Development of the Detector Control System for the LHCb RICH Upgrade

Summer Student Report

Created by:

Kevin Syc

Supervised by:

Giovanni Cavallero

Carmelo D'Ambrosio

August 16, 2019

Abstract

LHCb is one of the four main experiments running at the Large Hadron Collider (LHC). Its main purpose is to perform precise measurements to study CP violation and rare decays of b and c quarks. The RICH detectors are crucial components in identifying these decays by providing identification of charged hadrons. The LHCb experiment is currently undergoing an upgrade during the long shutdown (2019-2020) in order to run with a five-fold increase in the instantaneous luminosity. The LHCb detector will be readout at the full LHC bunch-crossing rate of 40MHz. This will allow to fully exploit the delivered luminosity of $\mathscr{L} = 2 * 10^{33}\ cm^{-2}s^{-1}$ during the upgrade [1]. The upgraded RICH detectors will have significant modifications to RICH1 optics and mechanics. Moreover, new photo detectors and frontend electronics will be installed in both RICH1 and RICH2. This will require a new Detector Control System (DCS) to ensure proper running of the upgraded RICH detectors. Both must run at the maximum performance, but at the same time be safe to operate without damaging any fragile electronics or mechanical parts. Namely three different environmental conditions need to be monitored within RICH detectors: temperature, pressure and humidity. These conditions are measured using different sensors which must be tested, calibrated and implemented within the DCS and Detector Safety System (DSS) before being hard-wired to the detectors. Level of precision in monitoring these conditions is directly related to the ability to properly determine the mass of a particle and thus determine its type. Charged particles pass through C_4F_{10} gas present in the vessel. Due to Cherenkov Effect, photons are emitted at an angle relative to the trajectory of the track, called Cherenkov angle. This angle is given by: $\cos\theta = \frac{1}{\beta * n}$ where $\cos\theta$ is the Cherenkov angle, β is the speed of a particle over the speed of light and n is the refractive index of a gas in the vessel. To identify mass of a particle, β is desired and thus, we also need to know the refractive index, n, precisely. Because $n = n(p, T)$, the extent to which we know n is determined by our ability to precisely measure pressure and temperature [2,3].

<u>Key words:</u> temperature, pressure, humidity, WinCC-OA, Detector Control System, Detector Safety System

Acknowledgements

I would firstly like to express my gratitude towards my main supervisor, Giovanni Cavallero, for supporting me throughout this project during the whole summer. He has taught me numerous things without which I would be unable to perform the experiments and conduct this summer project. His insights combined with passion within the field have made this experience much more enjoyable and educative.

Secondly, I would like to mention help of Silvia Gambetta and Thierry Gys who also greatly contributed to the success of this project by providing me lot of information about RICH detectors, LHCb as a whole and helped me with the technical aspects of this project.

I would also like to mention help of Antonis Papanestis who helped me with technical aspects of the project as well as provided lot of insights into the work behind the LHCb operations which were much needed for my project. I would further like to thank my secondary supervisor, Carmelo D'Ambrosio for providing feedback during presentation and supporting me throughout the project.

Last but not least, I would like to thank my whole office for making the working environment enjoyable, calm and relaxing.

Table of Contents

Abstract 2

Acknowledgements 2

Aims and Objectives of the Project 4

1. Background reading of DCS and DSS of RICH detectors 4

2. Familiarising with WinCC-OA 3.16 and C language 4

3. Connecting Temperature Sensors to ELMB 5

4. Building a WinCC-OA (3.16) monitoring panel, reading out temperature sensors and testing OPC-UA server on Linux platform. 6

5. Testing Different Configurations of Temperature Sensors 7

6. Connecting humidity sensor and implementing the data in WinCC panel 9

7. Connecting pressure sensor and implementing the data in WinCC panel 10

8. Implementation of sensors within RICH DCS Finite State Machine. 11

9. Conclusion 14

References 14

Appendices 14

Appendix 1: WinCC-OA control panel used to monitor the temperature, humidity and pressure data. 14

Appendix 2: Accuracy of temperature sensors. 16

Aims and Objectives of the Project

In order to understand the DCS of RICH detectors and implement monitoring devices of environmental conditions to ensure the proper running of the whole LHCb experiment, a number of objectives had to be completed as part of this project:

1. Background reading of DCS and DSS of RICH detectors.
2. Familiarising with the WinCC-OA SCADA tool, JOINT CONTROLS (JCOP framework) and C language.
3. Connecting temperature sensors to the Embedded Local Monitoring Board (ELMB).
4. Building a WinCC-OA (3.16) monitoring panel, reading out temperature sensors and testing OPC-UA server on Linux platform.
5. Testing different configurations of temperature sensors.
6. Connecting humidity sensors and implementing the data in WinCC-OA panel.
7. Connecting pressure sensors and implementing the data in WinCC-OA panel.
8. Implementation of temperature sensors within the RICH DCS and DSS Finite State Machine (FSM).

1. Background reading of DCS and DSS of RICH detectors

Background reading of DCS and DSS of RICH detectors began over the summer, prior to arrival on CERN site [4]. The background reading provided lot of insights into the architecture of DCS and DSS in LHCb and different sensors with their respective locations used to monitor the environmental conditions to ensure proper running of all electronics. Information about ELMB and wiring used to connect sensors to ELMB were obtained from the document which was very helpful from the beginning of the project. The document concerned can be found under the following link:

https://www.researchgate.net/publication/267972338_LHCb_Note_LHCb_The_LHCb_RICH_Detector_Control_System_Requirements_for_Monitoring_and_Control_of_the_RICH_Detectors

2. Familiarising with WinCC-OA 3.16 and C language

WinCC-OA is a standard software used by the LHC experiments to monitor and control the detectors. The production version for the upgraded LHCb experiment control system will be WinCC-OA 3.16 on Linux platform. The software uses a C language to connect data from ELMBs with graphical interface and build functional panels. There is a standard set of widgets and functions in use within JCOP framework (CERN standard) which can be used to remove the need to write custom scripts. Within this framework, it is easy to connect and manage multiple ELMBs and their inputs/outputs with fairly fast implementation.

Set of slides and exercises were given at the beginning of the project to help explain the software. These exercises were part of the course ‘PVSS Service Training PVSS-JCOPFw Course’ created by Manuel Gonzalez Berges. The slides and exercises were particularly helpful in explaining the basic features of WinCC like building a panel, using graphical interface, writing scripts, using JCOP framework, archives, alarms, log-in credentials and many more.

3. Connecting Temperature Sensors to ELMB

The Embedded Local Monitoring Board (ELMB) is a general purpose plug-in Input/Output module for the monitoring of front-end electronics and DCS devices of subdetectors based on industry standard CANbus [5]. The ELMB used to connect the temperature sensors is depicted in *Figure 1*. It has 64 multiplexed ADC channels (16bit). Channels 0-15 were used for 4 wire configuration of temperature sensors, while channels 16 – 63 were used for 2 wire configuration (See section 5). The board required 2 power supplies of 10V each (*Figure 2*). The information was transmitted via CANbus to CANbus-to-USB converter, following to the computer. CAN driver together with the OPC-UA server were successfully tested and used for this project on Linux computer which will be a new standard software framework used for the LHCb.

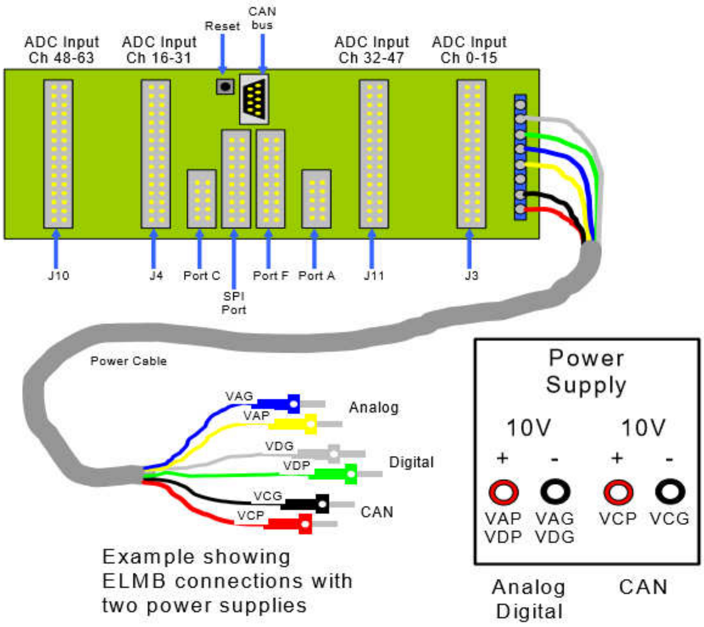


Figure 1: ELMB used to read out temperature data [4].

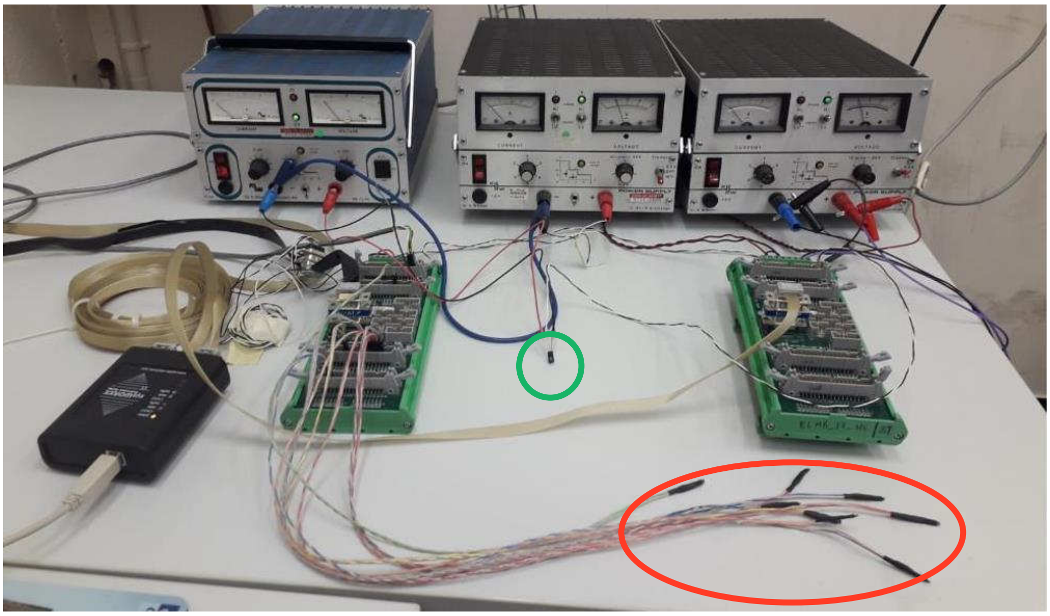


Figure 2: ELMB setup in the lab. ELMB on the left is used to gather input from temperature sensors (circled in red). ELMB on the right gathers the input from humidity sensor (circled in green).

A total of 7 temperature sensors (Pt1000) were used in the setup (circled sensors in Figure 2). These temperature sensors measured the temperature of air in the lab and only served as a reference to have values while developing WinCC-OA panel for later use.

4. Building a WinCC-OA monitoring panel, reading out temperature sensors and testing OPC-UA server on Linux platform.

The first step in building a WinCC-OA panel to monitor the sensor data was to set out some basic requirements of the panel. Based on the experience from the exercises provided before and after consultation with Giovanni, the following design requirements were worked out:

1. The Panel needs to have live feed of data from ELMB.
2. The Panel needs to be able to archive data of arbitrary time scale.
3. The Panel needs to have the option to plot data.
4. The Panel needs to have alarms implemented.

These were the basic functionalities in order to collect and analyse the data. Refer to *Appendix 1* for visual representation of the panel. Firstly, live feed of data had to be implemented. This was done using a ‘Table’ widget, which contained all 63 channels of ELMB with their corresponding wire configuration, which was either 2 or 4 wires. This table was updated every 5 seconds, with the read-out rate of sensors being 10 seconds. The alarms were implemented in the very same table where the colour behind the values in the table changed according to the alarm status together with the text displayed. There were 3 alarm statuses set: Ok, Warning and Error. In real operating conditions, error status would trigger a Low Voltage and High Voltage power cut, yielding the RICH detectors and eventually the whole LHCb unable to collect data for safety reasons.

The top right corner of the panel was dedicated to the data archiving. User could choose to archive a specific channel from the drop-down menu or simply archive all the channels at once. The user could also choose a format (txt or csv) for exporting the data. Time period settings were chosen by clicking on the button and browsing the calendar to choose the initial value and final value of time period. Option of writing table data was implemented as well, which enables users to write the exact table data in txt format. TREND HISTORY button is a standard widget in JCOP framework opened on second panel to display the graphs and history of channels. This was implemented in the archive settings to give user a quick access to the data.

Middle section contains a custom graph which is not part of JCOP framework, but rather written with custom script. As opposed to JCOP framework ‘TREND HISTORY’, this custom plot area is able to display more than 8 channels at once, which comes useful for debugging and commissioning. User can choose which channels to plot from the menu on the right. If the channel chosen is currently not in use, the debug window prints ‘Channel inactive’. The user can change plot settings like time period, y-axis, colour, grid and type of the display data from the menu right above the graph.

Few extra functionalities were added on top. User can choose to show only active channels in the table, or turn off the live feed of the data. Moreover, there are simple analytic solutions like maximum, minimum and average value of the current displayed values.

5. Testing Different Configurations of Temperature Sensors

There are 2 types of temperature sensors to choose from for the final detector. Pt100 and Pt1000. On top of that, there are also 2 configurations in which these can be wired. It is either 2 wires configuration or 4 wires configuration. In order to understand the difference between data taken by these 2 configurations and 2 sensors, all possible configurations were tested and data later compared. The temperature data were taken on real, assembled and operational column as it will be in the final LHCb (*Figure 3*). The temperature sensors used were also standard temperature sensors which will be used in the final detector (*Figure 4*).

Figure 3: Left side is the CAD of the column as designed before the upgrade, while right side is the real, operational column. The real column has 2 temperature sensors connected in it for convenient swapping between sensors. In reality, left sensor will be replaced by mechanical, thermo-switch.

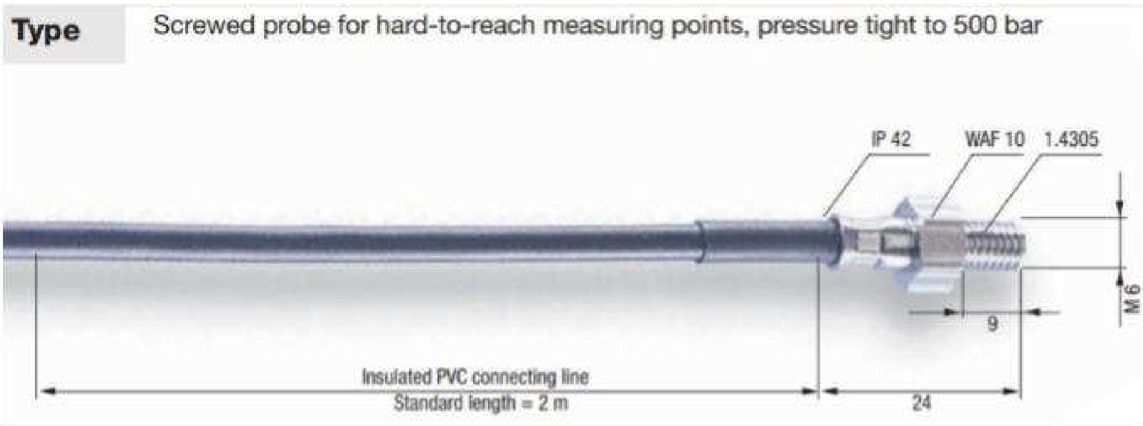


Figure 4: Standard temperature sensors used in the final detector [6].

This temperature sensor works by measuring the resistance of the active element in it, in this case platinum. For Pt100 the resistance at 0°C is equal to 100Ω, while for Pt1000, the resistance at 0°C is 1000Ω = 1kΩ. As the temperature rises, the resistance increases. In the case of Pt100, the resistance increases by roughly 0.39Ω/1°C, while for Pt1000 the resistance increases by roughly 3.9Ω/1°C. The difference between 2 wires and 4 wires configurations is that the 2 wires configuration does not account for the resistance of cables, while the 4 wires configuration does, thus making the results more accurate.

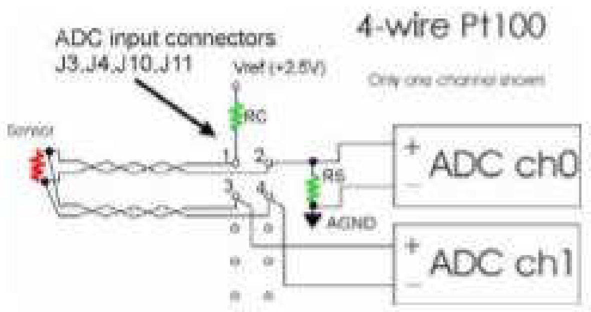


Figure 5: 4 wires configuration of temperature sensor [5].

In the case of 4-wire configuration, the voltage drop across the cables is taken into consideration and the sensor resistance is given by the following equation:

$$R(t) = \frac{ch_1 * R_s}{ch_0}$$

Where R(t) is the total resistance of sensor, ch_1 is the resistance as read by channel 1, ch_0 is the resistance as read by channel 0 and R_S is the value of resistor near Channel 0. The performance of this configuration is essentially determined by this resistor and its value was measured to be **100Ω**. The resistor RC determines the current flowing through the setup. This value is **3.9kΩ**. In order to convert the resistance readings into temperature readings, the following formula must be used:

$$t = \frac{-a + \sqrt{a^2 - 4b(1 - R(t)/R_0}}{2b}$$

where the values *a* and *b* are constants provided by the manufacturer and their typical values are **$3.9083*10^{-3}$** and **$-5.775*10^{-7}$** respectively. R_0 is the resistance of sensor at 0°C, which is 100 and 1000 for Pt100 and Pt1000 respectively.

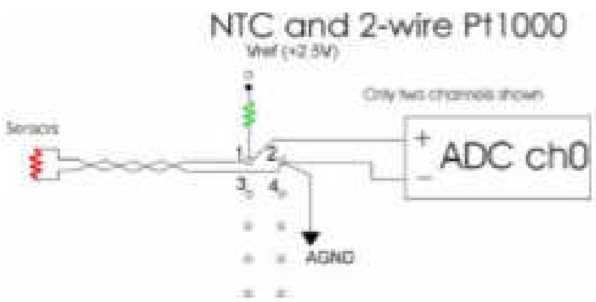
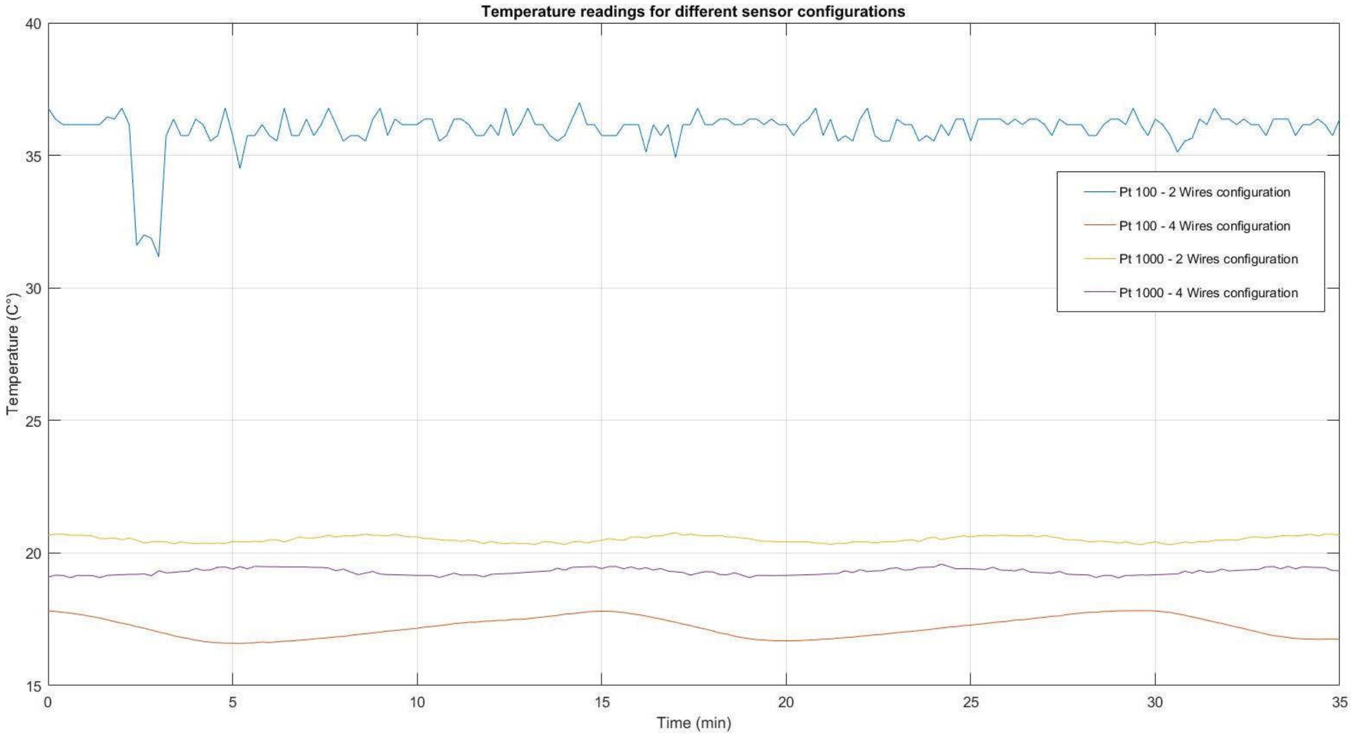


Figure 6: 2 wires configuration of temperature sensor [5].

In the case of 2 wires configuration, the voltage drop across cables is not taken into consideration and therefore the results are less accurate. The equation to calculate the temperature is the same as in the case of 4 wires configuration, however, the calculation for the resistance R(t) is given by:

$$R(t) = \frac{ch_0 * R_A}{2.5 - ch_0}$$

Where ch_0 is the resistance measured by channel 0 and R_A is the resistance of adaptor. In the case of ELMBs in the lab, the value of R_A is 200kΩ [2]. Refer to *Appendix 2* for detailed calculations of expected precisions of both configurations. All 4 different configurations were tested in the lab (Pt1000 -2 and 4 wires and Pt100 2 and 4 wires) and the results are summarised in *Graph 1*.



Graph 1: Comparison of performance of different temperature sensor configurations on column.

From Graph 1, it can be observed that the worst performer turned out to be Pt100 in 2 wires configuration. This configuration is showing the biggest temperature and the biggest amount of fluctuations. This is caused by the resistance in cables being too large and not accounted for. All of the other configurations were roughly of the same precision. The best performer turned out to be Pt100, 4 wires configuration which was also showing the smallest temperature measured. In Pt1000, 4 wires configuration it was also necessary to change the Analog-to-Digital Converter (ADC) range from 0.1V to 1V. This ensured that the measured voltage was within the range, and thus measurable by the ELMB. The ADC range is also directly related to the precision that one can achieve. The larger the range, the larger the steps and thus less precise the readings. Even though Pt1000 2 wires configuration did not account for cable resistance, the resistance of the element is large compared to the resistance of cables themselves (1000Ω compared to roughly 4.5Ω from cables), thus the readings were rather precise and overall resistance was not significantly affected.

6. Connecting humidity sensor and implementing the data in WinCC-OA panel

Figure 7 shows the type of humidity sensor used for measurements in the lab. It is a Honeywell HIH-4000-004 relative humidity sensor. It is powered by 5V from a separate power supply. As opposed to temperature sensors, the adaptor used in ELMB to read the data from this type of sensor reads out the voltage only and then the user changes the voltage in JCOP framework directly, while temperature sensor data are changed directly in the OPC-UA server.

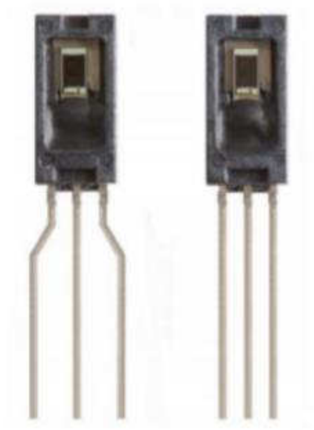


Figure 7: Humidity sensor used for measurements in the lab [7].

The connections were wired in the lab of building 16 at CERN site. Once connected, the data readings were implemented in WinCC-OA panel created before by placing a drop down button to choose different set of data to be displayed in the table (See Appendix 1). Alarms and archive were implemented for the humidity data as well and the result can be seen in *Figure 8*. In later stage, 2 humidity sensors were used to cross check the values. The humidity sensor was later installed in the box hosting the test of the column on to check that the temperature is above the dew point. If the temperature is below the dew point temperature, condensation occurs and this can damage electronics. In this case, HV and LV are switched off.

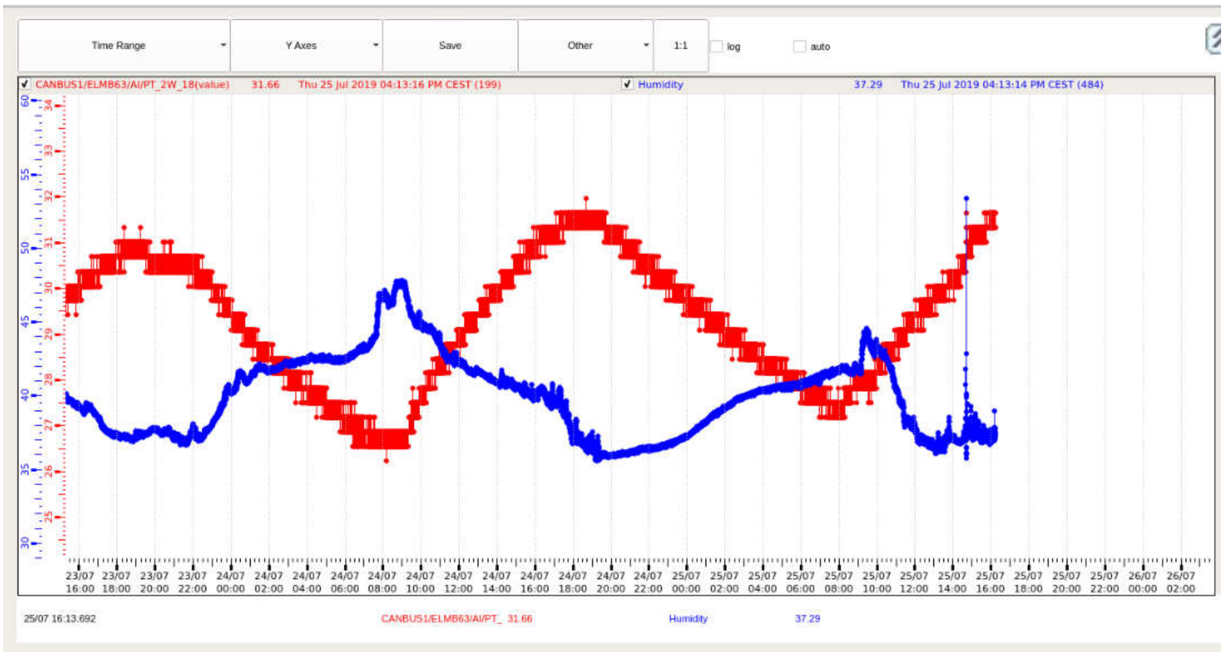


Figure 8: The dependency of humidity of the air compared to the temperature of the air in the lab over several days. Red data correspond to temperature readings while blue data correspond to relative humidity readings.

7. Connecting pressure sensor and implementing the data in WinCC-OA panel

Absolute pressure sensor shown in *Figure 9* was used to connect and measure the pressure in the lab. At later stage, 1 more pressure sensor was connected to cross check the values. The readings indicated reasonable measurement of roughly 101000 Pa which corresponds to the atmospheric pressure as expected. 32 Channels (Channels 32 – 63) of ELMB_17 were dedicated for pressure readings.

Figure 9: Absolute pressure sensor used in the lab [8].

Chnl. number	Alias	Status	Value	Wires config.	Alarm Status
Channel 32		Inactive	0.00	2 Wires	ressure Very LOV
Channel 33		Active	102068.09	2 Wires	Pressure OK
Channel 34		Inactive	0.00	2 Wires	ressure Very LOV
Channel 35		Inactive	0.00	2 Wires	ressure Very LOV
Channel 36		Active	101736.68	2 Wires	Pressure OK
Channel 37		Inactive	0.00	2 Wires	ressure Very LOV
Channel 38		Inactive	0.00	2 Wires	ressure Very LOV
Channel 39		Inactive	0.00	2 Wires	ressure Very LOV
Channel 40		Inactive	0.00	2 Wires	ressure Very LOV
Channel 41		Inactive	0.00	2 Wires	ressure Very LOV
Channel 42		Inactive	0.00	2 Wires	ressure Very LOV
Channel 43		Inactive	0.00	2 Wires	ressure Very LOV
Channel 44		Inactive	0.00	2 Wires	ressure Very LOV
Channel 45		Inactive	0.00	2 Wires	ressure Very LOV
Channel 46		Inactive	0.00	2 Wires	ressure Very LOV
Channel 47		Inactive	0.00	2 Wires	ressure Very LOV
Channel 48		Inactive	0.00	2 Wires	ressure Very LOV
Channel 49		Inactive	0.00	2 Wires	ressure Very LOV
Channel 50		Inactive	0.00	2 Wires	ressure Very LOV
Channel 51		Inactive	0.00	2 Wires	ressure Very LOV
Channel 52		Inactive	0.00	2 Wires	ressure Very LOV
Channel 53		Inactive	0.00	2 Wires	ressure Very LOV
Channel 54		Inactive	0.00	2 Wires	ressure Very LOV
Channel 55		Inactive	0.00	2 Wires	ressure Very LOV
Channel 56		Inactive	0.00	2 Wires	ressure Very LOV
Channel 57		Inactive	0.00	2 Wires	ressure Very LOV
Channel 58		Inactive	0.00	2 Wires	ressure Very LOV
Channel 59		Inactive	0.00	2 Wires	ressure Very LOV
Channel 60		Inactive	0.00	2 Wires	ressure Very LOV
Channel 61		Inactive	0.00	2 Wires	ressure Very LOV
Channel 62		Inactive	0.00	2 Wires	ressure Very LOV
Channel 63		Inactive	0.00	2 Wires	ressure Very LOV

Figure 10: Table from WinCC-OA panel showing pressure readings.

8. Implementation of sensors within RICH DCS Finite State Machine.

RICH DCS Finite State Machine (FSM) is a control environment within WinCC-OA where sensors can be implemented and controlled. FSM monitors the states of its children and reports an overall state to its parent. It monitors WinCC-OA datapoints representing hardware, in this case our sensors and it is fully integrated within JCOP framework.

In order to implement the sensors, parent control node named 'Environmental Conditions' was first created in Device Editor and Navigator in JCOP framework. 3 nodes with temperature, humidity and pressure readings were then created under this parent, each one containing its corresponding children, the datapoints or in other words, the sensor readings (*Figure 11*).

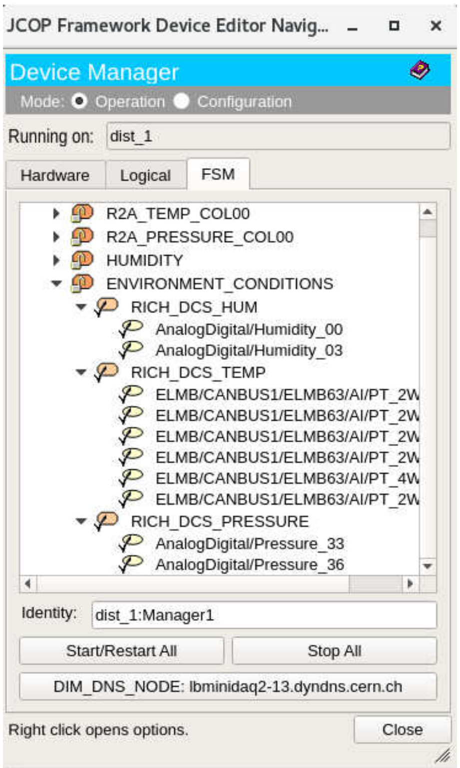


Figure 11: FSM architecture of temperature, pressure and humidity sensors.

Parent environmental conditions can be opened, where all 3 children and their status can be seen. There are 4 states that a children node can have. “Ready”, “Warning”, “Invalid” and “Error”. Ready state corresponds to good readings within limits and perfectly working sensors. Warning state corresponds to values that are outside of operating range but still within safe limits. Error state is a state when the readings are way off the normal operating range and this could cause significant damage to the electronics and photon detectors. Even if one of the children within any of the nodes is in Warning state, the warning propagates and eventually turns off the ENVIRONMENT_CONDITIONS parent to prevent damage of electronics. Invalid state corresponds to inputs that have invalid value, for example “-nan”, thus disconnected or somehow corrupted sensors.

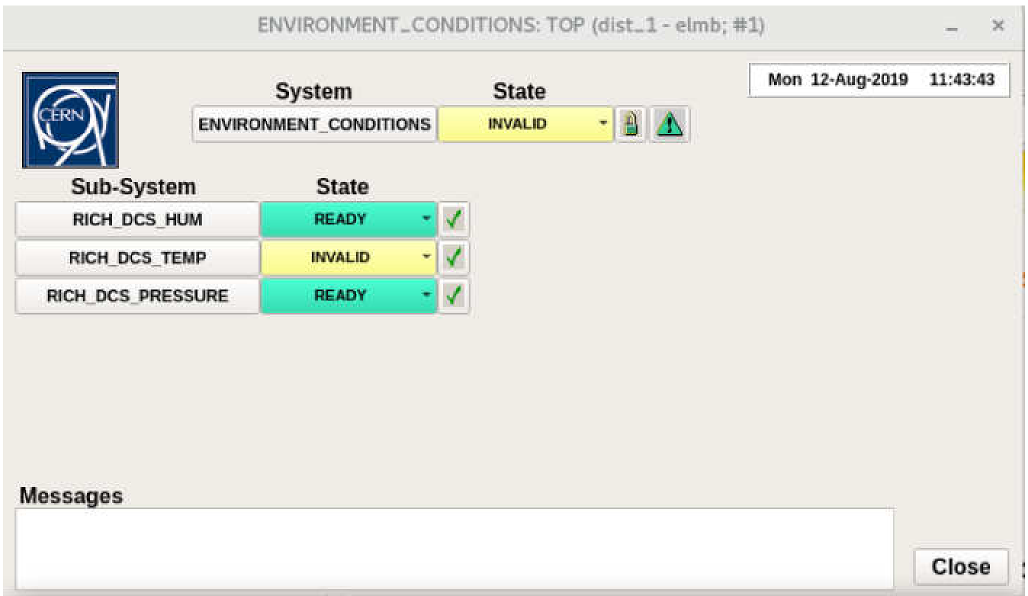


Figure 12: Parent node with children and their corresponding states in FSM.

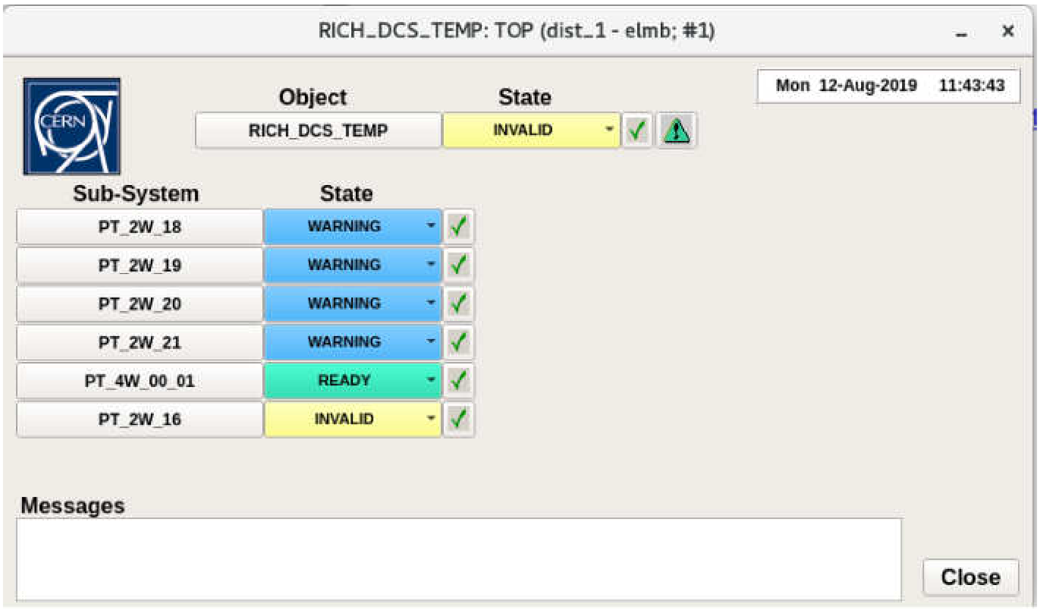


Figure 13: Temperature node with its children. PT_4W_00_01 corresponds to column temperature reading.

By clicking on any of the temperature readings within this node, operator has also chance to see the trend and check the history of the values. In this case, one of the sensors was deliberately disconnected in order to demonstrate that the whole parent containing temperature sensors as its children will be INVALID if one of the sensors is INVALID. If, however, the operator chooses to disregard this sensor, there is an option to disable the sensor. In this case, the parent will ignore this sensor totally.

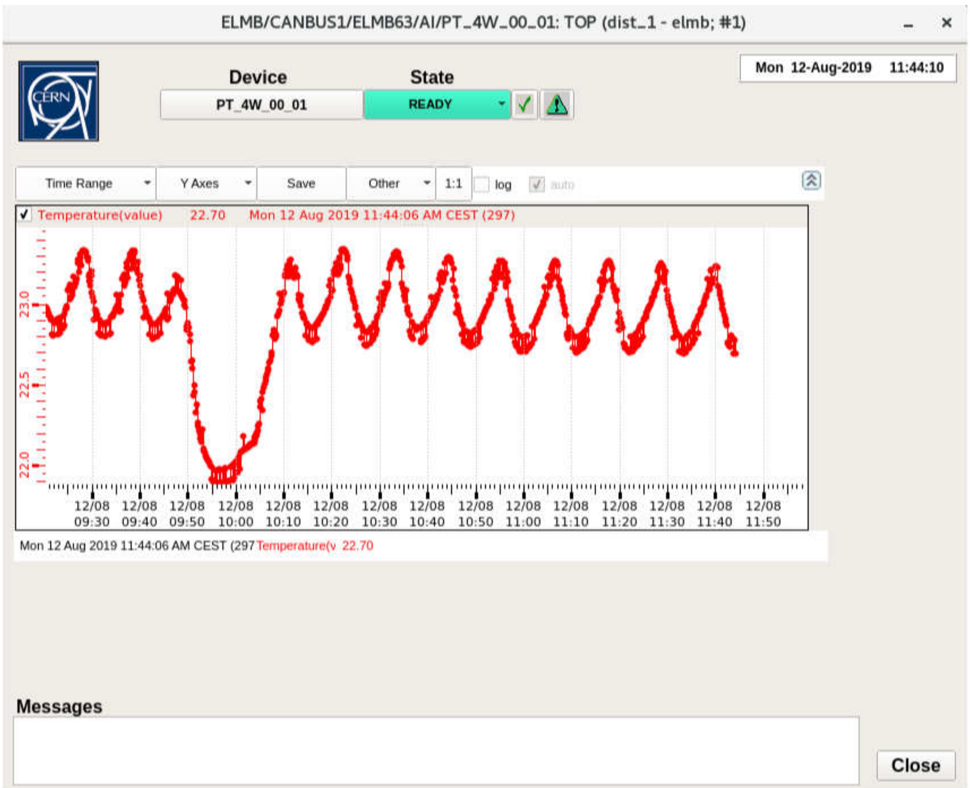


Figure 14: Trend history of temperature reading of column.

9. Conclusion

During my time at CERN, I have completed several parts of my projects. I have implemented, tested and wired temperature, pressure and humidity sensors into FSM architecture and built a convenient control panel to monitor the environmental conditions in the operating column. My technical skills in the field have greatly improved and I gained experience in slow control devices used in the LHC, standard software tools used in the Experiment Control System and the language of finite state machines with implementation of alarms, triggers and automation of data taking. I am now more confident in C language coding and implementing control systems in both HW and SW. Moreover, I improved my presentation and report writing skills as well as gained insights into the operations of LHCb and why the work performed is important.

References

1. Fiorini M, *The LHCb RICH Detector Upgrade*. 2017 European Physical Society Conference on High Energy Physics. 2017; 494-499

2. Inami K, *Cherenkov light imaging in particle and nuclear physics experiments.* Presented at; RICH2016, Slovenia.

3. Gambetta, S., Nuclear Instruments and Methods in Physics Research A (2017), http://dx.doi.org/10.1016/j.nima.2017.02.079

4. D'Ambrosio D, Metlica F, Papanestis A, Sannino M, Soler F.J.P., Ullaland O et al, *The LHCb RICH Detector Control System.* ResearchGate. 2004.

5. Cook JR, Thomas G, *ELMB128 Documentation: Everything you wanted to know about the ELMB128 but were afraid to ask.* 2008 Available from: http://lhcb-vd.web.cern.ch/lhcb-vd/ECS/ElmbUserGuide.pdf [Accessed 15th August 2019]

6. Testo, *Stationary temperature Probes,* 6, Available at: https://www.approvedbusiness.co.uk/files/companypdffiles/6912/Testo%20-%20Stationary%20temperature%20probes.pdf

7. Honeywell, *HIH-4000 Series Humidity Sensors,* 009017-5-EN GLO, February 2010, Available at: https://docs-emea.rs-online.com/webdocs/144e/0900766b8144ea72.pdf

8. Honeywell, *TruStability Board Mount Pressure Sensors,* 50099148-A-EN IL50, August 2014, Available at: https://docs-emea.rs-online.com/webdocs/14f2/0900766b814f2293.pdf

Appendices

Appendix 1: WinCC-OA control panel used to monitor the temperature, humidity and pressure data.

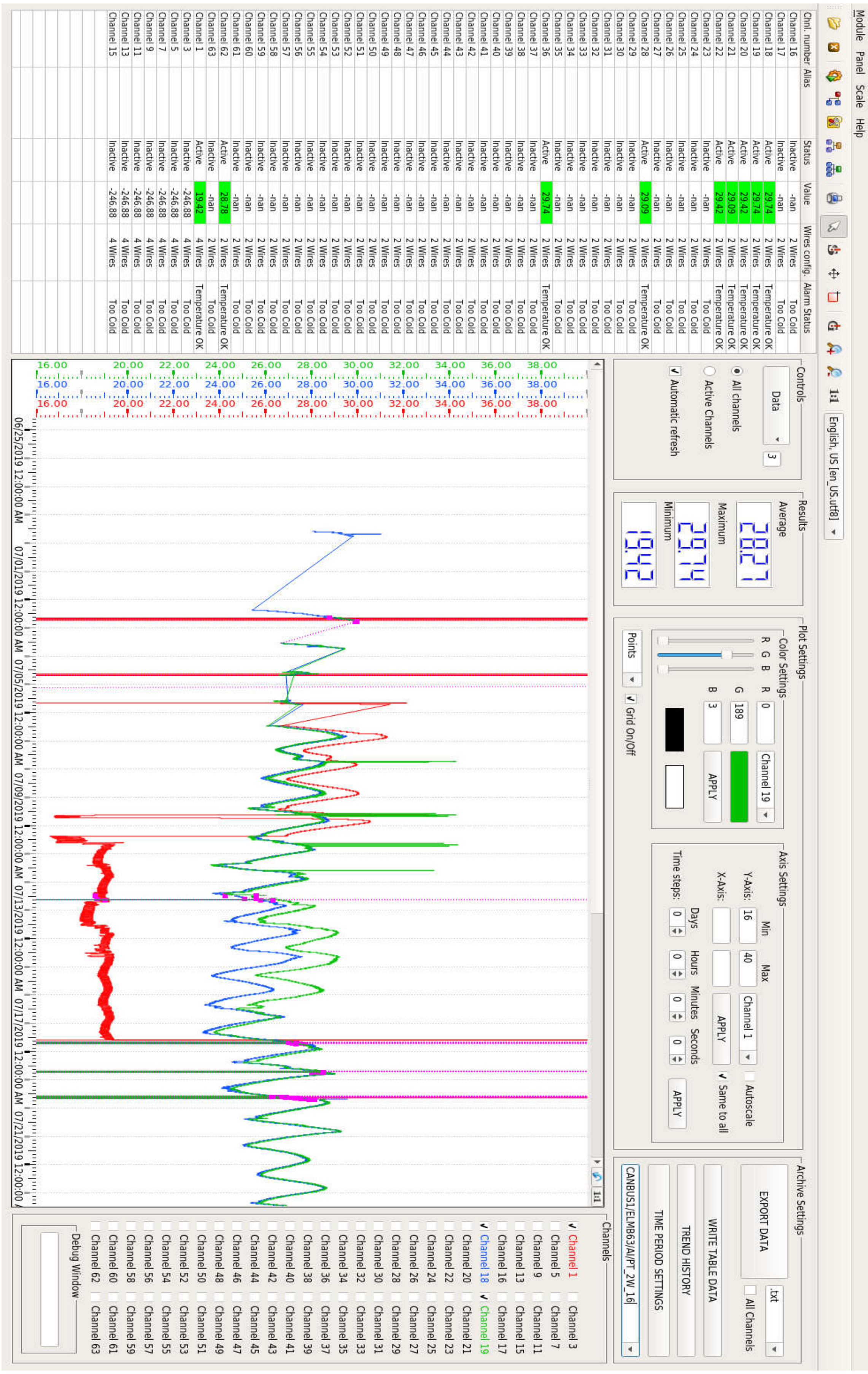


Figure 15: WinCC-OA monitoring panel.

Appendix 2: Accuracy of temperature sensors.

The accuracy of temperature sensors is dependent upon their configuration. For 4 wires configuration, there are 3 resistors that need to be taken into account in the circuit. Diagram from Figure 5 can be redrawn into the circuit below:

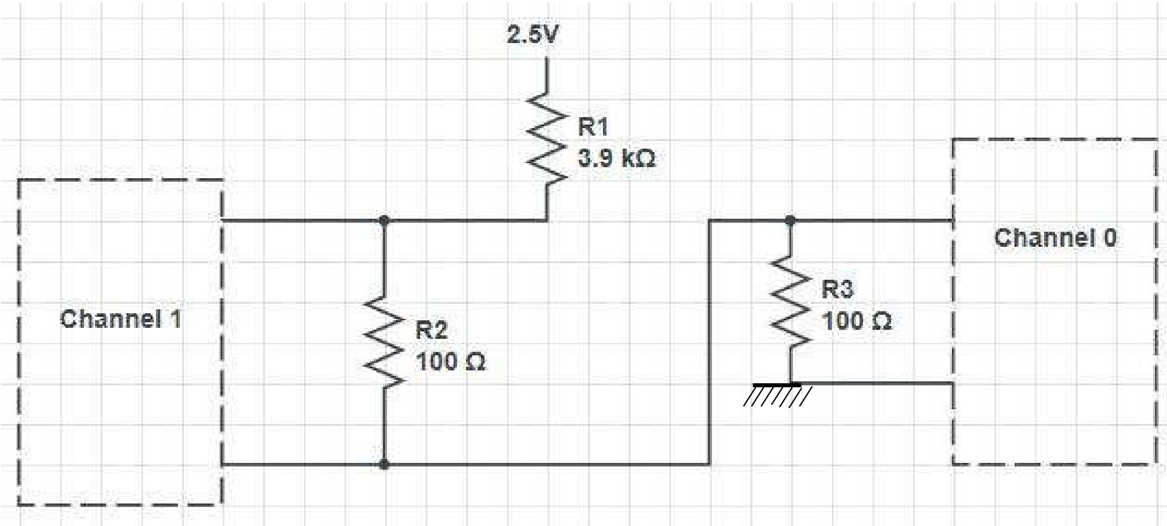


Figure 16: Schematics of 4 wire configuration.

The calculation given to find out the precision achieved is given as following. Firstly, current at 0°C through the circuit is calculated:

$$I_0 = \frac{2.5V}{3.9k\Omega + 100\Omega + 100\Omega} = 6.098 * 10^{-4}A = 609.8\mu A$$

Secondly, Voltage across R3 is calculated:

$$V_{R3} = R_3 * I_0 = 60.98mV$$

Thirdly, voltage across R2 is calculated:

$$V_{R2} = R_2 * I_0 = 100\Omega * 609.8\mu A = 60.98mV$$

Current across circuit at 20°C is calculated:

$$I_{20} = \frac{2.5V}{3.9k\Omega + 107.79\Omega + 100\Omega} = 6.086 * \ 10^{-4}A = 608.6\mu A$$

Voltage across R2 at 20°C is calculated:

$$V_{R2} = R_2 * I_{20} = 107.79\Omega * 608.6\mu A = 65.6mV$$

Assume linear interpolation to find the gradient:

$$\frac{65.6mV - 60.98mV}{20°C - 0°C} = 0.231mV/°C$$

Since ELMB is 16 bit board, there are 2^{16} levels, so 65536 levels. Assuming the ADC range of 100mV, we get 0.0015mV per ADC level. This in turn means that the precision of this sensor is given by:

$$Precision\ (@ADC\ range = 100mV) = \frac{0.0015mV}{0.231mV/°C} = \mathbf{0.0065°C}$$

2 Wires configuration can be redrawn as follows:

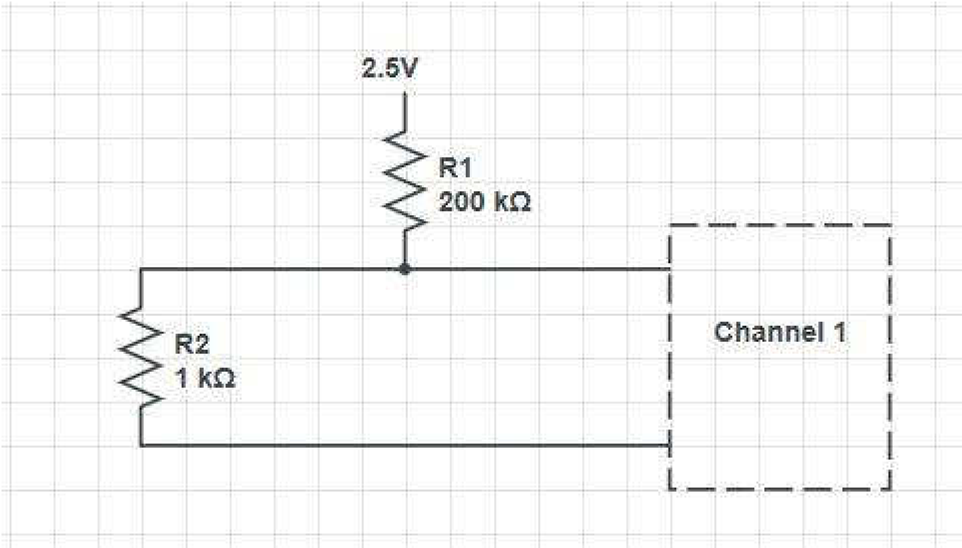


Figure 17: Schematics of 2 wires configuration.

The first step is again, finding the current through the circuit at 0°C.

$$I_0 = \frac{2.5V}{200k\Omega + 1k\Omega} = 12.44 * 10^{-6}A = 12.44\mu A$$

Second step, voltage across R2 at 0°C:

$$V_{R2} = R_2 * I_0 = 12.44mV$$

Thirdly, calculating current through the circuit at 20°C:

$$I_{20} = \frac{2.5V}{200k\Omega + 1077.9\Omega} = 12.433\mu A$$

Calculating voltage across R2 at 20°C:

$$V_{R2@20°C} = R_{2@20°C} * I_{20°C} = 13.40mV$$

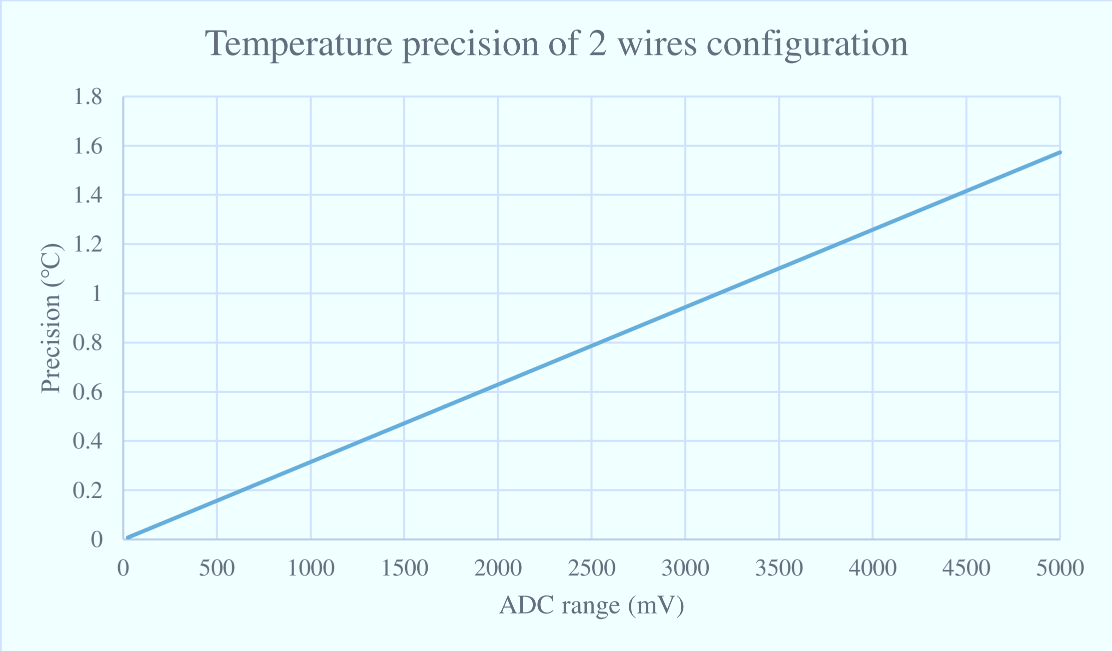
Assuming linear approximation, the change in voltage is given by:

$$\frac{13.40mV - 12.44mV}{20°C - 0°C} = 0.048mV/°C$$

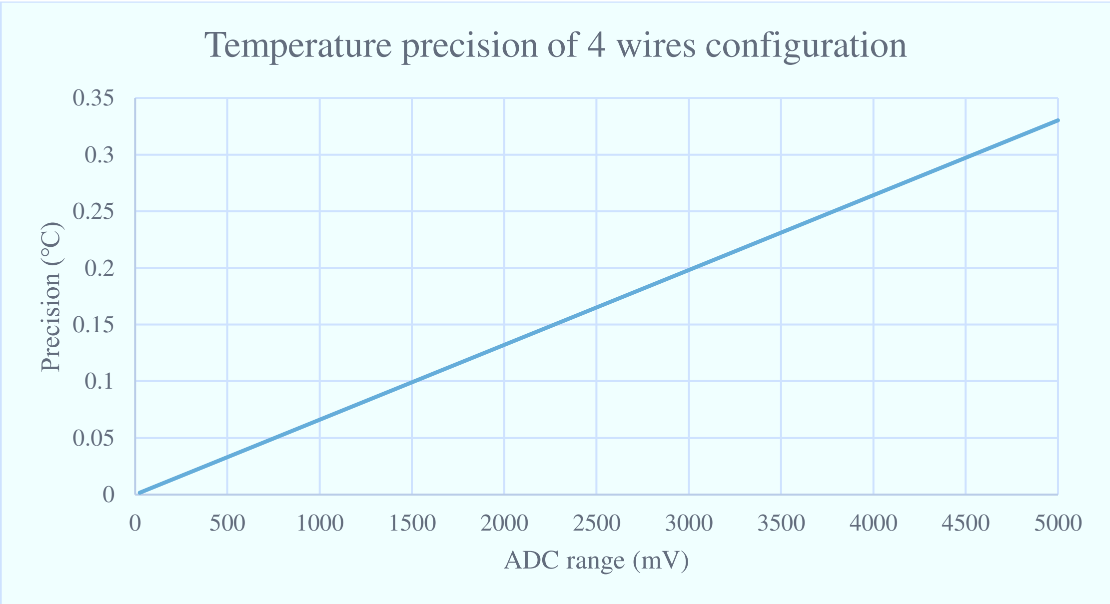
Assuming the ADC range of 100mV, the precision is given by:

$$Precision\ (@ADC\ range = 100mV) = \frac{0.0015mV}{0.048mV/°C} = \mathbf{0.03°C}$$

The effect of ADC range on the precision of Pt100 and Pt1000 can be seen in Figure 18.



Graph 2: Temperature precision of 2 wires configuration sensors.



Graph 3: Temperature precision of 4 wires configuration sensors.