 5	pF	
R_{ADIN}	Input series resistance		—	2	5	kΩ	
R_{AS}	Analog source resistance (external)	13-bit / 12-bit modes f_{ADCK} < 4 MHz	—	—	5	kΩ	3

Table 11 - ADC electrical specifications

V_{DDA} is determined by the system constraints, and is a 3.3[V] regulated voltage. Thus the possible measurement range V_{ADIN}, for the system is restricted to 0..3.3[V] +/- 100[mV]

The converted voltage results are read as an unsigned 16 bit number from the ADC blocks result register, corresponding to the active ADC input channel.

Conversion mode	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Format
16-bit differential	S	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	Signed 2's complement
16-bit single-ended	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	Unsigned right justified
13-bit differential	S	S	S	S	D	D	D	D	D	D	D	D	D	D	D	D	Sign-extended 2's complement
12-bit single-ended	0	0	0	0	D	D	D	D	D	D	D	D	D	D	D	D	Unsigned right-justified
11-bit differential	S	S	S	S	S	S	D	D	D	D	D	D	D	D	D	D	Sign-extended 2's complement
10-bit single-ended	0	0	0	0	0	0	D	D	D	D	D	D	D	D	D	D	Unsigned right-justified

Figure 23 - ADC result register layout

When a new conversion result is ready in the register, the processor will be notified by an *interrupt*, triggering an *interrupt service routine - ISR,* in the sampling software.

9.5.3.5.2 Functionality

The K64F's ADC with its functional blocks is shown in figure 24 and the intended signal-path is highlighted.

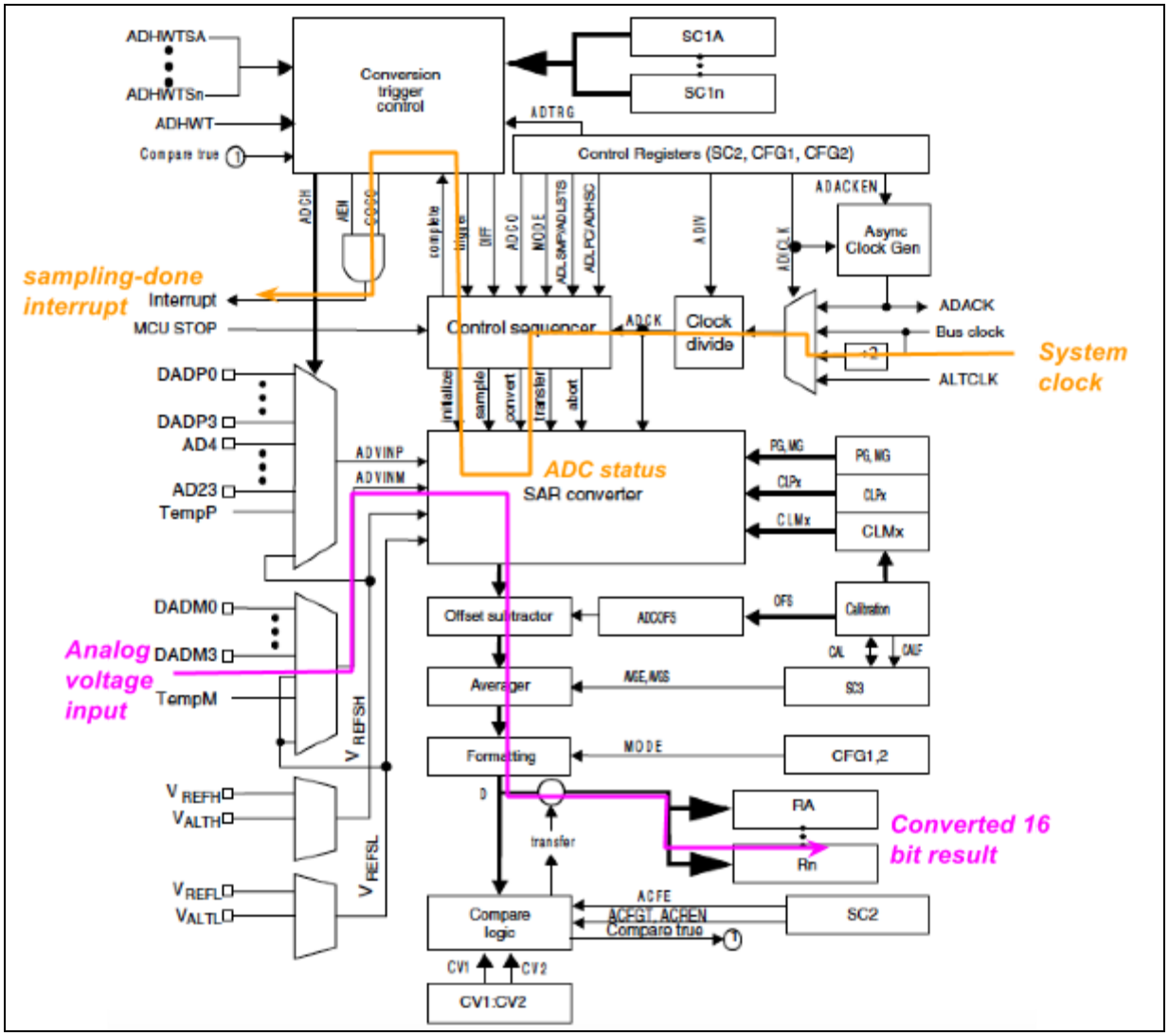


Figure 24 - ADC block diagram and datapaths

The SAR converter is the central component, performing the actual signal digitization, and presenting the result to the processor via a register in the processor's memory map.

4.5.3.6. Signal multiplexer and sample/hold

The input multiplexer combined with the sample hold circuit is intended to “select and hold” the desired signal, for the duration of the sampling process.
For each input channel on the ADC the circuit can be interpreted as shown in figure ***25***

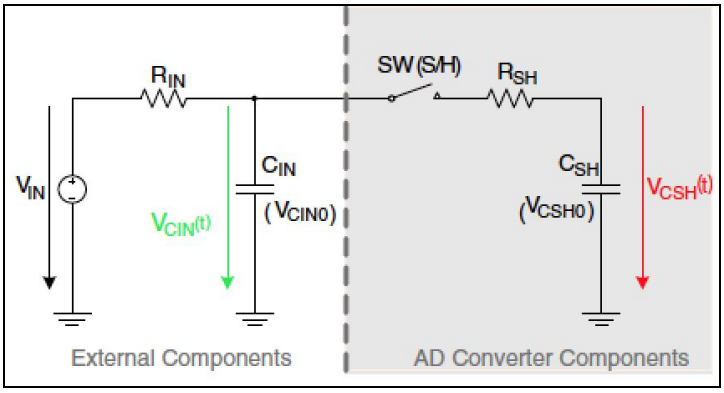


Figure 25 - equivalent diagram of one input channel

The *"AD Converter Components"* are the equivalent internal Capacitor and series resistor inside the ADC. R_{SH} and C_{SH} can be found as R_{AS} and C_{AS} in table N, derived from [19].

Parameter	Value
R_{AS}	5[KΩ]
C_{AS}	10[pF] (16 bit mode) 5 [pF] (8/10/12 bit mode)

Table 12 - The ADC's equivalent RC input circuits practical values

The sample hold switch ***SW***, will be closed, shortly, to charge the C_{SH} capacitor, and opened again, before sampling in the ADC is started. Due to the finite sizes of capacitors and resistors, the sample hold circuit will be charged over time, and approximate the input voltage over time. Thus the sampling time should be low enough.

The external components R_{IN} and C_{IN} can be calculated as equivalents from the analog front end. The thing to pay special attention to, around a circuit like this, is the time of sampling versus if the voltage across the *Sample Hold Capacitor* has stabilized. When the switch *SW* closes, the C_{SH} capacitor will be charged from the external circuit, the analog front end in this case. During charge of a series circuit with a resistor and capacitor, a waveform as shown in figure 26 will be present across the capacitor:

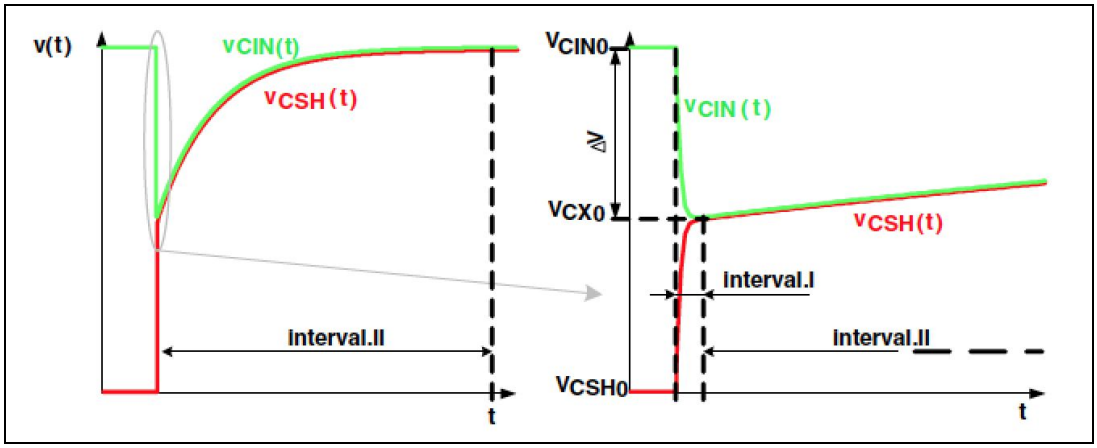


figure 26, voltage across the sample hold circuit

Since we are using the ADC's for measuring on two channels, both in the range 0..3,3[V], there is the risk that the voltage change will be from 0 to 3,3[V] between two successive samples, of the two different channels, this also means the voltage-change across the sample hold capacitor will have to be 3,3[V].

The voltage across the capacitor C_{SH}, as a function of time, in the sample hold circuit can be found as :

$$V_{CSH}(t) = (V_{CIN0} - V_{CSH0}) * \frac{\alpha}{\alpha+1} * (1 - e^{\frac{t}{\tau I}}) + V_{CSH0} \qquad (eq1)$$

Where

V_{CSH0} : *Initial voltage across the Sample Hold capacitor*

V_{CIN0} : *Initial voltage across the external capacitor*

$\tau I : R_{SH} * (C_{SH} + C_{IN})$

$\alpha = \frac{C_{IN}}{C_{SH}}$

The size of the equivalent *Rin* in figure 25 is a few tens of ohms, in the given application, and can be neglected. This allows a simplification of the expression

$$V_{CSH}(t) = (V_{IN}) * (1 - e^{\frac{-t}{RSH*CSH}}) \qquad (eq\ 2)$$

So we now can see the sample/hold circuit as a simple RC circuit. Due to the exponential term in *eq2*, the voltage V_{CSH} will only approximate V_{IN} over time. However the time constant τ given by $\tau = R_{SH} * C_{SH}$ allows us to approximate a full charge after 5τ .

To verify we solve and insert into *eq2* for the worst case condition, where the voltage difference is a full 3.3V gives us :

$\tau = R_{AS} * C_{AS} \Rightarrow \tau = 5K\Omega * 10pF \Rightarrow \tau = 50[nS]$

$V_{CSH}(5\tau) = 0,993 * V_{IN}$

Thus a Sample Hold time of 50[nS] in the controller software, will result in the SH circuit to be charged to minimum 99,3% of the voltage.

4.5.4. Design - Sampling controller

The sampling controller software is a combined timer and sequencer, responsible for taking samples on the correct channels at the right time.
Furthermore the sampling controller handles data organisation, so the number of samples required for performing Frequency analysis, FFT are ordered in memory buffers, that can be processed further, and sent via ethernet to the processing computer.

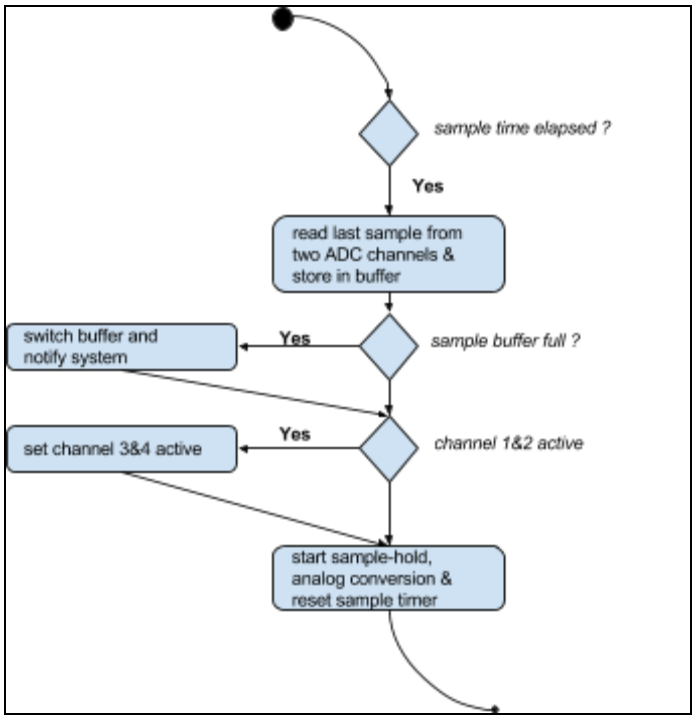


Figure 27 - sampling controller functionality

The sampling controller is implemented by means of a *periodic interrupt timer* - PIT, available in the K64 processor. The PIT timer is programmed with a prescaler value, determining the relation between the processor's system clock, and the desired time interval (the *sample frequency*) of the sampling system, every time the timer elapses, an interrupt is generated, triggering the sample routine.

The data generated by the sample controller are stored in sample buffers, one per channel. For each channel two sample buffers are created, one “active” for storing the current samples, and one “ready” for further processing and transmission.

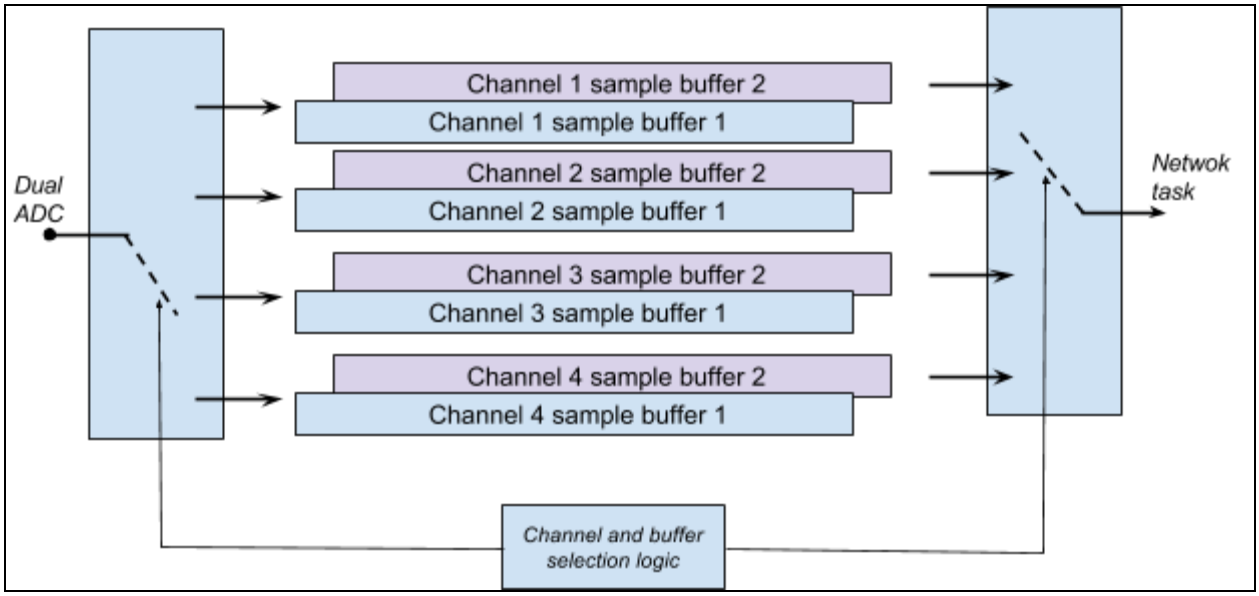


Figure 28 - sample buffers in the sample controller

The size of the sample buffers are determined by the required data length of the *FFT* function, to perform the frequency analysis, of the sampled signal. FFT’s always work of datasets organized in 2^n sizes, typically 1024, 2048, 4096, 8192 or 16384 samples.

The K64 processor has 256 KB RAM embedded on the chip. The LwIP stack for network communication is expected to take up 20-40 KB, and the remaining application, depending on the features implemented, might use 10-20 KB.
Performing FFT’s of 8192 (2) samples per channel, will require following available memory

$$MEM_{8Ksamples=} = NoChannels * Buff_{size} * bytes_{per\,sample} * 2 \Rightarrow 4 * 8192 * 2 * 2 = 131.072 \text{ [bytes]}$$

This accounts for roughly half the device’s available memory, and should be seen as the allowable maximum, if all 4 channels are to run simultaneously in parallel.

4.5.5. Performance estimations

The data collection systems design requirements dictates that 10KHz or more should be measured. The accelerometer is capable of measuring up to 20KHz.
The initial prototype (hardware & software) will be designed to run the sampling controller interrupts at 40KHz. Two channels can be read in parallel, and on every interrupt, a switch between *even (2+4)* and *odd(1+3)* channels is done.

With sampling running at 40KHz and data-frames are being transmitted from the device, performance will be measured to assess whether a higher sample rate is feasible.

4.5.6. Verification and test

A number of the functional tests has been performed, and are available as videoclips

Accelerometer front end, measured w spectrum analyzer, compared with IFM system
The orange curve is from the accelerometer amplifier, measured with a spectrum analyzer
The green curve is a vibration monitoring systems from IFM.
https://youtu.be/mUkxko6il8o
From this video it can be seen that the achievable accuracy from the ACH01 accelerometer seems comparable to the commercial system from IFM. The spectrum analyzer used, is an analog discovery[54] capable of sampling at 100[Msamples/sec], 50.000 times faster than the designed edge node.

4 Channel prototype vs IFM system:
Two tests are filmed, with bearings mounted in the testbench. Idealy data from both sources (the prototype and the IFM) would be combined in the same plot. This is not currently possible, and would require access to the data-communication protocol for the IFM system.

The test setup:
The tests are performed on a bearing test-rig[55]. The bearings are mounted with outer-rings permanent, and inner rings rotating. An IFM VSA001[56] accelerometer and the prototypes ACH01[14] accelerometer is fitted to the bearing mount, shown in figure 29.

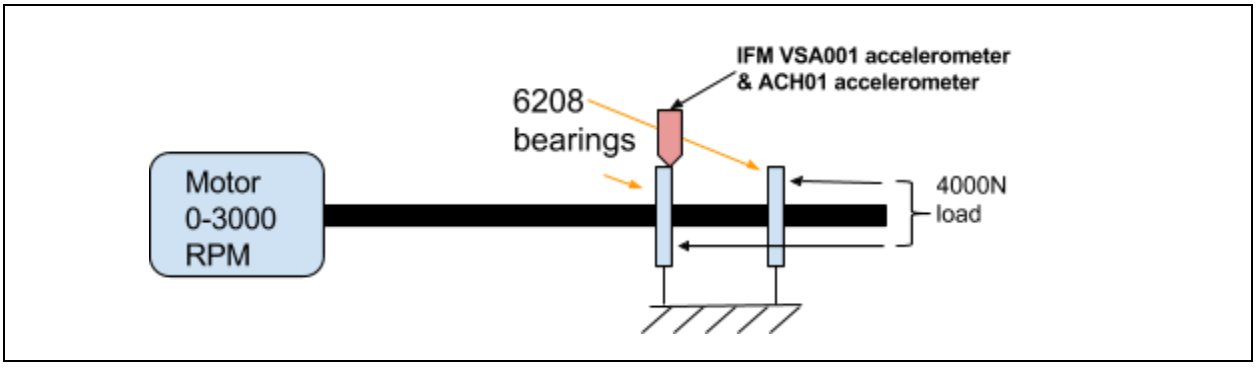


Figure 29 - bearing test-rig, and accelerometer (highlighted with yellow circle)

Two tests are performed, and a visual comparison is made.
Test1 : 0-700 Hz:
https://youtu.be/eZ-xl1lzcEc
In this test the frequency spectrum is zoomed in to 0-700 Hz, to allow looking at the characteristic bearing frequencies. The motor runs at a speed of 3000RPM (50Hz), and the 50[Hz] spacing can be seen in both frequency spectrums. The characteristics of both frequency spectrums are identical, thus the system is capable of detecting the same signal as the IFM system used for reference
The slightly more narrow frequency amplitudes, seen on the IFM system, is likely to be caused by a finer frequency resolution in the IFM.
The 0-700[Hz] test also reveals that some low frequency noise is present on the prototype. This can be due to aliasing, and should be investigated further.

Test1 : 0-10.000 Hz:
https://youtu.be/_1dMgg6-tNk
It can be seen that the lowest frequencies in the spectrum, below 6 KHz are more or less identical. The frequency spectrum between 6-10[KHz] contain much more energy, in the measurements on the prototype. This can be caused by two things:

1. The VSA001 accelerometers used on the IFM has a measurement bandwidth of 0-6[KHz], where the ACH01 used on the prototype can a measurement bandwidth of 2[Hz]-20[KHz]
2. Electrical noise from the frequency-converter, to the motor can cause noise, in this band.

4.5.6.1. Summary

Appendix 7 performs a simple visual comparison of the obtained measurements, compared to the IFM system. The achieved performance from the edge node is a result of the different signal processing-elements in the signal-chain, figure[19].
The achieved accuracy of the Frequency analysis-tests is comparable to the IFM system, but a more thorough comparison could be ideal, where both data-sources are included and evaluated in the same graph.
This overall indicates that all functional blocks, in the signal chain, after the accelerometer amplifier does perform satisfactory. The sampling systems overall performance could likely be improved, by optimizing the digitizing blocks, and increasing the size of the sample buffers, to increase accuracy of the measurements.

4.6. Network and connectivity

The build prototype is expected to be able to send collected data via TCP/IP on the wired ethernet interface, populated on the processor platform. For the initial experiments i will make use of the available *TCP/IP stack*, *LwIP[28],* including examples, that has been ported to the K64F processors ethernet controller.

The wealth of connectivity protocols, and available wired and wireless interfaces, that has emerged with the booming number of IoT devices and development solutions, presents a solution space to vast to explore, in a subchapter, as part of this project.
Data communication in the prototype, is based on *socket* communication on a simple TCP server example, from the *LwIP protocol stack* examples. The source code for the *LwIP* network-stack communication is included, but is not presented in detail here.

Due to the lack of focus devoted to optimizing network and connectivity features, in the project, the prototype device is implemented with a fixed IP address, and a fixed port-number, thus the client application communicating with the device, must be *hard-coded* with this information.

4.6.1. Communication protocol

The communication between the *Sense/Collect/Connect* device and the host-computer uses sockets. A socket can be perceived as a *communication-pipeline* between computers on each end.
When a network session is created a connection to a remote computer via a *network socket*, provided by the *transport layer* in the LwIP stack. A socket can provide a TCP or UDP based connection, depending on the requirements of the application, in this case TCP is selected, due to emphasis on data integrity.

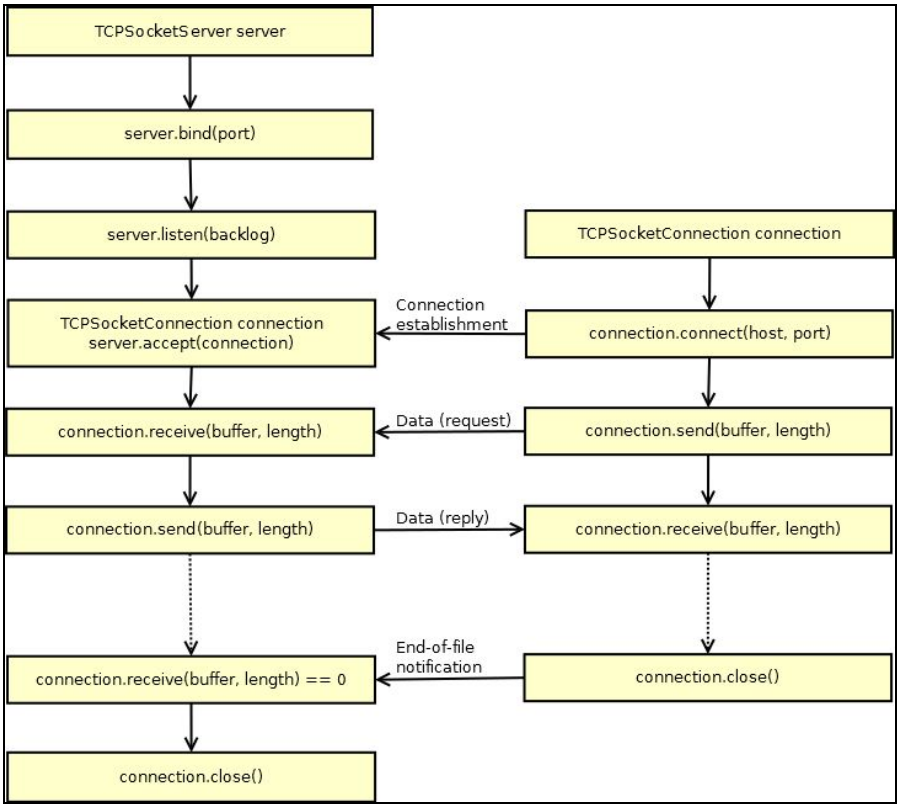


Figure 30 - lifecycle of a client/server system, connected via sockets

Once the communication has been established, the *server* runs on the measuring device, and the client is started on the host PC, or what may later become the cloud-solution, hosting the analysis-application. Data-exchange between client and server is done via a very simple protocol, allowing following interactions

client	type	server	type
<GET_SAMPLES>	request	<4x2x8K byte sampled raw data+timestamp>	response
<GET_RPM>	request	<2 bytes rpm data>	response

Table 13 - initial communication protocol

The raw data samples are sent as bytes through the *socket*, 2 bytes are required for each 16 bit sample, read from the ADC, and data from 4 channels are to be sent, resulting in 64 KBytes data packages.

4.7. Prototype conclusions

A functional prototype has been build, capable of measuring and digitizing the analog signals from the attached accelerometers is successfully build.
The achieved performance is not currently as good as the commercial IFM. The accelerometer amplifier seems suited, and with a sufficient dynamic to measure, at least vibrations, on the testbench for accelerated bearing lifetime-test.
It is believed that the sampling system, implemented in the initial prototype, can be improved, in order to get a better readout. Following improvements could be examined:

1. Change the sample-buffer layout. By sampling one single channels at the time, in the sequencer, instead of the current 4 channels, with shadow buffers, one could store 8 times more samples, resulting in a N= 64K *[KiloSample]* buffer.
2. Increase the sample-rate, this would allow a finer grained acquisition, and combined with ***1***, this would result in a finer frequency resolution (F_s/N)
3. Investigate a steeper aliasing filter. This might reduce the noise contribution, in the signal additionally.
4. Build a more comprehensive data-exchange protocol, to ensure data is sampled and transmitted in real time. Currently some delay is experienced, from the edge-node to analysis. Ideally the data visualization is done in real-time.

5. Data analysis and visualization

To make any use of the collected data, from the *sense/collect/connect* device, further postprocessing must be performed. The IIoT stack, in table 14, is used to frame the functionality in the prototype system to build. Layers 4 and 5 contains the required functionality to *retrieve, organize, store, analyze* and *visualize* the data from the ***sense*** layer.

Layer	Functionality	
6:Act	Process and tools for taking action based in the insights	
5:Analyze	**Capabilities for data to develop insight**	
4:Store	**Systems for storing and organizing data**	
3:Connect	Physical connection to pass data to the cloud / central system	
2:Collect	Edge hardware and software for gathering and formatting sensor data	
1:Sense	Sensors and endpoints digitizing physical world quantities into data	

Table 14 : IIoT machine monitoring "stack"

5.1. Functional requirements

The emphasis of functionality in the prototype is outlined in chapter 7.4 *Functionality to implement in the initial prototype.* The following chapters divides the functionality into three groups, *data-storage*, *data-analysis*, and *data-visualization*.

5.1.1. Data-storage requirements

The data storage must be capable of :

1. Maintaining two adjacent csv format files, for storing sampled and FFT data respectively
    a. Each file must be stored in the format ***FFT_HHMMDDMMYYYY.csv, RAW_HHMMDDMMYYYY.csv***
    b. Each entry received from the edge node must be saved with a timestamp in millisecond resolution
    c. Each file must contain a number of seconds data, *the duration*, that can be changed in the logging system
    d. All files must contain a header on the first line describing
        i. Samplerate
        ii. No. of channels sampled
        iii. IP address of the edge-node

iv. MAC address of the edge-node
v. Serial no. of the edge-node
vi. Sample time duration

Example:

<SR=40000><CH=4><IP=192.168.11.2><MAC=00:AA:BB:11:22><SER:1234561234><DUR=3600>

2. The Raw data 16 bit sampled values:
    a. Must be saved in as blocks of consecutive samples, from each channel, as follows

<timestamp><channel1 samples 0...N-1>
<channel2 samples 0...N-1>
<channel3 samples 0...N-1>
<channel4 samples 0...N-1>

3. The FFT data calculated:
    a. Must be saved in as blocks of consecutive samples, from each channel, as follows

<timestamp><channel1 samples 0...N-1>
<channel2 samples 0...N-1>
<channel3 samples 0...N-1>
<channel4 samples 0...N-1>

5.1.2. Data-analysis requirements

The data analysis must implement following functions

1. Data windowing, to avoid discontinuities, by providing a suited windowing function
2. FFT (Fast Fourier Transform) on the windowed data.
    a. Number of samples are to be 1024/2048/4096/8192/16384
    b. Ability to detect frequency changes of +/2% across the entire measurement range.
3. Averaging of the FFT over a configurable number of FFT periods shall be possible
4. Detection of thresholds
    a. A number of frequency intervals shall be programmable
    b. For each interval a threshold time shall be programmable
    c. Two threshold values (warning and critical) shall be programmable
    d. When the FFT value (raw or averaged according req 3) exceeds the programmed threshold a given alarm function shall be triggered

5.1.3. Data-visualization requirements

Data visualization is to fulfill following requirements

1. Visualization shall be browser-based
2. FFT visualization shall be displayed in an X/Y graph
    a. Frequency shall be visualized on the X axis
    b. Vibration Magnitude shall be on the Y axis
3. It shall be possible to zoom in in given frequency ranges
4. Historical data shall be available

a. displayed either as a *3D plot* a *spectrogram,* or other suited visualization.

5.1.4. Available functions via Python

"hey, it comes with batteries included", Quote from a python programmer, about the included python standard library,

Python will be the programming language for implementing the functions in the upper layers of the prototypes. Python is a versatile and high level interpreted programming language, that requires an interpreter to run. Python programs does not require compilation, but are translated as they are run on their host machine. Python interpreters are available for all popular operating systems, some even run on small embedded devices. Furthermore python interpreters are available on a lot lot of cloud-platform, which makes it suitable for application that needs to be scaled and deployed via a cloud solution.
The python standard library is quite comprehensive and includes file handling, network libraries, and math functions [32], suitable for implementing the prototype functions in this project.

5.1.4.1. Network library

From the included network utilities in the python, the ***socket*** library for accessing low level network interface functions will be utilized. This allow an efficient implementation of a *TCP client*, capable of retrieving data from the edge-node, that is acting as *TCP host*.

5.1.4.2. File handling

The standard library has functions included for working on *csv* (comma separated files), which will be utilized for simple data storage in the prototype.

5.1.4.3. Scientific computing library

The ***scipy*** libraries, for scientific computing, contains a number of packages, one of theses being ***numpy*** a library for numerical computing, and functions, such as Fourier transform, and data-windowing. We need the Fast Fourier variant FFT, available in the library for performing frequency analysis. Numpy is not part of the standard library, and needs to be downloaded and imported on the host machine.

5.1.4.4. Data visualization

A number of python-compatible libraries for data visualization exists. Since the prototype is targeted for deployment on a scalable cloud infrastructure, a browser based interaction would be prefered, hereby users of the system will avoid having to install an *application* on whatever device they are interfacing the system through. ***Bokeh****[32]* is a popular data visualization framework, that can be customized to visualize in a lot of novel and elegant ways. Bokeh plots are displayed in an internet browser, that would be expected available on any interfacing device. The *plotting-engine* of bokeh can be run as a standalone server, interfacing with web-clients, which seems like a good way of scaling up and deploying on a hosted platform.

5.2. Data analysis algorithms

This chapter presents the mathematical background, and the utilized functions necessary to perform frequency analysis. The frequency analysis is based on fourier transformation. Fourier transformation decomposes a time-domain signal into the frequencies making the signal up, thus the result will be values in the frequency domain.

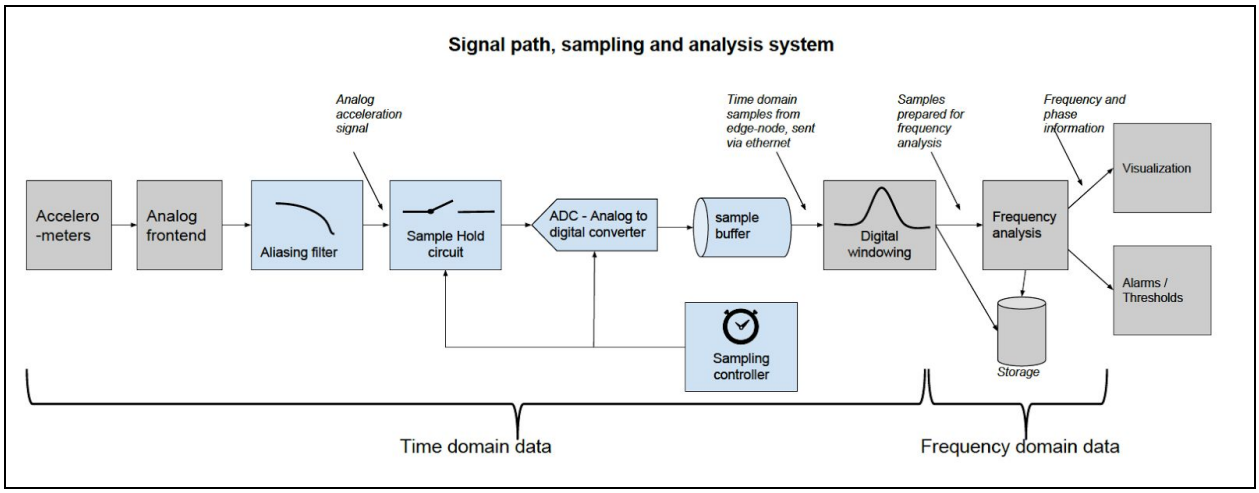


Figure 31 - entire signal path from time to frequency domain

The depicted figure 31 shows the functional blocks, in the signal path from accelerometer, leftmost, to the visualization and alarm handling on the right.
The data-analysis will be performed by the functional blocks digital windowing, and frequency analysis. The implemented functions assumes consecutive blocks of samples, gathered with a fixed sampling frequency

5.2.1. Fourier transform

Is a mathematical conversion, converting a signal from the time-domain into the frequency domain[32]. The fourier transform builds on the property that any periodic signal can be decomposed into a sum of different frequencies, the following animation shows this efficiently:
https://upload.wikimedia.org/wikipedia/commons/7/72/Fourier_transform_time_and_frequency_domains_%28small%29.gif

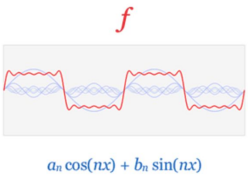
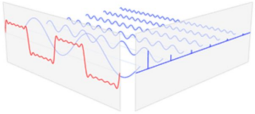
			
A periodic signal in the time domain	*Is made up from a sum of sinewawes with different amplitude and phase*	*When each individual frequency contribution is analyzed*	*The blue spectrum shown the amplitude for each frequency component of the red time-domain signal*

Figure 32 - Fourier transform principle, applied to a time domain signal

The mathematical expression for a fourier series, of a periodic signal of the form :

$$x(t) = a_n * cos(n\omega) + b_n * sin(n\omega)$$

can be described by the continuous time integral :

$$X(\omega) = \int_{-\infty}^{\infty} x(t) * e^{-i\omega t} dt \qquad - \ equation\ 1$$

Where ω *is the angular frequency* $\omega = 2 * \pi * f$

Thus for a *continuous* signal, like the *analog* signals generated by the analog frontend, for the accelerometers, the fourier series will also be a continuous spectrum. The simplified example, in table ***n*** above, shown only six frequency contributions making up the periodic signal, for simplicity.

Once the *analog* signals, in the accelerometer front end has been *digitized* and *quantified,* by the analog to digital converter in the processor, it is no longer *continuous,* but instead *discrete*, or *sampled.* Now *equation 1* no longer applies, and a *Discrete Fourier Transform - DFT* shall be applied instead, to perform frequency analysis. DFT's are implemented by performing *sample-based* calculations, on the discrete samples.

5.2.2. Discrete Fourier Transform

A *Discrete Fourier Transform* on a dataset of ***n*** samples requires $\boldsymbol{n^2}$ complex calculations, thus if 1024 samples are acquired from the *edge-node,* approximately 10^6 calculations is required, for 4096 samples (the selected buffer size for the prototype), approximately $17*10^6$ calculation is required.

5.2.2.1. The Fourier Transform - FFT

The Fast Fourier Transform - FFT is one such DFT function.
The popularity of the *FFT* is based on the fact that FFT calculations on ***n*** samples is accomplished by ***n x log(n)*** calculations. For 1024 samples, this requires *3082* calculations, and for 4096 samples, approximately 15.000 calculation is required.

The performance improvement in calculation-speed, as a function of samples, can be visualized graphically

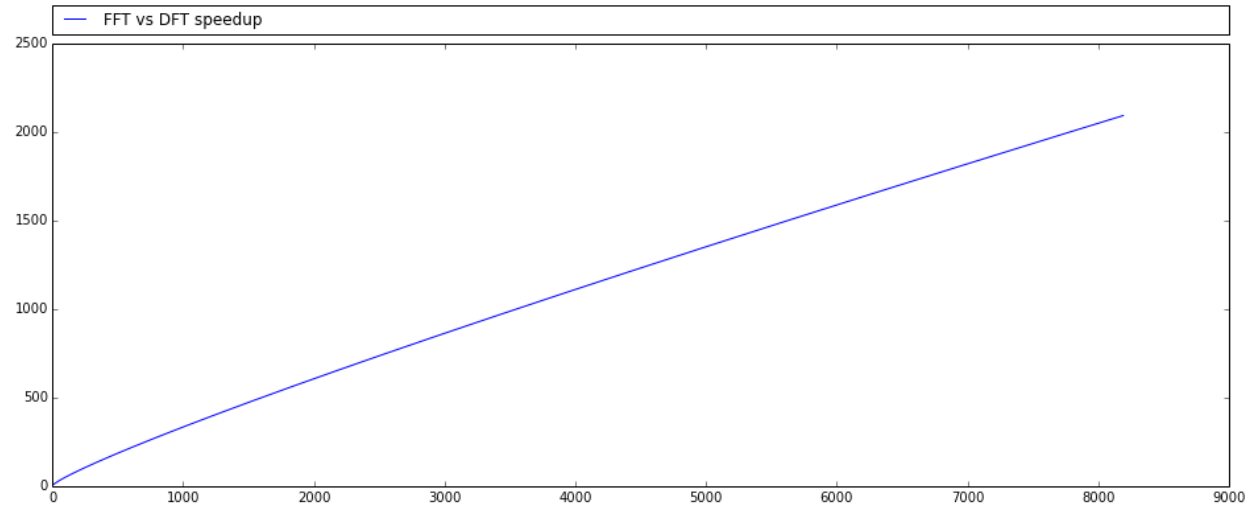


Figure 33 - FFT performance improvement over DFT
(X-axis is sample, Y-axis is improvement, in times the FFT is faster)

Calculation of the FFT is found as the sum calculation, for each sample in the dataset *x[n]*.
Any DFT expression, that performs discrete conversion from time to frequency domain, given by the equation:

$$X[k] = \sum_{n=0}^{N-1} x[n] * e^{i2\pi kn/N},\ n = 0, 1, ...N-1$$

The FFT achieves its performance improvement by an efficient implementation of the DFT, that is reordering the data in the sample block, evaluating the frequency spectra content for each sample point, reversing the sample reordering, performing a complex-number multiplication aka.“butterfly calculation”, and recombining the calculation into a frequency domain.

“Because the discrete Fourier transform separates its input into components that contribute at discrete frequencies, it has a great number of applications in digital signal processing, e.g., for filtering, and in this context the discretized input to the transform is customarily referred to as a signal, which exists in the time domain. The output is called a spectrum or transform and exists in the frequency domain.”[35]

Thus the frequency domain output is what is directly used to assess the severity of vibrations measured from an instrumented machine or device.
Since then frequency spectrum is derived from *discrete samples*, the *frequency spectrum* is also *discrete.* The discrete frequency-spectrum can be perceived as the combined outputs from a set of bandpass filters, each with the bandwidth Δf given by the sample frequency F_s divided by the number of samples N.
$\Delta f = \frac{Fs}{N}$ = This is also known as frequency lines, or frequency-bins.

Practical implications in the prototype:
F_s, the *sampling frequency* is a vital parameter in the systems, since all signals samples are governed by the *nyquist sample criterion*, stating that any signal of interest must be sampled at least twice the maximum frequency of interest, and that frequencies above $\frac{Fs}{2}$ must be filtered by an aliasing filter, to avoid aliasing in the discrete samples. From this we can see that the frequency resolution of an FFT, under the assumption that $\boldsymbol{F_s}$ is fixed, can only be affected by adjusting the number of samples ***N.***
The size of the sample buffers are ***N*2*** *bytes*, for each of the four channels, and governed by the available amount of memory in the processor platform.

5.2.2.2. FFT in Pythons Scientific computing library

The *numpy* library, as part of the scientific libraries for python, has several discrete fourier transform functions, implemented as FFT's available.

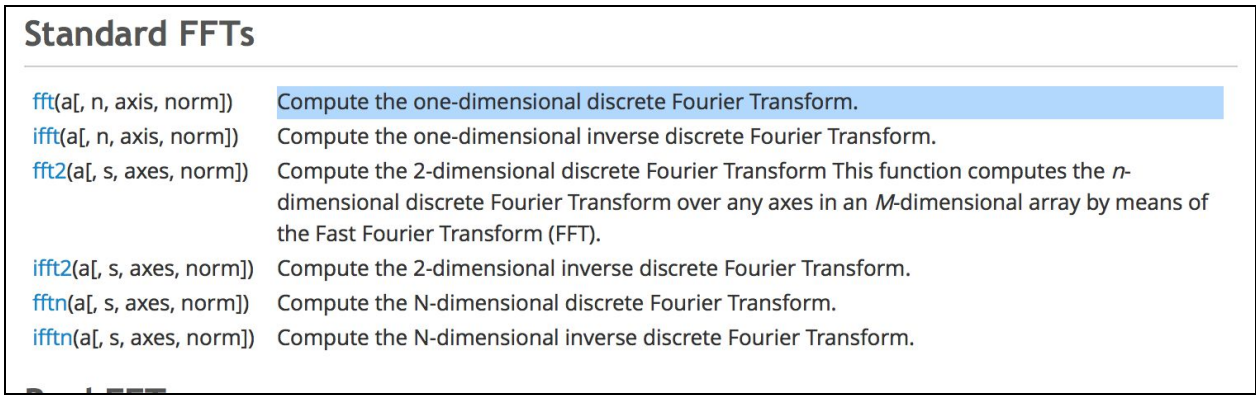


Figure 34 - available Standard FFT functions in numpy[35]

Numpy's Standard FFT's allows forward and inverse FFT's computation on data arrays. For the frequency analysis we will need The *one-dimensional* ***fft*** function.

5.2.2.3. Testbench for initial FFT

A testbench is written in *python* (available as appendix n), in order to asses the ***fft*** functions on a known simulated dataset. The testbench works on a dataset of ***N=4096***, similar to the initial prototype, and with a sampling rate $\boldsymbol{F_s}$ of ***20[KHz]***.

A number of test signals is generated to simulate working conditions, where a sensor might be deployed.

1. 300 Hz sine wave, amplitude 1.0
2. Simple signal, a sum of two sine wave of 300[Hz] with amplitude 1.0 and 500[Hz] with amplitude 0.5
3. Test-signal 2, added random noise, with an amplitude between 0 and 1

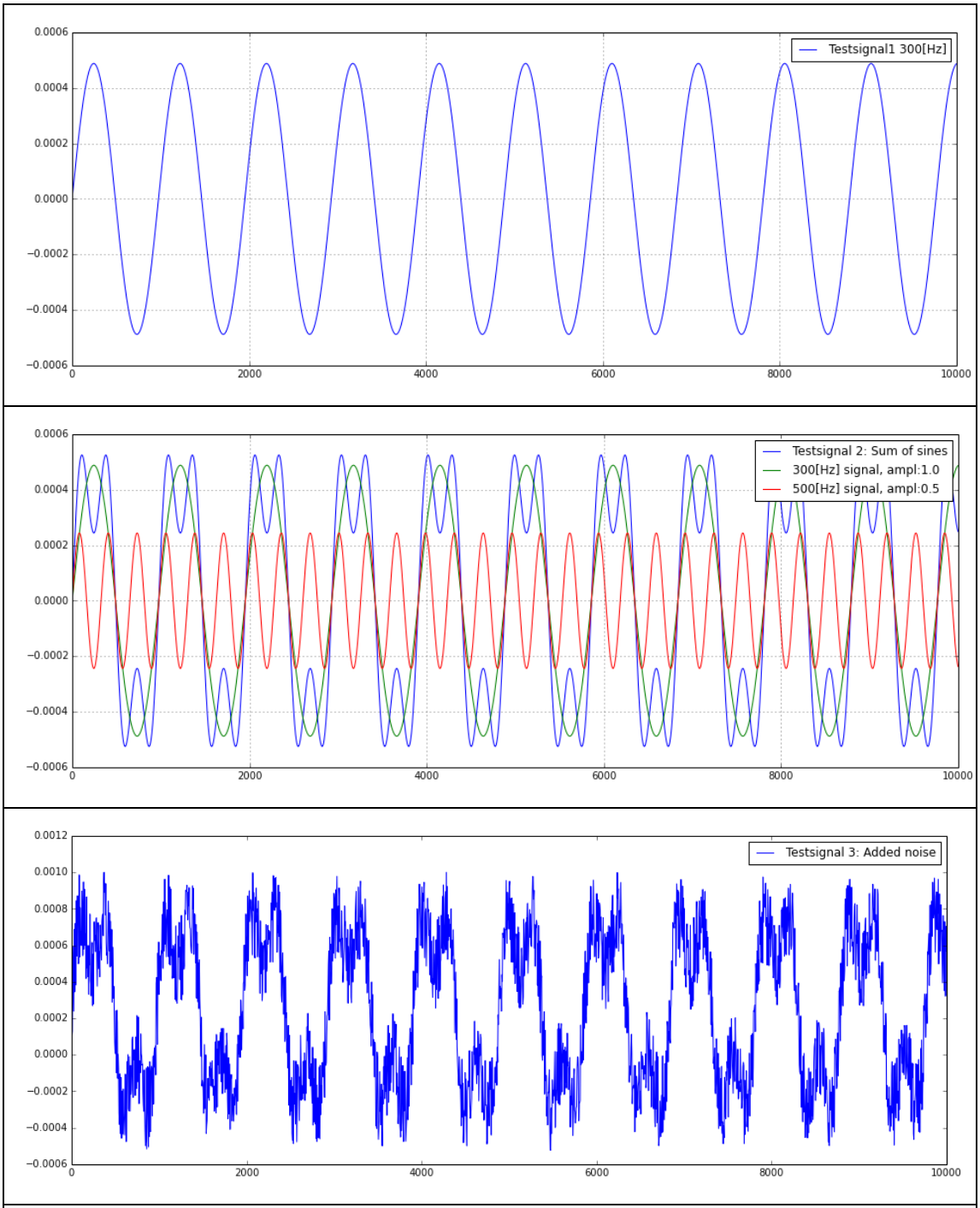


Figure 35 -Testsignals in the time-domain

When working with sampled discrete data produced by a 16 bit *Analog to Digital Converter - ADC*, we can gather data across a wide dynamic range, the output range of the converter *R* is equal to $\boldsymbol{R=2^n}$
For ***n=16***, this is equal to ***65.536*** steps.
Typically, amplitude spectrums is shown in logarithmic units, expressed as *decibels (dB)*. By using dB as a unit of measure, a wide dynamic range can be expressed in a compact graph. A logarithmic scale allows small contribution to appear, together with a large contribution. Decibels (dB) for an amplitude spectrum, are calculated as follows:

$dB = 20log_{10}(\frac{A}{A_{ref}})$ *where A is the reference amplitude, and* A_0 *is the measured amplitude*

When performing the ***fft*** on *testsignal 1* and plotting the logarithmic result in dB, the resulting frequency spectrum is seen.

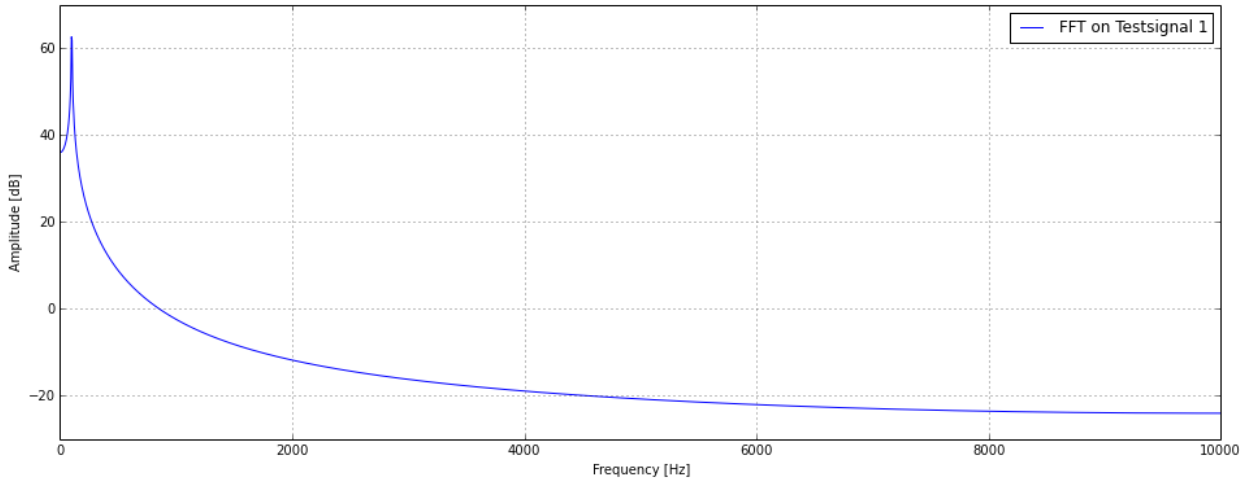


Figure 36 - FFT on 4096 generated samples, the test-signal 1

From *test-signal 1,* a 300[Hz] sine we would expect the FFT to display only one single *frequency-bin*, a "stick" located at 300[Hz]
The additional contribution, the *side-lobes,* seen in the frequency spectrum, on the FFT for *test-signal 1* is caused by a phenomena called *spectral-leakage.*

5.2.2.4. Spectral leakage

"For an accurate spectral measurement, it is not sufficient to use proper signal acquisition techniques to have a nicely scaled, single-sided spectrum. You might encounter spectral leakage. Spectral leakage is the result of an assumption in the FFT algorithm that the time record is exactly repeated throughout all time and that signals contained in a time record are thus periodic at intervals that correspond to the length of the time record. If the time record has a nonintegral number of cycles, this assumption is violated and spectral leakage occurs. Another way of looking at this case is that the nonintegral cycle frequency component of the signal does not correspond exactly to one of the spectrum frequency lines".[33]

Since our data generated by the *edge-node* (the prototype hardware), is sampled at a fixed rate, we know that no periodic relation exists, between the timedomain signal sampled, and the *length* ***N*** and *samplerate* $\mathbf{F_s}$. The signal is said to be discontinuous, when "the ends don't meet" at each end of the sample buffer, illustrated in figure 37.

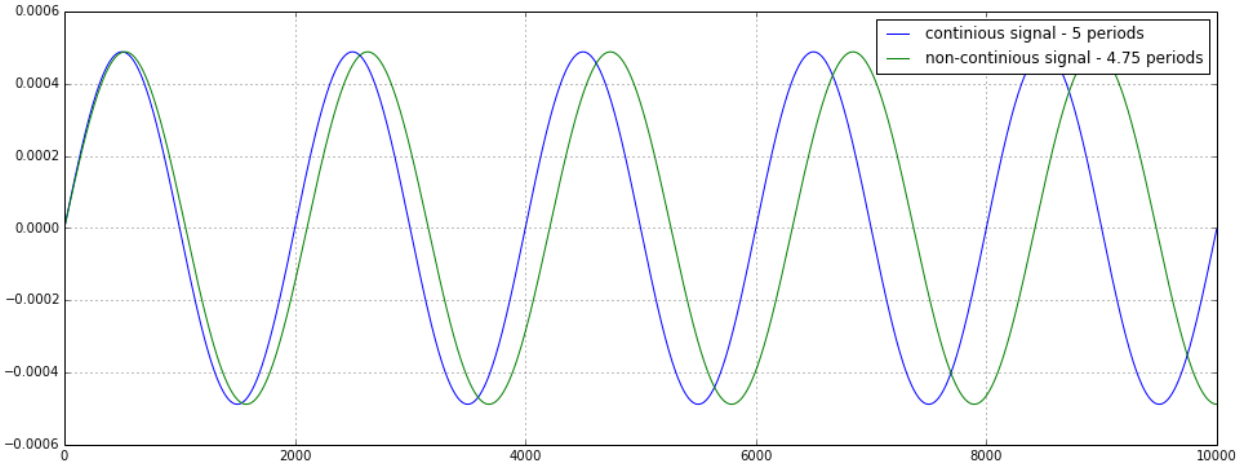


Figure 37 - discontinuity in a sampled dataset, the green samples are discontinuous they only represent 4.75 periods "the ends, head and tail, don't meet", where the blue signal represents exactly 5 periods

Spectral leakage leads to distortion of the desired FFT, as seen in figure 36 ideally the frequency response would be a thin "stick" representing single frequencies. Spectral leakage results in the amplitude for different *frequency-bins* to be spread across adjacent *bins*, leading to distortion of the result.
To accommodate this common phenomenon of spectral leakage, a compromise between loss of *dynamics* in the signal and amplitude information towards increasing the accuracy of the FFT has to be made, this is known as signal windowing.

5.2.2.5. signal windowing

Signal windowing basically is about multiplying a mathematically described curve, *a window*, with the samples of interest (containing the discontinuity). The multiplication is done in the time domain, sample by sample, as described by the sum:

$$x_{win}[n] = \sum_{n=0}^{N-1} x[n] * w[n]$$ *where* ***x[n]*** *are time domain samples, and* ***w[n]*** *is the window coefficients*

The *Scientific Computing Package, numpy,* in python contains functions for calculating 5 common used window functions, with different characteristics.

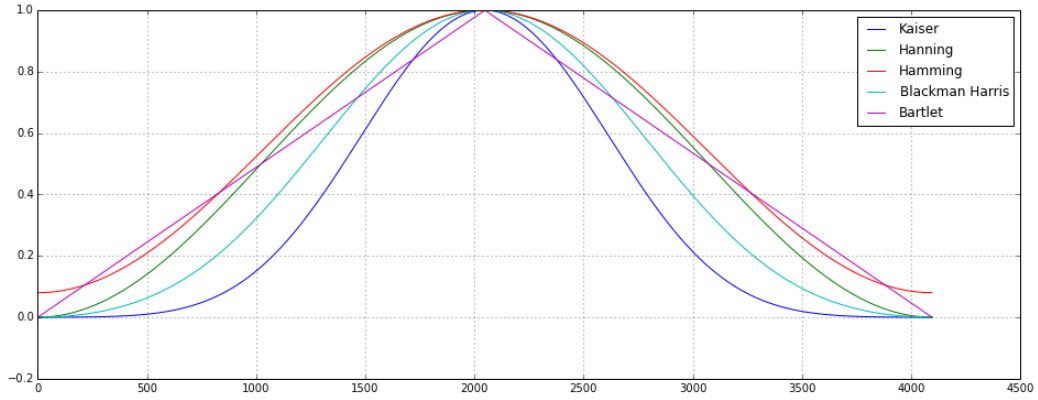


Figure 38 - window functions, to reduce spectral leakage, the 5 available window functions in numpy

When multiplying samples with a window function, some signal information is lost. Actually when processing a block of raw samples, these are captured "as a snapshot", until the sample buffer is full. This corresponds to using a rectangular window.
When multiplying *test-signal 1*, with a *Hamming-window*, the signal will be "hidden" in a *Hamming* envelope shape.

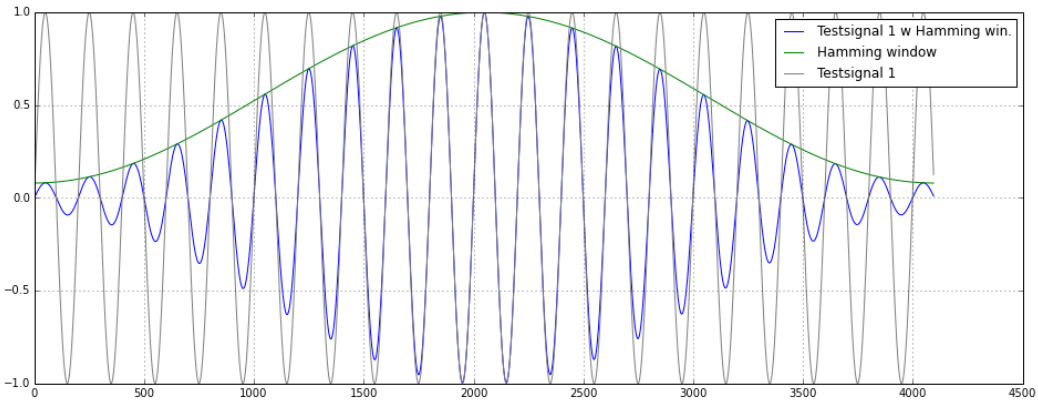
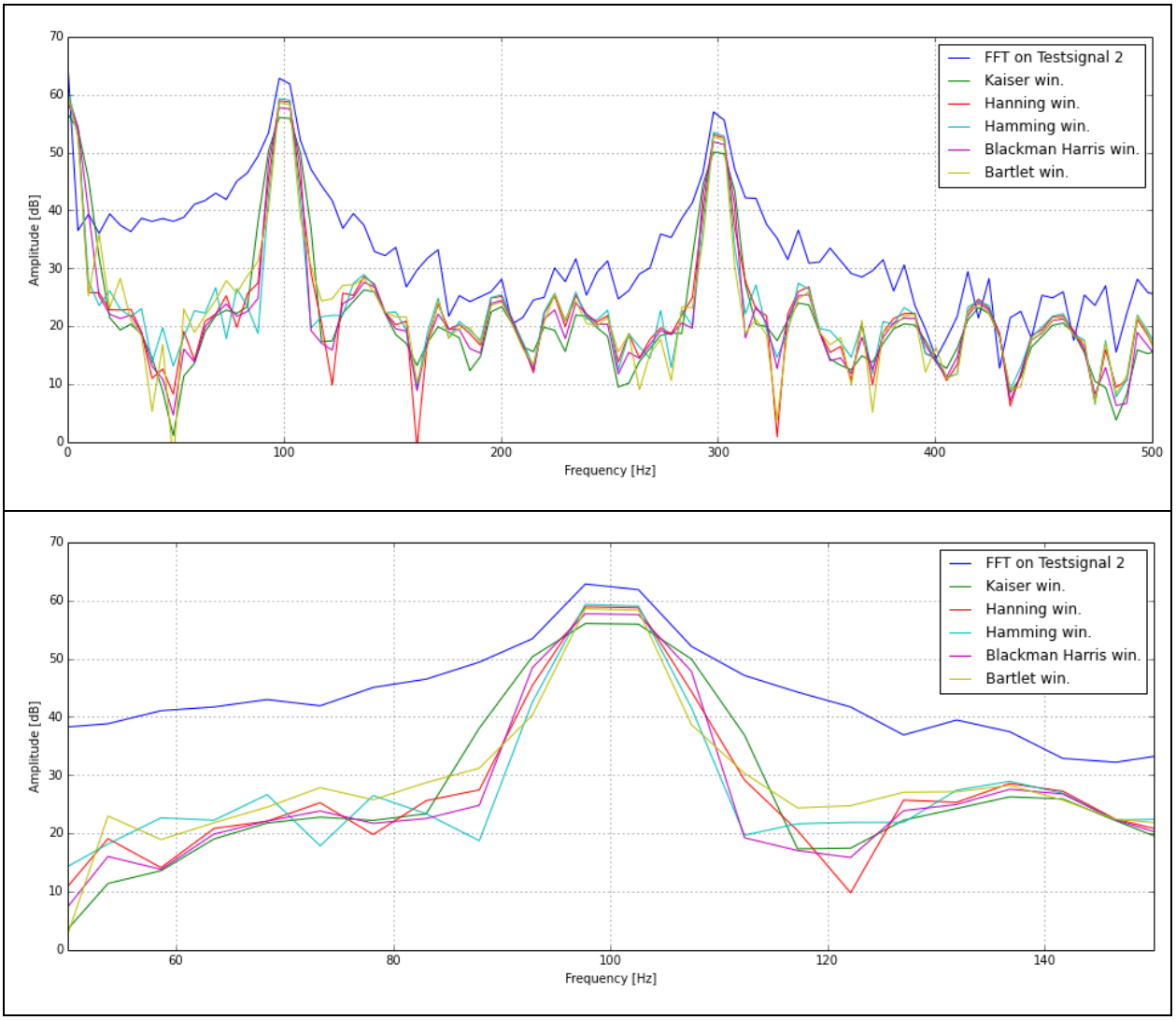


Figure 39 - Testsignal 1 windowed by a Hamming window, compared to the original Testsignal 1

When comparing the windowed signal to the original signal, it can be seen that some information is lost by performing the window function. Depending on the type of window, various amounts of amplitude or frequency information is lost, but overall the accuracy of the FFT is improved.

What characterizes the desired frequency analysis, in vibration monitoring system, in this report is following characteristics.
Absolute amplitude accuracy is not import, as we look for relative change over time
Good frequency precision is desired, since small deviations in frequency can indicate wear, and hereby changes in the bearings geometry, or missing lubrication.

When performing ***fft*** with a simulated aignal containing 100[Hz] and 300[Hz] added, with the 5 available windows listed, and zooming in on the spectrum between 0-500[Hz] we get following



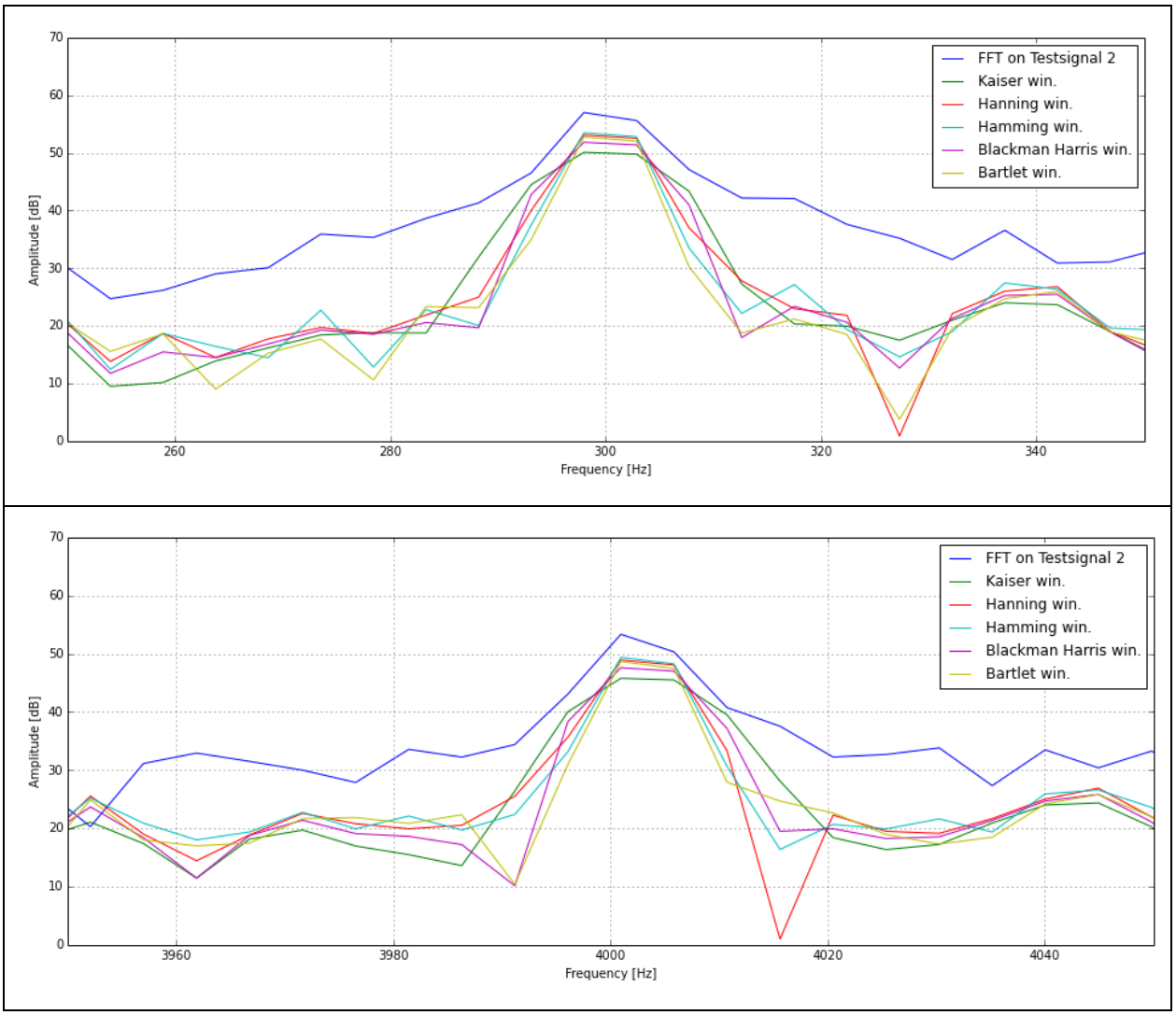


Figure 40 - comparison of FFT around the frequencies of interest

When observing the measured amplitudes, no significant differences are observed , thus any of the signal windows could be used, if the nature of the measured signal is what is observed.

From [33] a number of recommendations are made, depending on the application and nature of the vibrations.

Signal content	**Window type**
Sine wave or combination of sines	Hanning
Sine wave (amplitude matters)	Flat top
Narrowband random signal (Vibration data)	**Hanning**

Broadband random (white noise)	Uniform
Closely spaced sine waves	Uniform, Hamming
Excitation signals (hammer blow)	Force
Response signals	Exponential
Unknown content	Hanning

Table 15 - recommended window selection based on content [33],
A Hanning window is recommended for vibration signals

The Hanning window will be used for initial testing on the prototype.

5.2.2.6. Frequency resolution of the FFT

The bearings mounted in machines, where vibrations are to be measured, can have a diverse nature. Some machines may be low speed, with large bearings, where other high RPM machines would typically have smaller bearings, thus the geometry is different. As described previously, any bearing application will exhibit vibrations proportional to the rotational speed of the application.

The ability to interpret measurement, relies on not only on the processing and visualization of the data, but also the quality, more specific on the frequency resolution and dynamic range of the sampling system. The dynamic range is given by the 16bit Analog to digital converter, and the gain and range of the front end amplifier.
The frequency resolution determines the granularity of the frequency bins, that can be observed and measured.
From the applications analyzed previously it is known that in the **low speed range**, results in inner- and outer- ball pass frequencies of ***4,2[Hz]*** and ***5,8[Hz]*** can be expected and in the **high speed range**, in inner- and outer- ball pass frequencies of ***571[Hz]*** and ***678[Hz]*** are expected. Furthermore, individual high frequency contributions may appear, as result of damaged and pitting roller elements.

The achievable frequency resolution of any FFT analysis is known to be $\Delta f = \frac{Fs}{N}$, and in the initial prototype, we have a ***4K*** *(N=4096)* samples buffer, updated with a ***20[KHz]*** sample-rate, resulting is a *frequency-bin-width* $\Delta f = \frac{Fs}{N}$ of ***20.000/4096 = 4.88[Hz]***. This is not sufficient for detecting the two lowest inner- and outer-ball pass frequencies of 4.2 and 5.8 [Hz], as illustrated with a test-signal, generated to be a sum of the two frequency contribution.

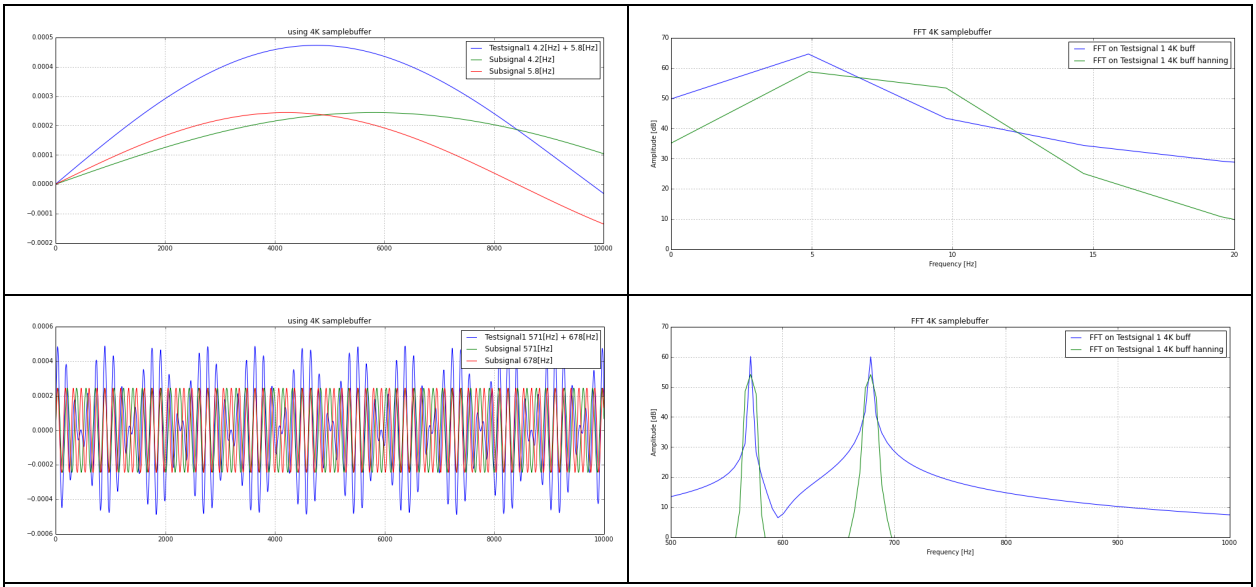


Figure 41 - generated low frequency test-signal (top-left) and FFT (top-right), high-frequency test-signal (bottom-left), and FFT (bottom-right) with N=4096 sample-buffer (se appendix n for calculations, and simulation)

The current design with four parallel samplebuffers, sampling each accelerometer, does not allow detection of the lowest frequency inner- and outer-ball pass frequencies.
One possible modification, based on the allowable buffer sizes, governed by the available RAM in the processor could be sampling the channels consecutive, one by one, thus allowing each channel to take up 16K samples. This modification would change the *frequency-bin-width* $\Delta f = \frac{Fs}{N}$ of ***20.000/16384 = 1.22[Hz]***.

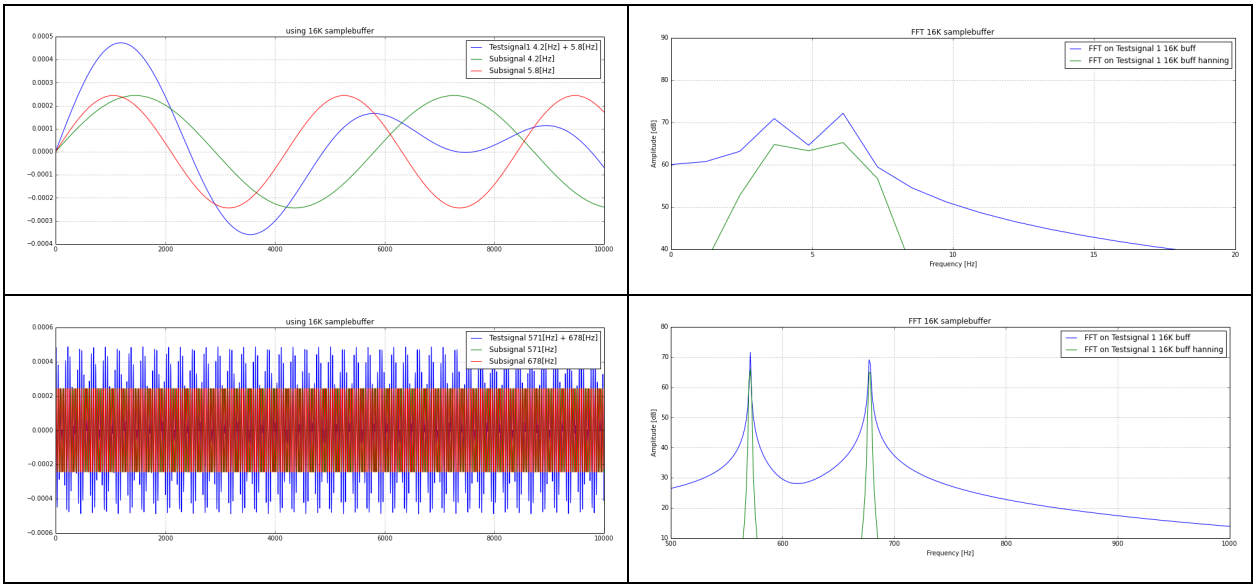


Figure 42 - generated low frequency test-signal (top-left) and FFT (top-right), high-frequency test-signal (bottom-left), and FFT (bottom-right) with N=16384 sample-buffer

FFT on 16K samples allows detection of the presence of the two low frequencies, but the *frequency-bin-width* does not leave room for detecting changes of a few percent, that may indicate beginning wear.

An even finer resolution, allowing detection of smaller changes, on the low frequencies, via the FFT can be achieved in three ways:

1. Increase the amount of available memory for samplebuffers
2. Decrease the resolution from 16bits (65536 steps) to 8bits (255 steps), thus allows twice the amount of samples to be stored
3. Decrease the sample-rate, to decrease the width of the *frequency-bins.*

5.2.3. Conclusion, data analysis

The flexibility of *Python* combined with the *Scipy Scientific Computing Packages*, provides a lot of useful features for analysing and processing data. Rapid simulations of the signals expected from real-life applications reveals some challenges when aiming to cover a broad range of measurement on different bearing applications.
The initial prototype will be designed to gather data in 4 parallel sample buffers, despite the fact that a limited frequency resolution can be achieved.
Further refinement of the prototype, might result in a more advanced sample controller, capable of adjusting the utilization of the available sample buffers, as well as adjusting the sample-rate f_s depending of the frequency range to measure, or emphasize.

5.3. Data visualization design

The analyzed data from the *frequency analysis* block have a multitude of endpoints, one of which is visualization. The visualization currently is what is used for decision support.

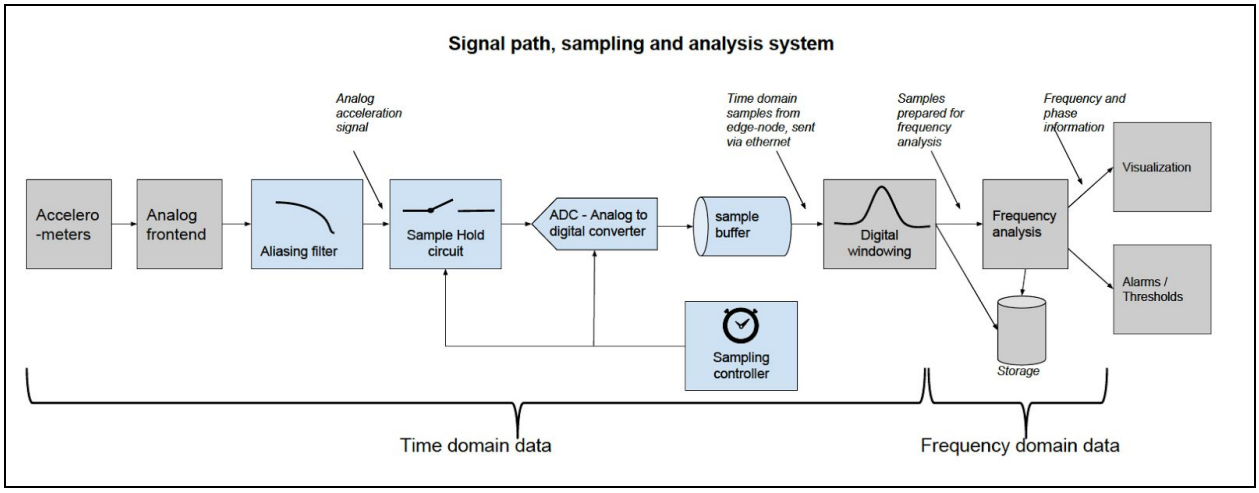


Figure 43 - entire signal path from time to frequency domain

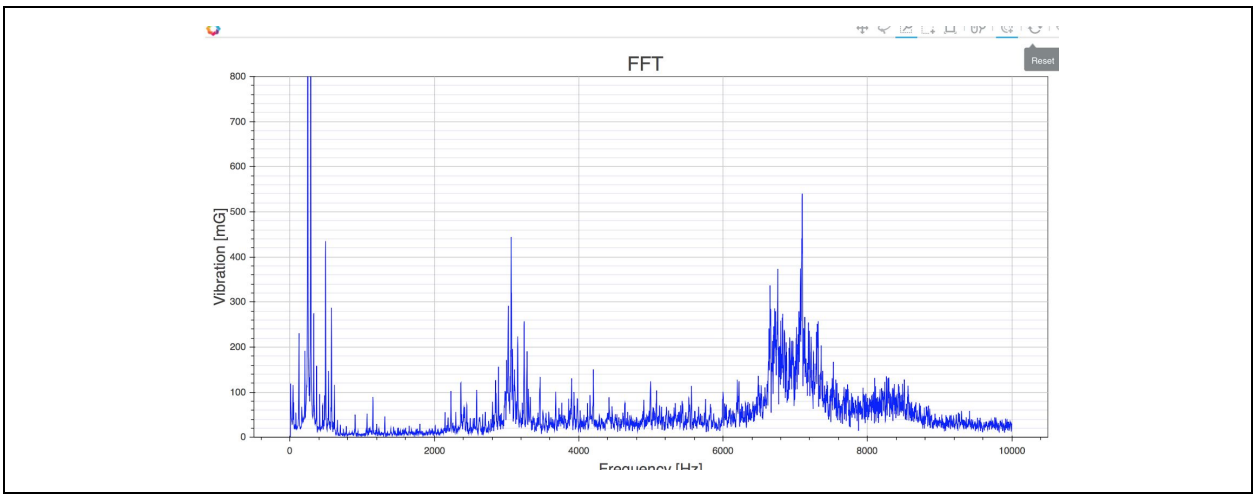
As seen in the previous chapter, analysis and insight requires a clear insight and overview of the observed and analyzed data.
Furthermore, when working with a broad frequency spectrum, and the aim of detecting small changes in frequency, in current and historical data, this involves the ability to navigate in three dimensions, namely *Frequency, Amplitude* and *Time*.
Frequency and *Amplitude* data is generated in real time from the *edge-node*, where the ability to navigate in historical time data, involves access to stored data samples.

5.3.1. Bokeh visualization framework for python

The test signal and analysis performed in previous chapters are done with *matplotlib[36]*, a popular 2D plotting data library, for visualizing data. *Matplotlib* can be used with various python utilities to visualize data in graphs. Interaction is primarily done via modification on the source code, and thus, is suited for scientist and programmers.
Bokeh[37] is an emerging visualization framework, intended to visualize both static and dynamic data, in modern web-browsers. Initial experiments has revealed it to be rather intuitive and capable for a project of this shape.
The *bokeh* framework allows visualization of simple 2D graphs, for frequency and amplitude, with additional interaction for zoom and pan in the data.
Furthermore when running a dedicated *bokeh-server,* additional interaction and *user-interaction* elements are available.
For the initial prototype, a bokeh-server will be running on a local-machine, interacting with the *edge-node*.

5.3.1.1. Initial FFT layout

A prototype layout has been build, as seen in figure 44.



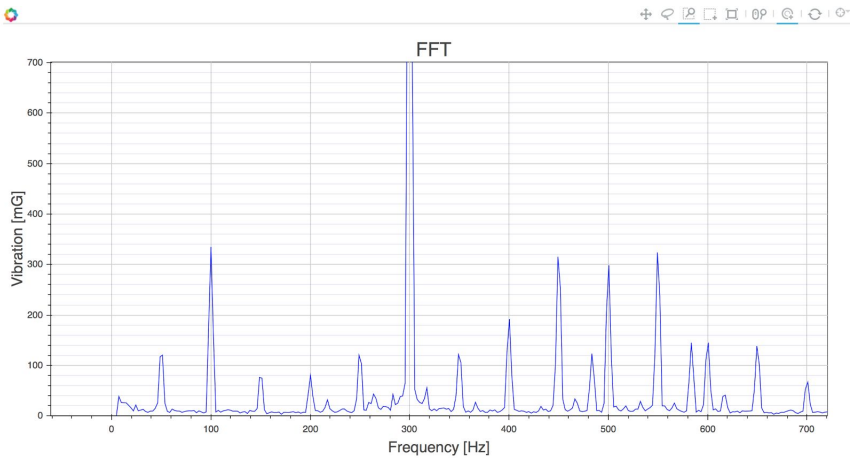
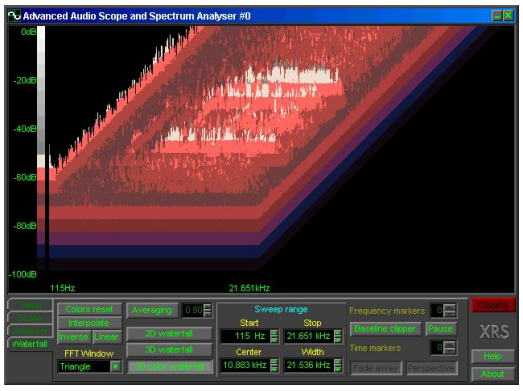


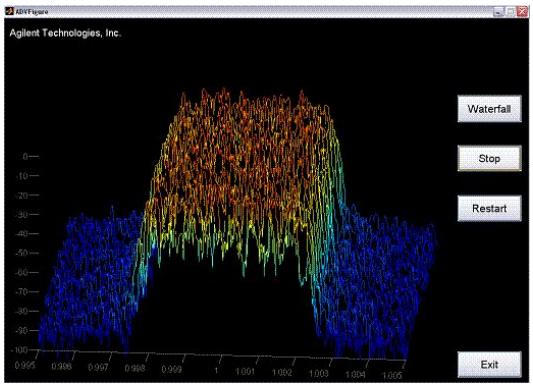
Figure 44- one channel FFT layout (top), data zoomed in (lower)

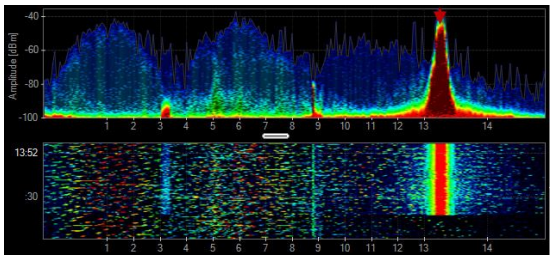
In the upper right corner, a toolbar is build, allow zoom and repositioning of the graph, to zoom in on the frequencies of interest. The lower frequency-spectrum in figure 44 shows a zoomed in visualization of the frequency range between 0-700[Hz]
A number of indicators showing relevant information, as inner and outer ball pass frequencies, based on machines measured rotation speed will be build into the visualization window as well.

5.3.1.2. browsing Historical data

Browsing historical data can be implemented in many ways, either as a 2D graph, as the current FFT, as a waterfall, spectrogram, or something different.







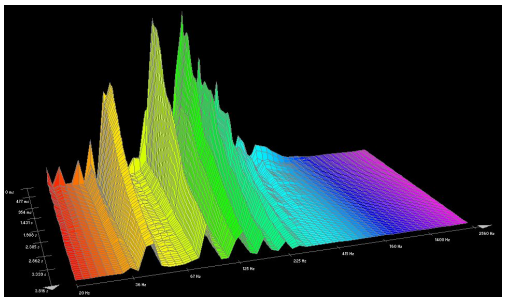


Figure 45 - different 3D visualization techniques

The optimum technique for browsing historical data will not be covered in depth in the current project. The goal is to be able to browse historical data, to discover trends.

5.4. Data storage design

The incoming data is to be stored in plain text files, in the initial prototype. The requirements for the basic data storage are outlined in the requirements section.
Assessing candidates for, and designing a suited final data-storage solution, requires significant more insight to how the system will perform when deployed. A deeper insight into database architecture, combined with constraints given by online storage solution, and the cost of data transport, is to be undertaken to present suited solution-candidates.

5.5. File-based storage implementation

The text-file based data storage is not implemented, currently.

6. Bibliography

1. Brüel & Kjær Sound and Vibration Measurement A/S, *Introduction to shock & vibration:* http://www.bksv.com/doc/bn1330.pdf ,1998
2. M.Jakobsen, *Assessing the use of low cost MEMS accelerometers for vibration measuring in rotating machines,* AU Herning, November 2015
3. N.Tandon and A. Choudhurry *A review of vibration and acoustic measurement methods for the detection of defects in rolling element bearings,* Tribology International 1999
4. Isac Brown, *Predictive Maintenance: The Art of Uptime*. Lux Research March 2016
5.
6. http://www.thingworx.com/thingworx-analytics
7. http://www.rs-online.com/designspark/electronics/knowledge-item/eleven-internet-of-things-iot-protocols-you-need-to-know-about
8. Thor Olavsrud, *GE launches industrial Internet PaaS,* http://www.cio.com/article/2957194/cloud-computing/ge-launches-industrial-internet-paas.html, august 2015
9. Saul Klein, *How to transform your business when the old rules no longer apply,* http://www.wired.co.uk/magazine/archive/2015/12/features/how-do-you-build-a-business-in-2016, november 2015
10. Amritt Tiwana, *Platform Ecosystems, Aligning architecture, governance and strategy*, p55-59, Elsevier, 2014
11. Geoffrey Parker et. al., *Pipelines, Platforms, and the New Rules of Strategy,* https://hbr.org/2016/04/pipelines-platforms-and-the-new-rules-of-strategy, April 2016

12. ELFORSK 346-011 – *Anvendelse af energieffektive lejeløsninger i industrien,* http://www.elforsk.dk 2014
13. S.J.Lacey, *An overview of Bearing Vibration Analysis,* Maintenance & asset management, vol 23, Nov 2008
14. *Technical datasheet Accelerometer ACH-01* , Measurement specialities 2008, http://www.meas-spec.com/downloads/ACH_01.pdf
15. *Application specification for Accelerometer ACH-04-08-05*, Measurement specialities 1998, http://resenv.media.mit.edu/classarchive/MAS836/Inertialnotes/ach04.pdf
16. 2N4117 JFET datasheet, http://www.digchip.com/datasheets/parts/datasheet/451/2N4117-pdf.php
17. Joel L. Dawson, *Current Sources and Current Mirrors* , MIT Department of Electrical Engineering 2011, http://www.mit.edu/~6.301/rec13.pdf
18. Ron Roscoe, *course materials for 6.101 Introductory Analog Electronics Laboratory,* MIT Department of Electrical Engineering, 2007
19. Freescale semiconductors, *K64 Sub-Family Reference Manual, http://cache.freescale.com/files/microcontrollers/doc/ref_manual/K64P144M120SF5RM.pdf* p:825-879 .
20. Frescale Semiconductor *Kinetis K64F Sub-Family Data Sheet* https://developer.mbed.org/media/uploads/GregC/k64f_ds_rev6.pdf
21. Texas Instruments *AN-1540 Power Measurement of Ethernet Physical Layer Products* http://www.ti.com/lit/an/snla089b/snla089b.pdf
22. Texas instruments, *PCB thermal calculator,* http://www.ti.com/adc/docs/midlevel.tsp?contentId=76735&DCMP=PCB_Thermal_Calc&HQS=Other+OT+pcb_calc_tb
23. Texas Instruments, *LM22675 simple switcher datasheet,* http://www.ti.com/lit/ds/snvs591k/snvs591k.pdf
24. Texas instruments *Web Bench Power Calculator*, http://www.ti.com/lsds/ti/analog/webench/power.page
25. Vishay semiconductors,SFH6202 dual optocoupler datasheet http://www.vishay.com/docs/83675/sfh620a.pdf
26. Infenion, *BSP77 datasheet,* http://www.infineon.com/dgdl/Infineon-BSP77-DS-v01_03-en.pdf?fileId=db3a3043271faefd01274cff04045d40
27. Rastislav Pavlanin, Cookbook for SAR ADC measurements, Freescale semiconductors, 2014 https://developer.mbed.org/media/uploads/GregC/an4373-cookbook_for_sar_adc.pdf
28. LwIP TCP/IP stack for microcontrollers: http://savannah.nongnu.org/projects/lwip/
29. *Freedom-K64F ultra-low-cost development platform for Kinetis K64 MCUs.* https://developer.mbed.org/platforms/FRDM-K64F/
30. Isaac Brown, *A Tale of Two IoT Titans: The Curious Case of the GE - PTC Partnership* Lux Research May 2016
31. Python 2.7, *The Python Standard library*, https://docs.python.org/2/library/index.html
32. *Fourier integral*
33. *The Fundamentals of FFT Based Signal Analysis and Measurement in LabVIEW* National instruments 2008, http://www.ni.com/white-paper/4278/en/

34. Steven W Smith, *The Scientist and Engineer's Guide to Digital Signal Processing,* http://www.dspguide.com
35. Scipy, *Python Scientific Libary* http://docs.scipy.org/doc/numpy-1.10.1/reference/routines.fft.html
36. *Python Matplotlib a 2D python data-visualization library*, http://matplotlib.org/
37. *Bokeh, interactive python data-visualization library, intended for browsers*, http://bokeh.pydata.org/en/latest/
38. William D Eggers, *Data as the new currency*, Deloitte review, 2013 http://deloitte.wsj.com/riskandcompliance/files/2013/11/DataCurrency_report.pdf
39. https://en.wikipedia.org/wiki/Absorptive_capacity
40. https://www.accenture.com/t20150523T023647__w__/us-en/_acnmedia/Accenture/Conversion-Assets/DotCom/Documents/Global/PDF/Dualpub_11/Accenture-Industrial-Internet-of-Things-Positioning-Paper-Report-2015.pdf
41. Craig Resnick, *IIoT and Industrie 4.0, It Takes a Village*, https://industrial-iot.com/2016/02/iiot-and-industrie-4-0-it-takes-a-village/ Feb 2016.
42. https://opcfoundation.org/markets-collaboration/
43. http://www.smartfactory-kl.de/
44. http://www.iiconsortium.org/
45. Frank burkitt, *A Strategist's Guide to the Internet of Things* , http://www.strategy-business.com/media/file/00294_A_Strategists_Guide_to_the_Internet_of_Things.pdf 2014
46. M Porter, J. Heppelmann *How Smart connected products are transforming companies,* HBR october 2015
47. M Porter, J. Heppelmann, *how-smart-connected-products-are-transforming-competition,* HBR november 2014
48. PTC, *ThingWorx IoT Platform Tutorials* http://learningexchange.ptc.com/tutorials/by_sub_product/sub_product_id:37
49. Cornelius Baur, Dominik Wee, Manufacturing's next act, http://www.mckinsey.com/business-functions/operations/our-insights/manufacturings-next-act , June 2015
50. NEW CHIP ALERT: THE ESP8266 WIFI MODULE (IT'S $5) http://hackaday.com/2014/08/26/new-chip-alert-the-esp8266-wifi-module-its-5/
51. Ben Kepes, *Understanding the Cloud Computing Stack: SaaS, PaaS, IaaS,* https://support.rackspace.com/white-paper/understanding-the-cloud-computing-stack-saas-paas-iaas/ 2016
52. Wikipedia on Microservices https://en.wikipedia.org/wiki/Microservices
53. *ARM Cortex M4 architechture, DSP instruction features* http://infocenter.arm.com/help/topic/com.arm.doc.ddi0439b/DDI0439B_cortex_m4_r0p0_trm.pdf section 3-8, ARM Ltd.2010
54. Analog discovery, technical reference manual https://reference.digilentinc.com/_media/analog_discovery:analog_discovery_rm.pdf
55. Eelpraxx 80 bearing tester, http://www.elgeti-engineering.de/media/pdf/Veroeffentlichungen/Publication-2016-05-Bearing-testing-for-incoming-inspection.pdf

56. IFM VSA001 accelerometer, http://www.ifm.com/products/dk/ds/VSA001.htm
57.