Expertise
and insight
for the future

Teemu Raivonen

Predictive Condition Monitoring in Intralogistic Systems

Metropolia University of Applied Sciences

Bachelor of Engineering

Automation engineering

Bachelor's Thesis

8 December 2019

Author Title Number of Pages Date	Teemu Raivonen Predictive Condition Monitoring in Intralogistic Systems 28 pages + 9 appendices 8 December 2019
Degree	Bachelor of Engineering
Degree Programme	Automation engineering
Professional Major	Automation Technology
Instructors	Jani Eronen, Service area manager, SSI Schäfer Timo Tuominen,Senior Lecturer, Metropolia university of applied sciences

This thesis focuses on research and implementation of condition monitoring concerning industrial warehouse automation systems.

The research part of this thesis aims to explore the necessity and possibilities of condition monitoring in relation to the state that technology and markets currently reside in. The research also focuses on the pre-requisites of implementation concerning condition monitoring and IIoT systems.

The latter part of this thesis concentrates on the design and implementation of a prototype system specifically designed for a client of SSI Schäfer. This prototype aims to provide applicable solution for reoccurring bearing failures in automated transport shuttles, included in the warehouse automation system of the client.

Lastly, the data provided by the prototype system included in this thesis is compiled to offer analytic insights about the bearing conditions for maintenance and further condition monitoring purposes.

Keywords	Condition monitoring, intralogistics, automation industry 4.0, IoT, IIoT, predictive maintenance, MEMS, NodeRed, Python

Table of Contents

Appendices

List of Abbreviations

ARC	An advisory company specialized in technology market research for industries
CSS	Customer Service & Support
CSV	Comma Separated Values, a plain text file format where individual values are stored in a list and separated by a comma sign “,”
FFT	Fast Fourier Transform, a mathematical concept used in spectral analysis
GPIO	General purpose I/O, a collection of assignable input and output ports for various uses
I/O	Input/output module
I²C	Inter-Integrated Circuit, a serial communication bus for processors and microcontrollers
IIoT	Industrial Internet of Things, industrial adaptation of IoT
IoT	Internet of Things, a concept related to smart systems and information technology
LHD	Load Handling unit. Linear servo driven extenders in Navette shuttles that transport storage containers between Navette shuttle and storage positions
MEMS	Microelectromechanical Systems
NPM	Node Package Manager, a package manager for JavaScript
OS	Operating system
PLA	Polylactic acid, a renewable thermoplastic aliphatic polyester often used in 3D-printing
SI-units	International system of metric units
SMBUS	System Management Bus
SSH	Secure Shell, a cryptographic network protocol used for remote access over an unsecured network
SSI	Schäfer Systems International
UI	User Interface
Unix-time	A System for tracking time, refers to seconds elapsed since 00:00:00 Thursday first of January 1970, often used in computational applications as the base time
URL	Uniform Resource Locator, specifies web resource location and retrieval protocol on a computer network

WMS	Warehouse Management System. Software for managing the flow of orders and fluent communication with the storage system

1 Introduction

This thesis is written in response for a demand that arose in one of SSI Schäfer customer sites concerning the local warehouse automation system. The purpose of this thesis study was to research the possibilities of condition monitoring and IIoT on industrial applications, specifically addressing the issues concerning Navette shuttle bearing failures and providing practical solution for the said issues. Due to the high frequency and cost of these issues, a variety options regarding an applicable solution are taken into consideration.

This thesis is composed of two parts. The first part describes and addresses the advantages and benefits of the predictive maintenance model and condition monitoring, while the second part addresses the basis, design and implementation of a condition monitoring system, provided for the mentioned client.

The first part expands some background on the subjects of this thesis, providing a basis on which the condition monitoring prototype is designed, as well as clarifies the necessities for further research and applicable solutions by examining the details in the fault history of the bearing breakdowns on the clients site.

The second part examines the constraints posed by the existing systems, addressing a set of pre-requisites and limitations concerning the implementation of the prototype system. The design of the functionalities and the scope of the prototype are composed around the said constraints of the existing system and the functional needs of effective condition monitoring.

2 State of the Industry

2.1 Industry 4.0

Changes happening in technology greatly affect industries and markets as well. Decrease of component prices, advancements in digital technologies and software availability, all reinforce huge technological possibilities for industrial automation. These new fields create opportunities, initiate competition, and force companies to evolve.

IoT (Internet of things) is a relatively new and popular concept. IoT broadly refers to a branch of new digital technologies and concepts that have emerged in past decades, such as smart devices, big data, machine learning and cloud computing. Another term linked to this is IIoT (Industrial internet of things), or Industry 4.0, that contains the same concepts as IoT, but with greater focus towards manufacturing and industrial use. (1.)

Though these concepts have gained much of their popularity due to successful marketing and bold estimations, they still contain valuable resources for the industry.

2.2 Predictive Maintenance

One of the most crucial aspects of production is maintenance. Inefficient maintenance has consequences. In the case of a company that provides automation systems and related services, a client chooses suppliers on the basis of reliability, efficiency and the worth towards their assets.

Unnoticed fault in a system often leads to defects, which may in turn, lead to damaged components, excessive repair costs and unnecessary downtime for the client.

There are three main strategies of maintenance to prevent these unnecessary costs

- Reactive maintenance
- Preventive maintenance
- Predictive maintenance

Reactive maintenance aims to correct arising malfunctions when they initially occur. This strategy aims to eliminate any unnecessary maintenance procedures by repairing the system or its components after they break. Though this strategy saves in direct maintenance costs, it increases the amount breakdowns and the downtime caused by the repair processes.

Preventive maintenance relies on maintenance cycles, on which the system and its components are checked for damage and malfunctions. In addition to this, components that are exposed to wear and detrition (such as bearings) are being replaced systematically during these cycles.

While this strategy drastically reduces the amount of malfunction, it greatly increases the maintenance costs.

Predictive maintenance is based on extensive monitoring solutions that can detect upcoming failures before they occur. This strategy decreases both maintenance costs and excessive repair costs, by scheduling the component replacements based on the actual state of the components.

In earlier decades preventive maintenance has been an industry standard. The excessive repair costs and downtime caused by reactive maintenance strategies and the heavy investment requirements of predictive solutions left these strategies to disadvantage compared to the preventive model.

Over the last 20 years technology has evolved rapidly. While the preventive model is still being embraced by majority of the industry, new solutions are providing more realistic terms for predictive monitoring solutions.

IIoT and predictive maintenance provide systems with broad monitoring and communication possibilities. Different diagnostic solutions have capabilities, not only to reduce the unnecessary costs related to maintenance, but to improve the quality of the whole system and to eliminate downtime as well.

A study done by ARC (2) states that Only 18 percent of assets have a failure pattern related to age. This leaves the preventive model to a disadvantage compared to the predictive model as the component prices of intelligent systems are drastically getting lower (3). To keep up with the competition companies need to re-evaluate their assets and strategies related to maintenance in accordance with the changes in the industry.

3 Background Overview

3.1 Associate parties

3.1.1 SSI Schäfer Finland

SSI Schäfer was originally founded in Germany 1937. The company started by manufacturing steel storage containers. The company has since expanded towards logistics and automation systems. The company currently has 70 subsidiary companies over the globe. SSI Schäfer provides intralogistic solutions with a comprehensive product family containing storage and

retrieval systems, containers, conveying, warehouse management software, work stations, as well as related services and manual warehousing. (4.)

SSI Schäfer Finland headquarters are located in Vantaa, in which sales, project managing, IT and human resources are centralized. In addition to this, the local maintenance sector operates primarily in the Vantaa area. The operations are concentrated primarily to serve Finnish customers, but the resources between Nordic subsidiaries are frequently shared to serve the customers efficiently.

3.1.2 Stockmann

Stockmann Oyj Abp is a retail company founded in 1862. The company currently maintains six department stores in Finland and seven department stores located in Baltic countries. In addition to the departments stores in Finland, the company has an online store for the products sold by their company. The companys brand focuses on cosmetics, clothing, luxury and household products. In 2017 the company had revenue of over one billion and over 7 000 employees. (5.)

3.2 Client Site

3.2.1 Project

The site of the client is a distribution center for Stockmann Oyj. retail services. The site is located in Tuusula and distributes to local Stockmann department stores in Finland and Baltic countries.

The center was finished in 2016 and has been operational since. The center consists of both automated and manual warehouse units. The automated unit uses Navette storage & retrieval system, which is a product system provided by SSI Schäfer. The Navette system at Stockmann site consists of nine picking stations, 72 Navette shuttles, 45 Navette lifts, Navette rack and a conveyor system connecting all mentioned units.

3.2.2 Production overview

The automated unit of the warehouse is based on organization of identical storage containers referred as “totes”, which contain smaller products ranging from makeup products to small collapsible furniture, where as in the manual unit handles everything above the dimensional restrictions of the totes.

The automated unit (operating on Schäfer equipment) holds total capacity of approximately 130 500 storage tote positions, in which the products are contained inside the totes. The totes are stored in an organized rack system, in which the totes are stored and retrieved automatically using Navette storage & retrieval system.

The system is designed for a daily production performance of 55 000 order lines and 180 000 single orders. (6.)

3.2.3 Provided services

The contract that was composed for the project between Stockmann and SSI Schäfer includes CSS resident maintenance that operates in two shifts during production. The service contains constant upkeep and preventive maintenance of the system, implemented by continuous system monitoring and annual maintenance cycle for the Navette-system and it’s included components.

4 Client system

4.1 Process overview

4.1.1 Wamas

Wamas is a WMS (Warehouse Management System) provided by SSI Schäfer. Wamas compiles the intralogistics areas and components of the warehouse management to a singular system. These areas include the automation system, picking management, platforms of SSI Schäfer and other suppliers, warehouse monitoring and information systems.

Wamas works as an interface for the customers production needs and is tailored for each customer specifically. Through Wamas, the customer has a direct access to managing all orders as well as the working stations related to the sites production. (7.)

4.1.2 Navette Rack

Navette rack refers to the physical storage unit of the Navette storage & retrieval system. The rack at the Stockmann site consists of eight floors, each containing nine aisles. Each aisle is reserved for the operation of one Navette-shuttle, which transports totes along the aisle. The totes are exchanged between the shuttles and the lifts (located between the aisles) in specific exchange locations, one for each lift along the aisle on both sides. A picture from one of the aisles is shown in figure 1.

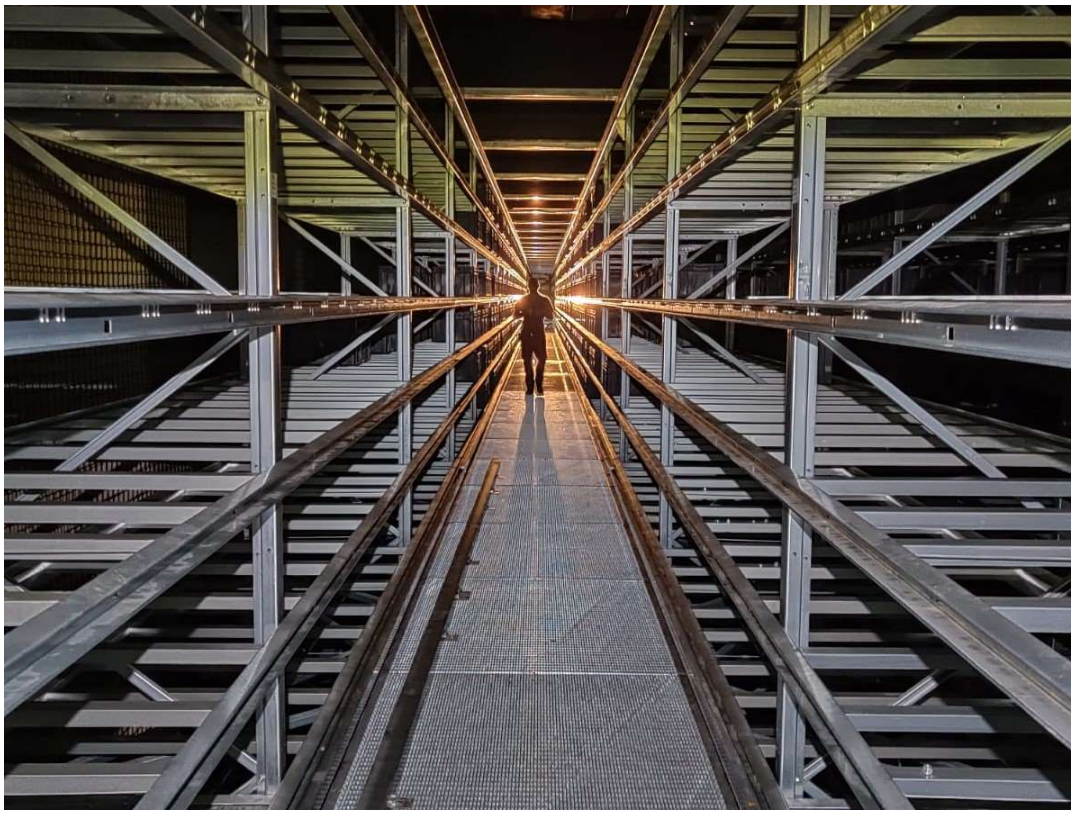


Figure 1. Navette rack aisle.

4.1.3 Navette Shuttles

The operation of a Navette shuttle is based on linear movement along the aisle, with additional LHD-units (Load handling device), which are telescope extensions that manage the horizontal positioning of the storage containers.

The shuttle is attached to guiding rails that secure the linear movement throughout the aisle. The two rails are located on the left side of the shuttle, supported firmly by guiding wheels positioned against the rails. A picture of Navette shuttle N1112 is shown in figure 2.

Figure 2. N1112 Navette shuttle.

4.2 Fault History

In 2018 a total of five separate Navette shuttles suffered undetected bearing failure, all of them occurring on the primary drive units, each resulting in repair process varying from one to two weeks. During each repair process the shuttle had to be locked virtually from the WMS, making all the products on the sector of that shuttle, unavailable to the customer for the said period. Since the bearing failures were not noticed before the initial breakdown, the friction and imbalance caused by the malfunctioning bearings eventually broke the bearings and damaged the structures surrounding them resulting in further mechanical damage and unnecessary component replacement. If these defects were spotted earlier, downtime for each replacement could have been reduced to four hours or less. Pictures of a main drive bearing unit of a Navette shuttle after similar malfunction are shown in figure 3.

Figure 3. Broken bearing on the left, damaged bearing socket on the right. Taken after a bearing breakdown in august of 2019.

5 Condition Monitoring

5.1 Bearing Overview

5.1.1 Principles

In mechanical systems where distinct parts are exposed to detrition or strain caused by rapidly moving elements, friction can be minimized by using supporting mechanical elements such as bearings. In addition to the elimination of excess friction, they constraint the movement of attached components to specific spatial motions, typically in linear or circular trajectories. (8)

5.1.2 Bearing Types

Bearings come in different sizes and forms, depending on the use and implementation. Bearings are in their simplest form a combination of two surfaces separated by a medium that prevents direct contact of the two surfaces. Ideally, the other surface works as a static counterpart to the surface to which the motion is applied so that the applied force is divided evenly along the intended constraints, ensuring smooth movement throughout the trajectory along the axis. The medium between these surfaces is usually one of the following types:

- plain bearing
- rolling element bearing
- fluid bearing
- magnetic bearing
- flexure bearing
- jewel bearing

Though the bearing types differ greatly in terms of the mediating substance and the direction of the spatial movement, the basic principle remains the same. The subject of this thesis study concerns exclusively mechanical rolling element bearings, in which the medium consists of

multiple cylindrical rolls between two cylindrically shaped elements, where the movement is restricted to circular movement around the primary axis. (8.)

5.1.3 Causes of Malfunction

As force is applied to the bearing continuously over long periods of time, the friction, even though minimal, eventually causes damage to the structures of the bearing. The bearing must be replaced eventually to prevent breaking and further damage to the system.

Each bearing has a calculated "lifecycle", often referred as L10 life value, which can be determined through numerous factors regarding the size, speed, materials and general assembly of the system and the parts it consists of. L10 life represents the calculated travel or runtime that the bearing can manage while operating in specified conditions until the inevitable fatigue due to detrition reaches the set limits. The L10 life does not take into account the damages caused by external faults like mechanical imbalance, misalignment, looseness or sudden exterior impacts. The calculated L10 life for an individual bearing is guaranteed to last for 90 percent of the cases, but the remaining ten percent of these cases often fail due to the complexity of the system and due to a number of influencing factors affecting the systems state. (9.)

5.2 Advantages of Condition Monitoring

Sometimes the bearing may suffer unintended damage through several things like impacts, faulty manufacturing, misaligned elements or excessive load applied unevenly to the supporting elements. This is where the analysis of these parts and their mechanical qualities come advantageous, or in some cases, necessary. By analyzing the state of a component like a bearing one could determine sudden spikes in detrition, unexpected resonance, impacts (transient shock) and other anomalies before the part initially breaks or causes further damage to the surrounding system.

5.3 Common types of Condition Monitoring

5.3.1 Temperature

When strain is applied to a bearing, a raise in temperature is caused by the friction. Even though the temperature remains minimal and only slightly above the temperature of the surrounding structures while the bearing is strained within its boundaries, the temperature rapidly increases when the bearing suffers internal damage and causes the bearing to run imbalanced. (10.)

Therefore possible internal faults can be diagnosed by monitoring the general temperature of the bearing or the structures in its immediate vicinity. It is very important to position the temperature sensors in very close proximity from the monitored bearing to avoid excessive heat-draining surfaces and thus to achieve the most accurate temperature measurement.

Temperature is a very simple an inexpensive monitoring solution and is widely used for these reasons. This method, however, does not indicate failure until the damage is critical and requires immediate action for component replacement, as increase in temperature requires significant increase in friction. Even though temperature monitoring may not be efficient method alone, it can be used along with other methods to confirm the mutual trends in each method.

5.3.2 Acceleration

Acceleration is one of the most common methods in bearing monitoring due to its accuracy and broad diagnostic features. It is used to measure axial movement to determine possible faults by monitoring unusual vibrations and impacts caused by the circular movement of the bearing. The measurement itself operates on a single axis and measures the acceleration linearly along that axis. Acceleration sensors of this type often consist of three separate axes to measure the movement and velocities on three dimensions.

MEMS-technology (Microelectromechanical systems) has created more advanced means to monitor acceleration more precisely. MEMS acceleration sensors are found in various price ranges, but implementation through industrial manufacturers remains expensive compared to other methods. (11.)

5.3.3 Ultrasound

Ultrasound monitoring is based on localized measurements of sonic emissions. This type of measurement offers useful advantages due to its capability to separate multiple frequencies from each other for broader monitoring purposes.

Ultrasound monitoring is one of the more accurate forms of condition monitoring, but it is also the most expensive one. This option serves well individual bearing units with critical tasks or short-term measurements during maintenance, but due to their price they are often an unrealistic solution regarding continuous monitoring on larger scale adaptations. (12.)

5.4 Challenges & Constraints

5.4.1 Operational Restrictions:

Regardless of the necessity of additional condition monitoring, the customer cannot afford excessive downtime. The applicable solution needs to take into account several restricting factors.

Each of the 72 shuttles are being used simultaneously, picking orders from the aisles and sending them to picking stations. Locking a shuttle virtually from the WMS restricts the customer from accessing any products contained on that aisle. This does not affect the main process or picking, since the system then prioritizes other orders from the other aisles. It is still none the less unpreferable to maintain shuttles unavailable for longer times than necessary.

5.4.2 Spatial Restrictions

Since the Navette system was designed as an entity, there is not much room for retrofit or post-installation. The bearing blocks for the main drives are located at the top of the shuttle structure on the same level as the upper guiding rail. The proximity from the bearing block to the floor of the aisle above is four millimeters. (With exception of the eighth floor) This leaves very little room for expansion.

5.4.3 Electrical Restrictions

The shuttles operate using a three phase 400VAC voltage supply for the main drives as well as 24VDC and 48VDC for the other components. The transformers are located inside an electrical cabin attached to the shuttle. The cabin has very limited space for further installation but provides connectors for a 24VDC power cable.

5.4.4 Networks and Platforms

The shuttles operate on an aisle length of 50 meters. The shuttles are controlled remotely using a wireless network. A local network is provided on the site with three base stations located throughout the rack system. This network can be used for wireless communication between condition monitoring and the client.

5.4.5 Mounting Restrictions

Due to the structure of the bearing block, no additional holes for mounting can be drilled. The block contains, however, two unused M12 mounting holes that can be used for expansive purposes.

5.4.6 Costs

Since the system consists of 72 Navette shuttles, each containing four bearings connected to their main drives, it is unrealistic to invest in extensive condition monitoring for each of these bearings without proper research into the possible solutions.

5.5 Conclusion

The solution needs to be implemented to cover these restrictions and to be applied within the limits accordingly. By focusing a prototype system on single Navette-shuttle, as to analyze the possibilities and efficiency of these hypothetical systems in practice, more valuable information and data can be acquired. This data can then be compared to the other shuttles to determine further configuration and necessity of condition monitoring on a larger scale.

In addition to the analytic benefits of a prototype system, it can be applied as a framework for a larger scale adaptation. Functionalities of the prototype used in this thesis study are not limited to any specific components or platforms and can be extended using a variety of different hardware and software implementations. This thesis aims to provide practical information, examples and alternatives for future condition monitoring possibilities.

6 Prototype

6.1 Functional Pre-requisites

Considering the given restrictions of the system and the qualities of the different types of condition monitoring, a combination of temperature and acceleration monitoring seems to be the most ideal solution in this instance, mainly because of the reliability and cost efficiency of the said combination.

Temperature measurement works in this instance as a general indicator for the friction of the bearing unit. Increase in friction has a direct correlation with the temperature of the structure. Long term monitoring of the temperature can provide a general state of the bearing strain. In contrast to this, real-time triaxial acceleration monitoring provides information about the actual impacts or excessive vibration and resonance, caused by either detrition or external mechanical faults.

As the bearing faults are very localized to certain pressure points, the measurement needs to be focused on very specific positions to guarantee accurate monitoring, especially when monitoring acceleration. Though it is not possible to eliminate the noise caused by the surrounding structures completely, the measured vibration and impacts originating from specific points can be amplified by positioning the sensors in short proximity from the bearings.

To reduce further installation and configuration to a minimum, the monitoring process needs to work as a stand-alone adaptation that operates separately from the main automation system, but can be integrated if deemed necessary.

In addition to this, the mobility of the shuttles restricts the use of cables between the shuttles and the system, with the exception of the electrical cabin, which is implemented locally to the structure of each shuttle. For these reasons the condition monitoring needs to communicate wirelessly to the user.

A wireless network can be used to transmit sensor data to a server platform from which the data can be processed and illustrated accordingly. Connecting sensors to a wireless network requires a microprocessor platform, which in addition to data processing, can host a small server application. Such application could work as a medium between on-site maintenance and the sensor data to notify the maintenance about the state of the bearings or sudden failures. The data can then be logged and processed for further analysis by using methods such as FFT (Fast Fourier Transform) and spectral analysis (8).

The prototype solution needs to provide real time data concerning the condition of the bearings (containing triaxial acceleration and temperature measurements), preferably on a visual platform. To provide necessary resources for further monitoring, the prototype needs to log the measured data to a format that can be used in post-process analytics.

Additionally, the prototype needs to be designed in such a way that it can be used as an indicative tool for maintenance purposes, meaning that it should be interchangeable between the shuttles with minor effort. Since the application is limited to one shuttle at a time, the installation process and reattachment to other shuttles needs to be both easy and swift to reduce the unwanted downtime during relocation. Naturally these further implementations cannot interfere with the other processes or functions of the shuttle or cause any hazards during operation.

Mockups of the prototype system are presented in appendix 1.

6.2 Design

6.2.1 Main Functions

As mentioned earlier, data logging is needed to provide reliable means for predictive condition monitoring through analytics. Effective data logging requires the data to be stored in a way in

which it can later be recalled and processed extensively. The logged sensor data needs to be recorded with a sample rate high enough to guarantee precise results. Using a relational database in this instance would be unnecessary, since time is the only valid relation towards each type of measurement included in this prototype. Instead, a CSV file (Comma Separated Values) provides a simple and structured alternative that can be easily accessed through analytic tools and software alike. The downside of CSV logging is the opening of larger text files, but this can be countered with specific CSV editors, such as Excel.

Effective condition monitoring also requires real time monitoring to predict malfunction. Utilizing condition monitoring with corrective methods in response to these signs of malfunction, rules out the bearing failures before they occur. To provide effective representation of the actual condition of the bearings, the monitoring needs to be accurate, easy to access and up to date. This can be achieved with an easy to read visual interface.

A server-based platform is ideal, since the prototype needs to communicate to the maintenance wirelessly. A graphical representation in relation to time, derived from the sensor measurements, paired with numeric readings using SI units, provides a simple, but powerful diagnostic tools for condition monitoring. This server can be run by a microprocessor connected to the prototype.

A working prototype with condition monitoring capabilities in place can be used to monitor shuttles individually. The results can then be compared to derive key values and limits, from which additional thresholds can be calculated. The list containing all the shuttle bearing unit replacements performed can be used in aid to define these final thresholds. These thresholds can then be used for alarm triggers to inform maintenance about anomalies related to the bearing units.

6.2.2 Components

MPU-6050 Is combinatory MEMS sensor that contains a triaxial accelerometer, a gyroscope and temperature measurement all on one chip. The readings of the sensor are stored in the registers of the chip momentarily and can be read from the sensor using I^2C based communication

platform. I²C (Inter-Integrated Circuit) is a synchronous serial communication bus developed in 1982 by Philips. (13.)

For this prototype, two breakboards containing MPU-6050 chip are being used (one for both bearings in the drive unit). The breakboard contains eight pins:

- 3.3VDC +
- GND
- SDA
- SCL
- XDA
- XCL
- ADD
- INT

The pins that are essential to the functions of this prototype are the voltage supply pins (3.3VDC+ & GND), the I²C communication pins (SCL & SDA) and the address pin (ADD).

Raspberry Pi 3B + is a commercial 64-bit quad core microprocessor with 1,4GHz clock speed and 1GB memory. The connections include four USB 2.0 ports, an ethernet port, a micro SD slot, wireless LAN, Bluetooth, a 40-pin GPIO, and a HDMI-port. The board is powered with 5VDC input voltage supplied either with a micro USB-connector, or directly to the GPIO. In this instance, the board is powered using the micro USB-connector. (14.)

As mentioned before, a microprocessor is required to handle essential calculations and functions for monitoring, as well as maintaining a server application. Raspberry Pi 3B+ suits this purpose well due to its programmability, processing power, adaptability to wireless networks and I²C communication capabilities. Raspberry Pi 3B+ also has USB ports, which can be used to log data to an USB drive with a CSV file. USB drive with 32GB memory is adequate for this purpose.

In order to configure Raspberry pi 3B +, an operating system needs to be installed. Raspbian is a Debian Linux based operating system designed specifically for Raspberry Pi. This operating system can be installed on the Raspberry pi using a micro SD card configuration. (15.)

24VDC power supply can be derived from the shuttles electric cabin, but the voltage needs to be converted to 5VDC to match the power supply needs for the Raspberry Pi. This can be achieved by using a step-down DC-DC converter with a regulator.

6.3 Assembly

6.3.1 Mounting

The MEMS sensors require the installation to be aligned accordingly to correspond all three axes. For this reason, the prototype will be mounted on top of the main drive bearing unit with two M12 bolts to ensure correct alignment. The bearing unit already has proper slots for these bolts and the prototype can be installed swiftly to its correct position without additional alignment.

6.3.2 Power Supply

The electric cabin of the shuttle is right below the main drive bearing unit. A power cable with two wires can be connected between the cabins 24VDC power supply and the prototypes step-down converter.

6.3.3 Case

The case is designed to host all the circuitry and connections of the prototype with added cable connection for the power supply cable from the shuttles electric cabin. The case includes heightened corner screw positions for the boards, holes for the M12 bolt attachment and screw holes for a plastic cover. The case is designed to fit within the limits of the shuttle aisle without a risk to collide with the surrounding structures.

The case was drawn using SolidWorks software and printed at Metropolia automation laboratory at Myyrmäki using Ultimaker 3D printer. The case was printed using PLA plastic with 40 percent infill. 3D model of the case is in appendix 3.

6.3.4 Connections

The boards are connected using jumper connection wires and an additional breadboard. The breadboard has four connection points on to which the power supply (3.3VDC+, GND) and the I²C communication bus (SDA, SCL) are connected. The power supply and the I²C bus of the bradboard is connected directly to the RaspberryPi GPIO.

The MPU-6050 has a static I²C address (0x68), but it can be changed (to 0x69) by assigning current to the address pin (ADD) on the breakboard. Since the application of the prototype uses two MPU-6050 breakboards, the addresses need to be differentiated in order to ensure simultaneous communication on the I²C bus. This can be applied by connecting a jumper wire from the 3.3VDC+ pin to the address pin on either one of the breakboards. Electric diagram of the prototype is presented in appendix 2. and Raspberry Pi 3B+ GPIO in appendix 6.

6.4 Setup & Configuration

In order to create an application for the prototype, few essential conditions must be in place. First step is to install an operating system on the Raspberry Pi, which in this instance will be Linux Debian based Raspbian. This was done using 16GB memory card with Raspbian image file stored on it. After booting the Raspberry with the memory card attached, Raspbian could be installed by following the streamlined configuration process of the OS.

After configuring local wireless connection and setting up the right input language, the next step was to download all essential libraries and drivers as well as set all the necessary configurations in place. The necessary installations include:

- Linux I²C drivers (i2c-bcm2702 & i2c-dev, written in to the directory "/etc/modprobe.d/raspi-blacklist.conf")

- Latest Raspbian system update
- Latest Python 3 update (3.7.3)
- I2C-tools
- Python-smbus module
- NPM update
- NodeRed dashboard (Installed using NodeRed UI)
- NodeRed Python 3 function node (Installed using NodeRed UI)

In addition to the installation of essential libraries and drivers, SSH needed to be enabled from system configuration to allow remote accessto the Raspberry Pi.

6.5 Application

6.5.1 NodeRed

NodeRed is a development and programming environment originally developed by IBM. NodeRed is based on a flow-based browser editor that streamlines the interfacing and connections of hardware components and processes of IoT applications by providing a visual wiring platform. NodeRed operates on Node.js, a JavaScript-based runtime environment that is constructed on an event driven, non-blocking I/O model. (16.)

Similiarly to Node.js, Node-Red divides its components into nodes, which operate as functional objects that communicate with each other. Node-Red has a selection of built-in nodes, providing essential tools for basic interfacing applications. In addition to this, a variety of custom nodes provided by the community exist to serve more specific needs. One example of these nodes is the Python code function node, which is an alternative to the built-in function node that uses JavaScript programming language (17). This node is used in this application as a function node for the main program.

Communication between the nodes is carried out by using JavaScript-based MGS objects with freely determinable properties. These objects usually have a pre-determined payload property, which operates as the default property used by most of the preset nodes.

NodeRed dashboard is a downloadable node library on NodeRed that contains utility nodes designed specifically for UI (User Interface) purposes. These nodes provide tools for presenting data visually in forms of graphs, gauges and charts. Additionally, the dashboard contains nodes for user input, including buttons, sliders and input fields. (18.)

The dashboard can be opened using the same URL that the NodeRed configuration interface uses with the exception of "/ui/" added to the end of the URL.

In appendix 4. we can see the NodeRed flow that is used in the application of the prototype. The flow presented here communicates with both MPU-6050 sensors used in this prototype. The flow consists of six different sections:

1. Timestamp is a node that outputs Unix time related data. In addition to this it can be used as a trigger to activate other nodes or functions. In this application, the Timestamp node triggers the function nodes (appendix 4., section 2.) and provides the CSV nodes (appendix 4., section 6.) with Unix time. The triggering can be set to cycle on a freely determinable sample rate.
2. The function nodes used in the flow are contributed Python 3 nodes. They contain a small Python program (appendix 5.) that reads the sensor data from the MPU-6050 directories and converts it to applicable format.
3. The MSG topics declared in the python program (appendix5.) need to be converted to msg.payload properties in order to interface them with the graph nodes (appendix 4., section 5.) and the CSV nodes (appendix 4., section 6.), since these nodes use the msg.payload property by default. This is done by using a change node to convert each sensor topic to an individual payload that communicates with its corresponding output (graph and CSV nodes).
4. Nodes indicated in this section are gauge nodes that provide visual output data for the UI-platform. The data is derived from the sensor data provided by the function nodes (section 2.).
5. These graph nodes appear on the UI platform as well. Each MSG topic is paired with a graph node to provide time relational data in visual form.
6. These nodes compile the sensor data and Unix time into a CSV format. This is done using the join nodes and a CSV node. The CSV file is then saved to a file using file

nodes. The directory appearing on the file nodes refers to a USB-drive on to which the CSV files are stored.

The UI view hosted by the server of this application is presented in appendix 9.

6.5.2 Python Code

The python program presented here is a modification of an existing program, originally created by Andrew Brikett (19).

The code used in this application is presented in appendix 5. And consists of eight sections:

1. Libraries that are needed in this program are smbus, math and decimal. Smbus is a library needed for I²C communication, math library is used for mathematical operations and decimal is needed for rounding calculated results.
2. Power management register are used for configuring power modes and clock sources. They are defined here for later use in the code.
3. This section defines the read functions and conditions
4. This section determines the buses and the right I²C address to read. (0x68 for sensor 1. and 0x69 for sensor 2.)
5. In this section the program reads the raw accelerometer and temperature data from the sensor
6. Here the the program converts the sensor data to appropriate SIunits: Celsius and Gravitational force equivalent (g). Additionally, earths gravitational acceleration is factored out by subtracting one unit from the Y-axis. This is because the sensors are oriented vertically in relation to the gravitational force.
7. This section creates the msg-object containing sensor data.

6.6 Installation

First installation of the prototype was done 12.9.2019 by the local resident maintenance team for the shuttle number N1118 on the eighth level of the Navette rack. For this procedure, the shuttle needed to be locked virtually from Wamas.

The prototype was sealed with a see through plastic cover with diameters 182x127x1 in millimeters. The prototype was mounted with two M12 bolts on top of the bearing unit as it was originally designed. The 32GB USB drive was attached to one of the ports and the rest of the exterior were covered with tape to insulate the ports and internal circuits from dust. A 2x0.75 mm. power cable was pulled in to the electric cabin of the shuttle through the upper inlet of the cabin. The power cable was routed directly to the 24VDC power supply. The installed prototype and routing in the electric cabin are presented in figure 4.

Figure 4. Installed prototype and routing inside the electric cabin to the 24VDC power supply.

After the cable was connected and secured with cable ties along the routings in the electric cabin the power was switched on. The shuttle was moved manually along the aisle to ensure that the installation did not cause a risk of collision to the other structures of the aisle. Tests were made to confirm the successful operation of the prototype and the server hosted by it. The functionalities and connections worked as intended and the UI (see appendix 9.) could be accessed from the control room via local wireless network.

After the installation process was complete the virtual lock of the shuttle was unlocked and the shuttle was put on automatic mode.

7 Analysis

7.1 Data Preparation

On 14.9.2019 more detailed analytics were carried out on the Navette N1118 by a scheduled logging phase on the span of five hours from 12:00PM to 17:00PM during production. The sampling on the timestamp node presented in the flow of appendix 4. was set on one second interval and deployed on 11:57AM. The logging was halted at 17:00PM and the USB drive was removed on the following day for further inspection of the data acquired during the monitoring phase.

The local automation system logs the total distances travelled on each axis of the shuttle. The distance travelled on the X-axis along the aisle was at the total of 19 631 213 meters at 12:00PM. The distance was checked again at 17:00PM with a value of 19 645 034 meters. The subtracted difference of these values equals 13 821 meters (13,821) kilometers, which is the total distance the shuttle travelled along the aisle from 12:00PM to 17:00PM.

The data generated during the monitoring phase was automatically saved on the USB drive in CSV format by the NodeRed application (see appendix 4.). The data was later retrieved from the USB drive and graphed in Excel. The data conversion from Unix time to readable format was

done using the following equation, where "A" represents the registered Unix timestamps from 12:00PM to 17:00PM with one second interval and "T" represents the time difference from UTC, which in Finland is UTC+3/2. The equation (1) is an altered version of an equation presented by Allen Wyatt (20).

$$A_1 \ldots A_n/1000 \,/\, 86400 \; + \; 25569 \; + \; T/24$$

The original CSV files retrieved from the USB drive needed to be spliced in order to eliminate measurements outside the timeframe of the monitoring phase. The Excel files containing the sensor data are presented in appendices 7. and 8. The order of measurements is organized on the columns as follows:

- Column A, Unix timestamp
- Column B, Temperature readings
- Column C, X-axis acceleration (g)
- Column D, Y-axis acceleration (g)
- Column E, Z-axis acceleration (g)
- Column G, calculated time format using formula Allen Wyatt's formula
- Column V, X-axis conversion from gravitational force equivalent to m/s²
- Column W, Y-axis conversion from gravitational force equivalent to m/s²
- Column W, Z-axis conversion from gravitational force equivalent to m/s²

7.2 Data Analysis

From the sensor data provided by appendices 6. and 8. can be observed the behavior of the temperature and the axial movements of the shuttle on automatic mode. The units of the temperature are presented here in Celsius and the units of acceleration are in the form of m/s². The bipolar increase in acceleration indicates the movement of the shuttle along the aisle, whereas the stabilization of these readings indicates downtime of the shuttle, as is seen at 13:00PM.

Some noise in the temperature readings was caused by subtle changes in the temperature of the internal circuits, but the essential trends are visible none the less. Correlation between temperature and runtime can be verified by comparison of the graphs: A steady increase in temperature caused by long runtime of the shuttle (12:00PM to13:00PM), followed by a drop in the temperature due to the downtime of the shuttle (13:00PM to13:30PM).

The graph containing the acceleration data from the X-axis contains the most intensive readings. This is due to the fact, that the X-axis is the shuttles main direction of movement, whereas the other axes indicate movement caused by transient shock originating from vibration and collisions of the shuttle. These transient shocks manifest as short peaks of acceleration on the Y- and Z-axes.

Although some vibration is expected to occur during the movement of large machines such as the Navette shuttle, the sensor data provided from the monitoring phase revealed some concerning insights. The acceleration on Y and Z axes peaks from eight to ten m/s² occasionally. This amount is over four times higher than the actual running speed on X-axis along the aisle.

7.3 Summary

7.3.1 Condition Monitoring Conclusions

From the information and data acquired during this thesis study, several things can be concluded.

- Condition monitoring is starting to become a necessity in industrial applications containing extensive mechanical systems, due to the effectiveness of predictive maintenance.
- The localized bearing failures at the Stockmann site are too cost heavy to be remedied by using preventive maintenance model.

- Due to the cost efficiency of the new generation sensor technology, condition monitoring is becoming more available and efficient solution for implementation on maintenance strategies.

7.3.2 Prototype Conclusions

The succession of the prototype can be examined by the correspondence to the original needs, design and scope, which can be divided in to individual categories:

- The outcome of the prototype corresponds the original functional pre-requisites and designs and thus provides the essential condition monitoring capabilities.
- The prototype was installed in a short time window to minimize the downtime of the production.
- The prototype works as a stand-alone application, functioning separately from the local automation system, but can be integrated in the future.
- The prototype fits to the required dimensions appropriately
- The power supply of the shuttle cabin was sufficient for the operation of the prototype.
- The prototype maintains the wireless connection to the local network during monitoring
- The mounting proved sufficient and provided stable and well aligned base for the MEMS sensors and monitoring.
- The prototype is interchangeable between the shuttles and can be used as an indicative tool during maintenance and service
- Entirety of the material costs did not exceed a budget of 100 euros, which is a minimal sum for the utility and benefit provided by the condition monitoring.
- The prototype functionalities are not limited to specific components or platforms and can be revised according to the standards of the company and the needs of the system.

7.3.3 Future possibilities

The data and research contained in this thesis advocates greatly towards future investments and condition monitoring. That being said, such investments, while being more affordable than ever

before, are still large investments due to the scale of the mechanical systems and the multitude of components involved. However, considering the costs caused by the localized bearing failures, some counteractive measures are in order. The prototype of this thesis study, while not in itself a realistic solution for larger scale implementation, provides the groundwork and essential functionalities for the design of applications consisting of greater multitudes, like in the case of the Stockmann warehouse site, consisting of 72 Navette shuttles.

The methods of spectral analysis and FFT require data with much higher sample rate to provide information. A separate monitoring phase will be arranged for this purpose by a shorter monitoring period with high sampling frequency of 100 samples per second for the span of 15 minutes during automatic runtime. This will provide representation of subtle low frequency resonance frequency response, which can be used for further analysis.

In addition to the data presented on this thesis (appendices 7. and 8.), further monitoring needs to be carried out on the other shuttles as well to guarantee more accurate results and conclusions. To provide authentic condition monitoring, a firm base for reference needs to be in place. Comparison between at least five shuttles will be organized in the future to present reliable threshold values for further implementation. These thresholds are advantageous considering the possibility of integration to the main automation system, providing functions like alarms to trigger processes and interfaces used by the system.

References

1 What is IIoT? Inductive automation [internet]. 2018 Jul [cited 2019 Sept.]; available from: https://inductiveautomation.com/resources/article/what-is-iiot

2 Ralph Rio. Optimize asset performance with industrial IoT and analytics. ARC. 2015 Aug [cited 2019 Sept]; available from: https://www.ibm.com/downloads/cas/AVMWDMGZ

3 James Carbone. Sensor market to grow despite price declines. Electronics sourcing . [internet]. 2018 Mar [cited 2019 Sept]; Available from: http://www.electronics-sourcing.com/2018/03/28/sensor-market-to-grow-despite-price- declines/

4 SSI Schäfer. homepage [internet]. 2015 Jul [cited 2019 Sept]; available from: https://www.ssi-schaefer.com

5 Stockmann Oyj. homepage [internet]. 2014 [cited 2019 Sept]; available from: http://www.stockmanngroup.com/en/en

6 SSI Schäfer. Stockmann Navette system [internet]. 2015 [cited 2019 Sept]; available from: https://www.ssi-schaefer.com/fi- fi/market- sectors/fashion-logistics/case-study- stockmann-130400

7 SSI Schäfer. Wamas [internet]. 2015 [cited 2019 Sept]; available from: https://www.ssi-schaefer.com/fi-fi/ohjelmistoratkaisut/ohjelmistotuotteet/wamas-132044

8 Brian P. Graney & Ken Starry. Rolling Element Bearing Analysis. The American society for nondestructive testing. 2012 Jan

9 Danielle Collins. What is L10 life and why does it matter? Linear motion tips [internet]. 2017 Feb [cited 2019 Sept]; available from: https://www.linearmotiontips.com/what-is-l10-life-and-why-does-it-matter/

10 EP Editorial staff. Methods for monitoring bearing performance. Efficient Plant [internet]. 2013 Oct [cited 2019 Sept]; available from: https://www.efficientplantmag.com/2013/10/methods-for-monitoring-bearing-performance/

11 Mark Looney. MEMS Vibration Sensing: Velocity to Acceleration. Analog Devices Jun 2017

12 Alan Bandes. Ultrasound condition monitoring. UE systems, Inc; 2014

13 InvenSense Inc. MPU-6000 and MPU-6050 product specification: Revision 3.4. InvenSense; 2013

14 Raspberry Pi foundation. Raspberry Pi 3 model B+ product brief. 2019

15 Raspberry Pi foundation. Raspbian documentation [internet]. 2019 Jan [cited 2019 Sept]; available from: https://www.raspbian.org/

16 OpenJs Foundation. NodeRed Documentation [internet]. 2019 Sept [cited 2019 Sept]; available from: https://nodered.org/docs/

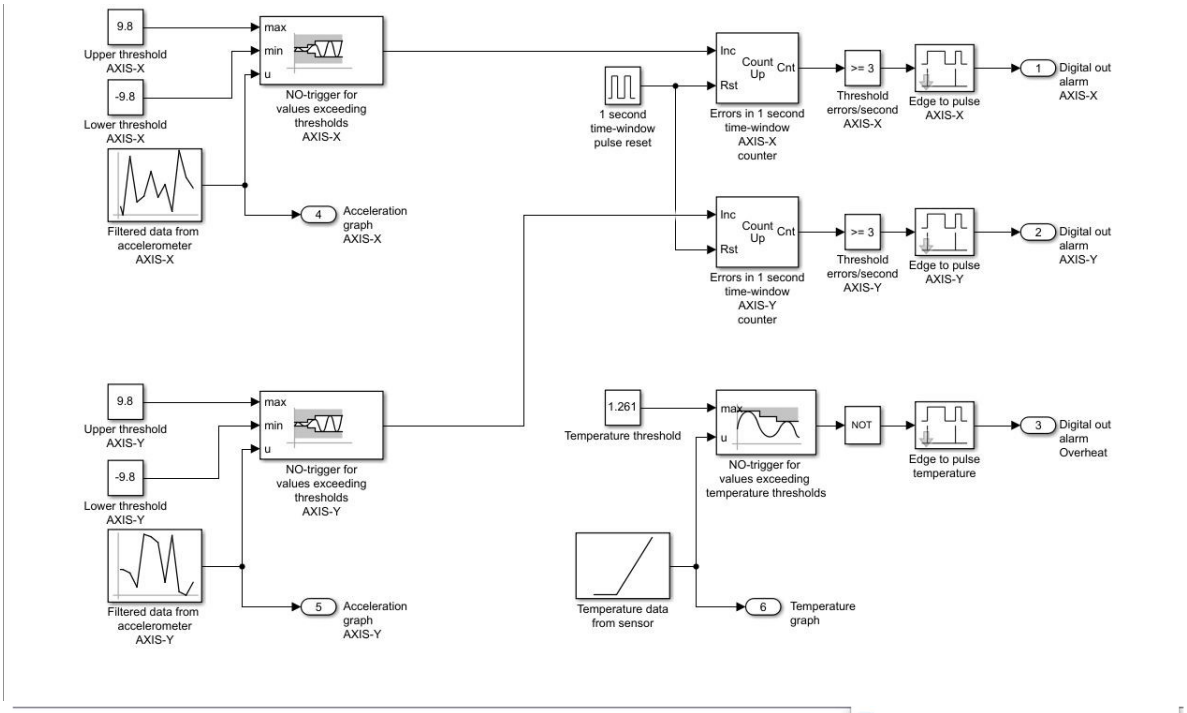
17 Dejan Rizvan. NodeRed Contrib. Python 3 Function node [Internet]. 2017 Jan [cited 2019 Sept]; available from: https://flows.nodered.org/node/node-red-contrib-python3-function

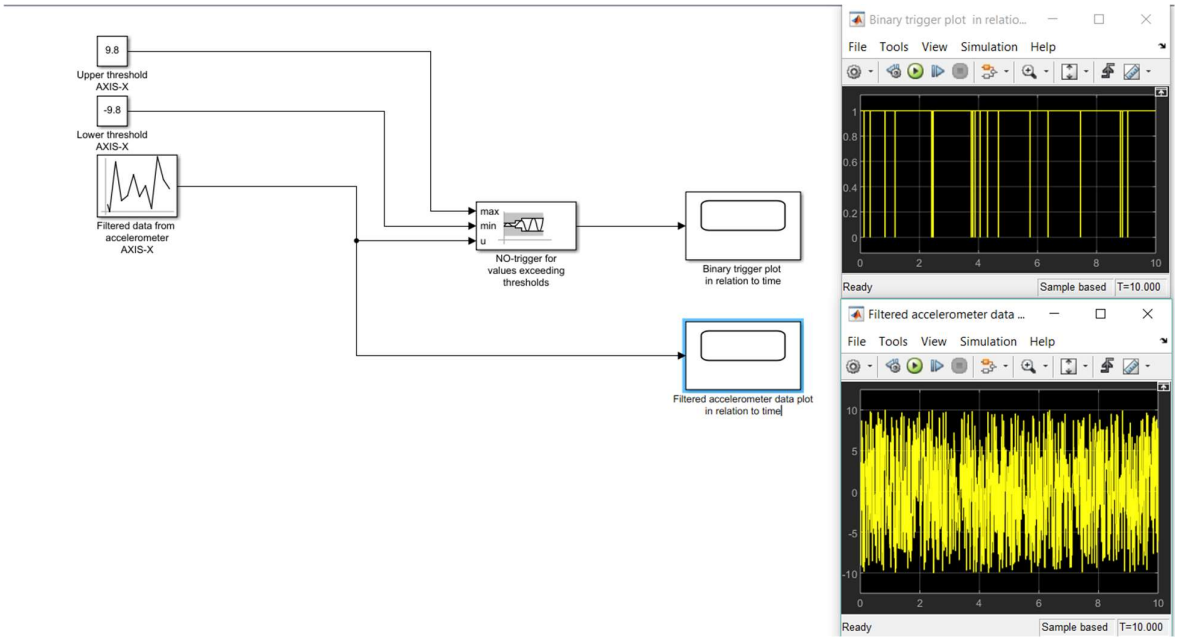
18 NodeRed. Dashboard-node documentation [internet]. 2019 Sept [cited 2019 Sept]; available from: https://flows.nodered.org/node/node-red- dashboard

19 Andy Brikett. Reading Data from the MPU-6050 on Raspberry Pi. Bitify [internet]. 2013 Nov [cited 2019 sept]; available from: http://blog.bitify.co.uk/2013/11/reading-data-from-mpu-6050-on-raspberry.html

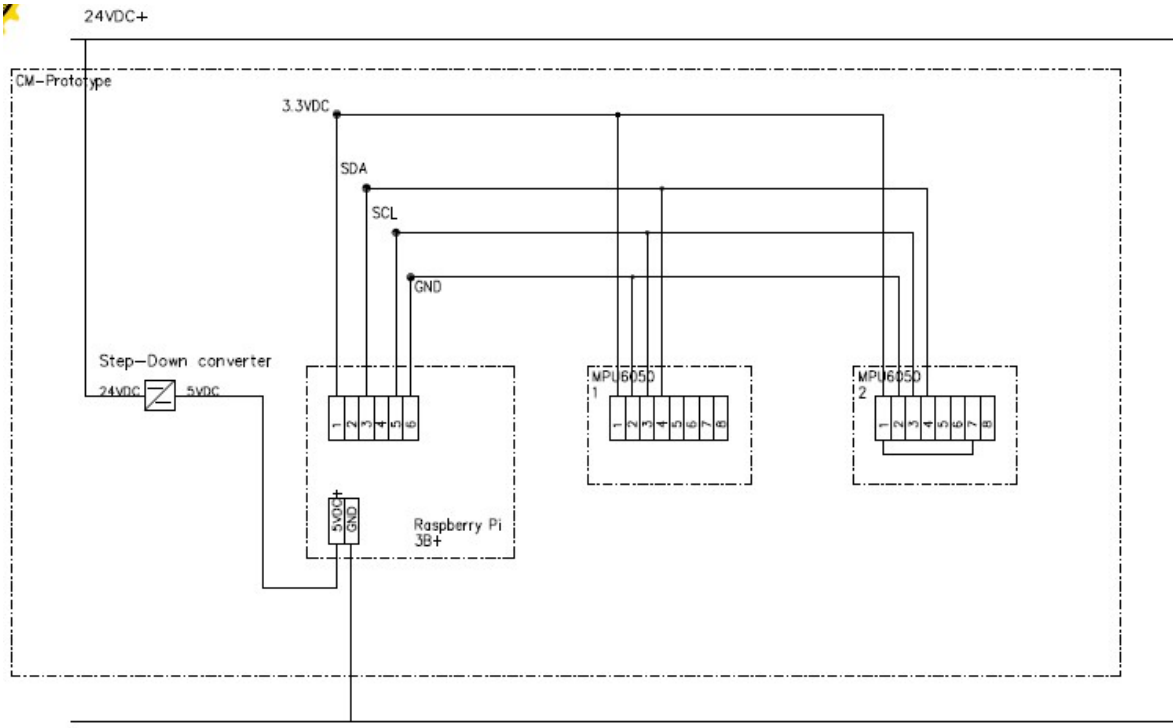
20 Allen Wyatt. Converting UNIX date/Time stamps; Excel.tips.net [internet] . 2018 Feb [cited 2019 Sept]; available from: https://excel.tips.net/T002051_Converting_UNIX_Date_Time_Stamps.html

Simulink mockup and simulation of the prototype functions

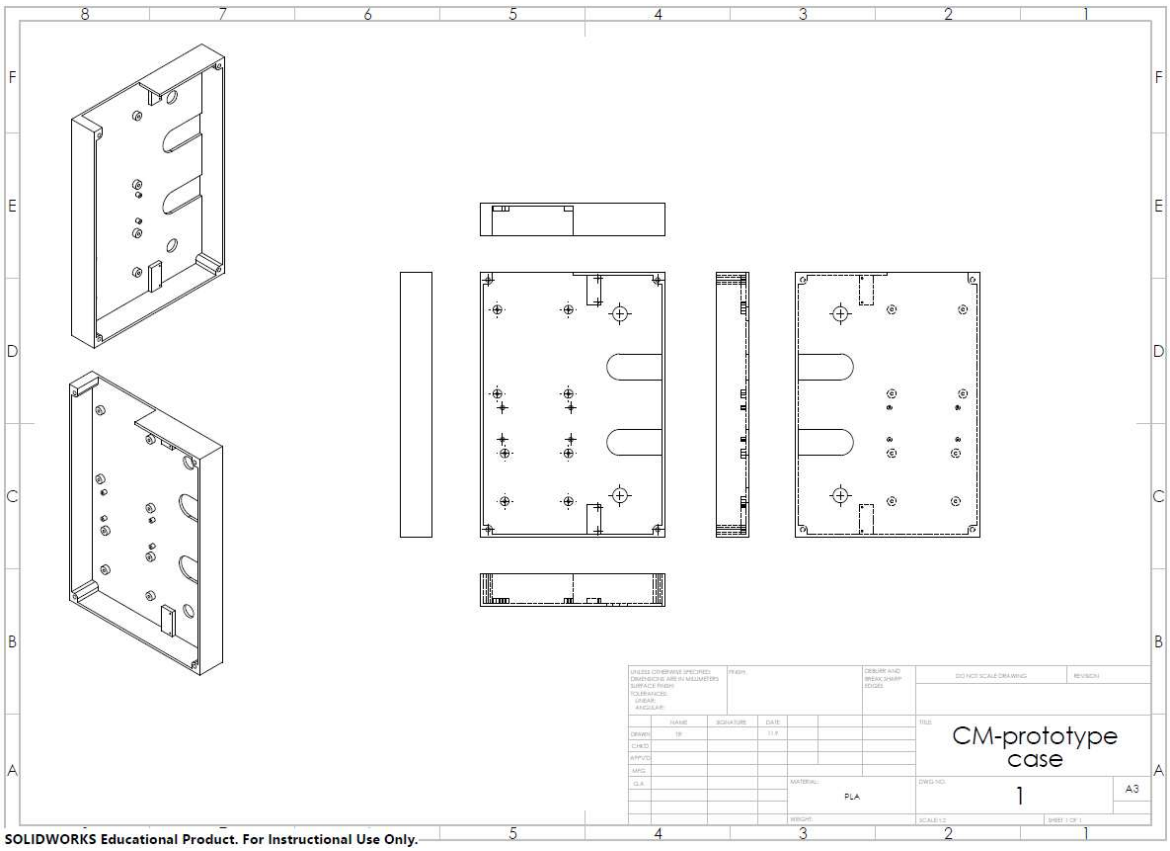




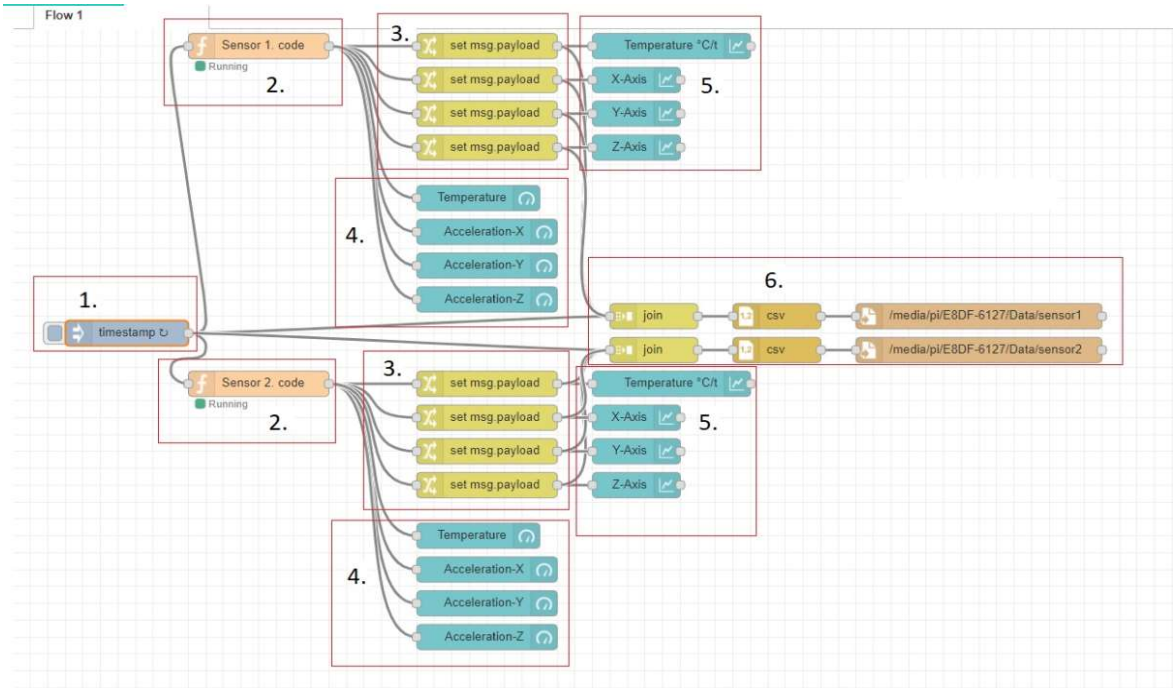
Prototype wiring diagram



Prototype case drawings & 3D model



NodeRed flow



Python code

1.

```
#importing all essential libraries
import smbus

import math

from decimal
import Decimal
```

2.

```
#Assigning power management registers

power_mgmt_1 = 0x6b

power_mgmt_2 = 0x6c
```

3.

```
#Defining read-conditions

def read_byte(adr):

    return bus.read_byte_data(address, adr)


def read_word(adr):

    high = bus.read_byte_data(address, adr)

    low = bus.read_byte_data(address, adr+1)

    val = (high << 8) + low

    return val


def read_word_2c(adr):

    val = read_word(adr)

    if (val >= 0x8000):

        return -((65535 - val) + 1)

    else:

        return val
```

4.
```
bus = smbus.SMBus(1) # assigning bus
address = 0x68       # This is the address value read via the i2cdetect command. Alternatively 0x69
# Waking up the 6050 up as it starts in sleep mode

bus.write_byte_data(address, power_mgmt_1, 0)
```

5.
```
#Here the program reads the raw accelerometer and temperature data from the sensor

accel_xout = read_word_2c(0x3b)

accel_yout = read_word_2c(0x3d)

accel_zout = read_word_2c(0x3f)

temperature = read_word_2c(0x41)
```

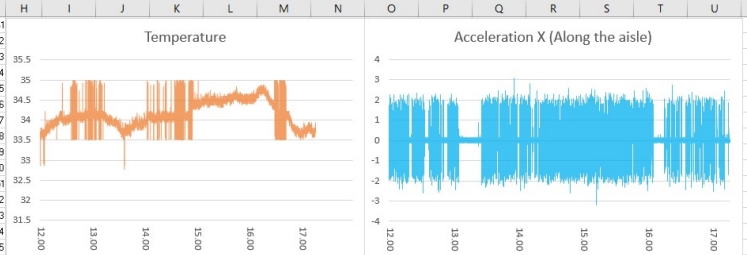
6.
```
#Here the raw data is converted to SI-units. Namely to m/s^2 and Celsius.


accel_xout_scaled = Decimal (accel_xout / 16384.0)

accel_yout_scaled = Decimal (accel_yout / 16384.0)

accel_zout_scaled = Decimal (accel_zout / 16384.0)

temperature_scaled= Decimal (temperature / 340.0 + 36.53)
```

7.
```
#Here values are rounded to set decimal limits
accel_xout_scaled2 = round(accel_xout_scaled,4)

accel_yout_scaled2 = round(accel_yout_scaled,4)

accel_zout_scaled2 = round(accel_zout_scaled,4)

temperature_scaled2 = round(temperature_scaled,2)
```

8.
```
#Here the msg output is formed topic by topic

msg =   {

 "x": accel_xout_scaled2,

 "y": accel_yout_scaled2,

 "z": accel_zout_scaled2,

 "t": temperature_scaled2
}

return msg
```

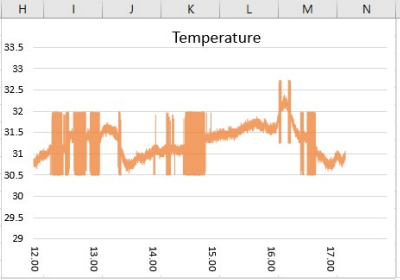
Raspberry Pi 3B+ GPIO diagram

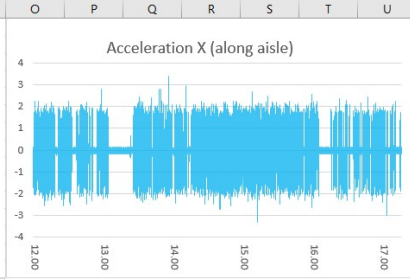
Sensor 1. Data from 13.9.2019

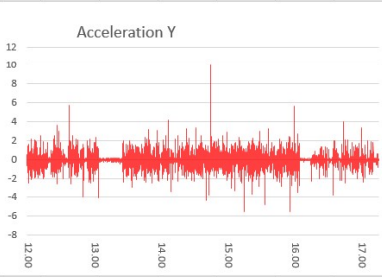


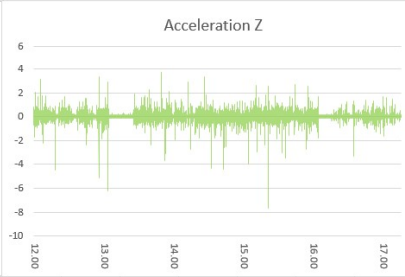
	A	B	C	D	E	F	G	V	W	X
1	1.568E+12	33.61	-0.1091	-0.0906	-0.0488		11:58:41	-1.522453	-0.5606	-0.1497
2	1.568E+12	33.61	-0.041	-0.0542	-0.0928		11:58:42	-0.85303	-0.20279	-0.58222
3	1.568E+12	33.66	0.0457	-0.001	-0.0598		11:58:43	-0.000769	0.32017	-0.25783
4	1.568E+12	33.61	0.075	-0.0435	-0.075		11:58:44	0.28725	-0.0976	-0.40725
5	1.568E+12	33.71	0.0447	-0.0178	-0.033		11:58:45	-0.010599	0.15503	0.00561
6	1.568E+12	33.57	0.0515	-0.0364	-0.05		11:58:46	0.056245	-0.02781	-0.1615
7	1.568E+12	33.52	0.053	-0.0557	-0.0256		11:58:47	0.07099	-0.21753	0.07835
8	1.568E+12	33.66	0.0515	-0.0251	-0.0354		11:58:48	0.056245	0.08327	-0.01798
9	1.568E+12	33.61	0.0537	-0.0356	-0.0747		11:58:49	0.077871	-0.01995	-0.4043
10	1.568E+12	33.66	0.053	-0.0168	-0.062		11:58:50	0.07099	0.16486	-0.27946
11	1.568E+12	33.66	0.2261	-0.0459	-0.1399		11:58:51	1.772563	-0.1212	-1.04522
12	1.568E+12	32.86	-0.1201	0.0552	-0.0862		11:58:52	-1.630583	0.87262	-0.51735
13	1.568E+12	33.71	0.0352	-0.0352	-0.0342		11:58:53	-0.103984	-0.01602	-0.00619
14	1.568E+12	33.66	-0.05	-0.0901	-0.0405		11:58:54	-0.9415	-0.55568	-0.06812
15	1.568E+12	33.66	0.0962	-0.0239	-0.0862		11:58:55	0.495646	0.09506	-0.51735
16	1.568E+12	33.61	0.0522	-0.0413	-0.0706		11:58:56	0.063126	-0.07598	-0.364
17	1.568E+12	33.57	0.0359	-0.0305	-0.0176		11:58:57	-0.097103	0.03019	0.15699
18	1.568E+12	33.66	0.05	-0.0454	-0.0591		11:58:58	0.0415	-0.11628	-0.25095
19	1.568E+12	33.61	0.0544	-0.0535	-0.0625		11:58:59	0.084752	-0.19591	-0.28438
20	1.568E+12	33.66	0.0667	-0.0049	-0.0457		11:59:00	0.205661	0.28183	-0.11923
21	1.568E+12	33.52	-0.1563	-0.0154	0.0164		11:59:01	-1.986429	0.17862	0.49121
22	1.568E+12	33.66	-0.0049	0.053	-0.0293		11:59:02	-0.498167	0.85099	0.04198
23	1.568E+12	33.52	0.0344	-0.147	-0.0986		11:59:03	-0.111848	-1.11501	-0.63924
24	1.568E+12	33.71	0.0728	-0.0352	-0.0747		11:59:04	0.265624	-0.01602	-0.4043
25	1.568E+12	33.66	0.0374	-0.0776	-0.0806		11:59:05	-0.082358	-0.43281	-0.4623
26	1.568E+12	33.61	0.0386	-0.0198	-0.0234		11:59:06	-0.070562	0.13537	0.09998
27	1.568E+12	33.66	0.0764	0.1335	-0.0479		11:59:07	0.301012	1.64231	-0.14086
28	1.568E+12	33.71	0.0232	0.0442	-0.1257		11:59:08	-0.221944	0.76449	-0.90563
29	1.568E+12	33.66	0.0159	0.0698	0.0298		11:59:09	-0.293703	1.01613	0.62293
30	1.568E+12	33.71	0.0581	-0.0298	-0.0383		11:59:10	0.121123	0.03707	-0.04649
31	1.568E+12	33.61	0.0647	-0.1245	-0.0466		11:59:11	0.186001	-0.89384	-0.12808
32	1.568E+12	33.71	0.0486	-0.074	-0.042		11:59:12	0.027738	-0.39742	-0.08286
33	1.568E+12	33.61	0.1204	-0.0459	-0.0715		11:59:13	0.733532	-0.1212	-0.37285
34	1.568E+12	33.61	0.054	-0.1243	0.0046		11:59:14	0.08082	-0.89187	0.37522
35	1.568E+12	33.61	0.0789	-0.0242	-0.0144		11:59:15	0.325587	0.09211	0.18845
36	1.568E+12	33.8	0.0586	-0.0654	-0.0415		11:59:16	0.126038	-0.31288	-0.07795
37	1.568E+12	33.75	0.0364	-0.0898	-0.0337		11:59:17	-0.092188	-0.55273	-0.00127
38	1.568E+12	33.66	0.0667	-0.0002	-0.0366		11:59:18	0.205661	0.32803	-0.02978
39	1.568E+12	33.71	0.2498	-0.0186	-0.074		11:59:19	2.005534	0.14716	-0.39742

Sensor 2. Data from 13.9.2019









	A	B	C	D	E	F	G	V	W	X
1	1.568E+12	30.88	-0.1401	-0.0354	-0.0715		11:58:41	-1.70718	-0.34798	-0.70285
2	1.568E+12	30.74	-0.0339	-0.0085	-0.0022		11:58:42	-0.66324	-0.08356	-0.02163
3	1.568E+12	30.79	0.0283	0.0208	-0.0198		11:58:43	-0.05181	0.20446	-0.19463
4	1.568E+12	30.79	0.05	-0.0005	-0.0142		11:58:44	0.1615	-0.00492	-0.13959
5	1.568E+12	30.79	0.0447	0.0293	0.0005		11:58:45	0.1094	0.28802	0.00492
6	1.568E+12	30.84	0.041	-0.012	-0.0134		11:58:46	0.07303	-0.11796	-0.13172
7	1.568E+12	30.79	0.0374	-0.0313	-0.0273		11:58:47	0.03764	-0.30768	-0.26836
8	1.568E+12	30.79	0.0308	-0.0146	0.0127		11:58:48	-0.02724	-0.14352	0.12484
9	1.568E+12	30.84	0.0366	-0.0151	-0.0015		11:58:49	0.02978	-0.14843	-0.01475
10	1.568E+12	30.79	0.0361	-0.0103	-0.0117		11:58:50	0.02486	-0.10125	-0.11501
11	1.568E+12	30.74	0.2087	0.0337	-0.0305		11:58:51	1.72152	0.33127	-0.29982
12	1.568E+12	30.88	-0.1709	0.0505	0.0105		11:58:52	-2.00995	0.49642	0.10322
13	1.568E+12	30.74	0.0298	0.0408	-0.0037		11:58:53	-0.03707	0.40106	-0.03637
14	1.568E+12	30.79	-0.1047	-0.0686	0.0029		11:58:54	-1.3592	-0.67434	0.02851
15	1.568E+12	30.79	0.0818	0.0242	-0.0259		11:58:55	0.47409	0.23789	-0.2546
16	1.568E+12	30.84	0.0381	-0.0071	-0.0095		11:58:56	0.04452	-0.06979	-0.09339
17	1.568E+12	30.88	0.0327	0.0134	0.0215		11:58:57	-0.00856	0.13172	0.21135
18	1.568E+12	30.84	0.0466	-0.0098	0.0061		11:58:58	0.12808	-0.09633	0.05996
19	1.568E+12	30.79	0.0449	-0.021	0.0125		11:58:59	0.11137	-0.20643	0.12288
20	1.568E+12	30.69	0.0464	-0.0076	-0.0071		11:59:00	0.12611	-0.07471	-0.06979
21	1.568E+12	30.84	-0.1616	0.0591	-0.0269		11:59:01	-1.91853	0.58095	-0.26443
22	1.568E+12	30.84	-0.0171	0.1011	0.0825		11:59:02	-0.49809	0.99381	0.81098
23	1.568E+12	30.84	0.0283	-0.0913	-0.0493		11:59:03	-0.05181	-0.89748	-0.48462
24	1.568E+12	30.84	0.0505	0.0249	-0.0464		11:59:04	0.16642	0.24477	-0.45611
25	1.568E+12	30.84	0.0303	0.0271	-0.0024		11:59:05	-0.03215	0.26639	-0.02359
26	1.568E+12	30.69	0.0518	-0.0073	0.0334		11:59:06	0.17919	-0.07176	0.32832
27	1.568E+12	30.84	0.0464	0.1431	0.0266		11:59:07	0.12611	1.40667	0.26148
28	1.568E+12	30.69	-0.0027	0.0977	-0.1716		11:59:08	-0.35654	0.96039	-1.68683
29	1.568E+12	30.74	0.0083	-0.0505	0.0361		11:59:09	-0.24841	-0.49642	0.35486
30	1.568E+12	30.74	0.0161	0.02	0.0613		11:59:10	-0.17174	0.1966	0.60258
31	1.568E+12	30.79	0.0522	-0.0408	0.0137		11:59:11	0.18313	-0.40106	0.13467
32	1.568E+12	30.84	0.0251	-0.0427	0.0566		11:59:12	-0.08327	-0.41974	0.55638
33	1.568E+12	30.74	0.0752	-0.0066	-0.033		11:59:13	0.40922	-0.06488	-0.32439
34	1.568E+12	30.74	0.0432	-0.1011	0.0032		11:59:14	0.09466	-0.99381	0.03146
35	1.568E+12	30.84	0	-0.0615	-0.0066		11:59:15	-0.33	-0.60455	-0.06488
36	1.568E+12	30.79	0.0479	-0.0935	-0.0439		11:59:16	0.14086	-0.91911	-0.43154
37	1.568E+12	30.74	-0.0144	-0.02	0.0044		11:59:17	-0.47155	-0.1966	0.04325
38	1.568E+12	30.74	0.0308	-0.01	0.0273		11:59:18	-0.02724	-0.0983	0.26836
39	1.568E+12	30.79	0.2087	-0.0715	-0.0015		11:59:19	1.72152	-0.70285	-0.01475

CM-Prototype UI