Olli Johannes Raudaskoski

Embedded solution for fast axial measurement

Embedded IOT system applied to paper indstry

Helsinki Metropolia University of Applied Sciences

Engineering degree

Information Technology

Embedded solution for fast CD measuring

24 April 2018

Author Title Number of Pages Date	Olli Johannes Raudaskoski Embedded solution for fast axial measurement 39 pages 3 appendices 5 June 2017
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Smart Systems
Instructor(s)	Thomas Widmaier, Project Manager Kimmo Saurén, Lecturer

The thesis implements a handheld measuring device for measuring roll diameter. The device is built to international paper industry consultant company. The device creates a measurement that gives a preliminary assessment about the shape of the roll. Measurement is performed by moving the measurement device parallel to the rolls height. Therefore, measurements done in the thesis only measure diameter variations perpendicular to the height of the cylinder-shaped roll.

The thesis device has two rods. Movable mount is attached to each rod so that the device can be adjusted for different roll sizes. attached to the other rod is a dial indicator. When the contact point of the dial indicator is moved the distance is passed to the device in a micro meter precision. in the other rod, there is an adjustable counterpart that keeps the diameter constant.

When measurement is performed the device gives a two-dimensional graph of the roll's surface. The measurement does not require the paper roll to be moved and the device gives an instant report of the measurement. The device built in this thesis only has one measurement point but the software is built so that maximum of four measurement points can later be added to the device. Each measurement point needs to be manually configured before use.

A touchscreen display is used for HMI. The software runs on Rapberry Pi 3 platform in a Windows IOT Core operating system. The Windows was selected as an operating system so that algorithms developed by the ordering company can be implemented easily. The visual compatibility with other software from the company that ordered the thesis was considered when selecting the operating system.

The device was tested with a small roll. And later in paper factory. The test concluded that the device operated as wanted and the results resembled measurements performed by larger machines. The ordering company then took the machine in use.

Keywords	paper industry, dial indicator, rotational encoder, CD-measurement

Tekijä Otsikko Sivumäärä Aika	Olli Johannes Raudaskoski Sulautettu paperitelan mittaus 39 sivua 3 liitettä 5.10.2017
Tutkinto	insinööri (AMK)
Tutkinto-ohjelma	tieto- ja viestintätekniikka
Ammatillinen pääaine	Smart Systems
Ohjaajat	Projektipäällikkö Thomas Widmaier Lehtori Kimmo Saurén

Insinöörityössä rakennettiin ratsumittalaite kansainväliselle paperiteollisuuden asiantuntijayritykselle. Ratsumitta on käsin pideltävä laite, jota vedetään telan päällä, jolloin se antaa alustavan arvion telan tai valssin muodosta. Tavallisesti ratsumittalaitteet eivät tallenna mittaamiaan tuloksia. Laitteet, jotka tallentavat telan pituusmittaustulokset, ovat kalliita, ja mittaaminen vaatii aikaa ja telan siirtämistä.

Insinöörityössä kehitettyä laitetta vedetään telan päällä telan pituuden mukaisesti. Laite antaa kaaviokuvaajan telan halkaisijan muutoksista. Laite mittaa ainoastaan telan pituuden suuntaan risteäviä epämuodostumia.

Insinöörityönä tehdyssä laitteessa on kaksi vartta, joista voi säätää ratsumittalaitteen mittaväliä. Toisessa varressa on kiinni heittokello. Kun heittokellon anturia työnnetään, se antaa mikrometrin tarkkuudella tietoa siitä, kuinka paljon heittokellon anturi on työntynyt sisään. Toisessa varressa sijaitsee vastakappale, jotta heittokellon lukemat pysyvät tarkkoina.

Laite mittaa kuljetun matkan pituutta rotaatiokooderilla. Kooderiin kiinnitetään kiekko, jonka halkaisija pitää syöttää ratsumittalaitteen asetuksiin.

Laitteen kanssa kommunikoidaan kosketusnäytön kautta. Ohjelmisto toimii Raspberry Pi3 -alustalla Windows IOT Core -käyttöliittymällä. Windows valittiin käyttöliittymäksi, koska haluttiin mahdollisuus lisätä laitteeseen tilaajayrityksen aikaisemmin Windows-ohjelmilla luotuja algoritmeja. Myös ulkonäön yhteensopivuus yrityksen muiden ohjelmistojen kanssa vaikutti päätökseen.

Laitetta testattiin ensin yrityksen hankkimalla pienellä telalla. Sitten laitetta testattiin paperitehtaalla. Testeissä todettiin, että laite toimi hyvin ja saadut mittaustulokset vastasivat isoilla koneilla tehtyjä samankaltaisia mittauksia. Insinöörityön tilaaja otti laitteen käyttöön.

Avainsanat	paperiteollisuus, ratsumittalaite, tela, valssi, CD-mittaus

Contents

List of Abbreviations

UWP – Universal windows platform

MVVM – Model, View, View Model -design pattern

API – Application Interface

CD – Cross machine direction

MD – Machine direction

AD – Active directory

IOT – Internet of things

HGC – Hybrid Grind Control software

HMI – Human machine interface

1 Introduction

This thesis inspects the implementation of an embedded Windows environment and requirement of embedded devices in paper industry roll measurement. Thesis also focuses on how to create user friendly measuring environment with accurate data and clear measurement reporting.

The thesis implementation happens with a creation of a measuring device. The device created on this thesis measures the surface of a roll, that is used in paper and metal industry. The device measures variations in roll's diameter. The data from the roll is acquired by using a dial indicator and rotational encoder. Data is displayed in interval scale in an amplitude chart where intervals occur when data is received from the rotational encoder. The amplitude chart will display the difference in roll diameter on Y-axis.

Figure 1: 3D-Model of the device

Figure 1 is a computer model of the thesis device and it describes how the thesis device could be implemented. The structure is placed on top of the roll in such way that the rods are on both sides of the roll. The model was used to Visualize the device during the development of the thesis. Note that rotational encoder is not implemented on the model.

The software is coded with C# utilizing Microsoft UWP -library. The device created on this thesis has windows IoT Core 10.0.15063.* as operating system and uses Raspberry Pi 3 as a platform. The device uses two sensors to create the measurement. User interacts with the device via touchscreen display. Every measurement done with the device will be stored in database along with roll and user information. The report of the measurement can be saved to external mass storage device, sent via E-mail or be shared with Windows IoT Core's FTP server.

The device is scalable so that different sensors can be used to achieve the desired measurement speed. The interface is designed to be simple enough so that factory workers with industry experience will not have trouble using the device.

The project belongs to and was funded by RollResearch International Ltd.
This project was designed so that it stores to and retrieves from database using Factory API designed at RollResearch International Ltd.

2 Requirement for measurement

In paper industry, rolls are used to form the resulting product. The rolls are used for pressing, sizing, coating, drying and moving the paper in a paper machine. This thesis focuses on the rolls used in paper industry, but the device can also be used in the metal industry rolls. This chapter intends to explain what the rolls are used for, what type of rolls are there and why the rolls need to be measured.

2.1 Rolls wear out

To get the best possible paper quality the rolls must apply pressure evenly to the paper. The rolls wear out in use, and worn out roll can cause variations in the thickness of produced paper. Pressing the paper applies pressure to rolls. Because the shell of the roll has uneven thickness and density by default the roll wears out asymmetrically.

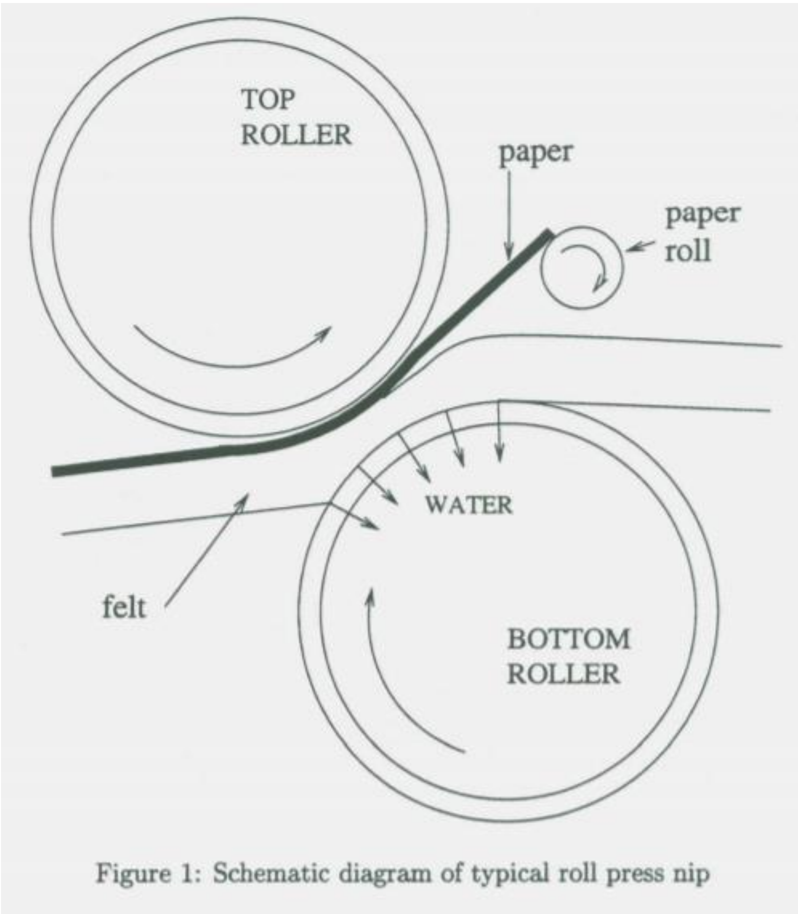


Figure 2: The Nip [1, s. 2.]

Figure 2 describes how the paper is pressed. Nip is the point where two rolls meet and apply pressure to the paper. Kuosmanen [2, s.19.] explains in his paper how the uneven nip pressure can cause regenerative problem. If a roll is off shape it can cause uneven pressure to the nip. This uneven pressure will cause more heat to the two rolls. This heat causes thermal expansion on the rolls. Expanded rolls will apply even more pressure in the nip causing the problem to regenerate. The pressure on the nip causes paper to be thinner. The differences in roll shape will expand and cause expanding differences in paper thickness.

The rolls in the paper industry have a big role in the paper quality and type. There are several rolls in a paper machine and each machine have different type of rolls for different sections of the papermaking process. Rolls are used for shaping and drying the paper. The rolls cannot be measured while they are attached to the paper machine.

2.2 Roll in a coordinate system

There are two things that are measured. First, the distance travelled on the Z- axis of the roll. roll change in thickness along the C-axis of the roll. Z- and C-Axis are used in CNC-machining coordinate system.

In roll grinding the roll is placed in the coordinate system so that it's length is along Z-axis. The Circumference of the roll is measured in C-axis, that places the diameter variations measured in this thesis to the C-axis as well. The Y-axis is not used in roll grinding, and the X-axis is used in grinding machine control, not in measurement [3]

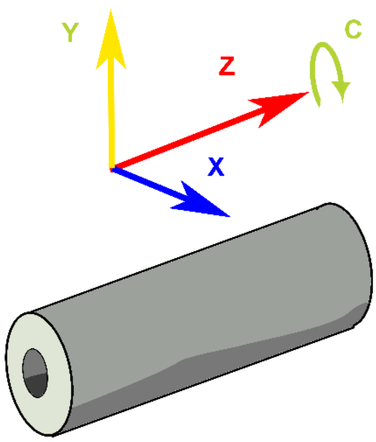


Figure 3: Roll placed along CNC coordinate system

Figure 3 describes how roll is placed along the 6DOF coordinate system. Note that the yaw has been changed to C and the pitch and roll aren't visible on the figure.

2.3 Roll structure

The rolls in the paper industry are designed in many ways. There are multiple parts in the paper making process where rolls are implemented. Forming rolls press the pulp into flat shape before the actual pressing rolls thin the paper out. Paper is then sized, Sizing is a material that is applied on top of the paper giving it different properties [3]. After sizing paper can be coated to add more durability [3]. Paper can go through different routes with different rolls along the way. Most of these routes and phases use different rolls.

There are different weighting mechanics implemented on roll depending on the roll type. Most of the rolls in paper industry have a surface that is separate from the rest of the roll mechanic. A paper roll usually has a coating on top of its metal surface. Not all rolls have coating [3].

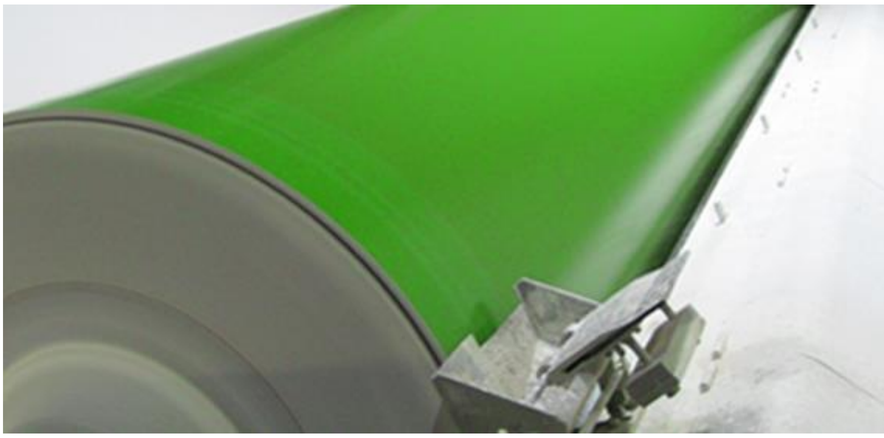


Figure 4: Coated roll [4]

Figure 4 shows coated roll, where the green part is the roll coating. Note that the Chamfer isn't visible in Figure 4. Commonly used coating materials are Polyurethane, Ceramic, Rubber and Composite [3]. The coating will have a chamfer at the both ends of the coating. This is to make sure that the paper stays in the right place. If the paper is moving over the chamfer, the pressure in the other end of the paper pulls the paper back.

2.4 Roll grinding

The rolls go through multiple rounds of measurement and grinding before the desired shape is gained. There are two ways rolls are shaped. Grinding is a process where the roll is grinded against a grinding rock.

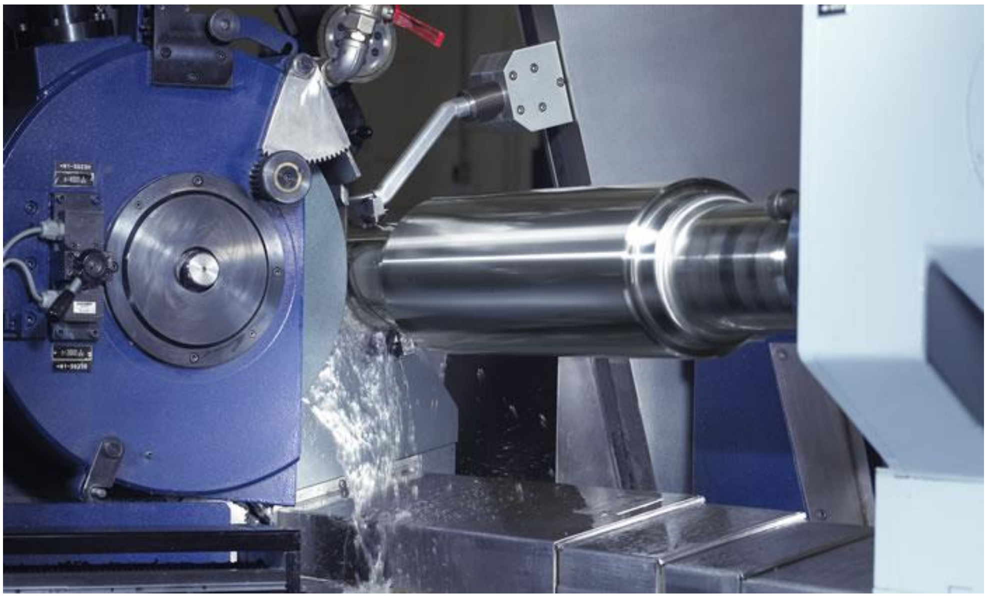


Figure 5: Herkules grinding machine [5]

Figure 5 shows how grinding rock is pressed against the roll. Note that the roll is cooled with water during the grind to prevent heat expansion. Grinding makes the roll straight. It is important to note that, when roll is rotated with high velocity its shape is changing due to the massive centrifugal force.

This bending property of a roll is matched by crowning the roll’s diameter deliberately uneven. This deliberate shaping of the roll should be taken into consideration when the roll measurement is inspected. The shape of the roll, when rotated is called the dynamic shape of the roll. A roll that isn’t moving will have static shape [6, s. 292-293].

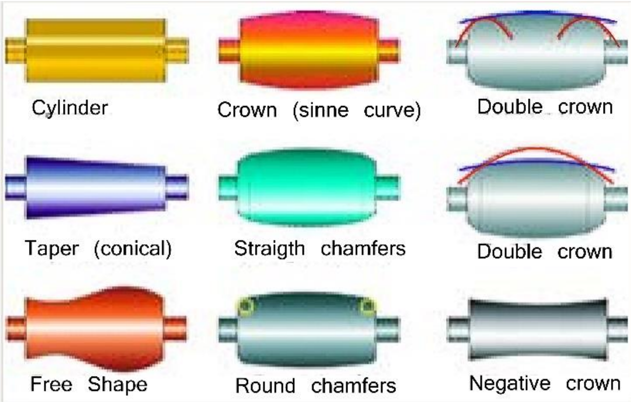


Figure 6: Different grindings for rolls [7, s. 1.]

Figure 6 shows how rolls can be formed to different shapes. Where the free shape is not grinded at all. Different crowning is used for different roll. The desired result is always to have the dynamic shape as close to a straight line as possible.

3 Roll Measurement

Kuosmanen [2, s. 14.] divides measurements into three generations. First generation performs measurements in the CD. The second generation of measurements measures variations in MD. Last generation that Kuosmanen mentions, considers dynamic shape of the roll, calculations to compensate the roll's shapeshifting. The three generations of measurements are meant to be layered, so that the second generation comes after the first generation and the calculations come after all the measurements. The process of shaping the roll takes usually multiple measurements. The grinding of the roll happens between these measurements. This thesis focuses on the measurements of the first generation. This chapter explains what types of measurements this thesis intends to do and what other measurements can be performed.

3.1 Axial measurement

Roll has a defined tolerance how much differentiation there can be on the roll surface. There are several ways a roll can be measured. This thesis only measures axial-differences in the roll. The axial difference gives a guiding estimation if the roll is misshaped perpendicular to the rolls Z-axis.

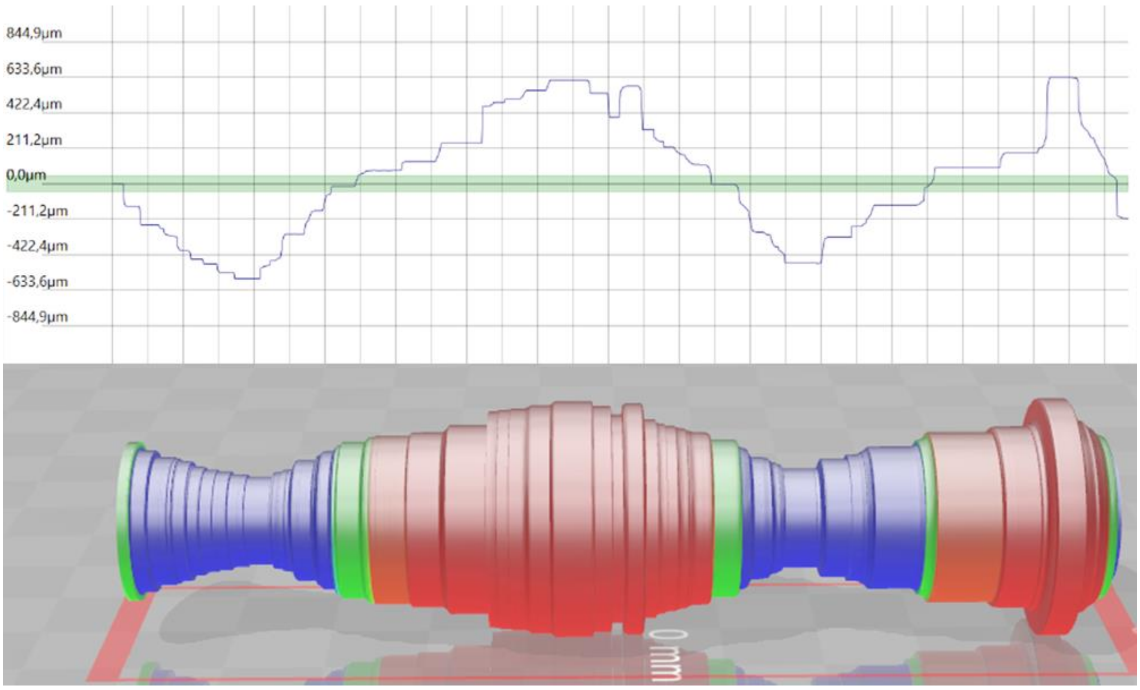


Figure 7: Measurement as a graph and below roll modelled after the measurement data.

Figure 7 shows an axial CD measurement performed by the thesis device. The measurement is visualized as a 3D model below the graph. The 3D model is exaggerated because the error scale to the size of the roll, is too small to be properly seen. The green area in the graph is tolerance which is set to 100µm. In 3D-model the areas that are below tolerance are coloured blue, and areas above are coloured red. There is a known diameter for the roll. The green areas of the 3D-model are the values of the diameter and the values which are within the tolerance.

3.2 Axial measurement problem

If there is difference in roundness of the roll, measuring its diameter in one Z-axis isn't enough. More extensive methods are required. Axial measurement does not give info if the roll is curved. It also cannot detect bumps in the roll if they're not in the way on the sensor.

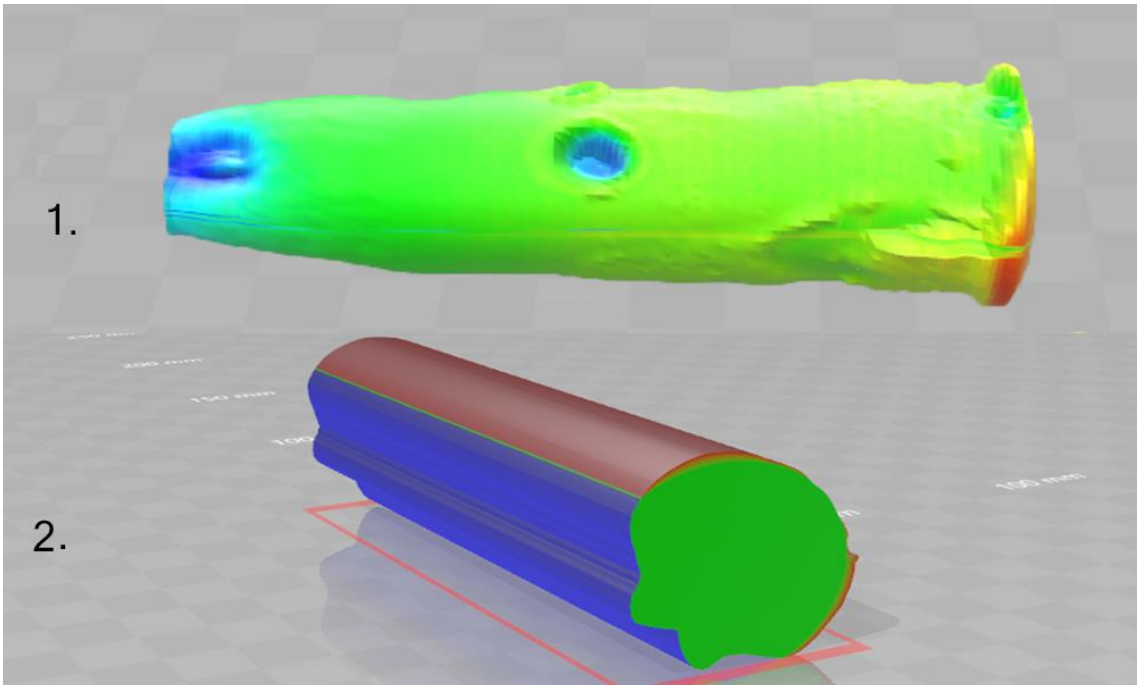


Figure 8: Unmeasurable rolls

The Figure 8 describes two rolls that the thesis device can't measure with accuracy. The sections where the roll is bulged are coloured red. Blue sections mean that the roll is imploded. Green areas are within the accepted surface variation tolerance. The first roll would give some data, but the large dents in the roll are detected only if they come across the dial indicator. The thesis device reveals that the roll might require some more extensive measurements, since it is misshapen at both ends. Measuring the roll number 2 would result it being perfectly straight because its diameter only changes perpendicular to C-axis.

4 Designing the device

The purpose of this thesis is to create inexpensive measurement device that can perform CD measurements. The device is handheld so marginal for human error must be considered and if possible minimalized. General structure of the device relied on already existing design of handheld CD measuring device. This chapter inspects the selected hardware and software components and tries to answer questions of why these parts were selected and what is their purpose for this device.

4.1 Choosing the hardware

The parts were assigned to this thesis by the company that ordered the thesis device. The thesis is handheld so the parts used are selected to make the device compact and the power consumption needs to be taken into consideration. The frequency for an accurate measurement requires at least 20Hz measurement frequency.

4.1.1 Raspberry Pi 3

Raspberry Pi 3 is a small computer meant for prototyping projects. It was selected for it's cheap price, availability and it supported a Windows operating system. Raspberry was also tiny enough to be placed in a mobile device.

Raspberry has quad-core ARM Cortex A53 cluster as its processing unit which has clock speed of 2GHz [8, s. 24.]. The ARM Core can run both 32-bit and 64- bit mode, but 32-bit mode is chosen by default. The speed change between the two modes are marginal at best due to the small 1GB RAM, so the 32-bit mode is used. [9, s. 8.]. Raspberry Pi 3 allowed speedy measurements to be displayed as an animated graph without lowering the measurement frequency. Raspberry also provided GPIO pins for resetting the rotational encoder. Multiple USB-hubs of Raspberry were used to communicate with the sensors and to store data to external mass storage.

4.1.2 Raspberry Pi 7” touchscreen

The use of the device happens through HMI. Touchscreen was selected to make the device mobile and compact. This touchscreen was chosen because it's compatible with Raspberry Pi. The screen resolution is 800x480. In the software created on this thesis it is presumed that the screen resolution is constant. Which results that changing the screen may change the layouts of the application. Which can have effect to the functionality of the application. Applying the touchscreen to the device does not require any software implementation in the code when using Raspberry Pi 3's DSI port

4.1.3 Sylvac S_Dial Work PLC

The device created in this thesis was tested with both S_Dial Work Basic and S_Dial Work PLC. The PLC model takes 100 measurements per second and can be operated through both RS232 and RS485 lines. The dial indicator communicates using ascii commands. The commands are listed on the appendix 2.

4.1.4 Wachendorff 36A magnetic rotational encoder

Just measuring the diameter variation isn't enough to create assessment of the roll's shape. The information where diameter variations occur is needed to create 2-dimesional graph.

Encoder is a sensor that measures rotation. There are two kinds of encoders. Incremental encoder will tell how much it has been turned, but not the relative position. Absolute encoders can tell the position encoder. Inside an encoder there is an optical sensor and a wheel. The wheel will contain a pattern that the optical sensor monitors. From the optical pattern, the sensor can observe the absolute position.

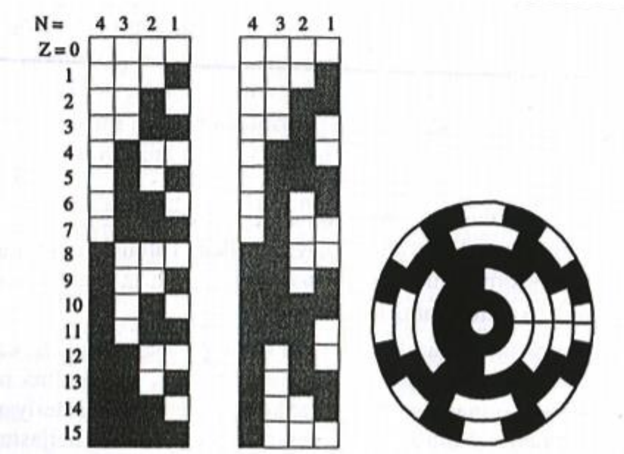


Figure 9: disc utilising Gray code [10, s. 98.]

Figure 9 describes how patterns can be converted into information using a code created by Frank Gray. On the right, there is a drawing of a simple 4-bit optical disc utilizing Gray code. The table on the centre is the Gray code disc folded open. The table on the left describes how binary code could be implemented similarly on the disc.

4.2 Software components

The development of the thesis happens on Windows platform. Visual Studio 2017 community edition is used for coding. UWP is a design tool developed by Microsoft. UWP is used when the code needed to be able to perform in any Microsoft platform, in this case Windows IoT Core.

The project is designed to be able to communicate with the API that is developed by the company that ordered this thesis. Data from the measurement is stored to the database in a way that is modelled in the API.

4.2.1 Windows IoT Core

This thesis does not require any hard-real-time operations so Raspberry Pi 3 with a Windows IoT Core is implemented. Windows IoT Core is a general-purpose operating system meant for IoT-projects. The programming happens through LAN. The Windows IoT Core uses an UWP library. That means that Windows IoT core is programmed with the same library that windows phone applications use. But the Windows IoT Core doesn't have the same hardware or drivers that a phone has. Because of this some classes and objects are available and compile, but do nothing when used. This caused a lot of difficulties with code debugging.

4.2.2 MVVM Design pattern

The thesis project follows the MVVM design pattern. MVVM is commonly used in WPF and UWP applications. Johnson [11, s. 306.] writes that the MVVM -pattern enabled better unit testing and helps differentiating the developer elements from the visual designer elements. The MVVM pattern is constructed from three components, Model, View and View Model.

The Model defines what type of data is used and how it is kept together. The model of the thesis is mostly defined by the database. The Model component does not define any functions. Thesis device needed to use same model on order to serialize the data that is desired to store or retrieve from the database. Serialization is a process where object is constructed to a different form. in this thesis communication with the database happened through an API that used JSON objects [12].

The View is the visual layer of the program. View component does not create or modify data. The View is unaware of the Model component. The View in UWP application is constructed of two files. The XAML defines how the visual elements are placed, and the code behind CS file defines operations to the said visuals. Note that in ideal case the code behind contains as little code as possible [12].

The View Model is the functional component of the program. The View Model does not know the existence of the View, but it can trigger events, if some data needs to be passed to the View without data binding. The View Model has access to the Model objects. The View is aware of the View Model component but it should not directly call any of the View Model’s functions. Instead communication between View and the View Model happens through data binding and events. Data binding is a process where PropertyChanged event is used to pass values to the XAML [12].

4.2.3 FactoryAPI

Factory API is a program that was developed at RollResearch. It is used to communicate with the database. FactoryAPI is implemented for parallel software development strategy. The thesis device will create data and Factory API stores it to the database.
The FactoryAPI takes request in a form of a post. The URL will define what information is requested. To receive the information from the database identification is required. This can be private key, bearer token or username and password. The username and password are used just to receive a bearer token. The bearer token can access the same data as private key but it has a predefined expiration time.

5 Hardware implementation

For measuring different kind of rolls the structure for the device needed to be adjustable. The device is designed so that you can have different dial indicators and absolute encoders. This chapter focuses on describing the how the hardware was attached to the device and how the wiring was done. Also what the hardware requirement need to be matched for the hardware to function.

5.1 The frame

The device frame has two rods and a centre piece. When placed on a roll the whole structure rests on the centre piece. The centre piece has wheels at all four corners so that it moves smoothly in the direction of the roll. The rotational encoder and dial indicator are attached to the rods. The dial-indicator is attached to movable mount on the other rod. On the opposite rod there is a small counterpart so that the dial indicator rod doesn't push the device and cause error to the measurement. The rotational encored is attached to a wheel that is pushed towards the roll. The wheel is rotated as the structure moves.

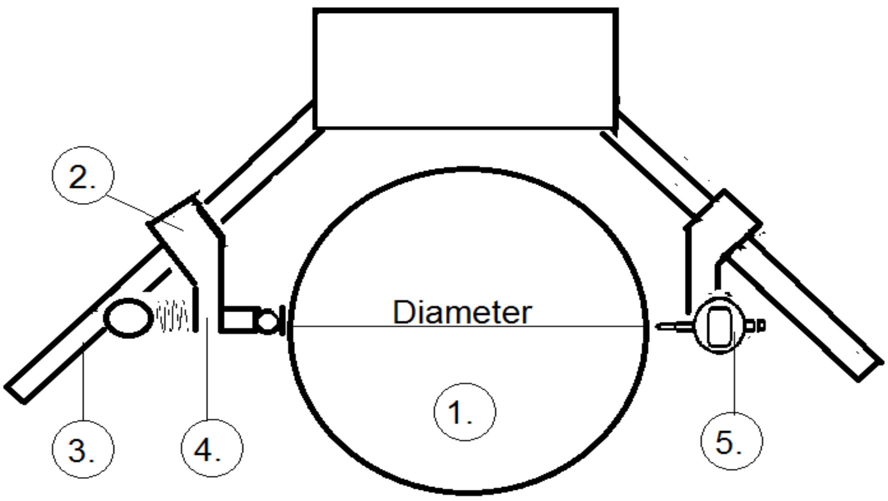


Figure 10: Cross-section of the thesis device. (1) Roll. (2) Movable mount. (3) A rod with a track for the mount. (4) Adjustable counterpart. (5) The dial-indicator.

Figure 10 describes how the device is placed on top of the roll. Note that the wheels that hold the structure on top of the roll are not visible.

The rotational encoder is attached to Raspberry Pi 3 with an USB to TTL adapter. The adapter grants 5-Volt power supply to the encoder. The encoder used in thesis could operate power supplied by USB port.

5.1.1 Dial indicator

Sylvac S_Dial -indicator detects the distance that the contact point has been moved with micro-meter precision. It is connected to a movable mount on the other rod of the measurement device. The dial indicator is properly attached when the mounts if both rods are in the same level with the rolls centre. The contact point of the dial indicator should be pushed inwards about half of its length when it is in resting position.

The minimum power requirement for PLC model is 9V and maximum is 24 volts. The final version of the thesis implemented the PLC model. When the 9-Volt dial indicator is used, external power supply is needed. PLC Model requires USB to TTL converter to operate at full speed.

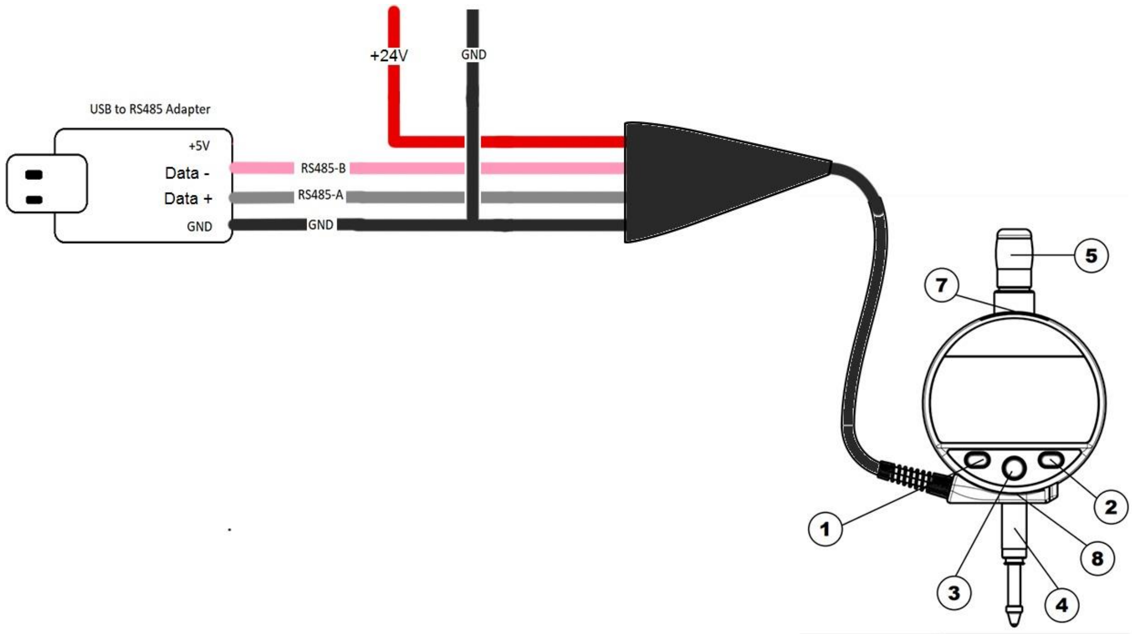


Figure 11: Dial indicator wiring to USB to RS485 -adapter

Figure 11 describes how the dial indicator is wired to the USB to RS485 -adapter. The Adapter is connected to the Raspberry Pi 3's USB port. Note that the power for the dial indicator must come from the separate power source.

5.2 Rotational encoder

The rotational encoder is attached to the same rod where the dial indicator is located. A special mount, designed by RollResearch International is used to attach the encoder to the structure. A wheel is attached to the shaft of the rotational encoder. The encoder wheel is the pressed against the roll by a spring so that when device is moved the wheel spins.

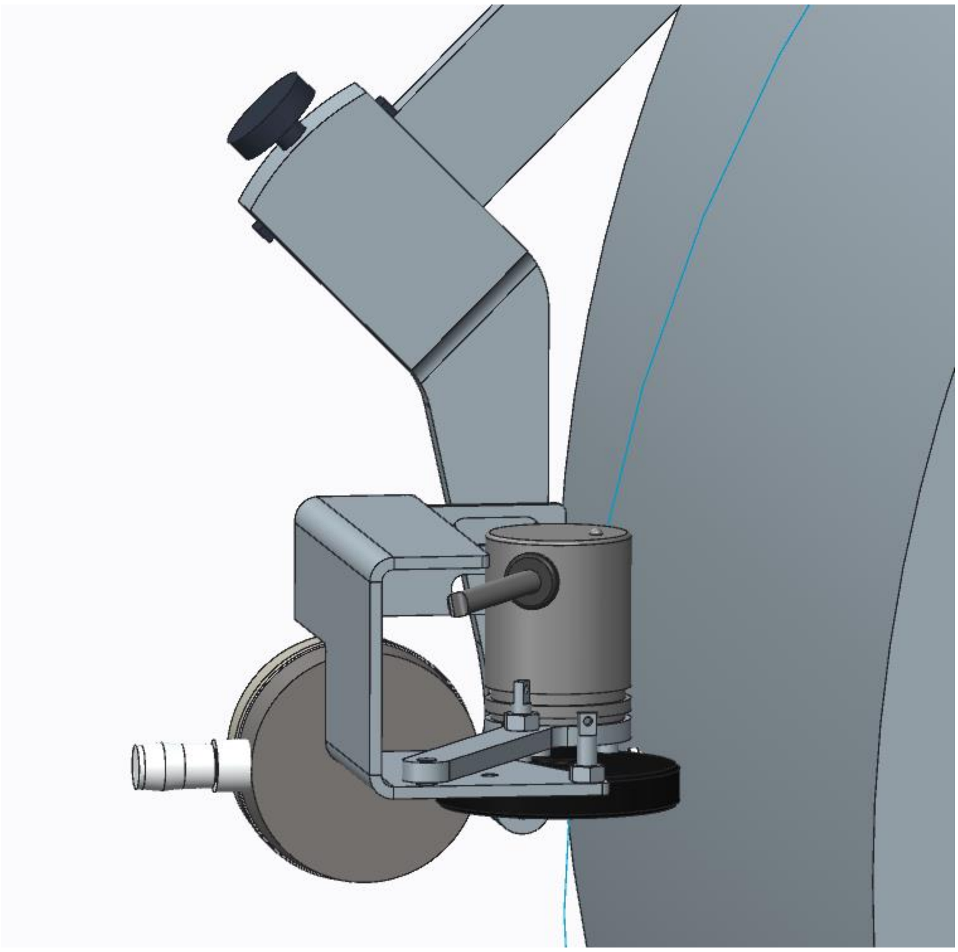


Figure 12: Attachment of the rotational encoder.

Figure 12 describes how rotational encoder is attached to the device frame. Note that the spring pulling the encoder against the roll is missing on the picture. The encoder needs to be attached to one of the rods because the centre piece of the structure is movable.

6 Impementing UI Functionalities

The device is designed to be able to be compatible with RollResearch HGC-software. The device can also operate as a standalone version. Both versions can produce and display the measurement data in a report. Measurement data is always saved in a ascii file. If Factory API is available and configured the measurement can also be saved to database. This chapter inspects what graphical aids were implemented to the thesis and how the implementation was executed. How the device is used is also inspected on this chapter.

6.1 Registering the device

The device can be used to store measurements to database if Factory API is running on the server. To use the device with the database, user with administrative rights to server must first register the device to the Factory API. The registration is valid until the device is unregistered from the server. Unregister Button can be found on the bottom right corner of the sign in page. When the unregistered device is restarted, or the user logs in, identification via username is still required. If user chooses to use the device as a standalone device. Measurements can be performed without the database. To use standalone mode “AD User” checkbox needs to be checked. The measurement can then be done without it being stored to database. A report from the measurement can still be obtained even if the database is not in use. This standalone feature is meant for the users who just want the device and not the API.

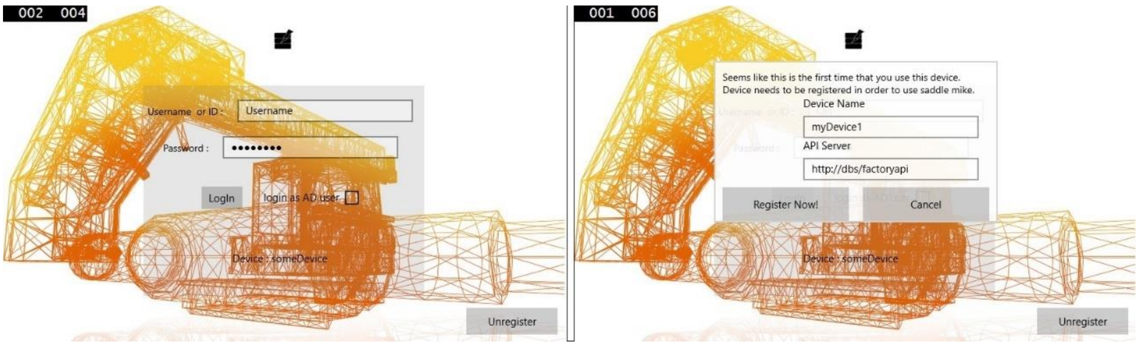


Figure 13: Login screen

Figure 13 shows the login screen and the popup that follows the unregistered login. The page for login contains two text inputs, login button, and checkbox. If the user wants to login as AD user the “Login as AD user” checkbox must be checked.

When pressing the “Login”-button software searches if the device has been registered by checking if the device memory contains private key. If no private key can be found registration popup appears. If private key can be found, software assumes the device is registered and proceeds to get the user information from the database. When the device is registered the password-field is not visible and the connection icon can be seen on the top centre of the screen.

If registration is required popup asking for device name and server address appears. The device name is any unique name the user chooses. No two devices with the same name can exist in the database. The server address field requires the address where Factory API is located. The device echoes to user all error messages that may occur during registration. The user information is stored to the NavigationContext.

Device can be unregistered by sending unregister command to the FactoryAPI. When the device is un-registered it will remove the private key, server address and machine number from its memory. The user preferences for reporting and the sensor configurations will remain the in the local settings.

6.2 MainPage

When logged in user first sees the main screen of the application. Main screen contains four buttons: Measure, Sensors, Options and Sign out.

Figure 14: Main page

Figure 14 describes the layout of the main panel. Note that the buttons are displayed as images rather than text. The name of the current measurer is displayed on the top-right corner. If standalone version is used the name will be “standalone version”.

The main page checks if every value needed to perform a measurement has been set. If devices aren’t configured the main page shows popup that tells user to go configure sensors. The navigation hierarchy of the UI is designed as a star- or hub and spoke topology. having a main page makes navigation easier because every page can navigate to the main page.

6.3 Configuring sensors

Before the device can measure anything, the sensors need to be configured. The device is designed in such way that the sensors may be changed. That is why it is left to the user to configure the sensors. The device saves the previous configuration so that configuration only needs to be done once per sensor.

The software retrieves the devices connected to USB-ports and lists them on a ListView -control. When an item on the ListView is tapped a popup appears. In the pop-up, the sensor can be configured. This method of configuration was chosen instead of static configurations so that in case that something breaks sensors could be changed to a different model without software changes. The configuration will be saved to local settings and taken to use when the measurement is performed. Tapping the disconnect button will clear all the configurations from the local settings.

Figure 15: Dial gauge configuration dialog

Figure 15 describes the dialog for sensor configuration. The software uses a default timeout value for all the sensors. Note that the configuration page does not tell if the sensors are configured properly.

6.4 Setting up preferences

Options page is where the user can set preferences for measurement reporting. The server address can also be changed here. When option is changed it is stored in the local settings immediately excluding the server address selection that requires a button to be pressed before saving the option.

6.5 Performing a measurement

When the sensors are configured properly user can navigate to the measurement page. The page needs a moment to reset the digital encoder. Pressing the “Play” button will start the measurement. And change the “Play” icon to “Pause”. “Pause” will pause the measurement and toggles the icon back to “Play”. The same measurement can be continued after the pause has been pressed. Pressing the “Stop” button will stop the measurement. After “Stop” has been pressed. “Report” button appears. User can either start a new measurement or go view the report.

The measurement page starts two threads alongside UI thread. One thread reads the encoder and the other reads the dial indicator. Even data is read, no data is saved until the user presses play button. Before every measurement there is 2 seconds wait dialog. This is because the rotational encoder will need to put 5V to reset line for 2 seconds before every measurement.

The drawing is handled by a while-loop in the UI thread. The sensor reading will be applied only if the rotational encoder value is changed. This is because no change in value should happen if the device is not moved. Once the sensor has received and parsed a value it will set a flag. Once flags for both sensors are set high the UI-loop will add the sensor values to double arrays. The Double array storing the dial indicator data is multidimensional array. Where every dimension is a separate sensor array. All the sensor array will always be the same length because they reflect a point in the surface of the roll. And there cannot be a point with just one value.

The Measurement will work as state machine. Once the play button is pressed the state is set to running. The first value will be taken as referential value and that value is subtracted from every measurement value that follows. Every measured value is therefore relative to the first value. When the measurement has been started the back navigation is disabled. The back navigation is enabled when the measurement is stopped.

The measurement can be paused. Pause state occurs when user taps the pause button that appears instead of play button when measurement has been started.

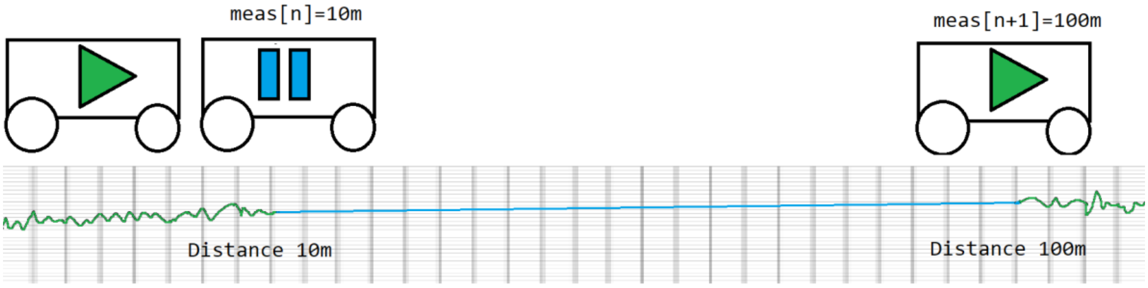


Figure 16: Pausing the measurement while moving the device

Figure 16 describes what happens to the measurement while it is moved while stopped. Note that the stopping happens only programmatically. If the wheel of the device is moved while measurement is paused the device will not store any measured movement but the sensor values will still change. This results that if the measurement is paused at

10-meters and moved 90-meters forward and the play is pressed again. There is just going to be a 90-meter difference between two points with no data in the middle.

When measurement is stopped a cancellationtoken is set to cancel and the state is changed to stopped. Both of the sensor threads have subscribed to the cancellationtoken so they will exit through the OpertionCanceledException catch clause and close their serial connection. It is important that serial connection is closed properly so that measurement can be redone. stopping the measurement will set report button visible and restarting the measurement will hide it again. The report button is used to navigate to the report page.

When cancellation is called both threads exit. The sensor threads have subscribed to a cancellation token. When cancellation token is in use, if the threads are currently working, the working function will throw task cancelled exception. The threads are then forced to exit by the catch clause. Lastly the serial lines are closed.

If there are some errors during the measurement the thread that gave the error exits. The error is then delivered to the user in a popup window.

6.6 Displaying the measurement

There are two types of graphs drawn in this thesis. Dynamic graph that is dynamically updated during the measurement. And a static graph that is drawn only after all the measurement values have been received. The graph is drawn on a canvas object using the path class. The path consists of points which are in pixel values. That means that to display the measurement as a graph, the millimetres need to be converted to pixels.

6.6.1 Dynamic graph

The animated graph that is drawn during the measurement is referred in this thesis as the dynamic graph. The dynamic means that its next value can be any point in the real coordinate system and the drawing needs to be able to scale itself accordingly.

Figure 17: Dynamic graph during a measurement

Figure 17 describes How the dynamic graph is displayed during the measurement. Where the latest measured value is displayed in the left side of the graph. Note that the measurement in figure 12 is paused and the. Also note that the received sensor value is displayed at the same y-position as the last point of the graph.

$$i = \begin{cases} 0\,, & j < k \\ j - k\,, & j > k \end{cases}$$

Formula 1: Selecting the last j measured values to be drawn.

Formula 1 describes how i is incremented once j is greater than k. where j is incremented every time new measurement value is drawn, i is the index of first value to be drawn and k is how many points is the dynamic graph going to have. The drawing of the received data during the measurement happens by taking the values between index i and j.

The measurement value received from the rotational encoder is not used when dynamically drawing the measurement. This is due to the centre of the graph moving when there is acceleration on the device. The data is easier to read when the movement stays constant. This will result to the data being distorted a little, but since the dynamic graph is drawn from just a few points this distortion is not noticeable. To get the value of x we need to divide the width of the drawing area w by k.

$$x = n * \frac{w}{k}$$

Formula 2: x-axis pixel calculation

Formula 2 describes how the pixel location is calculated from the encoder. Where x is the value where the horizontal value of the drawn point in pixels, n is the index of the point which goes from 0 to $j - i$, w is the width of the graph and k is how many points is the dynamic graph going to have.

The dial indicator value is scaled so that the absolute maximum that has been received from the dial indicator will be half of the height of the draw area. We need to scale it to half of draw area's height, because the dial indicator can also output negative values. When the dial indicator measures zero the value is placed in the middle of the graph.

$$y = \frac{h}{2} - (Y * \frac{\frac{h}{2}}{abs(\hat{Y})})$$

Formula 3: y-axis pixel calculation

Formula 3 describes how the pixel location is calculated from the encoder. Where y is the vertical value of the point in pixels, Y is the value received from the dial indicator, $\hat{Y}$ is the measured maximum and h is the height of the graph.

6.6.2 Static Graph

The graph drawn after all the values are received is referred in this thesis as static graph. The static graph knows all the maximum and minimum values so it can scale the all the values accordingly. There is an option to use filtering on the graph. The filter can be selected on the options page. There are None-, Median-, Moving Average-, and Gaussian filtering available. None meaning that there is no filtering at all. The filter strength can be changed from the options as well. The strength of the filter means how much effect using the filter will have on the shape of the graph.

In the static graph, a grid is drawn on the background. The Grid will be as tall as the draw area. The grid lines will always have same position. The values on the left side of the grid will define the scale of the grid. The millimetre value between two parallel horizontal lines is calculated.

$$mm\ (L_{n+1} - L_n) = \frac{\left(B * \frac{\hat{Y}}{\frac{h}{2}} + \hat{Y}\right)}{N * 0.5}$$

Formula 4: value between two horizontal lines

Formula 4 describes how to convert pixels to millimetres if the amplitude of the graph is known. Where the N is how many lines are drawn in total and B is the size of the buffer in pixels. The value of N is divided in half because the addition of horizontal lines is done in two separate loops. First loop draws lines above zero and the second loop below it. This way lines are mirrored and the location of the zero stays in the vertical centre of the graph. To get the values below the zero the result of the calculation needs to be negated.

6.7 Implementing data handling

When navigating to report page, popup informs if the measurement was saved to database successfully. Measurement will not be saved successfully on the standalone version. Measurement Report displays the acquired data as a XY-chart. Some information about the measurement can be found scrolling to the bottom of the page. User can choose to create a report of the measurement. Creating a report creates two files, one printable version of the measurement that is saved as HTML or pdf -form and an ASCII file that contains the measurement data.

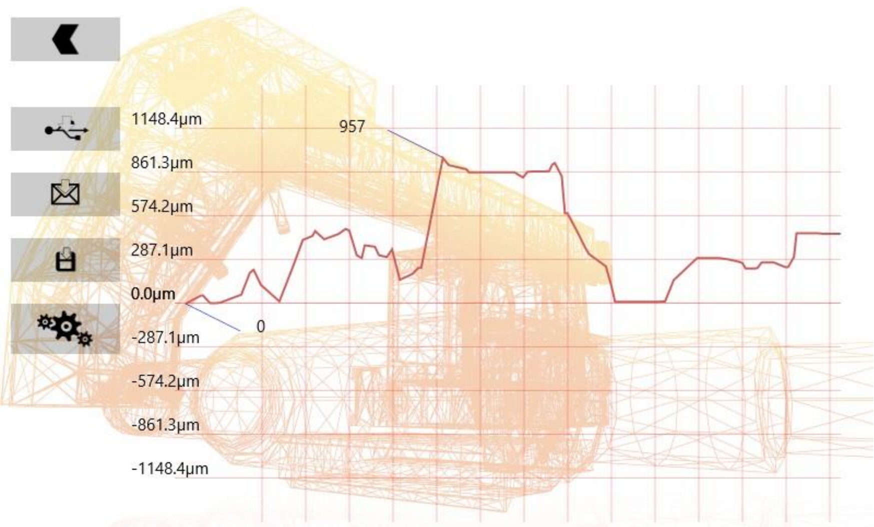


Figure 18: The preliminary report of the measurement

Figure 18 describes how performed measurement is displayed to the user. Note that the values in the graph are displayed as micrometres. Buttons are listed

Five buttons exist on report page. Appendix 1 describes what icons are used on buttons. Pressing the back -button will take you back to main page. When the USB -button is tapped, user can store the measurement report in mass storage device. User will be informed via popup, if no mass storage device is found plugged in to any of the USB-ports. Tapping the email -button will send the measurement report via e-mail to a selected address. Tapping the save -button will save the measurement locally to the documents folder of the Raspberry Pi 3. The documents folder can be accessed over Windows IOT Cores file sharing system.

7 Data storing and protection

The main objective of the device is to produce data. The data is interpreted by the user. Data about the user, database and sensors are also requires for user-friendly experience.

7.1 Database Model

The Database use is optional in the thesis device. The use of database allows easy to manage for measurements. The database is designed to be accessible by HGC. The database stores models defined in the MVVM -design pattern. The Model classes are described below.

Class	**Task**
Measurement	Can be thought as a parent of the model. Retrieving the measurement from the database will retrieve all the other models as well.
MeasurementData	Contains the measured values of the measurement.
User	Tells us who is the measurer of the measurement. The user who measures must be registered to database.
Roll	Contains the data about the roll that measurement was performed to. Roll must already be defined in the database. You cannot save measurement made to a roll that doesn't exist in the company database.

Table 1: Classes utilized in the model layer of MVVM.

Table 1 describes what the model classes are used for. Note That the individual members of the classes are described in figure 17. HGC defines greater number of model classes, but these are the only ones used in the thesis.

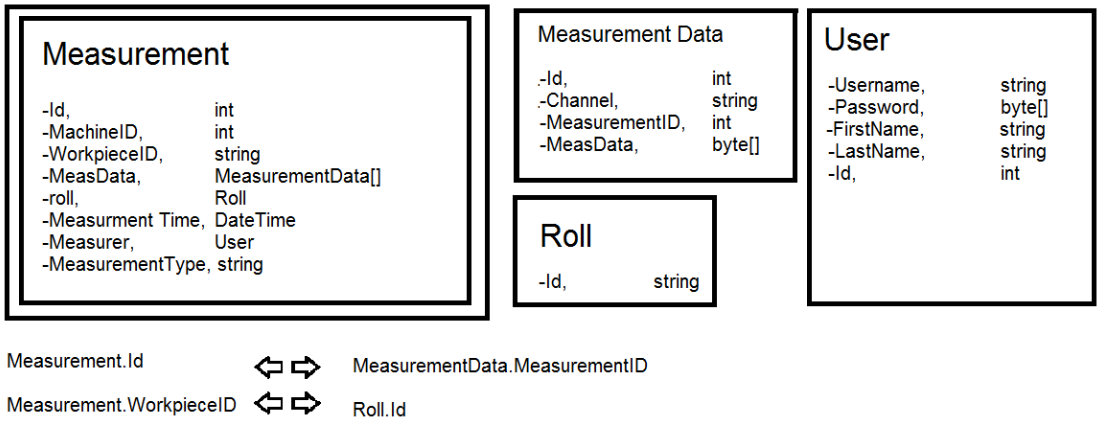


Figure 19: Model class members and relations.

Figure 19 describes What type of data the model classes store. The two-way arrows in the bottom of the image mean that those variables contain the same value. By retrieving measurement from database, it is known which roll was measured and by which user. This means for measurement to be saved in the database, the measurer will always have to provide username and roll id before measurement can be started.

7.2 Storing Data

For the thesis device to adapt to the user preferences and function user-friendly, the device must store data about the user actions, measured data and database keys. Since the device may contain data that must not be shared to everyone, or data that must always be seen, it is important to evaluate how to handle possibly sensitive data.

7.2.1 Evaluating the need for data protection.

If the database exists the measurement will be stored to the database automatically. In the standalone version, the measurement report must to be exported if a save of the measurement is needed. There is multiple way that the software developed in thesis stores data. This is due that some data has different priorities than other. In this thesis data is prioritized by privacy, availability and lifespan. The longer the data exists the more chance the is for it to be seen, altered or deleted.

Does not need protection	needs some protection	Needs a lot of protection		
	Lasts only one operation	**Lasts only one measurement session**	**Can be changed between sessions**	**Must stay the same until deleted**
Needs to be shown	•Popups	•The measurement report on the device screen	-	-
Should be available to everyone	•The current state of the device	-	•The sensor configurations •Report preferences	•The saved measurement report
Only for few users	-	•The Email address where the measurement was sent to	•User email	•The measurement data in database
Not to be shown to any people	-	•The model objects	•User password	•The private Key

Figure 20: Prioritization evaluation for the data.

Figure 20 describes how the data protection is prioritized. The green blocks are the data that the machine should not try to protect. Deleting or altering this data will not affect the functionality of the machine and in most cases, it is preferred for people to see this information. Yellow blocks are data that can affect the operation of the device. They are still meant to be changed but are protected by passwords or warnings. The Red blocks need some protective algorithm. Private key is the only data that is encrypted. Private key needs to be protected because it allows access to the company database. The private key is protected with a CryptographicBuffer and stored locally to the machine.

7.2.2 ASC

Software should be able to retrieve the data is from the database and other way uses *.ASC files. The HGC requires ASC files to be written in a very specific way. The header must define a roll that is located in the database, measurement type, Section, assemply options, Data Type and amount of data points. The HGC uses a constant amount of datapoints. The amount of datapoints must always be 1024.

```
Comment about data
Roll=RollName
Measurement type=CD
Section=0
Assembly options=10
Points=1024
Data type=ASCII
Data=
-0          0.00
-68.3       0.00
-132.7      0.02
.........
```

Listing 1: Header of the ASC file

Listing 1 describes the header that is used in measurement ASC -file. Because thesis only focuses on axial measurement the measurement type will always have to be CD. The amount of measured points must be scaled to have 1024 measured values. The measurement points must be stored four times so that in the data section of the file there will be 4096 measurement points. This is because ASC -files need to be uniform with the HGC that is designed to evaluate 4 -point measurements. Data points are saved so that first the negated Z-axis values are stored. Z-axis values are negated due to the HGC showing the CD -curve reversed. Then the data about the diameter variation is added, the data is separated by tabulator. Note that the data is in millimetres and that the listing only shows three first data points of the total 4096.

7.1 Communicating with the Factory API

The database is never contacted directly. Communicating with the database happens through Factory API. To communicate with Factory API the thesis device will first obtain a private key. Factory API creates the private key when the device requests it. The private key is used to identify a registered device. The private key is assembled by device from the network adapter id, name of the device and a hashed code created by The Factory API. The name of the device is given by the administrative rights user when the device is registered. When private key is obtained it is encoded and stored in system memory.

Private key was used because it has less chance of failing and it is reusable. It also makes the code handling and debugging easier. To register the private key to the Factory API a bearer token must first be obtained. Token is gained by sending a request with the username and password.

Bearer token can be obtained from factory API at any time. When requesting a bearer token from the Factory API password and username are put to the request stream and no authorization is needed.

When object is received from the Factory API, it is stored to a string variable. The received object is in JSON format. The object is parsed by

```
private T jsonToObject<T>(T myObject, string json)
```

```
        {
            using (MemoryStream ms = new MemoryStream(Encod-
ing.UTF8.GetBytes(json)))
            {
                DataContractJsonSerializer ser = new DataCon-
tractJsonSerializer(myObject.GetType());
                myObject = (T)ser.ReadObject(ms);
                return myObject;
            }
        }
```

Listing 2: Parsing a C# object from JSON file

Listing 2 describes how to parse JSON file to a C# object. Note That the function is a generic type function. The DataContractJsonSerializer is used to serialize object from stream.

8 Measurement reporting

Important part of the thesis was to create printable measurements. Different companies use different templates for their measurement reports. This thesis implemented 3 different templates.

8.1 Report Types

The reporting was done first with HTML file format due to the lack of apparent PDF libraries. Later the PDF reporting was enabled by a custom class.

8.1.1 HTML

HTML was chosen because it offered good graphical tools that could be read by most browsers and it was directly printable.

The report is built so that there is CSS header at the beginning of the file. The information about the measurement is divided in boxes with the <div> element. The canvas is added as an HTML object, and javaScript -function is used to draw the graph. The template headers come from resource files so the language of the report can be changed. In this English and Finnish reporting was implemented.

8.1.2 PDF

There are no libraries for creating a PDF in Windows IOT Core and the “PdfSharp” and “Windows.Data.Pdf” -libraries does not work with application in the time when this thesis is made. Instead the PDF -report is done by creating a separate class that writes the required data in the PDF 1.2 -standard. Older standard was selected for its simplicity and to make sure that PDF-report is compatible with older PDF -readers.

Thesis implemented 3 different templates for PDF reporting. The templates are described in the appendix 3. PDF 1.2 states that PDF is assembled of four items header, body, cross reference -table and trailer [13]. Focus for the class is to modify the elements in the body -part of the PDF. The element sizes, locations and rotations were altered to get different templates.

9 Filtering the Graph

If the measurement has jitter or unnatural spikes. The measured graph can be filtered. This thesis project has three available filters. Two of the filters were requested by the company that ordered the thesis. The gaussian filter was added because it would reduce well the jitter that may occur during the measurement.

9.1.1 Moving average filter

This filter calculates the average from a selected number of elements. the moving average Filter was implemented to remove noise from the measurement graph.

$$F_n = \begin{cases} \frac{W}{2} > n \quad or \quad D_{len} - \frac{W}{2} < n & \frac{(\sum_{k=0}^{n} D_k)}{n} \\ \frac{W}{2} < n & \frac{\sum_{k=-\frac{W}{2}}^{W} D_{n+k}}{W} \end{cases}$$

Formula 5: The Mathematical representation of the moving average filter

Formula 5 describes how to filter array F from data array D. Where D is the array that is to be filtered, W is the kernel of the filter and n is the index of the selected value. Note that the first and the last values of the array are averaged from a set, that is smaller than kernel. Note also that half of the members selected to the kernel are taken from behind the filtered point and half are taken from the following values.

9.1.2 Median filter

Median filtering is used to hide out any sudden spikes in the measurement. Median value is calculated by sorting the kernel number of measurements from the original array from lowest to highest. The middle value of the sorted array is the median. The filtered array is assembled from the median values. Note that half of the members selected to the kernel are taken from behind the filtered point and half are taken from the following values.

10 Implementing sensors

The thesis implements two sensors. One is a dial indicator that measures the dimeter variation on the roll surface. The other is a rotational encoder that detects the position of the diameter variation relative to the roll.

The measurement will always need data from both sensors before adding measurement point to the data array. Raspberry's ARM Cortex A53 has clock speed of 2GHz [8, s. 24]. Raspberry Pi 3 has four cores, which allows parallel programming. Both sensors can then be read in their own threads and the GUI updates on its own thread. Given that the operations performed by raspberry are much faster than the sensors that this thesis implements, it can be stated that the measurement speed of the slowest sensor will be the limit of measurement speed.

10.1.1 Software implementation of dial indicator

The Dial indicator used in this project communicated using ASCII protocol. The reading is happening in its own thread and is executed parallel to the encoder reading. Both threads raise a flag once reading is performed. Once both of the flags are up one measurement point is received.

ASCII protocol was chosen to communicate with serial device. The dial indicator used in the thesis can measure 100 measurements per second. The device is connected to Raspberry Pi 3 with a USB to TTL -adapter. The Sensor can communicate with both RS485 and RS232 -serial protocols. The ASCII commands for the dial indicator can be found in the appendix 2.

Configuration	**Value**
Write timeout	30ms
Read timeout	30ms
Baudrate	128000Bds
Parity	Even
Stop bits	One
Data bits	8
Handshake	None

Table 2: Sylvac S_Dial Work PLC configurations

Table 2 Lists the parameters for the Sylvac S_dial PLC. Where configuration tells what items are being configures and value tells the value used in the thesis. note that some of the values are enum -type.

Communicating with dial-indicator happens by first sending it a query string followed by newline character. The dial-indicator answers by sending total of 9 bytes to the host. The answer needs to be read to host's buffer with data reader class. Data is first read into the buffer. Buffer is then read one byte at the time and it is parsed at the same time. The number starts with '+' or '–' character whose ASCII numbers are 43 and 45. After either of those characters have been received only the dot with ASCII code 46 and numbers whose ASCII code is between 48 and 57 are accepted as message bytes. Message length is static so when enough acceptable characters have been received we can flush the buffer. Once full message has been received the measured value is stored to array. The array is the size of how many S-Sensors have been connected to the device of the thesis. If reading fails we read again. If cancellation token has been received the reading is cancelled.

10.1.2 Software implementation of rotational encoder

Thesis implemented absolute rotational encoder WDGA 36A. Absolute encoder gives a position value, where incremental encoder gives number of pulses for every turn. With absolute encoder it is easier to detect direction of the movement. The encoder communicates using RS485 protocol and it is connected to Raspberry Pi 3 via USB to TTL - adapter.

Configuration	**Value**
Write timeout	30ms
Read timeout	30ms
Baudrate	9600
Parity	None
Stop bits	One
Data bits	8
Handshake	None

Table 3: Rotational encoder setup

Table 3 describes what settings is used for the encored serial communication. Note that the smaller timeout value means greater possibility of unfinished messages. Larger timeout value means slower communication. The timeout was tested to find the most optimal value.

The encoder output data is an unsigned 32-bit number. if the rotation is incremented over 4294967295, which is the maximum value of 32-bit number, the encoder value resets. This reason thesis device resets the encoder before every measurement. The encoder communicates through RS485. The sensor is connected to raspberry's USB- port using USB to TTL converter. Write here how the encoder is attached to the device

A wheel connected to the shaft of the encoder and the wheel diameter is used to calculate the distance travelled with the following formula.

$$\frac{rotation}{16384} \cdot \pi * wheel\ diameter - \frac{previous\ rotation}{16384} \cdot \pi * wheel\ diameter$$

Formula 6: Encoder position to millimetres

Formula 6 describes how to calculate distance from the rotational encoder data. Where rotation is unsigned 32-bit number received from the encoder. And the wheel diameter is the diameter of the wheel that is connected to the shaft of the rotational encoder in millimetres.

The encoder is read on it's own thread and is executed parallel to the dial indicator reading. The threads are synchronized with a flag. Once both flags are up one measurement has been read.

The data from the encoder is placed to the hosts buffer by DataReader class. Encoder sends 6 bytes of data least significant bit first. 2 first bytes contain the preamble part. The 4 remaining bytes contain current position in a form of single-turn, and multiturn value.

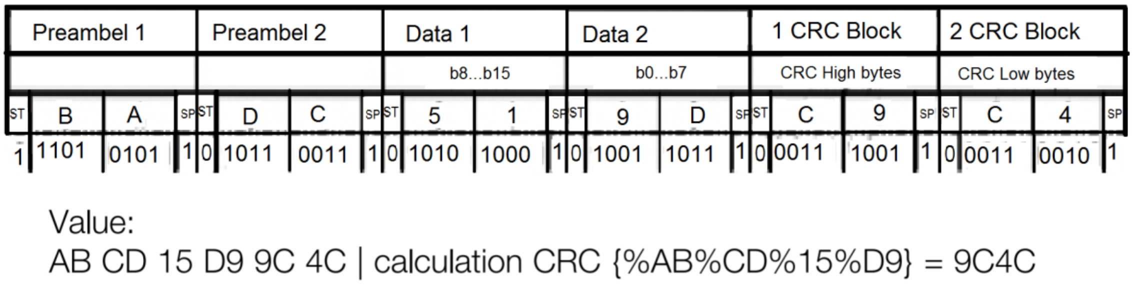


Figure 21: Encoder sample message

Figure 21 describes how bits are organized inside the encoder message.The preamble part stays constant. After preamble has been received the following four values contains the data. The four bytes contain the information how much the encoder has been turned in a form of a single-, and multi-turn part. Sigle-turn is the value how much the sensor has rotated on the current rotation cycle. Multi-turn is all the previous rotations added together. These values need to be added together to obtain the current position of the sensor. Note that the sample message come least significant bit first.

```
byte ReceivedBytes= new byte[6]{AB,CD,15,D9,9C,4C};

uint sigleturn = (uint)(0x3FFF & (ReceivedBytes[5] + (ReceivedBytes [4] << 0x08)));

uint multiturn = ((uint)(0x3FFFF & ((ReceivedBytes [4] >> 0x06) + (ReceivedBytes [3] << 0x02) + (ReceivedBytes [2] << 0x0A))) * 0x4000U);

uint encoderValue = singleturn + multiturn;
```

Listing 3: Constructing the data.

Listing 3 describes how the message is assembled in the code. The received bytes are the bytes used in the sample presented at figure 13. The bytes are shifted to two unsigned integer values. Note that the multiturn value uses two most significant bits of the single turn as it’s least significant bits.

WDGA 36A Reads 100 measurement per second to a buffer. Buffer is then read by the software. If buffer is read faster than it is filled, uncomplete messages can be received or the reading might start from the wrong byte. 12 bytes are read at one time to make sure that at least one comprehensible value is received at one read. Once the values have been read the rest of the buffer is detached.

11 Results

The average of 35 measurements per second was gained. This rate was approved by the project manager. The device feels natural to use and the UI is easy to understand. The PDF reports were formatted to the requested standard and some extra templates helped the evaluation of the roll shape and mathematical data. The database communication was performed successfully and the standalone version met the requirements. The CD form could be exported in multiple ways and could be read by the HGC.

12 Conclusion

This thesis will give a good example how to receive, store and handle data in UWP environment. The measurements done by the device are fast and give good evaluation of the CD form of the roll. The device doesn’t measure surface variation in MD. The device is also unable to detect if the roll curves with even diameter. If roll’s curve and MD variation are to be measured a separate machine is required.

Machine can save and report data with a professional looking layout. Preliminary report can be viewed from the touchscreen display. More extensive reports can be exported as HTML or PDF -file to email, USB-stick or saved locally to the machine’s FTP server. The option of sending the report directly to printer was left out because Windows IOT Core didn’t have printer drivers were declared at the time of creation of this thesis [14].

Device can save measurements to the database using the Factory API. Models defined by the Factory API. The stored measurements can be viewed with HGC. The older version of the HGC can view the measurements by importing the *.ASC-file.

References

1 Skelton, John. 1999. The Dynamics of a Roll Press Nip. 15th Annual Workshop on Mathematical Problems in Industry, University of Delaware.

2 Kuosmanen, Petri. 2004. PREDICTIVE 3D ROLL GRINDING METHOD FOR REDUCING PAPER QUALITY VARIATIONS IN COATING MACHINES. Teknillisen korkeakoulun Koneensuunnittelun julkaisuja 2.2004. s. 19.

3 Widmaier, Thomas, 2018. Doctor, Konetekniikan laitos, Insinööritieteiden korkeakoulu, Aalto-yliopisto, Espoo. Henkilöhaastattelu 29.3.2018.

4 Uncompromised roll cover performance for sizing and coating. Verkkoaineisto. <https://www.valmet.com/globalassets/services/board-and-paper-mills/roll-services/roll-covers/coating.jpg>. Luettu 15.2.2018

5 Roll Grinding to the Highest Standards. Verkkoaineisto. <https://www.herkules-machinetools.com/herkules/img/application/roll-grinding.jpg>. Luettu 22.3.2018

6 Juhanko, Kuosmanen, Porkka, Widmaier. 2010. Dynamic geometry of a rotating cylinder. Estonian Journal of Engineering 16.4.2010, s. 292-293.

7 Roll Grinding Solutions for the Paper Industry. Verkkoaineisto. <http://www.rollresearch.fi/portaali/images/Download_en/Solutions-for-Paper-industry_English_rev.18.8.2015.pdf>. Luettu 22.3.2018

8 ARM Cortex -A Series. 2015. Verkkoaineisto. ARM. <http://infocenter.arm.com/help/topic/com.arm.doc.den0024a/DEN0024A_v8_architecture_PG.pdf>. Luettu 22.3.2018

9 Kunst, Quade. 2017. Operating the Raspberry Pi 3 in 64-bit mode. Raspberry Pi Geek. Issue 23. Lawerence, USA:Linux New Media USA.

10 Aumala, Olli. 2003. Mittaustekniikan perusteet. Helsinki:Otatioeto.

11 Johnson, Bruce. 2013. Professional Visual Studio 2012. Indianapolis, USA:John Wiely & Sons Inc.

12 The MVVM Pattern. 2012. Verkkoaineisto. Microsoft. <https://msdn.microsoft.com/en-us/library/hh848246(d=printer).aspx>. Luettu 22.3.2018.

13 Bienz, Cohn, Meehan. 1996. Portable Document Format Reference Manual Version 1.2. Adobe Systems Incorporated. s. 65.

14 Hardware compatibility list. 2017. Verkkoaineisto. Microsoft. <https://docs.microsoft.com/en-us/windows/iot-core/learn-about-hardware/hardwarecompatlist>. Luettu 12.11.2017.

Wachendorff WDGA 36A technical specifications

Mechanical Data	
Housing	
Flange	synchro flange
Flange material	aluminum
Housing cap	steel case chrome-plated, magnetic shielding
Housing	Ø 36 mm
Shaft(s)	
Shaft material	stainless steel
Starting torque	approx. 0.3 Ncm at ambient temperature
Shaft	Ø 6 mm
Shaft length	L: 11.5 mm
Max. Permissible shaft loading radial	80 N
Max. Permissible shaft loading axial	50 N
Shaft	Ø 8 mm
Shaft length	L: 18 mm
Max. Permissible shaft loading radial	50 N
Max. Permissible shaft loading axial	50 N
Bearings	
Bearings type	2 precision ball bearings
Nominale service life	1.4 x 10'8 revs. at 100 % rated shaft load 2 x 10'9 revs. at 40 % rated shaft load 1.7 x 10'10 revs. at 20 % rated shaft load
Max. operating speed	12000 rpm

Machinery Directive: basic data safety integrity level	
$MTTF_d$	1000 a
Mission time (TM)	20 a
Nominale service life (L10h)	1.7 x 10'10 revs. at 20 % rated shaft load and 12000 rpm
Diagnostic coverage (DC)	0 %

Electrical Data	
Power supply/Current consumption	10 VDC up to 32 VDC: typ. 50 mA
Power consumption	max. 0.5 W
Power supply/Current consumption	4,75 VDC up to 5,5 VDC: typ. 80 mA
Power consumption	max. 0.44 W

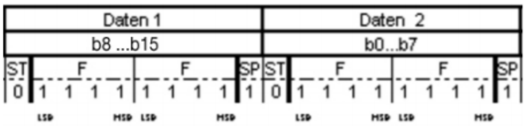
Sensor data	
Single-turn technology	innovative hall sensor technology
Single-turn resolution	up to 16,384 steps/360° (14 bit)
Single-turn accuracy	< ±0.35°
Single-turn repeat accuracy	< ±0.20°
Internal cycle time	≤ 600 µs
Multi-turn technology	patented EnDra® technology no battery no gear.
Multi-turn resolution	up to 24 bit.

Environmental data	
ESD (DIN EN 61000-4-2):	8 kV
Burst (DIN EN 61000-4-4):	2 kV
includes EMC:	DIN EN 61000-6-2 DIN EN 61000-6-3
Vibration: (DIN EN 60068-2-6)	50 m/s² (10 Hz up to 2000 Hz)
Shock: (DIN EN 60068-2-27)	1000 m/s² (6 ms)
Design:	according DIN VDE 0160
Turn on time:	<1,5 s

Interface	
Interface:	**RS485**
Positive direction of counting: (View on shaft)	DIR = GND -> cw DIR = +Ub -> ccw
Set to zero:	Preset = apply +Ub for 2 s
Baud rate:	Standard: 9600 bit/s Other baud rates on request
Polling cycle:	Standard: 20 ms (Tolerances: +/- 2 ms) Other polling cycles on request
Telegram lenght:	6 byte singleturn, 8 byte multiturn
Telegram composition:	2 Byte Präambel, 2 /4 Byte user data, 2 Byte CRC
Bytecomposition:	Startbit (0) and Stopbit (1), Bytes are Big-Endian and LSB first, no Paritybit

CRC-Definition:	Code: • CRC-CCITT 16 bit (X^16+X^12+ X^5+1) • Startvalue 0x1021, • Start/Stopbits aren´t included • Präambel (0xABCD) is included, • Bytewise orientation: per CRC-Refresh there is used 1 Byte
Protocol malfunction behaviour:	If encoder recognizes that it´s impossible to send a right positionvalue (e.G.: Magnet-loss), there will be send out a telegram with maximum value user Data at normalcycletime and normal Baudrate.

Protocol RS485



LED-behaviour:	
At Start / while booting:	- red gleam (< 2.3 s)
Malfunction:	- constant red gleam (> 2.3 s)
Normal function:	- constant green gleam
No supply:	- no gleam

General Data	
Weight	approx. 112 g
Connections	cable or connector outlet
Protection rating (EN 60529)	IP67, shaft sealed to IP65 (IP40 for K1)
Operating temperature	-40 °C up to +85 °C
Storage temperature	-40 °C up to +100 °C

More Information
General technical data and safety instructions http://www.wachendorff-automation.com/gtd
Options http://www.wachendorff-automation.com/acc

Sylvac S_dial Worl PLC technical specifications

E

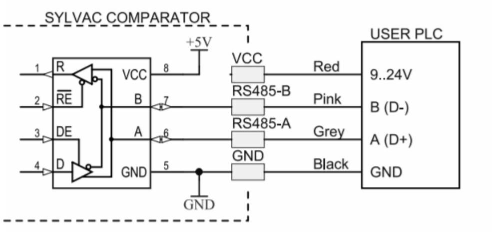
10.2 RS-485 databus

The instrument can be connected to a field bus, via the link cable. See page 2 for connection of the cables.
The data exchange between a master and the slave PLC comparator(s) is defined by a protocol (MODBUS by default). See the table of transactions and variable addresses in the appendix.

Important note : The RS-485 lines are not galvanically isolated from the Power Supply. The reference Ground of the RS-485 Master must be tied to the Ground of the Power Supply

10.3 Field bus address configuration

The comparators do not have pre-defined addresses. They therefore do not reply to any bus request. Before being able to dialogue with a comparator it must be configured with a unique bus address. Configuration can be done either by SLA remote command, or by the SLA menu. Assigning a zero address causes deconfiguration of the comparator.



16

E

10.6. Automatic controller (PLC) link cable

Extra flexible cable. Wire cross-section: 0.05 mm2 (26 x 0.05 mm Ø, AWG 30).

Wire colour	Signal name	Signal description
Black	GND (0 volt)	Supply (0 volt)
Red	VCC	Supply (9 … 24 VDC) Note: supply from a USB port (+5 V) is only possible temporarily for manual or Proximity cable configuration
Blue	Output 3	Outside upper tolerance (Re-work)
Green	Output 2	Within tolerance limits (OK)
Yellow	Output 1	Outside lower tolerance (Scrap)
White	Input 2	Favourite button function execution
Brown	Input 1	Preset / Zero recall
Pink	RS-485-B (Data-)	Bus Link - B (half duplex)
Grey	RS-485-A (Data+)	Bus Link - A (half duplex)
Braid	PE	Shielding (earth protection)

If more than one digital indicator is connected to the PLC via the bus lines, all GND, all VCC, all RS-485-B (D-), all RS-485-A (D+) and all PE wires must be tied together.

18

E

11. List of the main commands

Selection and configuration

CHA+ / CHA-	Change measurement direction
FCT0 ...9...A...F	Assign «favourite» function
MM / IN	Change measurement unit
KEY0 / KEY1	Lock / unlock keypad
MUL [+/-]xxx.xxxx	Modify multiplication factor
PRE [+/-]xxx.xxx	Modify preset value
REF1 / REF2	Change active reference
STO1 / STO0	Activate / de-activate HOLD
TOL1 / TOL0	Activate / de-activate tolerances
LCAL dd.mm.yy	Modify last calibration date
NCAL dd.mm.yy	Modify next calibration date
NUM xxxxxxxxxx	Modify the instrument number
TOL +/-xxx.xxx +/-yyy.yyy	Inputting tolerance limits
RS232 baud, N, P, S [CRLF]	RS-232 port parameter selection
RS485 baud, N, P, S [CRLF]	RS-485 port parameter selection
BUS MODBUS/ASCII/ORBIT	Protocol selection
SLA aaa	MODBUS address selection
MIN /MAX /DEL /NOR	Selecting MIN, MAX, Delta, Normal mode
CLE	Re-initialisation of MIN, MAX or Delta
UNI1 / UNI0	Activate / de-activate change of units
OUT1 /OUT0	Activate / de-activate contin. data transmission
PRE ON / PRE OFF	Activate / de-activate Preset function
PRE	Recall Preset
SET	Zero reset
RES2 / RES3	Change of resolution

Interrogation

CHA?	Measurement sense?
FCT?	«favourite» function active?
UNI?	Measurement unit active?
KEY?	Keypad locked?
MUL?	Multiplication factor?
PRE?	Preset value?
REF?	Reference active?
STO?	Status of HOLD function?
TOL?	Current tolerance limit values?
LCAL?	Date of last calibration?
NCAL?	Date of next calibration?
NUM?	Instrument number ?
?	Current value (mode Tol, value followed by <=>)
RS232?	RS-232 port parameter?
RS485?	RS-485 port parameter?
BUS?	RS-485 bus protocol?
SLA?	MODBUS address?
MOD?	Active mode (MIN, MAX, Delta or Normal)?
SET?	Main instrument parameters?
ID?	Instrument identification code?

Maintenance functions

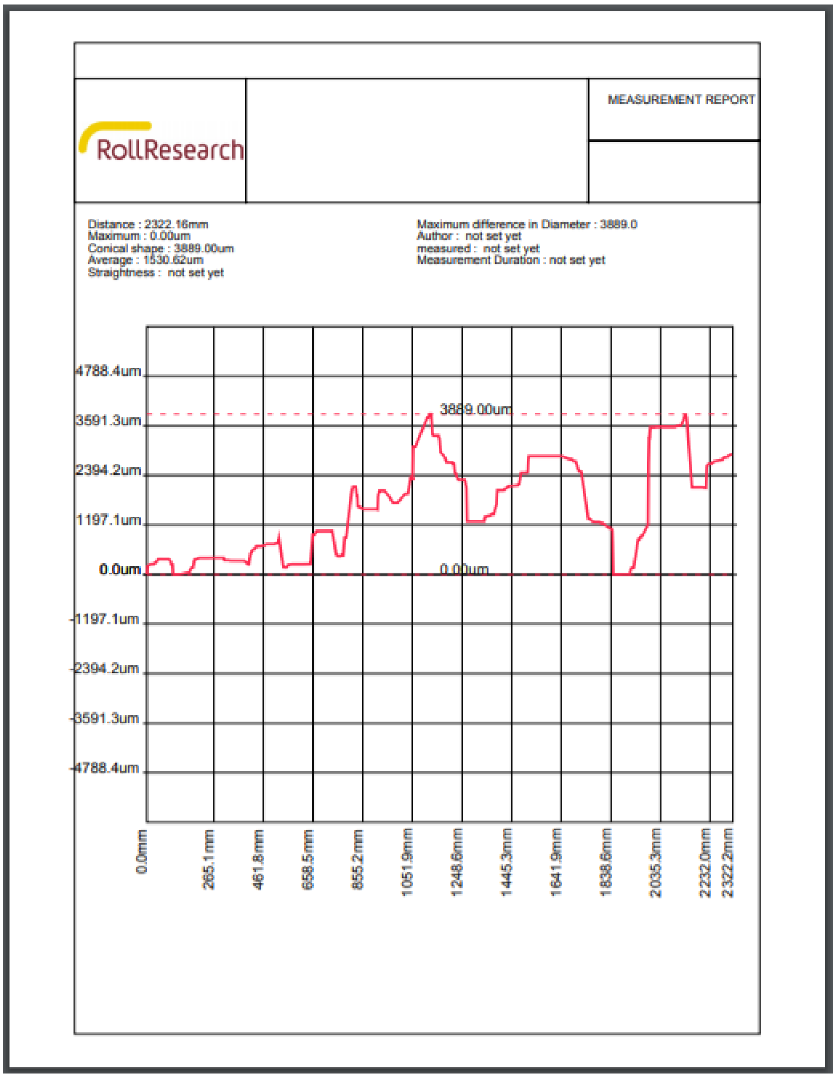
OFF	Switch-off (wake up using a button or RS)
RST	Re-initialisation of the instrument
VER?	Version No. and date of firmware

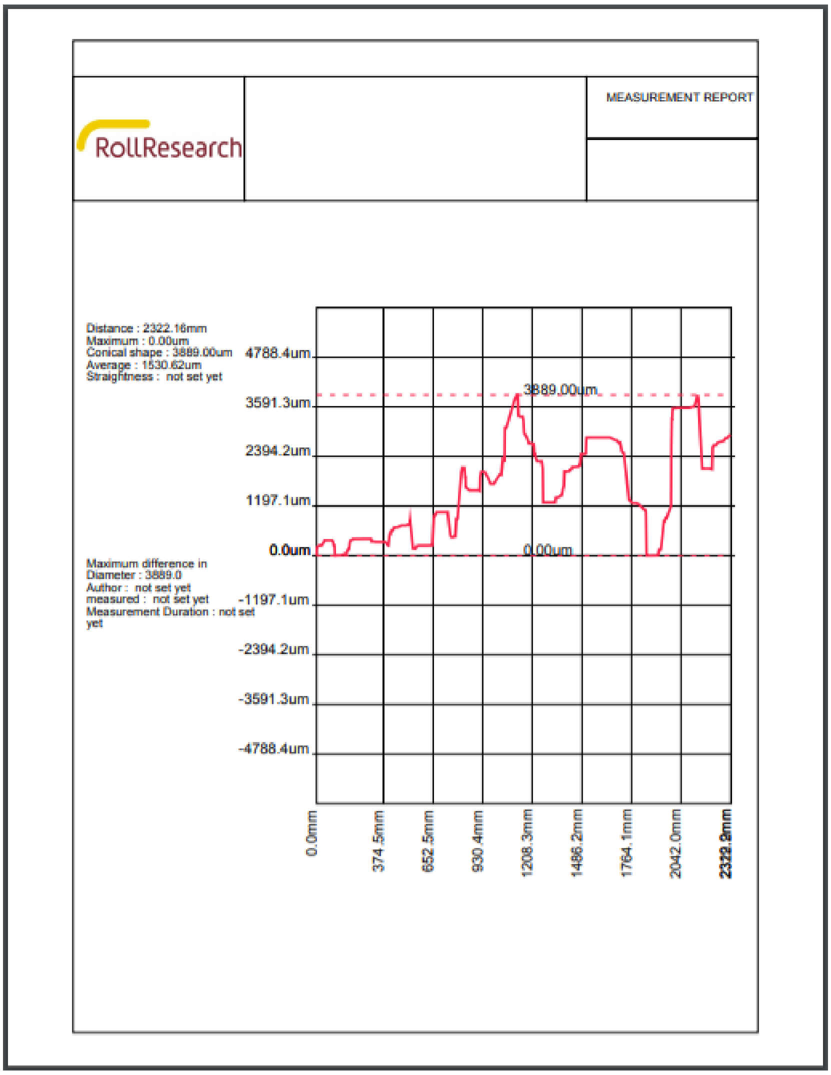
E

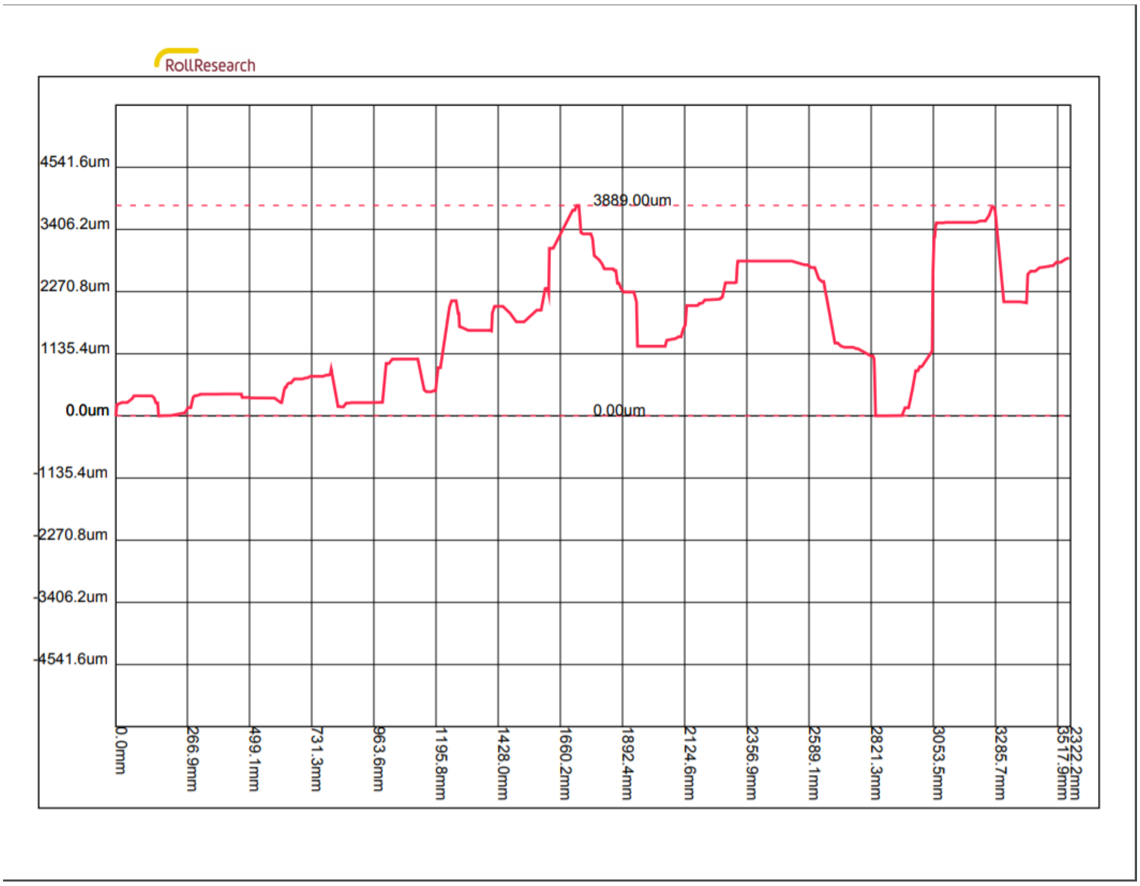
12. Specifications

Measurement range:	12.5mm	25mm	
Max error (0.01mm scale):	10µm	10µm	(±1 digit)
Max error (0.001mm scale):	4µm	5µm	
Repeatability:	2µm		
Weight:	260g	265g	
Measurement force (standard):	0.65-0.9N	0.65-1.15N	
Max. speed of travel:	1.7m/s		
No. of measurements/ sec:	measurement: 100 mes/s		
Measurement unit:	metric/english (Inch)		
Maximum Preset (0.01mm scale):	±9999.99 mm / ±399.9995 IN		
Maximum Preset (0.001mm scale):	±999.999 mm / ±39.99995 IN		
Measurement system:	Sylvac inductive system (patented)		
Power:	9....24VDC		
Average consumption:	4mA (24VDC)		
Data output on Proximity :	RS232/USB compatible (default : 19'200, 7bits, Even, 2stop)		
Data bus	RS485 compatible, charge 1/256 (default : 128'000, 8bits, Even, 1stop)		
Digital data output:	Open Collector (NPN). 24 VDC/250 mA max, integrated polarity reversal and overvoltage protection (Freewheel Diode)		
Digital data input:	Max 24 VDC, impedance 10 kΩ, integrated polarity reversal protection		
Working temperature (storage):	+5 to +40°C (-10 to +60°C)		
Electromagnetic compatibility:	as per EN 61326-1		
IP rating (in accordance with IEC60529):	IP 67 / IP 54 (depending on model)		
Fixing and space envelope:	Ø8h6 (3/8''), interchangable M2.5 (4-48-UNF) probe (as per DIN 878)		

Template samples




MEASUREMENT REPORT
RollResearch
Distance : 2322.16mm
Maximum : 0.00um
Conical shape : 3889.00um
Average : 1530.62um
Straightness : not set yet
Maximum difference in Diameter : 3889.0
Author : not set yet
measured : not set yet
Measurement Duration : not set yet
4788.4um
3591.3um
2394.2um
1197.1um
0.0um
-1197.1um
-2394.2um
-3591.3um
-4788.4um
3889.00um
0.00um
0.0mm
374.5mm
652.5mm
930.4mm
1208.3mm
1486.2mm
1764.1mm
2042.0mm


RollResearch
4541.6um
3406.2um
2270.8um
1135.4um
0.0um
-1135.4um
-2270.8um
-3406.2um
-4541.6um
3889.00um
0.00um
0.0mm
266.9mm
499.1mm
731.3mm
963.6mm
1195.8mm
1428.0mm
1660.2mm
1892.4mm
2124.6mm
2356.9mm
2589.1mm
2821.3mm
3053.5mm
3285.7mm
3517.9mm
3322.2mm