Vaasan yliopisto
UNIVERSITY OF VAASA

Hind Aldhaher

Evaluating Internal and External Data Points in Long-term Periodical Testing with Protection Relays

School of Technology and Innovations
Master's thesis in Automation and Computer Science
Energy and Information Technology

Vaasa 2022

VAASAN YLIOPISTO
Teknologian akateeminen yksikkö
Tekijä: Hind Aldhaher
Tutkielman nimi: Evaluating Internal and External Data Points in Long-term Periodical Testing with Protection Relays
Tutkinto: Diplomi-insinööri
Oppiaine: Automaatio ja tietotekniikka
Työn valvoja: Teemu Mäenpää
Työn ohjaaja: Anton Pääkkönen
Valmistumisvuosi: 2022 **Sivumäärä**: 73

TIIVISTELMÄ:

Suojarele on sähköverkon osa, jonka tarkoituksena on suojata jakeluverkkoa ja reagoida poikkeustilanteissa. Suojarele voi olla sähkömekaaninen, staattinen, mikroprosessoripohjainen tai digitaalinen eli numeerinen. Numeerinen suojarele on uusin suojareletyyppi, jossa käytetään digitaalista signaaliprosessoria. Suojareleet kehittyvät nopeata vauhtia ja niiden nykyaikaistumisen myötä suojareleiden sisällä olevien erilaisten vikojen havaitseminen on nykyään entistä haastavampaa. Tämän vuoksi on kehitettävä uusia testausmenetelmiä suojareleiden sisäisten vikojen havaitsemiseksi sekä niiden toimivuuden testaamiseksi.

Suojareleiden jatkuva testaus, laadunvarmistus ja kehittäminen vähentävät suojareleissä esiintyvien vikojen mahdollisuutta. Tutkimuksen tarkoituksena on vähentää suojareleiden vikatilanteita ja keskittyä varmistamaan ja parantamaan suojareleiden laatua niiden jatkuvasti pidentyvän käyttöiän aikana suorittamalla uutta testausmenetelmää, pitkäaikaistestausta osana pitkäaikaistestausjärjestelmää. Tämä tutkimus toteutetaan yhteistyössä ABB Oy:n kanssa. ABB on merkittävä maailmanlaajuinen teknologiayritys, jolla on toimintaa yli 100 maassa ja joka on erikoistunut automaatioon ja sähköistämiseen.

Tutkimuksen yksi tavoite on suorittaa pitkäaikaista jaksottaista testausta ABB:n REX640-nimiselle suojareleelle ja automatisoida ohjelmallisesti kokonaista testausprosessia, johon sisältyy datan kerääminen, tallentaminen ja analysointi. Tutkimuksessa kerätään dataa REX640:n sisä- ja ulkopuolelta. Tutkimuksessa on selvitettävä pitkäaikaisen testauksen aikana asiaankuuluvaa dataa, jolla saadaan arvioitua suojareleen hajoamisen ennustamista. Asiaankuuluvan datan selvittämiseksi määriteltiin ensimmäinen tutkimuskysymys seuraavasti:
Mikä data on relevantti pitkäaikaisessa jaksottaisessa testauksessa suojareleiden kanssa?

Relevantin datan löytämiseksi, on selvitettävä menetelmä, jonka avulla voidaan kerätä kyseistä dataa. Näin ollen, tutkimuksen toinen tavoite on löytää menetelmä, jonka avulla arvioidaan, tallennettaan sekä analysoidaan suojareleestä kerättävää dataa. Sopivan menetelmän löytämiseksi määriteltiin toinen tutkimuskysymys seuraavasti:
Mitä metodia tullaan käyttämään relevantin datan keräämiseksi?

Tutkimuksessa kehitetyt haastattelut, kirjallisuuskatsaus, vaatimukset sekä automatisoitu testiympäristö auttoivat löytämään vastauksia tutkimuskysymyksiin. Tuloksena selvitettiin pitkäaikaiselle jaksottaiselle testaukselle relevanttia dataa sekä sopivaa metodia havaitun relevantin datan keräämiseksi. Relevantin datan sekä kehitetyn metodin avulla on mahdollisuus tunnistaa suojareleissä ilmeneviä vikoja sekä arvioida suojareleiden vikojen ennustettavuutta pitkäaikaisen jaksottaisen testauksen avulla ennen kuin jakeluverkossa tapahtuu merkittäviä häiriöitä.

AVAINSANAT: suojarele, digitaaliset releet, numeeriset releet, pitkänaikaistestaus, testausmenetelmät, testiautomaatio, jaksottainen testaus, datankeruu

UNIVERSITY OF VAASA	
School of Technology and Innovations	
Author:	Hind Aldhaher
Title of the Thesis:	Evaluating Internal and External Data Points in Long-term Periodical Testing with Protection Relays
Degree:	Master of Science in Technology
Programme:	Automation and Computer Technology
Supervisor:	Teemu Mäenpää
Instructor:	Anton Pääkkönen
Year:	2022 **Pages:** 73

ABSTRACT:

A protection relay is a part of the electrical network intended to protect the distribution network and react in case of abnormal situations. The protection relay can be electromechanical, static, microprocessor-based, or digital, also known as numerical. The numerical protection relay is the newest type of protection relays, which uses a digital signal processor. With the modernization of protection relays, various failures inside the protection relays are nowadays becoming more challenging to detect. Therefore, new test methods need to be developed to detect protection relay's failures and to test the functionality of protection relays.

Continuous testing, quality assurance, and development of protection relay's testing methods will reduce the failures may occur in protection relays. The purpose of this study is to reduce the failure situations of protection relays and to focus on ensuring and improving the quality of protection relays through their continuously expanding lifetime by performing a new test method, long-term testing, as a part of long-term test system. This study is carried out in collaboration with ABB Oy that is a major global technology company with operations in over 100 countries that specializes in automation and electrification.

The first objective of this study is to conduct long-term testing with a protection relay called REX640 and to automate the entire testing process, which includes collecting, storing, and analyzing data programmatically. In this study data will be collected from internal and external data points of REX640. The research aims to discover the relevant data in long-term periodical testing with protection relays to evaluate the prediction of protection relay failures. To discover the relevant data, the first research question is defined as follow:
What data is relevant in long-term periodical testing with protection relays?

A proper method should be discovered to collect the relevant data. The second objective of this study is to discover a method to evaluate, store, and analyze the collected protection relay data. This will allow protection relay failures to be detected. To find the proper method, the second research question is defined as follow:
Which method will be used to collect the relevant data?

The interviews, literature review, and automated test environment developed in this study allowed finding answers to the research questions. As a result, the study identified the relevant data in long-term periodical testing and a proper method to collect the identified relevant data. The relevant data and the discovered method will allow identifying the defects may occur in the protection relay and evaluating the predictability of protection relay failures with long-term periodical testing before a major disruption occurs in the distribution network.

KEYWORDS: protection relay, digital relays, numerical relays, long-term testing, testing methods, test automation, periodical testing, data collection

Contents

Figures

Tables

Equations

Abbreviations

AIM	Analog Input Module
ANSI	American National Standards Institute
API	Application Programming Interface
BIO	Binary Input Output
CSV	Comma-separated Values
COM	Communication Module
CPLD	Complex Programmable Logic Device
CPU	Central Processing Unit
DUT	Device Under Test
DSP	Digital Signal Processor
FPGA	Field-programmable Gate Array
FTP	File Transfer Protocol
FTPS	FTP Secure
GPIO	General Purpose Input/Output
GOOSE	Generic Object-oriented Substation Event
I/O	Input/Output
IEC	International Electrotechnical Commission

IEC 61850	International Standard for Substation Communication and Modeling
IED	Intelligent Electronic Device
IoT	Internet of Things
IRF	Internal Relay Fault
IP	Internet Protocol
LAN	Local Area Network
LCD	Liquid-crystal Display
LTS	Long-term Test System
MMS	Manufacturing Messaging Specification
PVC	Product Verification Center
PC	Personal Computer
PCM600	Protection and Control IED Manager
PSM	Power Supply Module
RAM	Random Access Memory
RoHS	Restriction of Hazardous Substances Directive (2011/65/EU)
SMV	Sampled Measured Values
SSH	Secure Shell
TXT	Text file
Uaux	Auxiliary voltage
USB	Universal Serial Bus
WHMI	Web Human-machine Interface
XML	Extensible Markup Language

1 Introduction

The protection relay is one of the most important components of the electrical network. The purpose of the protection relay is to secure the distribution network and monitor the performance of the network to ensure that it is working properly. Therefore, even a small failure inside the protection relay will be greatly reflected in the power grid and might cause outages in the electrical network. For instance, simple protection relay failures related to corrosion of the hardware or faulty wiring can cause the electrical network to break down. As a result, this causes serious malfunctions, substantial financial losses, or user injuries. (Gurevich, 2009, p. 333–335)

The previously mentioned situations of protection relay's internal failures must be identified, prevented, or reduced before major failures arise and lead to dangerous situations. The study proposes a new testing method to test the protection relay. This new testing method is a part of evaluating the prediction of protection relay's failures, and it is called a long-term testing, which is a part of long-term test system (LTS).

The aim of the long-term testing is to automate data acquisition by developing an application that facilitates predicting the failures of the internal protection or control functions of the protection relay, which may lead the main functionalities of the device to malfunction. With the introduction of testing with LTS, the stability, quality, and safety of the protection relay can be increased to an even higher level from today's standards. In addition to customer satisfaction and user experience, this study seeks to increase customer safety and reduces major problems such as the disruption of distribution networks.

This study is conducted in cooperation with ABB Oy. ABB is a leading global technology company that operates in more than 100 countries and is focused on electricity and automation. ABB was founded in 1988 when the Swedish company ASEA and the Swiss company Brown Boveri merged. ABB is headquartered in Zürich, Switzerland and it is listed on the Zürich, New York, and Stockholm stock exchanges. The company combines software with its portfolio of electrification, robotics, automation, and motion. (ABB, 2022a)

The study is carried out in ABB Oy’s Distribution Solutions division, which is located in Vaasa, Finland. The division designs, sells, distributes, and manufactures protection relays, as well as control, automation, and monitoring devices for the electrical distribution network (ABB, 2022b).

1.1 Background and Motivation

ABB delivers a full range of IEC 61850 conformant protection and control products, such as the Relion series and related components, as shown in Figure 1 below. The Relion product family provides the most comprehensive variety of products for power systems measurement, control, protection, and monitoring. Relion product family utilizes both International Electrotechnical Commission (IEC) and American National Standards Institute (ANSI) application. (ABB, 2019, p. 1–5)

The International Standard for Substation Communication and Modeling (IEC 61850) has been used in ABB products since 2005 (ABB, 2020a, p. 9). According to the International Electrotechnical Commission (IEC, 2020, p. 13) IEC 61850 provides interoperability between direct relays of different manufacturers. Interoperable functions can be functions that represent connections to the process or substation automation functions, for instance protection functions. Furthermore, IEC 61850 establishes a foundation for substation communication systems and networks. Also, the standard has two major versions, Edition 1, and Edition 2. The Edition 2 describes a new functionality to the first edition. Most of ABB’s products are modeled after IEC 61850 standard and support both IEC 61850 Editions 1 and 2 (ABB, 2020a, p. 9–18).

As shown in Figure 1, ABB's Relion product family includes many different protection relays, such as Relion 605, Relion 611, Relion 615, and Relion 630 series relays. In addition to RIO600 Remote Input/Output (I / O) Unit, and REX640 protection relays. Protection relays can be divided according to their structure into electromechanical, static electronic, microprocessor-based, or digital, also known as numerical protection relay. Relays can also be divided according to their main function used in the protection of the distribution network. In this case, protection relays can be overcurrent, earth fault, or distance relays (Haila, 2016, p. 32). The protection relays shown in Figure 1 are mostly digital protection relays with a self-monitoring system (ABB, 2022d).

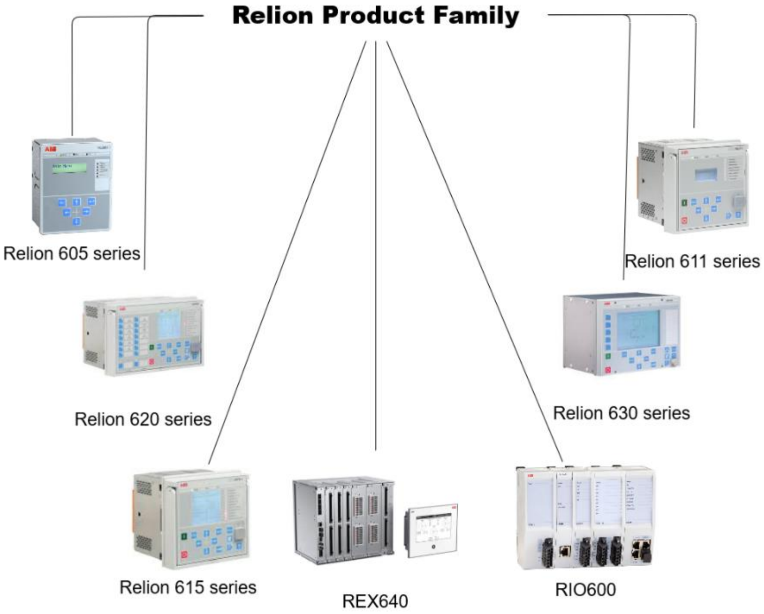


Figure 1. ABB Relion family products (ABB, 2019; ABB, 2020a).

A malfunction in the protection relays, for example, due to a design fault, a communication fault, or a setting error, can have significant consequences in the electrical network. The fault recorder and self-monitoring system of the protection relays can alert about faults, internal faults, and disable operation, but the configuration of the protection relay requires a proper testing system to determine its faultless operation (Welton & Knapek, 2017, p. 2). The purpose of long-term testing is to proactively prevent all possible failures of the protection relays. Long-term testing is also a quality assurance tool of interest to ABB customers. This study focuses on the development of LTS by implementing a long-term testing on REX640 protection relay periodically by analyzing the internal and external data points of the protection relay. The LTS will be used to proactively indicate the failure of protection relays. This will increase safety, protection, and quality assurance for users and to discover preventable problems and breakdowns.

Long-term periodical testing is a new area in ABB and relatively unexplored with protection relays. Long-term periodical testing includes data collection and analysis of the protection relay. By analyzing the collected data and discovering relevant data, it is possible to automate the data collection by developing an application programmatically that enables the prediction of protection relay internal failure situations.

1.2 Research Objectives

The study focuses on investigating the relationships, correlations, and importance of long-term test system data points and the device under test (DUT). The aim is to concentrate on data points that reflect proportionality to the actual events, and the abnormal conditions of the DUT. As a result, it is possible to transform the insights into development activities by improving the vulnerabilities of the device under test and building predictive analytical models for similarly controlled testing environments. Based on this, the key question of the study is:

RQ1: What data is relevant in long-term periodical testing with protection relays?

To obtain relevant data points, the precise methods in which those data points will be collected must be identified. In this case, the second primary question of the research is:

RQ2: Which method will be used to collect the relevant data?

The first step is to integrate the device under test to the long-term test system. The second step is to explore and clarify the data acquisition. The third step is to automate the data retrieval programmatically, so that the data can be retrieved and explored beyond this project. The final step is to generate insights from the collected data that can be used to predict the breakdown of the protection relay. In consequence, the breakdown situation of the protection relay could possibly be predicted after automated data collection for instance, as the temperature rises (A. Pääkkönen, personal conversation, 17.06.2022).

1.3 Research Structure

The research first introduces primary theoretical terms of the work, such as protection relays, long-term testing, and test automation. The second chapter presents the basics of protection relays and test automation. Chapter three introduces the literature review by examining previous studies and related situations to this study. Chapter four provides a better understanding of the research method and research approach. The study used the knowledge and skills of ABB professionals through a research interview as presented in chapter five. Furthermore, chapter five implements the work project and documents the different stages of the project, including requirements and test execution. The sixth chapter analyzes the results of the fifth chapter. The seventh and final chapter presents the conclusions and future considerations (Figure 2).

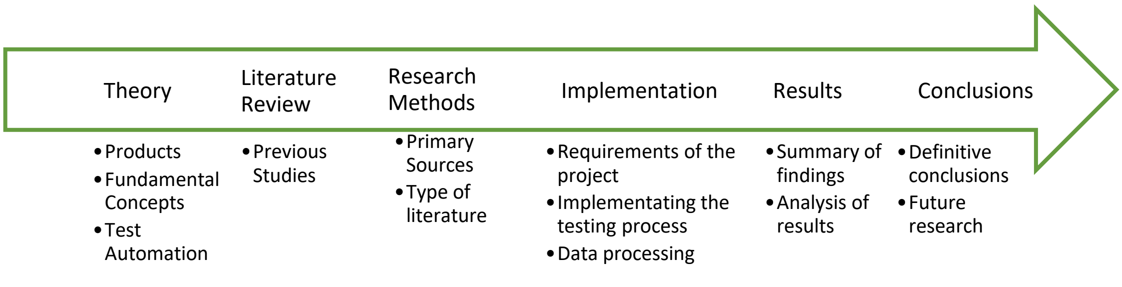


Figure 2. Research Structure.

2 Protection Relay in Long-term Test System

The primary purpose of this chapter is to cover the principal concepts and definitions used in the study. The chapter covers the basics of protection relays and provides an overview of their history. This chapter aims to explore the features of REX640 protection relay, including background, material, and operational capability. Furthermore, this chapter discusses the long-term test system and concludes with a review on test automation.

2.1 Fundamental Concepts and Definitions

The term protection relay can also be protective relay. In this work, it was decided to use the term protection relay to describe an electronic device that protects the distribution network. When an abnormal situation in an electrical circuit is detected by the protection relay, it closes its connectors. In turn, the connectors shut and complete the circuit breaker's trip coil circuit, causing the circuit breaker to break and isolate the malfunctioning portion of the electrical circuit from the rest of the faultless electrical circuit (Warrington, 2012, p. 18, 151, 364).

Long-term testing is defined in the literature as a term with different purposes. However, in this study, long-term testing is a part of ABB’s new testing system called long-term test system. Long-term testing is utilized to test the protection relay for a longer interval. This study focuses on developing the LTS by automating the long-term testing with protection relays.

Periodic testing, or periodical testing is testing the DUT between specific intervals. Test intervals can range from seconds to years, depending on the DUT and the overall testing methodology (Sravanthi et al., 2016, p. 134). However, in this study the term periodical testing will be used to describe a testing method that focuses on verification of the protection relay control system, verification of the IEC 61850 standard’s generic object-oriented substation event (GOOSE) communication for interlockings, performance validation, verification of conformance to configuration requirements, settings, and verification of those components that cannot be supervised and are considered as hidden faults in protection relays (Patki et al., 2018, p. 5–10).

In this research, a test case will be automated. The term test automation refers to the creation of a program out of a manual test case using general-purpose programming or scripting language. Test automation can also refer to the execution of a test by formulating keywords or calls to executable sub-programs as a series of relevant arguments. The driver program can automatically interpret this sequence. (Thummalapenta et al., 2012, p. 881)

2.2 Basics and Features of Protection Relays

Protection relay is a device that continuously measures the current of an electrical network utilizing current and voltage transformers or applicable sensors. The relay detects faults and other abnormal situations in the network and warns the user of faults by sending trip commands to the circuit breakers, allowing faulty parts of the network to be isolated from the rest of the electrical system. The protection relay can also send alarms and events to the network supervisor and control circuits subsequently. (Haila, 2016, p. 8, 28, 36)

The purpose of the protection relay is to monitor electrical systems through current, insulation, temperature, and voltage inputs and to detect and isolate electrical faults. Thus, a protection relay isolates major electrical faults before disasters can occur. The protection relay also has the function of triggering a trip signal in the event of an abnormal system condition or fault being detected by the operators (Werstiuk, 2012, p. 65–76).

The initial relay functions developed by ABB were included in the circuit breaker design, which served as an over-current trip. The first electromechanical protection relay was invented in 1905 and it is called the time-overcurrent relay. It was constructed with a bellow made of impregnated balloon fabric, coupled with the air valve, and dampened the movement of the solenoid to produce the necessary delay. The first stand-alone electromechanical protection relays were utilized to power industry and illuminate towns. (ABB, 2004, p. 3)

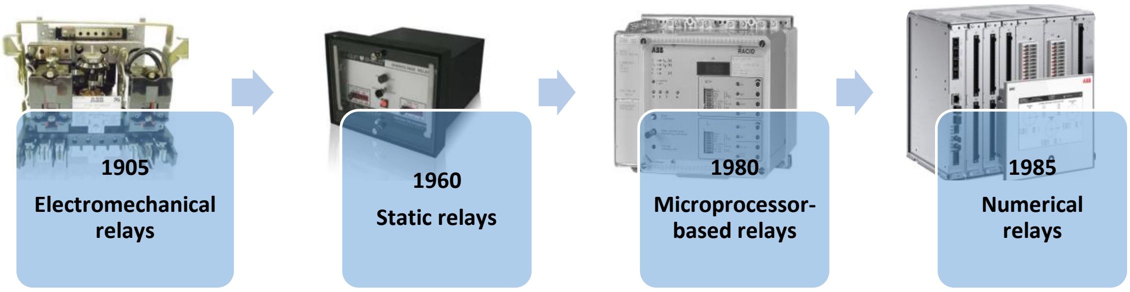


Figure 3. Different types of protection relays (ABB, 2004; ABB, 2019).

Protection relay can be divided into electromechanical, static, microprocessor-based, digital, or digital, also known as numerical (Figure 3). Electromechanical relays are simpler forms of protection relays and the basis of all relays. Because of that, electromechanical relay can have complex mechanical drive systems. On the other hand, static protection relays consist of non-moving parts, such as a capacitor, an integrated circuit, a transistor, and microprocessors (Rahebi & Al-Shalah, 2020, p. 550). Digital protection relays are more flexible than electromechanical and static protection relays, since digital protection relays utilize their software for relay protection and measurement (Werstiuk, 2012, p. 65–76). Digital protection relays also utilize a microprocessor which differs from the numerical protection relays. Whereas numerical protection relays utilize a digital signal processor (DSP) (Goh et al., 2010, p. 45). However, digital, and numerical protection relays are considered synonymous despite the differences between them (Lackovic, n.d, p. 24–25).

Finland’s transmission system operator Fingrid Oy utilizes 334 different relay models (Haila, 2016, p. 32). Most of the protection relays deployed on the Finnish electrical network have been digital, as they include a self-monitoring system, which reduces protection maintenance and extends test intervals compared to electromechanical relays that need to be tested manually (Lindblad, 2008, p. 465). Furthermore, digital protection relays have an internal fault recorder that can record events not detected by the protection functions or self-monitoring system. The fault recorder operates at 1 kHz in most of the relays. The exception is the 1.6 kHz recorder which is utilized in ABB’s Relion family protection relays (Haila, 2016, p. 35).

2.2.1 Relion Product Family

The majority of ABB protection relays come with software components called connectivity packages. The connectivity packages consist of relay-specific data, software collections, a comprehensive relay data model, and a single-line diagram template. The event and parameter lists comprise the data model of protection relays. ABB protection relays can be simply configured with their easy-to-use software called Protection and Control IED Manager (PCM600) utilizing the connectivity packages. (ABB, 2020c, p. 21, 27, 109, 110)

PCM600 communicates with intelligent electronic devices (IED) through IEC 61850 based communication protocols or file transfer protocol / file transfer protocol secure (FTP / FTPS). PCM600 is IEC 61850 certified, making it easier to manage IED protection functions, controls, and communications (ABB, 2019, p. 1–5). At the same time, PCM600 enables data exchange with other IEC 61850-compliant devices. PCM600 can also be used to configure and design protection relay applications using interface packages as they contain code and data (ABB, 2020a, p. 13).

2.2.2 REX640 Protection Relay

REX640 is a high-performance exclusive protection and control relay developed for advanced power distribution and generation applications. REX640 is a member of the

Relion product family (Figure 1), and its first release took place in 2018. The modular architecture of protection relay's hardware and software elements allows covering of any complete protection application requirements that may occur during the relay's and substation's entire life cycle (ABB, 2020b, p. 48–67).

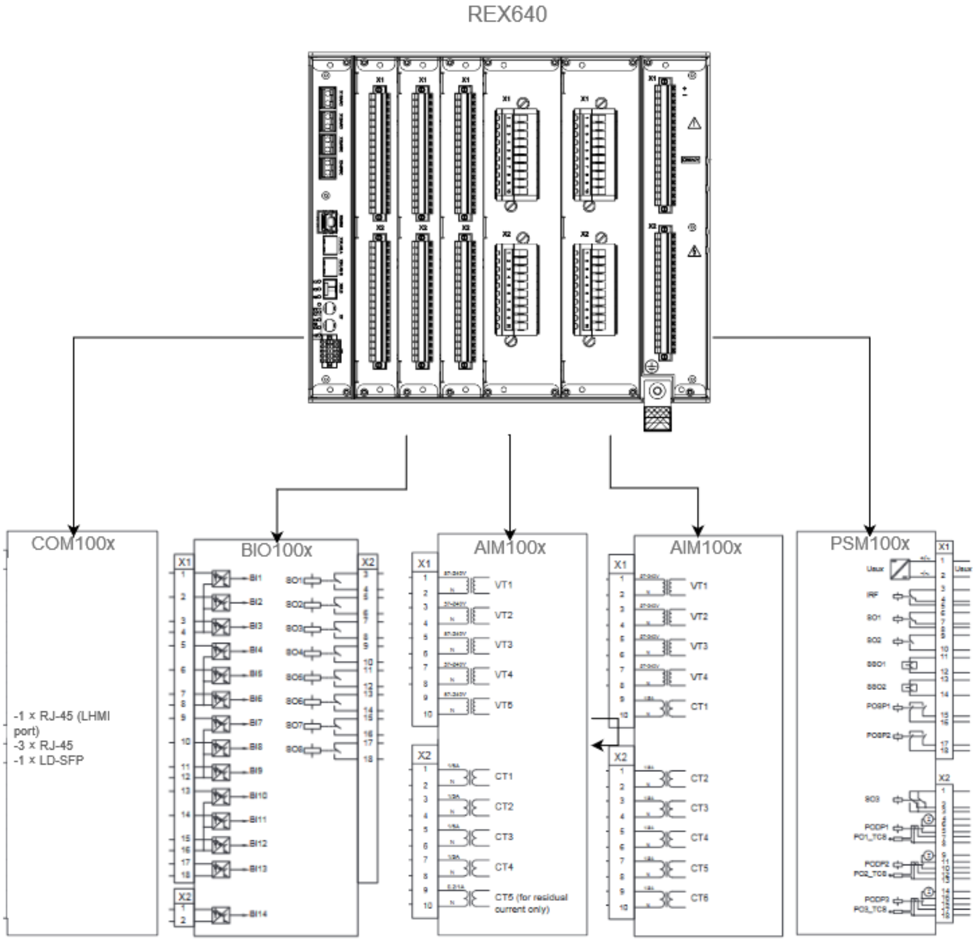


Figure 4. REX640 modules (ABB, 2020c, p. 5–6, 116–118).

REX640 hardware consists of obligatory and optional components depending on the composition. The protection relay in Figure 4 consists of five slots with obligatory modules, where each module has a function: COM100x is a communication module, BIO100x module is for binary input and output, the analog input module (AIM100x) provides voltage and current inputs, and PSM100x is relay power supply module that allows various current and voltage measurements (ABB, 2020c, p. 4, 14, 26, 32). These modules will be utilized in the implementation phase of this study.

2.2.3 Protection Relay Material

REX640 protection relay has been designed with sustainability aspects in mind, including long lifetime, reliability, disposal of the device, and environmental friendliness in manufacturing process. The materials of the protection relay have been selected by Restriction of Hazardous Substances Directive of the European Parliament (EU RoHS Directive, 2011/65/EU). The directive restricts the use of dangerous substances in the materials of protection relay. (ABB, 2020d, p. 19)

The protection relay has been extensively tested at the design and manufacturing stages to ensure long service life and reliability. Figure 5 shows the materials used in the different parts of REX640 protection relay. Aluminum is used in the casted enclosure and metallic plates of the protection relay. In addition, steel is used in screws, bushes, and plastic such as polycarbonate and liquid crystal polymer. The liquid crystal polymer is used in different plastic parts of the protection relay. (ABB, 2020d, p. 19–21)

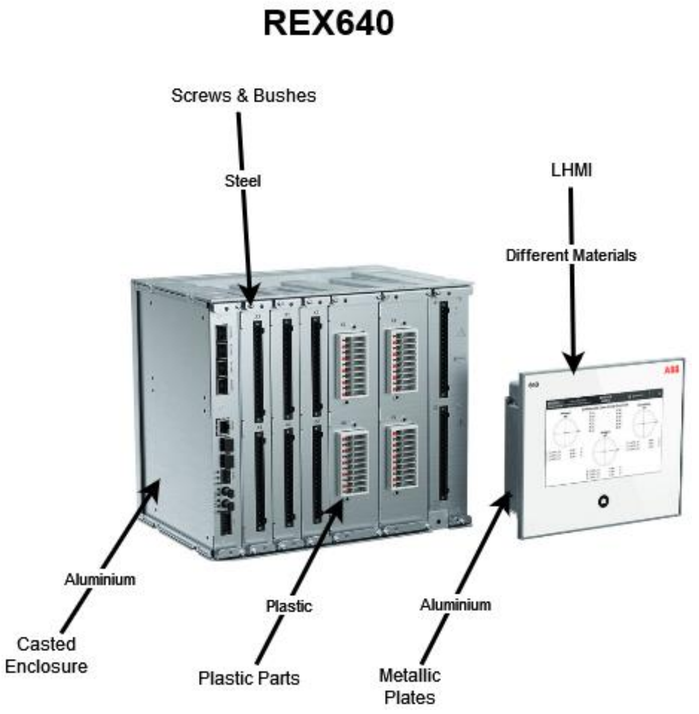


Figure 5. REX640 materials (ABB, 2020a; ABB, 2020d).

2.2.4 Operational Capability of REX640

REX640 relay has a built-in self-monitoring system that constantly monitors the status of the relay hardware and the operation of the relay software. When the built-in self-monitoring system detects a fault or malfunction, it notifies the user. The protection relay records the faults in a non-volatile memory which saves stored data even when power is turned off. The relay can keep track of the last 128 fault incidents. The records can be used to examine the occurrences of the power system. Each record contains information such as current, voltage, and angle values, as well as a time stamp. The start or trip signal of a protection block, or both, can initiate fault recordings. (ABB, 2020c, p. 23–27)

Fingrid uses a protection relay fault recorder at almost all 400 kV stations, as well as at the main 220 kV and 110 kV stations (Haila, 2016, p. 38). However, in 2021, Fingrid fixed, tested, and laid out about 1000 protection relays (Fingrid, 2022a). Possible faults include a fault-to-fault connection, where a differential relay must trip a delay, or a fault between a current transformer and a circuit breaker, where a communication signal is sent from the tripping of a busbar protector to a distance relay in the substation. In the worst case, faults in the protection relays can lead to the tripping of all feeder lines (Fingrid, 2022b, p. 5).

Since REX640 is a numerical protection relay, some of the internal components of the protection relay, such as analog-to-digital converters, power supplies, input cards, and output cards might fail without warning or detection, resulting in lack of device or system protection (ABB, 2022c; ABB, 2020b, p. 61, 184–191, 1182–1198). Furthermore, numerical, and digital protection relays are controlled by software, which may be vulnerable to unexpected system breakdowns (Werstiuk, 2012, p. 65).

To solve the problems mentioned above, it is necessary to evaluate the prediction of the internal defects that may occur in the protection relay. This will be achieved by collecting external and internal data points of REX640. Thereafter, analyzing the collected data and implementing a long-term periodical test allows the prediction of protection relay’s internal failures. This will be discussed in the next chapter.

2.3 Long-term Test System

Long-term test system is implemented by ABB, and it is intended for testing devices on a long-term basis. The duration of long-term testing varies from days to years. Long-term testing can be carried out, for instance, in the form of long-term idling, which increases the uptime of the protection relay. Idling is usually continued infinitely by continuously running a set of tests and validating that the device is working correctly. Long-term testing can also be performed as a long test sequence with or without the inclusion of loops. For instance, without loops the test case would not generate iterations of repeated steps. (A. Pääkkönen, personal conversation, 20.06.2022)

Long-term testing can also be executed in a periodical sequence. Periodical testing ensures that the protection system of the protection relay performs its function by validating that the protection relay is configured correctly, measuring accurately, and utilizing its built-in self-supervision system correctly. Periodical testing consists of test cycles, often known as iterations (Wester et al., 2013, p. 1, 5–12). A test cycle begins with the protection relay being restarted and iterations being repeated at regular intervals (Werstiuk, 2012, p. 25–42). This study handles long-term periodical testing, where the protection relay will be tested at periodical intervals over a longer term.

2.4 Test Automation

The purpose of the test setup of this study is to automate the testing process and data collection. The use of testing tools to apply automatic testing of a DUT during software development is known as test automation. Test automation consists of various testing functions, such as test case design, test scripting, test execution, test evaluation, and test result reporting (Wang et al., 2020, p. 315).

The test setup of this study aims to automate the testing process using Robot Framework and Python to facilitate the examination of relevant data points and the launch of long-term test system. Robot Framework is a Python-based open-source software licensed under Apache 2.0. Robot Framework is a keyword-based extensible automation framework that provides easy-to-read results and logs information in HTML format to help review the test execution results (Robot Framework, 2022). Robot Framework will be used in this study to automate the collection and storing of protection relay data.

The utilization of Robot Framework and Python technologies in the test automation of long-term testing process and data collection is based on the use of related technologies in the automation of the continuous integration phase of the protection relay. The integration of reusable technologies in the automation of the long-term testing process is an advantage in the future of the corresponding LTS testing area. Furthermore, Robot

Framework simplifies the automation of an embedded system such as a protection relay by enabling automated interaction for multiple interfaces of the system simultaneously as an abstraction-layer framework solution (A. Pääkkönen, personal conversation, 14.10.2022). As such, test case automation using keyword-driven techniques allows data, scripts, and services to be independent of each other (Na & Huaichang, 2015, p. 428).

3 Prior Research on Data Collection Methods and Previous Studies

The purpose of this chapter is to discuss various studies through a literature review. Literature review is a research method that aims to develop existing theory and build on it to construct a new theory. Literature review builds an overall vision of a particular issue and seeks to identify related problems and solutions (Salminen, 2011, p. 3–5). This chapter focuses on the second research question "which method will be used in collecting data?". The chapter also discusses previous studies related to the second research question, subject, and keywords of this study to get acquainted with protection relays and various ways of extracting data points from protection relays.

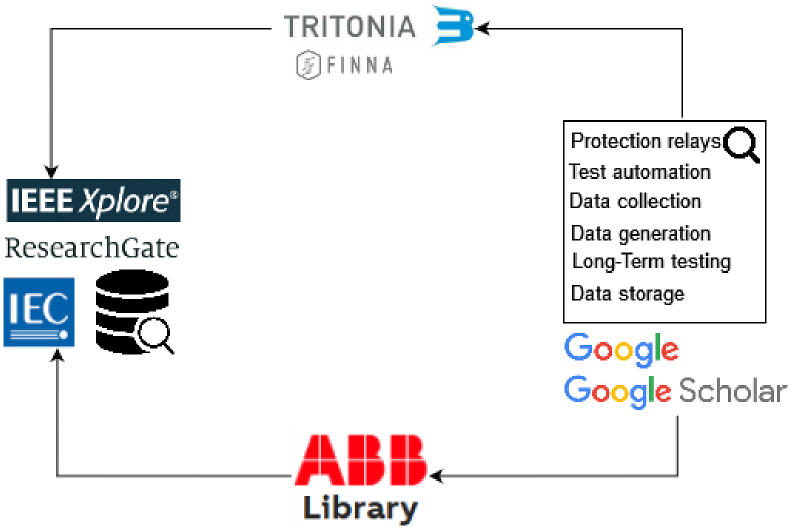


Figure 6. Framework of the literature search.

The keywords used to search the relevant previous studies are “protection relays”, “test automation”, “data collection”, “data generation”, “long-term testing”, “data storage”, “numerical relays”, “digital relays”, “periodical testing”, and “hidden failures”. Keywords were searched from different libraries such as the University of Vaasa Library (Tritonia Finna), ACM Digital Library, and ABB Library. Furthermore, different databases were used in searching the keywords above. These databases are Google Scholar, IEEE Xplore, ResearchGate, ScienceDirect, SpringerLink, and IEC. The searches were performed using a range of search standards including 1) articles published between 2008 and 2022, 2) articles related to numerical or digital protection relay testing, and 3) articles related to protection relay's data collection or data generation (see Figure 6).

Google Scholar search found around 3200 articles related to the keywords and search standards, while ScienceDirect found 47 articles. IEEE Xplore matched 83 articles, SpringerLink 51 articles, and ACM Digital Library identified 10 articles. ResearchGate did not show the amount of found papers, but relevant articles were found by sorting publications with relevance. When articles were searched using restricted search criteria, all databases resulted in a total of 20 publications related to the mentioned keywords. The publications were reviewed and a total of nine articles were selected based on their relevance to the search standards and the subject of this study. Table 1 contains these articles and the publications searched from ABB Library are related to REX640 protection relay discussed in this study.

Table 1. Selected research references.

#	References	Database
1	ABB. (2020b). REX640 Technical Manual. Retrieved 9.6.2022 from https://search.abb.com/library/Download.aspx?DocumentID=1MRS759142&LanguageCode=en&DocumentPartId=&Action=Launch	ABB
2	ABB. (2020d). Operation Manual. Retrieved 16.6.2022 from https://search.abb.com/library/Download.aspx?DocumentID=1MRS759118&LanguageCode=en&DocumentPartId=&Action=Launch	ABB
3	Abniki, H., Alipour, H., Keramat, M. M., Ansari, S., Abyaneh, H. A. & Razavi, F. (2022). Practical Test of Synchronization Relay. Retrieved 26.7.2022 from https://ieeexplore.ieee.org/document/9763133	IEEE Explore

4	Gajic, Z. (2009). Easy Method for Testing Transformer Differential Relays. Retrieved 19.7.2022 from https://www.researchgate.net/publication/289184445_Easy_Method_for_Testing_Transformer_Differential_Relays	Researchgate
5	Lindblad, P. (2008). Prolonging Periodical Testing Intervals of Numerical Protection Relays. Retrieved 19.7.2022 from https://ieeexplore.ieee.org/document/4497030	IEEE Explore
6	Ma, K. Huang, S. Chen, J. Yang, Y. (2010). Discussion of the Testing Technology for Protection Device Used in Digital Substation. Retrieved 22.7.2022 from https://ieeexplore.ieee.org/document/5736034	IEEE Explore
7	Ouadi, A., Chafai, M., Bentarzi, H., Zitouni, A. (2011). Microcontroller based protective relay testing system. Retrieved 29.7.2022 from https://www.researchgate.net/publication/262257924_Microcontroller_based_protective_relay_testing_system	Researchgate
8	Patki, S., Kamin, D. & Suyash, K. (2018). Numerical Relay Testing and Validation in Relay Life Cycle. Retrieved 27.7.2022 from https://www.researchgate.net/publication/329338067_NUMERICAL_RELAY_TESTING_AND_VALIDATION_IN_RELAY_LIFE_CYCLE	Researchgate
9	Turner, S. (2012). Using COMTRADE Records to Test Protective Relays. Retrieved 25.7.2022 from https://ieeexplore.ieee.org/abstract/document/6201248	IEEE Explore
10	Xia, Y., Zou, Z. Li, W., Zhang, X., Cui, S. & Liu, H. (2021). Research on Remote Operation Automatic Test of Relay Protection Based on MMS Communication Protocol Engine Technology. Retrieved 28.9.2022 from https://ieeexplore.ieee.org/document/9633833	IEEE Explore

11	Zhao, L., Li, Z., Ni, M. & Cheng, Y. (2019). Review and prospect of hidden failure: protection system and security and stability control system. Retrieved 13.12.2022 from https://link.springer.com/content/pdf/10.1007/s40565-015-0128-9.pdf?pdf=button%20sticky	SpringerLink

Protection relays have three different failure modes, the first mode is the lack of relay functionality, the second is relay functionality with a delay, and the third is incorrect functionality of the protection relays. The fault conditions can be caused by a mechanical fault in the cables going in or out of the relay, protection circuits of open connectors, internal faults in the relay, and setting or configuration errors (Lindblad, 2008, p. 468). Protection relays can also contain hidden faults that cannot be detected until their impacts are exposed to abnormal conditions in the electrical network, such as cascading faults or power blackouts (Zhao et al., 2019, p. 1735–1736).

Gajic (2009, p. 7–9) reported that it is possible to test a numerical, three-phase power transformer differential protection relay with a set of three currents or indicators. This expresses the set as a linear combination of symmetrical components, allowing the detection of an external or internal fault in the protection relay by a linear combination of positive, negative, and zero sequence current component sets. The testing can be performed by using an application called Wizard, which can be used to inject currents into the device under test and to expect the performance of the protection relay.

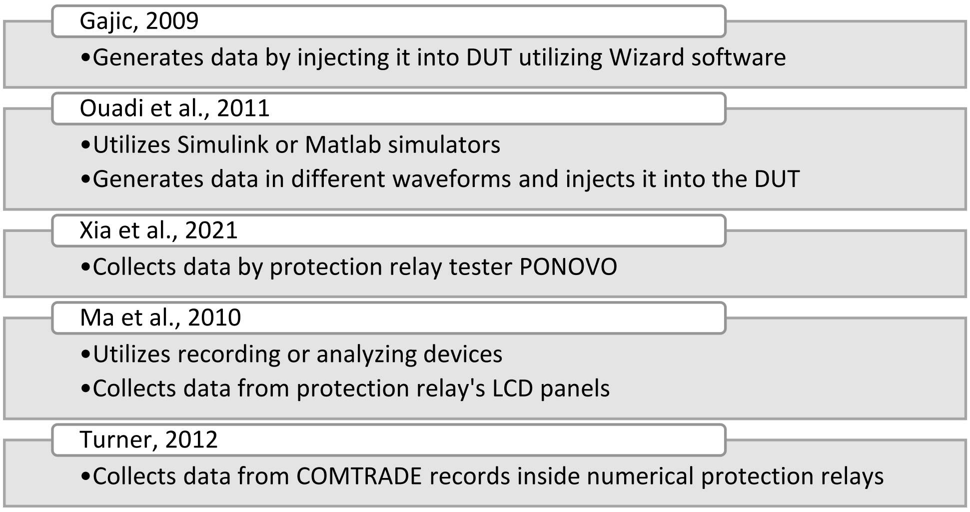
Ouadi et al. (2011, p. 88–90) tested the performance of a digital protection relay by developing a software fault simulator. The fault simulator is used to generate relevant data such as voltage and current representing signals of power system’s different conditions. The data can be produced by SIMULINK or MATLAB simulators utilizing a PC. In this case, relevant data signals are generated in different waveforms and injected into the device under test.

In addition to the fault simulator, Xia et al. (2021, p. 306) developed a remote automated test system to test protection relays. The test system used a protection relay tester called PONOVO to collect voltage and current data. The protection relay tester was connected to a personal computer (PC) for data analysis. The computer and protection relay tester were linked to a local area network (LAN) through a switch. Further, the tested protection relay was also connected to the switch by adopting the network port IEC 61850.

Ma et al. (2010, p. 3–5) carried out a performance test of digital protection relays. The test provided relevant data by simulating different electrical state failures that were done by changing the duplicating sampled measured values (SMV) message outputs. The SMV messages can be recorded by utilizing message recording and analyzing devices. The test was completed by having the test equipment software to analyze the operation messages and trip reports from the protection relays liquid-crystal display (LCD) panels. Furthermore, by utilizing a digital recorder, it can store waveform files in COMTRADE format to take advantage of the test instrument software's replay function and relevant data to the device under test. Turner (2012, p. 1–3) noted that it is challenging to create critical faults or failures in numerical protection relays with existing test equipment software. To address this, the COMTRADE records stored in numerical protection relays captures the digital fault records that occur during real system events and can be easily applied to test equipment software to aid in the analysis of critical data.

Defects can occur in protection relays for various reasons. These defects can be caused for instance, by protection circuits of open connectors, mechanical faults in the cables connected to the protection relay, setting, or configuration errors (Lindblad, 2008, p. 468). Defects can also be caused by protection relay's internal faults, component faults or hidden faults which are not easily detectable before they cause a major problem in the distribution network (Zhao et al., 2019, p. 1735–1736). To prevent different type of protection relay failures, a wide range of testing methods have been developed. Gajic (2009, p. 7–9) tested the performance of protection relay by injecting current to the relay utilizing an application called Wizard. Ouadi et al. (2011, p. 88–90) developed a software fault simulator that generates data in different waveforms to be analyzed and injected into the protection relay. Xia et al. (2021, p. 306) utilized a test device called PONOVO which enables data collection from protection relay and data analysis from PC. While Ma et al. (2010. p. 3–5) simulated electrical failure situations by modifying SMV messages and monitored the behavior of the protection relay utilizing protection relays software and LCD panel. According to Turner (2012, p. 1–3) COMTRADE event records stored in numerical protection relays captures fault records that occur during protection relays actual system events and can be applied to test device software to analyze critical data.

In short, there are numerous ways to test protection relays. Data generation is one method of obtaining testing data, while data collection is a part of the whole testing process. Data generation is for example injecting data into the device under test, which can be done by utilizing fault simulator as Ouadi et al. (2011) performed or as Gajic (2009) conducted with Wizard application. Data collection is, for instance, collecting data directly from the protection relay, such as Ma et al. (2010) and Turner (2012) performed their implementation, or utilizing protection relay tester which records the measured test data as Xia et al. (2021) did (see Figure 7).



Gajic, 2009
- Generates data by injecting it into DUT utilizing Wizard software

Ouadi et al., 2011
- Utilizes Simulink or Matlab simulators
- Generates data in different waveforms and injects it into the DUT

Xia et al., 2021
- Collects data by protection relay tester PONOVO

Ma et al., 2010
- Utilizes recording or analyzing devices
- Collects data from protection relay's LCD panels

Turner, 2012
- Collects data from COMTRADE records inside numerical protection relays

Figure 7. Data collection and data generating methods.

The data generating and collecting methods mentioned above are a few of many possible methods that could be utilized to test different protection relay types. These methods may be applicable in some way to the implementation phase of this work, although not all methods may be able to detect the protection relay's hidden faults. Protection relay's hidden faults are difficult to detect under normal conditions, as they may be triggered when the protection relay is in an abnormal pressure condition, such as a harsh operating environment or large voltage drop (Zhao et al., 2019, p. 1735).

This study focuses on the development of the long-term test system, which is a new area at ABB and generally unexplored within protection relays. LTS is focused on the condition of the protection relay over its entire lifetime. Advanced protection relays such as numerical or digital protection relays require verification and validation from the development phase to the entire lifetime with various perspectives (Patki et al., 2018, p. 23). Numerical protection relays require various tests during development, deployment, maintenance and configuration, and other operations, as there are hidden faults in protection relays that are found to lead to multiple outages. To prevent this, protection relays need to be tested periodically (Abniki et al, 2022, p. 1).

Therefore, long-term periodical testing is carried out in LTS. Long-term periodical testing is intended to evaluate the prediction of protection relay's failure situations. In long-term periodical testing, a pressure condition will be created on the protection relay at periodical testing intervals. The creation of pressure conditions may allow triggering protection relays hidden faults (Zhao et al., 2019, p. 1735). The data will be collected from external and internal data points of REX640 that is a numerical protection relay.

REX640 allows connecting to different devices by its binary outputs. The binary outputs can be used for connecting the protection relay to a notification device for detection, signaling, recording, tripping, and performing local or remote-control functions of the circuit breaker or disconnector (ABB, 2020b, p. 176). Despite REX640 protection relays including a built-in self-supervision system, which allows the protection relay to report internal faults and warnings, there is still a need to check for hidden faults that may appear in protection relays. REX640 allows collecting data from its internal COMTRADE format files (ABB, 2020d, p. 84, 90–93). However, this study automates data collection and storing programmatically to minimize the time spent on the testing phase and to facilitate the data storing, which makes it difficult to automate the data collection from COMTRADE file format data in long-term periodical testing.

Based on the previous studies (Abniki et al., 2022; Gajic, 2009; Lindblad, 2008; Ma et al., 2010; Ouadi et al., 2011; Patki et al., 2018; Turner, 2012; Xia et al., 2021; Zhao et al., 2019) discussed in this chapter and the results of the research gap presented earlier, long-term periodical testing is relatively unexplored testing method with protection relays. This chapter also demonstrated that protection relays need to be tested periodically for a longer term of period to detect the hidden faults that may occur inside the protection relay. It was found that changing the environmental conditions enables the detection of hidden faults that may occur in protection relays and long-term periodical testing involves testing the protection relay under different environmental conditions (Zhao et al., 2019). Changing the environmental conditions causes stress to the protection relay and aids in the identification the relevant data. The relevant data should be collected with the help of sensors, such as voltage, current, and temperature sensors, data loggers, and various protocols, such as GOOSE, SMV, or Manufacturing Messaging Specification (MMS). Data collection, storing, and analysis methods should be suitable for the test setup to be carried in this study, long-term periodical testing as well as research requirements of this study that will be further discussed in chapter 5.

4 Research Method and Approach

A research method is a practical technique or tool for collecting data, analyzing it, and carrying out research. The nature of the research problem indicates how the data will be collected and analyzed. Therefore, it is necessary to understand the research process and the application of various research methods (Walliman, 2011, p. 7–10).

Research sources utilized in this study have been retrieved from Google Scholar, ABB Library, and University of Vaasa Library, Tritonia Finna portal. Tritonia Finna portal provides access to various databases, such as IEEE Xplore, SpringerLink, ScienceDirect, ACM Digital Library, and ResearchGate. These databases were used to search for various articles and papers as sources of this study. ABB provided access to the IEC database, which offered further information about the standards used in this study. In addition, the use of Google provided access to Google Scholar articles and various books from Google Books.

Besides the previous ones, the primary references of this study are the interviews taken with ABB experts (Appendix 1). The references and the interviews of this study guarantee the validity of the research construct and theoretical input. In addition, the practical application of this study will be implemented based on the theoretical input, knowledge, and insights of ABB experts.

In addition to the research method, this research is based on the constructive research approach, which develops new innovative structures to solve real-life problems while advancing the theory of its field of application. The constructive approach is to produce an innovative construction, which could be a plan, information system model, organization, diagram, or other construction. Constructive research aims not only to solve a real-life problem but also to implement the construct with a new structure that incorporates both a high level of practical application and theoretical input (Lukka, 2003, p. 84–93).

Based on the previous description and the data reviewed in this study, the constructive research approach is suitable for this study. The construct of this study is the previously mentioned ABB’s new testing method long-term testing. This innovative new testing method prevents or reduces the occurrence of real-life problems such as user injuries, serious accidents, and economic losses by ensuring the safer protection of distribution networks (Gurevich, 2009, p. 333).

The constructive research approach has a seven-step process (Figure 8), which is applied to this study. The first stage of the research approach is finding the relevant problem that fits in the first chapter of this study, where the research problem and research questions RQ1 and RQ2 are addressed.

The second step is the identification of the possibilities for research collaboration that also takes place in the first chapter of this study, where the research collaboration with ABB is discussed. The third step is to obtain a general and comprehensive understanding of the subject, which is suitable for chapters 2 and 3 of this research. Chapter 2 discusses ABB's products, design, and fundamental concepts. In turn, chapter three covers the literature review, which brings an in-depth knowledge of the subject to this study.

The fourth stage is the solution construction step that is suitable for the implementation phase of this research, chapter 5. Chapter 5 also carries out the solution construction process discussed in the previous chapters. The fifth stage is the implementation and testing of the solution, which is also appropriate for chapter 5 of this paper, as the solution process will be implemented, including its preliminary validation in that chapter.

The sixth step is to examine the applicability of the solution, which is the sixth chapter of this study, where the results of the implementation phase that is, the solution and its application to the testing process, will be examined. The seventh and final step in the constructive research approach is the analysis and identification of the theoretical contribution, which is also suitable for chapter 6 and 7 of this study. Chapter 6 analyses the answers to the questions RQ1 and RQ2 presented in the first chapter and chapter 7 discusses the future possibilities of the research.

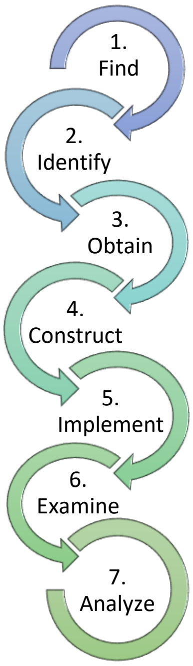


Figure 8. Constructive research process overview (Kasanen et al., 1993; Lukka, 2001).

5 REX640 Protection Relay in the Practical Test Execution

The purpose of this chapter is to implement previously discussed long-term periodical testing on REX640 protection relay. The chapter considers the overall testing process and the execution of internal and external data points of the protection relay in detail by utilizing the expertise of ABB colleagues through a research questionnaire. Furthermore, the chapter analyzes the implementation requirements as well as applies the theoretical framework to the execution phase. Finally, the method of data collection will be discussed, and relevant data will be analyzed. This implementation phase will be carried out in the Distribution Solution's Product Verification Center (PVC) laboratory located in Vaasa.

5.1 Test Formation

The test formation process aims to find out first a suitable method for collecting relevant data as part of the implementation phase of this research. After this, a failure situation must be developed for the protection relay. In this case, long-term periodical testing can be applied on DUT by analyzing the collected data at certain time cycles. In consequence, it is possible to manually perceive the failure situation and automate the testing using Robot Framework and Python. Finally, the automated script will be implemented to the device under test which is REX640 protection relay in this study (see Figure 9).

Because LTS is a new setup at ABB, the documentation of the system is still under development and therefore the implementation phase of the testing process is challenging. Thus, similar LTS tests have not been previously developed for REX640 protection relay. Based on this and the lack of supporting research material, the study benefited from the expertise of ABB's professionals, Frej Suomi and Petteri Liikkanen by conducting interviews. This chapter will describe the steps of the testing process and utilize the answers of the interviewees (Appendix 1).

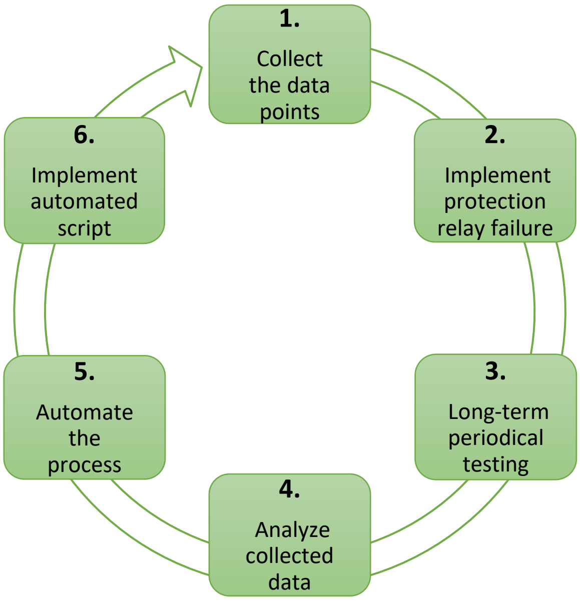


Figure 9. Test formation process.

The first step of the testing process is to collect the relevant data. According to Suomi and Liikkanen, it is possible to measure the temperature of different data points by connecting thermocouples to the internal and external data points of the protection relay. The thermocouples make it possible to read temperature measurements directly from the data logger by utilizing a temperature logger such as PicoLog. PicoLog is used for tracking temperatures over longer periods. It is also possible to read temperature measurements from the protection relay, for example by placing eight temperature sensors, such as the PT100 sensor in specific physical spots inside the protection relay.

Liikkanen states that protection relay's input voltage and current can be obtained by measuring protection relay's auxiliary voltage (Uaux), which is best suited for long-term periodical testing. Suomi claimed that collecting protection relay's input voltage and current requires reading over MMS in connection with the software, such as PCM600. Also, this enables accessing data surfaces that can be collected from REX640 with PCM600.

The purpose is to generate failure situations in the protection relay. Suomi declared that it is possible by utilizing packet storms, corrupted time synch messages, wrongly configured switches, GOOSE messages, or SMV. In this case, it is also possible to analyze the data points that would be available directly from the application of the relay to be applied in the long-term periodical testing setup. Retrieving these data values needs a cyclic reading of logs or customer support data.

The final step of the testing process is implementing the test automation, as shown in Figure 9. The automated script will be implemented by using Robot Framework and Python programming languages. The purpose is to automate both the data collection from the protection relay and the long-term testing process. Automating the testing process reduces the potential for human error and minimizes the delay of the process. On the other hand, automating the data collection is based on the automation of the entire testing process, since the data collection is a simple operation that involves extracting outputs and artifacts from a system (A. Pääkkönen, personal conversation, 14.10.2022).

5.2 Requirements Related to Testing Process of the Protection Relay

This chapter addresses requirements based on the interviews discussed in chapter 5.1 in relation with RQ1 and RQ2 introduced in chapter 1.2. The aim is to identify the requirements to provide the relevant data for long-term periodical testing, and then to apply the remaining requirements to find out the method to be used for collecting the relevant data. The testing process will be started by measuring protection relay’s temperature, voltage, and current data with the aim of discovering the relevant data that would allow evaluating the prediction protection relay’s failures. Finally, the method for collecting and monitoring the discovered relevant data periodically over a longer period of time will be investigated.

Based on previous theory, the first requirement of the testing process is to measure data over a long-term period from days to years. The intention is to analyze the measured data, which is one step toward finding out relevant data in long-term testing with REX640. This implies that data must be collected in real-time to detect and verify different failure situations that will be created to test the behavior of the protection relay. Based on this, the second requirement is to collect real-time data continuously.

Storing the data in local-hosted database refers to the management, monitoring, and maintenance of stored data in a local-hosted server. The data storage in cloud-hosted database outsources the data storage to an external cloud service provider which maintains the supporting infrastructure in through its data centers. This study involves long-term periodical testing, which includes analyzing data over a longer period of time, the collected data must be stored in a local or cloud-hosted database. Therefore, the third requirement of this study is to store the data permanently, which refers to the long-term data storage. (Diamond, 2020)

The fourth requirement is utilizing a data logger or figuring out a measuring device that enables saving the data directly in a raw file format, such as comma-separated values (CSV), text file (TXT), or extensible markup language (XML) file formats without the need to utilize a software for intermediate steps in order to re-format the data. This facilitates the automation of the testing process. (A. Pääkkönen, personal conversation, 08.08.2022)

The fifth requirement in this study is finding out a suitable data collection method by considering ABB’s cyber security guidelines (P. Eskelä, personal conversation, 19.08.2022). Cyber security is the ability to protect or defend the use of cyberspace against cyber-attacks and is related to critical infrastructure, cloud security, application security, network security and many other areas where ensuring security is critical (Nguyen et al., 2019, p. 5). Cyber security in automation is primarily maintaining secure operations and data, considering various cases, such as human operator mistakes, remote access situations, maintenance operations, control, support and backup system failure, and online support (ABB, 2017, p. 16).

Thus, the first three requirements are more related to the first research question of this study to find out the relevant data in long-term periodical testing. The rest of the requirements are more related to discovering the method of data collection that is, the second research question of this study (see Figure 10).

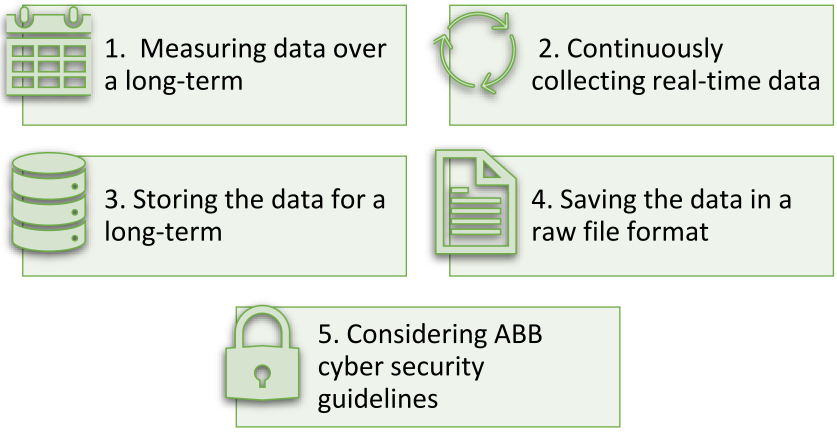


Figure 10. The five requirements of the testing process.

5.3 Testing and Data Collection Process Setup

This chapter implements the testing steps discussed earlier in chapter 5.1:

1. Collect internal and external data points of REX640
2. Implement protection relay failure to monitor the behavior of DUT
3. Long-term periodical testing with the protection relay
4. Analyze the collected data from the protection relay
5. Automate the entire testing process
6. Implement automated script for data collection and storage

The chapter first discusses the implementation of temperature data collection. This is followed by an outline of voltage and current data collection. In addition, this chapter reports the implementation of programming the automation approach of data collection and storing utilizing Python and Robot Framework.

5.3.1 Data Collection Using PicoLog TC-08

The testing process began by examining PicoLog TC-08 data logger proposed by Petteri Liikkanen. The implementation involved attaching two t-type thermocouples to PicoLog data logger’s channels and then connecting PicoLog logger to PC via a universal serial bus (USB) cable. PicoLog data logger includes a free software called PicoLog, which is downloadable on a PC. PicoLog software enables real-time data monitoring, temperature measuring, and collecting data periodically (Pico Technology, 2021, p. 3–7).

However, it was noted that PicoLog data logger produces data directly in a “picolog” file format, which indicates that there is no raw file format available in PicoLog TC-08. The data needs to be exported to other file formats by utilizing PicoLog software (Pico Technology, 2021, p. 3–7). Exporting the files by software complicates the fifth step of test formation which is automating the process (see Figure 9).

5.3.2 Data Collection Using Raspberry Pi 4

The testing process continued by experimenting Raspberry Pi computer to collect and store the temperature data. Raspberry Pi is a single-board computer created in 2012 by Raspberry Pi Foundation that uses Python, Scratch, and C programming languages (Fernandes, 2018, p. 8–10). Python is usually installed by default on Raspberry Pi, which is useful for the test automation phase of this study. As cited in chapter 5.1, temperature data collection is possible, for example, by placing thermal sensors in the hot-spots inside the protection relay to which Raspberry Pi can connect to directly through General-Purpose Input Output (GPIO) pins that enables measuring the hot-spots inside REX640.

The test setup required purchasing of Raspberry Pi 4, SD card, charger, Raspberry Pi protective case, 4.7 kΩ resistor, and eight DS18B20 thermal sensors to measure the temperature from the internal and external data points of REX640 protection relay (see Figure 11). The test setup involved downloading the Raspbian OS software onto the SD card and inputting it into Raspberry Pi 4. A keyboard and a mouse were then attached to Raspberry Pi. Raspberry Pi was connected to the network by plugging an Ethernet cable into the Ethernet-based AFS677 router (ABB, 2010, p. 5). Then a Secure Shell (SSH) connection was established on Raspberry Pi and firewall was configured. In this case, the automation of data collection can be programmed remotely (Ziemann, 2018, p. 134–135). The wiring in Figure 11 illustrates the implementation in the laboratory. In addition, the wiring can be applied to a permanent bread board.

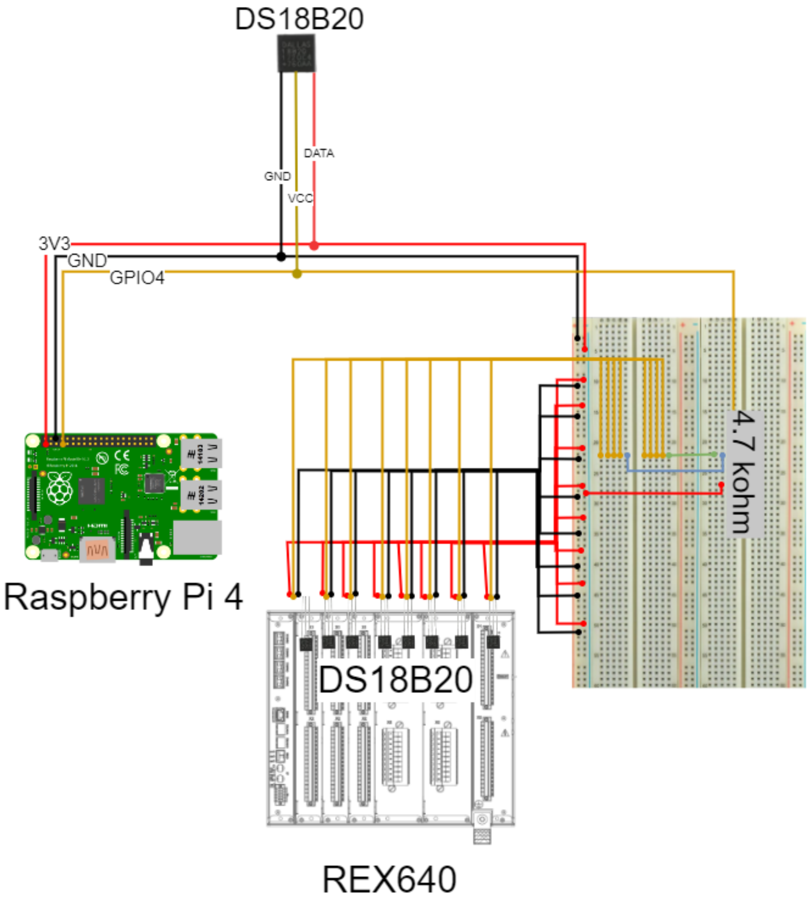


Figure 11. Wiring to obtain the thermal data.

The data collected from DS18B20 thermal sensors was performed by writing a program with Python language that utilizes Raspberry Pi operating system, GPIO pins, and thermal sensors. The program included a class called Sensor_DS18B20 to read the collected temperature data from the path inside Raspberry Pi. Once the thermal sensors were found, the collected temperature data was stored in Celsius and Fahrenheit degrees. In addition, the program specified the date and time at which the temperature data would be printed every 15 seconds. Finally, the class defined how to print the data and sent it in a CSV file format to a database (see Appendix 2).

5.3.3 Data Collection Using DataTaker 80 Series 4

After investigating numerous data collecting methods, it was discovered that the data measurement will be carried out using a data logger called dataTaker 80 (DT80). DT80 is designed for data collection, measurement, and storage. Furthermore, DT80 includes an application programming interface (API) that allows the automation of the testing setup. Also, the data logger includes an internal memory that enables data to be stored for a longer period. DT80 features real-time data to be sent in both CSV and data logger binary file formats, which meets the requirements of this study (Figure 10). DT80 is more suitable for long-term testing since Raspberry Pi does not support analog inputs for voltage or current measurements and it is not designed to measure current and voltage without complex add-on circuits (Sklar, 2012). DT80 has 5 analog channels to measure the temperature, current, and voltage values from the protection relay. In addition, DT80 has 5 digital channels, which enable the connection to the internal relay fault (IRF) output in the PSM100x module (Figure 4).

The test environment was built using DT80 Series 4, a charger, remote SSH connection to test setup over Internet, and DT80's interfaces DeTransfer and dex 2.0, to allow configuring the data logger remotely. Furthermore, Ethernet based LAN was cabled and established between LTS's setup and devices including REX640 and DT80 through AFS677 network and switch router (ABB, 2010, p. 5). Hence, obtaining the temperature was enabled by connecting k-type thermocouples to the analog channels of DT80 and gluing them to the hot-spots of REX640.

Collecting temperature data requires an identification of the modules inside REX640. While necessary wiring is required to the protection relay for collecting current and voltage data. To evaluate the input voltage and input current data of REX640, a different test setup must be built. The voltage data would be measured with a voltage divider by connecting the wiring to the Uaux input of the PSM100x module of REX640 protection relay (Figure 4; ABB, 2020c, p. 117). Thus, measuring of the output voltage of the power supply (U_{out}) can be implemented with the voltage divider which is equal to the input voltage (U_{in}) multiplied by a resistor ratio (R_2/R_1+R_2) according to the following formula (Khan Academy, n.d.; P. Liikkanen, personal conversation, 31.08.2022):

$$U_{out} = \frac{R_2}{R_1 + R_2} \times U_{in}$$

Equation 1. Voltage divider.

The current measurement range of DT80 is ±30 mA and the input current to be measured from the protection relay is over 30 mA, which explains utilizing a shunt resistor. Also, the voltage divider (Equation 1) is utilized for the output voltage of the power supply U_{out} measurements above 30 volts (Thermo Fisher Scientific, 2010).

Hence, the test setup is built to measure the input voltage and the input current of REX640. The measurement was conducted by building the voltage divider circuit and connecting the PSM100x module’s Uaux input channel. Measuring the input current of REX640 was implemented across 0.21 Ω shunt resistor and 24 volts of direct current (VDC) from Powernet power supply. Measuring the output voltage of the power supply required the voltage divider wiring (Equation 1), where the 24 VDC Powernet is the input voltage (U_{in}) of REX640 and the output voltage (U_{out}) of the power supply is measured between the resistors R_1=56 kΩ and R_2=4.7 kΩ (see Figure 12).

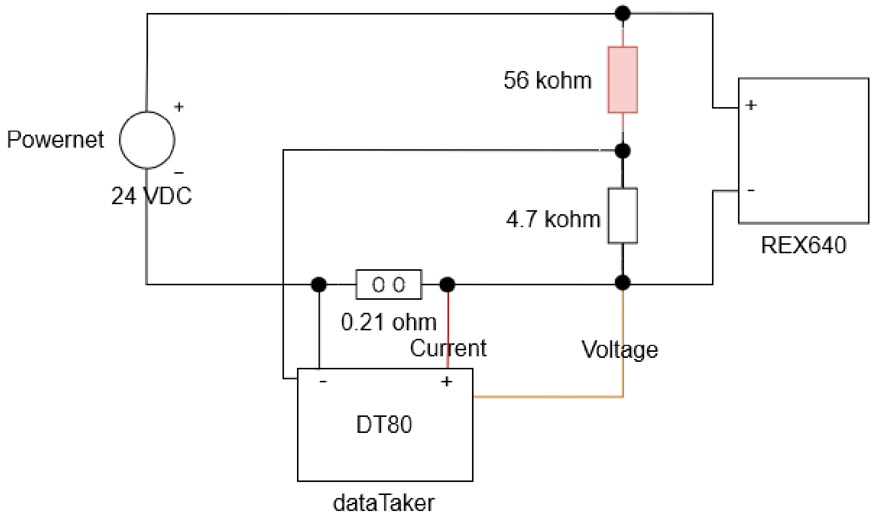


Figure 12. Voltage and current measurement's circuit diagram.

5.3.4 Long-term Data Storage

This chapter discusses the use of a database or alternative services to store and analyze data for longer periods of time. For instance, the data can be collected by DT80 device and stored in PostgreSQL, Azure Internet of Things (IoT) Central, or any other cloud service. However, the database or the alternative services should meet the requirements of this study (Figure 10).

CavyIoT is an open-source IoT platform, which includes triggers that respond automatically to changes in temperature. CavyIoT utilizes an HTTP RESTful API for data storing and retrieving, which is a third-party provided solution (CavyIoT, 2019). However, according to ABB's cyber security requirements, third-party provided solution for storing and retrieving data requires more process overhead due to target company information system compliance. Azure IoT consists of cloud services managed by Microsoft cloud services and IoT devices that communicate with back-end services hosted in the cloud (Microsoft, 2022a). However, Azure IoT Central keeps data only for the last 30 days, and therefore in any case requires an external storage to be taken into use to achieve the requirements of this study (Microsoft, 2022c; J. Lipponen, personal conversation, 08.08.2022).

In order to address the limitations of the cloud-based IoT solutions, a local-hosted PostgreSQL database can be considered as more suitable option to meet the requirements of this study. PostgreSQL is an open-source object-relational database system that utilizes SQL programming language to store data workloads for a longer period (Postgresql, 2022). Storing data in PostgreSQL can be managed by pgAdmin, which is an open-source management and development platform that supports Web browser (PgAdmin, 2022).

The data stored in local-hosted PostgreSQL database can be imported into the open-source data visualization and analytics platform, Grafana to analyze, alert, and compare the state of the protection relay (Grafana Labs, 2022a). Furthermore, to run the automated tests at certain intervals, an automation pipeline solution, for instance, a feature of Microsoft's Azure DevOps product, Azure Pipeline may be scheduled to facilitate the execution of a separately curated test process (Microsoft, 2022b). This allows the detection of the failures of the entire test by monitoring the status of the Azure Pipeline.

5.4 Test Automation Process

This chapter covers one of the most important steps in the overall testing process, the test automation implementation process. Applying test automation to the test environment facilitates the detection of errors, increases the timeliness of the testing process, and enhances the potential for improving the test environment.

The test involves collecting data from the protection relay using DT80 device. The data collected from the device is sent to the locally hosted PostgreSQL database in CSV format. The database is connected to the Grafana platform, which includes a built-in PostgreSQL data source plugin (Grafana Labs, 2022b). Grafana allows the data to be monitored and analyzed. Azure DevOps is also utilized to run the entire test process as pipelines and by facilitating error detection for a longer period of time. Figure 13 illustrates the overall process, where the data is imported and stored in PostgreSQL table by Python and SQL.

With the established connection between PostgreSQL database and Grafana, the data can be continuously synchronized. Test automation also included sending data and connecting to the development and operations server, Azure DevOps using Robot Framework (Microsoft, 2022d). Azure DevOps dashboard enables a single-point view for test execution statuses and data line charts (Microsoft, 2022e).

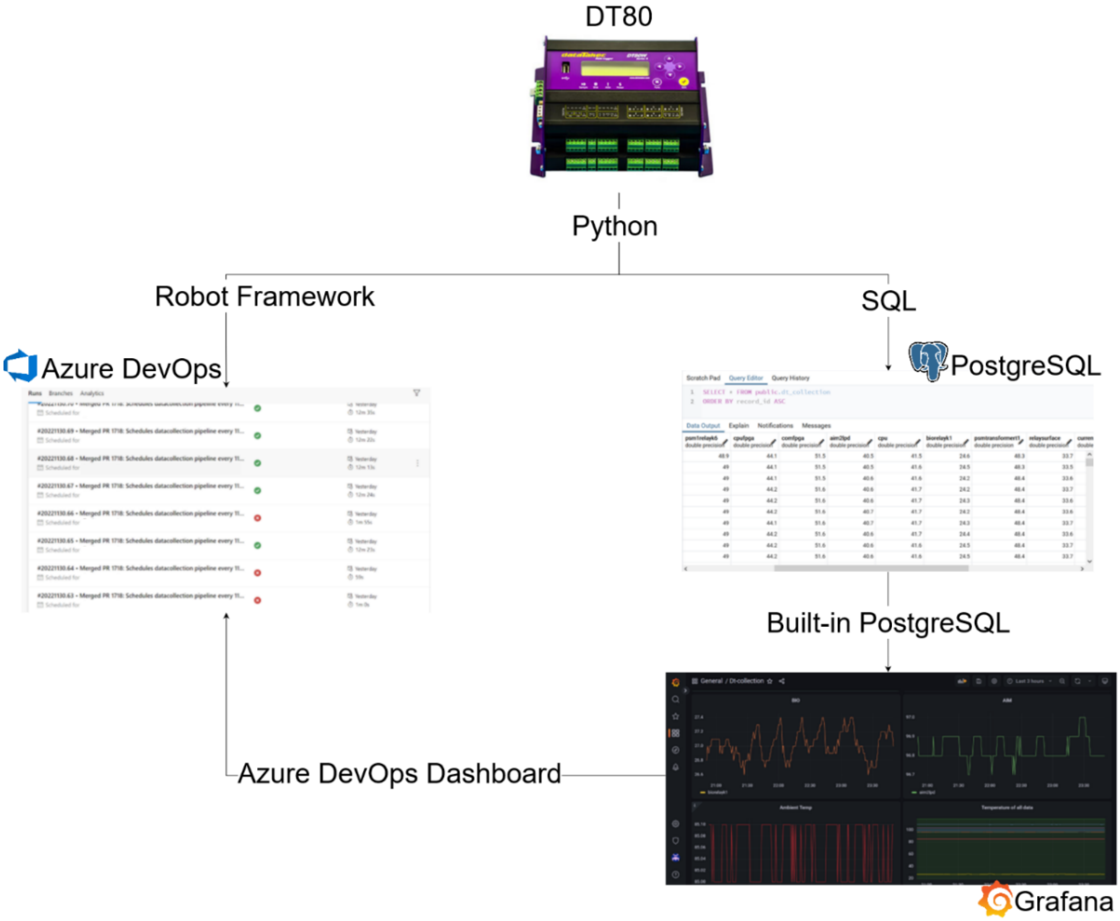


Figure 13. Test automation with Robot Framework and Python.

The measured temperature, voltage, current, and IRF data from the protection relay is imported by writing a Python program with a class called DT80Measurements. The class creates a connection to DT80 device using the device's host internet protocol (IP) address and port. Moreover, the class utilizes DeTransfer commands to configure the collected data of DT80 into CSV format (see Appendix 3).

Furthermore, a connection to the PostgreSQL database is defined in the class using external Python library with date and time stamp, which allows data to be pushed in one-minute intervals directly to a dt_data table located in the PostgreSQL database. The table created in PostgreSQL is established using the SQL programming language and the measured data from DT80 is inserted into this table with SQL's INSERT INTO statement utilizing an external Python library (see Appendix 3).

For each attribute or method in the Python code, various keywords to Robot Framework were defined, such as “Configure DT80 to CSV Mode” or “Read DT80 Values” (Appendix 3). These keywords enable calling the Python methods from Robot Framework program. Azure DevOps Pipeline then runs Robot Framework program in every 11 minutes interval as presented in the Robot Framework script (Appendix 4). This approach is most suitable for long-term testing, as Azure DevOps limits the number of scheduled runs to around 1000 runs per week (Microsoft, 2022b).

5.5 Preparation for Test Execution

The testing setup maintains that data is collected in PostgreSQL database and run in Azure DevOps Pipeline. Once the protection relays component overheats, DT80 reports "OverRange" and Azure Pipeline indicates it as an error. Also, data synchronization to Grafana is continuous as long as PostgreSQL is storing the test data incessantly. Furthermore, a direct connection to the protection relay is created, allowing to monitor of the events reported by the relay in detail. Events can be observed by the protection relay’s Web human-machine interface (WHMI) connection through a Web browser (ABB, 2020d, p. 31). Simultaneously, the internal faults of the relay could be detected using the IRF channel located in the PSM1001 module, which is directly connected through DT80.

As part of the testing process, thermocouples are attached to different hot-spots from internal and external data points of the protection relay. The hot-spots refers to the internal components of the protection relay that are most sensitive to temperature rise and may potentially fail at high temperatures. These hot-spots are identified by launching REX640 and utilizing FLIR i50 thermal imaging infrared camera. The thermal camera measures the temperature when its spot marker points to any REX640 component (FLIR, 2008). When the thermal camera measures component temperature as 40°C or more, the component is considered hot-spot point. For instance, Figure 14 represents the hot-spots discovered in the central processing unit (CPU), where the spot marker of the thermal camera points to the field-programmable gate array (FPGA) component (left), and the CPU component is located on the right.

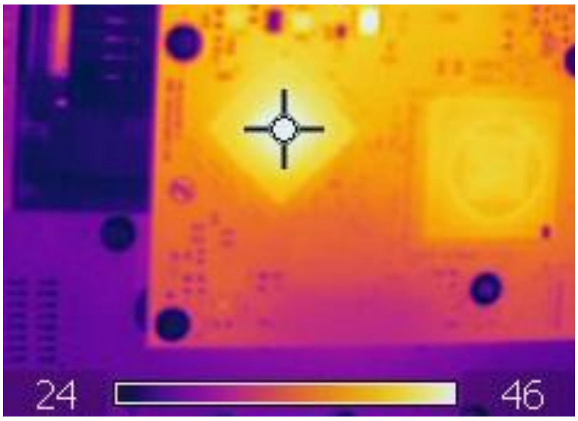


Figure 14. Hot-spots discovered with thermal infrared camera.

There are nine thermocouples utilized altogether. One thermocouple measures the external data point from REX640 surface, and another thermocouple measures the temperature of the test chamber. The other thermocouples measure the internal data points inside REX640. The thermal camera identified a total of seven hot-spots inside REX640 modules and thermocouples were glued on each of them. The identified hot-spots are CPU module’s CPU and FPGA components, COM1 module’s FPGA component, BIO1 module’s Relay K1 component, AIM1 module’s Lattice complex programmable logic device (CPLD) component, and PSM1 module’s Relay K6 and Transformer T1 components. Table 2 below contains a list of discovered hot-spots inside REX640.

Table 2. The internal hot-spots of REX640.

CPU	COM1001	BIO1003	AIM1002	PSM1001
CPU	FPGA	Relay K1	Lattice CPLD	Relay K6
FPGA				Transformer T1

The Weiss Umwelttechnik WK3 temperature test chamber shown in Figure 15 is designed for testing a variety of products at different temperatures and allows long-term testing at temperatures ranging from -42°C to +180°C (Weiss Technik, 2010, p. 8). The purpose is to test the behavior of REX640 in the test chamber by adjusting the temperature over a longer duration at periodical intervals. At the same time, the measured data will be analyzed from Grafana graphs, and the events reported by the protection relay will be monitored from the WHMI of REX640.

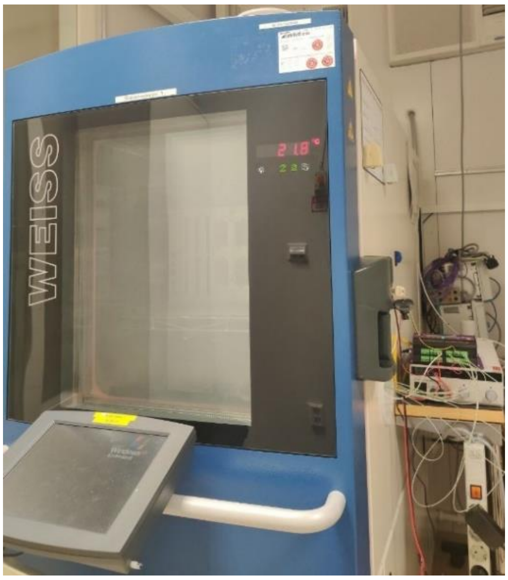


Figure 15. Temperature test chamber and DT80.

The maximum environmental temperature range of REX640 depends on the test to be performed. Table 3 presents different types of temperature tests such as Dry heat, Dry cold, Damp heat, Change of temperature, and Storage tests. Each type test has a specific temperature range limitation and duration (ABB, 2020c, p. 38–41). In long-term periodical testing, the environmental temperature range of REX640 is from +55°C to +85°C, as the protection relay will be tested periodically in different temperatures for a longer period of time.

Table 3. Environmental temperature range in various protection relay type tests (ABB, 2020c, p. 41).

Type test	Environmental temperature range	Maximum duration (h)
Dry heat	+55°C	96
	+85°C	16
Dry cold	-25°C	96
	-40°C	16
Damp heat	+25°C …+55°C, with humidity >93%	6 cycles (12h + 12h)
Change of temperature	-25°C …+55°C	5 cycles (3h + 3h)
Storage	-40°C	96
	+85°C	96

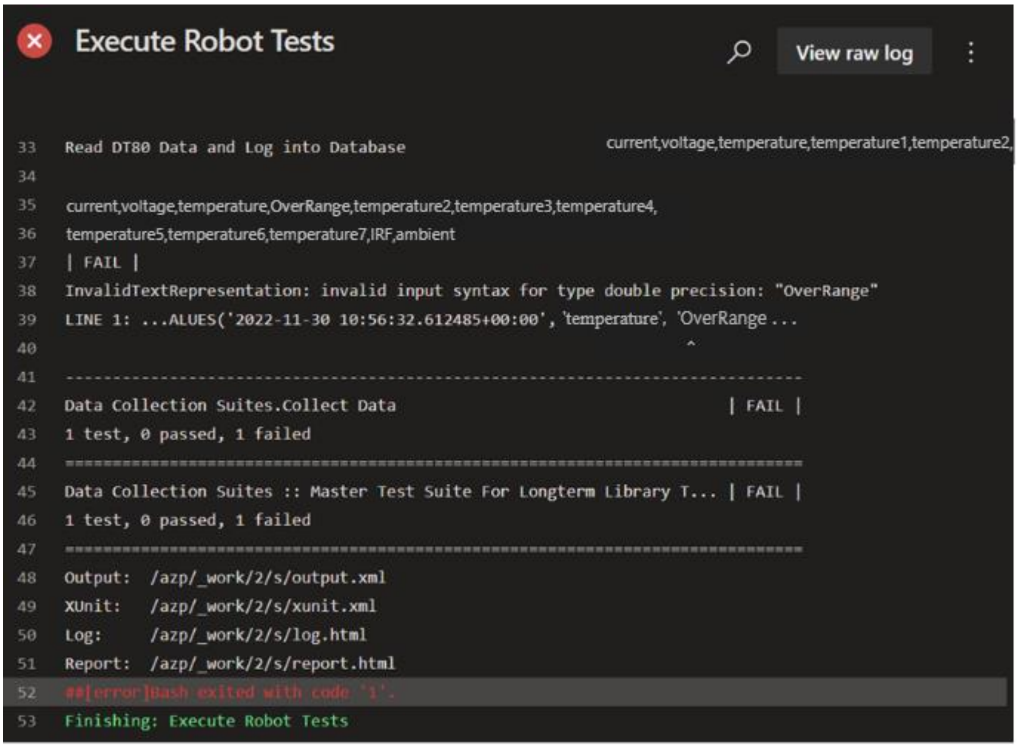
The extreme temperature of the protection relay components was verified by finding out the junction temperature, which is measured from the silicon surface of the board. The junction temperature is from -40°C to +125°C (Freescale Semiconductor, 2012, p. 1, 108–109). This implies that the extreme temperature of REX640 components is +125°C.

REX640 was placed in the temperature test chamber after verifying the functionality of Azure Pipeline, PostgreSQL, and Grafana. Furthermore, the accuracy of the thermocouples was examined with ten measurements at different temperature ranges by comparing the ambient temperature reported by the test chamber with the ambient temperature measured by the thermocouple, resulting in an accuracy of ±1°C. Based on this, the data measured by thermocouples is sufficiently accurate for the needs of this study. Thus, temperature adjustments were scheduled in the test chamber for 12 days, with the goal of identifying the relevant data and estimating the predictability of the protection relay failure situation. Testing was then started a couple of days after REX640 was placed in the test chamber until the measured data seemed stabilized as observed from Grafana's graphs.

5.6 Test Execution Process

The test execution process consists of six steps: data collection, data input, data processing, data storage, data analysis, and data output (Geetha, 2014, p. 3). This study collected data by measuring temperature, voltage, and current, collected the measured data from DT80, processed the data with automation, and stored the data in PostgreSQL database. This chapter describes the output of the collected data at different temperatures and analyzes the graphs from Grafana. However, the collected data will be analyzed further in Chapter 6.

REX640 was kept for a few days at the normal temperature of the test chamber, 25°C, to allow the temperatures of the components to stabilize. After stabilization, the lowest temperature was measured from the BIO modules Relay K1 component, while the highest temperature was measured from the COM modules FPGA component. At the same time, the input voltage of REX640 showed 24 V and the input current was about 666 mA. When the test chamber temperature was raised to 55°C, the temperature of REX640 components and the input current increased, while the voltage remained unchanged. Nevertheless, at 55°C, one of the thermocouples failed and the failure was detected when Azure Pipeline showed an error (Figure 16).



```
Read DT80 Data and Log into Database                    current,voltage,temperature,temperature1,temperature2,

current,voltage,temperature,OverRange,temperature2,temperature3,temperature4,
temperature5,temperature6,temperature7,IRF,ambient
| FAIL |
InvalidTextRepresentation: invalid input syntax for type double precision: "OverRange"
LINE 1: ...ALUES('2022-11-30 10:56:32.612485+00:00', 'temperature', 'OverRange...
                                                                     ^
------------------------------------------------------------------------------
Data Collection Suites.Collect Data                                   | FAIL |
1 test, 0 passed, 1 failed
==============================================================================
Data Collection Suites :: Master Test Suite For Longterm Library T... | FAIL |
1 test, 0 passed, 1 failed
==============================================================================
Output:  /azp/_work/2/s/output.xml
XUnit:   /azp/_work/2/s/xunit.xml
Log:     /azp/_work/2/s/log.html
Report:  /azp/_work/2/s/report.html
##[error]Bash exited with code '1'.
Finishing: Execute Robot Tests
```

Figure 16. Azure test Pipeline detects an overrange value and fails the test case.

After repairing the thermocouple and the temperature of REX640 components stabilized at 55°C, the temperature of test chamber was raised to 75°C. The current graph responded to the temperature rise by increasing. When the temperature of the components stabilized at 75°C, the temperature of the test chamber was increased to 85°C. Subsequently, the current increased more and showed negligible fluctuations with an accuracy of 3 mA. REX640 components became warmer as the test chamber’s temperature increased as shown in Figure 17 below. Figure 17 illustrates the temperature of all the previously mentioned hot-spots (Table 2), the ambient temperature, which is the temperature of the test chamber, and the current graph.

Figure 17. Temperature of all components and current graph after gradual increases of ambient temperature from 55°C to 85°C.

The ambient temperature was then raised gradually to over 85°C, where COM modules FPGA component exceeded the junction temperature, that is 125°C and continued operating by responding to the temperature increase. The temperature of the test chamber was adjusted periodically 12 days for a total of 20 times with eight testing periods, during which REX640 reacted considerably over the supported 85°C by rebooting three times in a row as shown in Figure 18 with huge fluctuation occurring on the line graphs between timestamp of 14:00 and 14:10. In consequence, the protection relay was no longer contacted through WHMI, even though the IRF of REX640 was on, indicating that there are no failures inside the protection relay. Therefore, the temperature of the test chamber was decreased to determine the defect that occurred. At the same time, the input current graph declined.

The situation of the protection relay remained the same until the temperature of the test chamber decreased significantly below 85°C and the current graph suddenly reached a peak, further analyzed in chapter 6. The connection to REX640 was re-established through WHMI, where the protection relay indicated in the event log that it had been rebooted and resumed normal operation. The situation can be observed in Figure 18, where the current graph increases significantly after 16:10 timestamp.

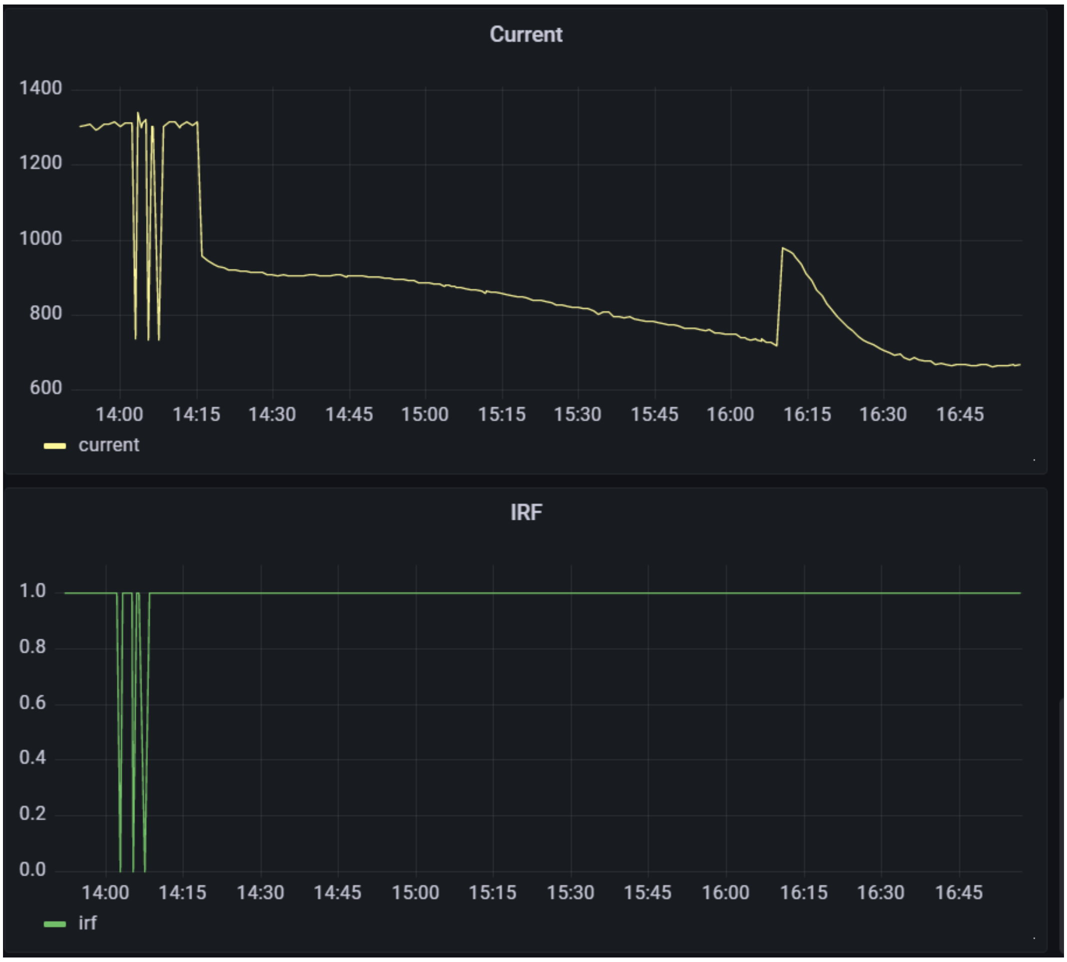


Figure 18. REX640 reboots at high temperature and recovers after decreasing the ambient temperature.

The temperature was then increased back to 85°C to compare the data before and after the rebooting of the protection relay. The temperatures of the components reacted to the test chamber temperature increase more sensitively than before REX640 rebooting.

When the temperature stabilized at 85°C, the components increased to the same temperature level as before the rebooting of REX640. The current curve was more volatile than before protection relays rebooting. The current fluctuated with an accuracy of 10 mA, while before the rebooting it fluctuated with an accuracy of 3 mA (Figure 19). The measured voltage stayed at the same level as before the rebooting. The analysis of the discussed test results, and answers to research questions RQ1 & RQ2 can be found from chapter 6 below.

Figure 19. Fluctuating current curve after the rebooting of REX640 at 85°C.

6 Analysis of Results and Answers to Research Questions

The purpose of this chapter is to analyze the test results presented earlier and to answer the research questions of this study. When there is a failure in an electrical system, the system's protection relay is supposed to react by cutting off the fault. However, when the protection relay fails, the consequence can be a major malfunction or a refuse operation that extends the failure zone (Chen et al., 2021, p. 1–2). Therefore, the primary purpose of this study is to subsequently reduce the probability of protection relay failures by monitoring the collected data during the long-term periodical testing.

6.1 Analysis of the Results and Answers to the First Research Question

The long-term testing was carried on for a total of 12 days during which the temperature of the test chamber was adjusted periodically 20 times with eight testing intervals. Based on the test results discussed in the previous chapter, REX640 failed when the temperature exceeded the maximum supported environmental temperature of the protection relay. After investigating the reason for REX640 failure, it was determined to be caused by long-term thermal runaway. Thermal runaway occurs when the device overheats and the critical temperatures of the device are exceeded (Conte et al., 2009, p. 4). This resulted in a sharp rise in the input current and an increase in component-specific temperature data as the test chamber temperature increased above the critical temperature of the protection relay.

Subsequently, it was concluded that protection relay's input current data, and component-specific temperature data are extremely relevant as part of other approaches to evaluate the predictability of the protection relay's failure. This is explained by the behavior of the data, including the behavior of the input current during the rebooting of the protection relay, when it indicated a failure in the protection relay.

Additionally, the IRF of REX640 did not show the recovery of the protection relay and stayed on, indicating that there was no failure in the protection relay, even though the protection relay was no longer communicating through WHMI after the rebooting. While the recovery of REX640 was shown only in the input current after decreasing the ambient temperature, which confirms the input current observation. On the other hand, after the recovery of REX640, the component-specific temperature data reacted more sensitively to changes in ambient temperature as the temperature increased. This indicates that at high temperatures, it is extremely probable to estimate the prediction of the protection relay failure through the component-specific temperature data as well as through the input current data.

Hence, if a failure occurs inside the protection relay, its occurrence can be observed by continuous input current data and component-specific temperature data monitoring. Continuous monitoring can be done utilizing the test automation developed in this study by analyzing the data with Grafana graphs and evaluating the test with Azure Pipeline. As a result, RQ1 can be answered by the fact of utilizing the input current data and the component-specific temperature data as relevant data in long-term periodical testing with protection relays.

6.2 Answers to the Second Research Question

In this research, it was known that the data to be collected is temperature, current, and voltage, but it was not known what data is relevant in long-term testing. Conducting long-term periodical testing allowed the identification of the relevant data to evaluate the prediction of the protection relay’s failure. Since it was known that the data to be collected is protection relay’s temperature, current, and voltage data, an approach was taken to find a suitable method to collect data on these bases. This study explored various methods for the collection of relevant data to meet the research requirements: 1) Measuring data over a long-term, 2) Continuously collecting real-time data, 3) Storing the data for a long-term, 4) Saving the data in a raw file format, and 5) Considering ABB cyber security guidelines.

This study looked for a data collection method, by exploring first an appropriate device to collect the relevant data. The device should be able to collect temperature, voltage, and current data, measure data over a long-term, support test automation and continuous real-time data collection, and allow data to be exported in a raw format to meet the research requirements. The study considered utilizing PicoLog data logger, but it was found that PicoLog allows saving data only in picolog file format and does not meet the study requirement 4, where the data must be saved in a raw file format.

Furthermore, utilizing Raspberry Pi for data collection was also researched in this study, and a program was developed on a Raspberry Pi for data collection. However, LTS requires a simple solution that can be utilized for a long-term to collect the relevant data and Raspberry Pi demands complex add-on circuits for current and voltage measurements, which complicates the test implementation for a longer term. Consequently, it was decided to use the dataTaker 80 device to collect data and to improve test automation. This enabled the data to be measured and collected by DT80 and linked to the developed test automation for continuous data monitoring. This solution supported long-term testing best since it does not require specialized configuration and complicated wiring for data collection.

DT80 allowed measuring the data over a long-term and provided real-time data collection continuously with minute intervals. DT80 enabled temperature data collection by a direct connection with the thermocouples. It also enabled the collection of voltage and current data by simple wiring to the protection relay’s Uaux input channel. In addition, DT80 enabled the connection to the IRF channel of REX640 to monitor protection relay’s internal failures.

Once a suitable data logger was found, it was then searched for a suitable database for data storage, which should also meet the requirements of the study. This study investigated storing the data in CavyIoT platform, but it was observed that the platform utilizes a third-party provided solution that does not meet ABB’s security requirements and the fifth requirement of this study, where the data collection method should consider ABB cyber security guidelines. Eventually, the locally hosted PostgreSQL was considered as more suitable option as the requirements per study can be equally met.

The data was collected and stored in PostgreSQL and Azure DevOps databases by the test automation developed in this study. The test automation utilized Azure DevOps Pipeline to execute Robot Framework program every 11 minutes as Azure DevOps restricts the number of scheduled runs to around 1000 runs per week. In addition, a table was created in a local PostgreSQL database using SQL programming language and the measured data from DT80 was inserted in one-minute intervals into the created table utilizing a Python library. To improve the data analysis process of this study, a connection was created between PostgreSQL and Grafana that allowed the relevant data to be visualized and further analyzed from Grafana's graphs.

Finally, a suitable method for long-term periodical data collection was found to meet all requirements of this study. DT80 enabled measuring the data over a long-term (research requirement 1), collecting real-time data continuously (research requirement 2), and saving the data in a raw file format (research requirement 4). Furthermore, PostgreSQL allowed storing the data for a long-term (research requirement 3) with the considering of ABB cyber security guidelines (research requirement 5). As a result, RQ2 can be answered by stating that the suitable method for collecting the relevant data requires the functionalities of collecting data in real-time, connectivity to generic measurement sensors, and further processing enables by a raw data format. All of these were found to be met by DT80, which was the selected data collection device in this study.

7 Concluding Remarks

Protection relay is an important component in the electrical network, intended to protect the distribution network and react in the event of abnormal conditions. This study discussed different types of protection relays, including static, electromechanical, microprocessor-based, and digital, also known as numerical. Each protection relay type can be tested using different testing methods. However, protection relays are developing in a rapid pace and the more modern the protection relays are, the harder it is to detect their internal failures. Therefore, continuous testing, quality assurance, and development of protection relay's testing methods reduce the potential for the internal failures that may occur in protection relays.

There are different ways to test protection relays, for instance, using a fault simulator to generate and inject data into DUT and to analyze protection relay failures (Ouadi et al., 2011) or using a protection relay tester to collect data from DUT and detect protection relay failures (Xia et al., 2021). This study noted that there are hidden faults in protection relays that can be revealed when the protection relay is under abnormal conditions, such as a harsh operating environment (Zhao et al, 2019, p. 1735). This research reviewed previous related studies and concluded that long-term periodical testing is a relatively unexplored area with protection relays through a research gap. The study stated that long-term periodical testing can be utilized to evaluate the prediction of protection relay failure. In addition, an interview was conducted with ABB experts to gain insights on the practical application of this research.

This study covered long-term periodical testing as a part of LTS, which focuses on improving and ensuring the condition of the protection relay during its constantly increasing lifetime. A method to collect, process, store, and analyze the relevant data was explored in this study. Furthermore, the study developed a program to automate the data collection and storage process. This concluded with the implementation of the long-term periodical testing with REX640 protection relay and the identification of the relevant data in long-term periodical testing.

In general, the study achieved its primary objectives. The first objective was to determine the relevance of the data in long-term periodical testing with protection relays. The use of the input current as part of the assessment of predicting the protection relay failures is possible, but the use of component-specific temperature data in addition to the input current at high temperatures could possibly be used to evaluate the prediction of the protection relay failures. The second objective was to discover a method to collect relevant data. This research investigated many possible methods which can be adapted to different practices, but the most appropriate method for this study included the use of DT80, which allowed the measured data to be collected over long-term periodical testing, data storage in the local-hosted database, and test automation of the entire testing process.

The thesis provided insights about long-term testing, protection relays, the significance of data points in relays, and the importance of protection relays as part of the electrical network. This study shall inform researchers that the protection relay's input current and component-specific temperature data are relevant in evaluating the prediction of protection relay failures. Furthermore, the study will inform practitioners that evaluating the prediction of protection relay failure has implications for improving the performance of the distribution network, as predicting the failure of protection relays can be evaluated in advance before the occurrence of major disasters in the distribution network.

LTS will continue conducting long-term periodical tests in the future, to obtain more resources, collect more data and answers over a longer period of time, and provide opportunities to improve and extend the test automation developed in this study. Future research should investigate the use of long-term testing to examine and compare the behavior of multiple protection relays with longer-term tests to collect further data that guarantee the research benefits. In addition, another area of future research could be the use of developed technologies, for instance, artificial intelligence to perform and improve long-term periodical testing. This enables comparative analysis by conducting long-term periodical testing on a wide range of protection relays that could increase the value and verification of test results.

References

ABB. (2022a). ABB:stä lyhyesti. Retrieved 30.05.2022 from https://new.abb.com/fi/abb-lyhyesti

ABB. (2022b). ABB Oy, Distribution Solutions. Retrieved 17.6.2022 from https://new.abb.com/fi/abb-lyhyesti/suomessa/liiketoiminnat/distribution-solutions

ABB. (2022c). ABB Oy, Protection relays. Retrieved 16.8.2022 from https://new.abb.com/medium-voltage/digital-substations/protection-relays

ABB. (2022d). ABB Oy, Relion protection and control. Retrieved 21.9.2022 from https://new.abb.com/medium-voltage/digital-substations/relion

ABB. (2020a). REX640 IEC 61850 Engineering Guide. Retrieved 30.05.2022 from https://search.abb.com/library/Download.aspx?DocumentID=1MRS759116&LanguageCode=en&DocumentPartId=&Action=Launch

ABB. (2020b). REX640 Technical Manual. Retrieved 9.6.2022 from https://search.abb.com/library/Download.aspx?DocumentID=1MRS759142&LanguageCode=en&DocumentPartId=&Action=Launch

ABB. (2020c). REX640 Product Guide. Retrieved 9.6.2022 from https://search.abb.com/library/Download.aspx?DocumentID=1MRS759144&LanguageCode=en&DocumentPartId=&Action=Launch

ABB. (2020d). Operation Manual. Retrieved 16.6.2022 from https://search.abb.com/library/Download.aspx?DocumentID=1MRS759118&LanguageCode=en&DocumentPartId=&Action=Launch

ABB. (2019). Distribution substation products – Global product offering. Retrieved 13.6.2022 from https://search.abb.com/library/Download.aspx?DocumentID=1MRS758911&LanguageCode=en&DocumentPartId=&Action=Launch

ABB. (2017). Technical guide. Cybersecurity for ABB drives. Retrieved 26.10.2022 from https://library.e.abb.com/public/e033f84022a4422fad29bcfd9109ada3/EN_Cybersecurity_guide_B_A4.pdf

ABB. (2010). ABB FOX Switch (AFS) Family for utility applications. Utility Communications. Retrieved 12.9.2022 from https://library.e.abb.com/public/c9b5f3b4ee491ef8c125784000532ea2/AFS%20Flyer%202011.pdf

ABB. (2004). Excellence, Safety and Comfort — 100 Years of Relay Protection In Sweden. Retrieved 13.6.2022 from https://library.e.abb.com/public/d9ed154fb67595f5c12574b9002efb97/1MRK500063-GEN_en_Excellence__safety_and_comfort_-_100_years_of_relay_protection_in_Sweden.pdf

Abniki, H., Alipour, H., Keramat, M. M., Ansari, S., Abyaneh, H. A. & Razavi, F. (2022). Practical Test of Synchronization Relay. Retrieved 26.7.2022 from https://ieeexplore.ieee.org/document/9763133

Chen, Q., Zhou, X., Sun, M. & Zhang, X. (2021). Fault Tracking Method for Relay Protection Devices. Retrieved 11.12.2022 from https://www.researchgate.net/publication/351490183_Fault_Tracking_Method_for_Relay_Protection_Devices

CavyIoT. (2019). Retrieved 16.12.2022 from https://www.cavyiot.com/#learn-more

Conte, V. F., Gollob, P. & Lacher, H. (2009). Safety in the battery design: the short circuit. Retrieved 10.12.2022 from https://www.researchgate.net/publication/297162170_Safety_in_the_battery_design_The_short_circuit

Diamond, P. (2020). Cloud storage vs. on-premises servers: 9 things to keep in mind. Microsoft. Retrieved 16.12.2022 from https://www.microsoft.com/en-us/microsoft-365/business-insights-ideas/resources/cloud-storage-vs-on-premises-servers

Fingrid. (2022a). Kunnossa kaiken ikää. Retrieved 26.9.2022 from https://www.fingridlehti.fi/kunnossa-kaiken-ikaa/

Fingrid. (2022b). Kantaverkon ja asiakasliityntöjen relesuojaus. Retrieved 26.9.2022 from https://www.fingrid.fi/globalassets/dokumentit/fi/palvelut/kulutuksen-ja-tuotannon-liittaminen-kantaverkkoon/kantaverkon-ja-asiakasliityntojen-relesuojausohje.pdf

Fernandes, S. L. (2018). Raspberry Pi 3 Cookbook for Python Programmers: Unleash the Potential of Raspberry Pi 3 with over 100 Recipes, 3rd Edition. Retrieved 25.8.2022 from https://finna.fi/Record/jamk.993540254806251

Freescale Semiconductor. (2012). P1012 QorIQ Integrated Processor Hardware Specifications. Retrieved 8.12.2022 from https://community.element14.com/products/devtools/technicallibrary/m/files/16768

FLIR. (2008). Thermal Imaging InfraRed Cameras. Retrieved 7.12.2022 from https://www.instrumart.com/assets/FLIRi40_i50_i60dataVFLIRnoprice.pdf

Grafana Labs. (2022a). Dashboard anything. Observe everything. Retrieved 17.11.2022 from https://grafana.com/grafana/?plcmt=footer

Grafana Labs. (2022b). PostgreSQL data source. Retrieved 5.12.2022 from https://grafana.com/docs/grafana/latest/datasources/postgres/

Geetha, M. A. (2014). Privacy Preservation Using Intelligent Techniques. Retrieved 26.10.2022 from https://www.researchgate.net/publication/327551947_Privacy_Preservation_Using_Intelligent_Techniques

Gajic, Z. (2009). Easy Method for Testing Transformer Differential Relays. Retrieved 19.7.2022 from https://www.researchgate.net/publication/289184445_Easy_Method_for_Testing_Transformer_Differential_Relays

Goh, Y. L., Nagi, F., Ramasamy, A. K. & Abidin, A. A. Z. (2010). Numerical relay for overcurrent protection using TMS320F2812. Retrieved 26.6.2022 from https://www.researchgate.net/publication/228901568_Numerical_relay_for_overcurrent_protection_using_TMS320F2812

Gurevich, V. (2009). Tests of Microprocessor-based Relay Protection Devices: Problems and Solutions. Retrieved 17.6.2022 from https://www.researchgate.net/publication/265064771_Tests_of_Microprocessor-based_Relay_Protection_Devices_Problems_and_Solutions

International Electrotechnical Commission. (2020). Communication networks and systems for power utility automation - Part 7-1: Basic communication structure - Principles and models. Retrieved 26.10.2022 from https://library.iec.ch/ieclib/libweb2.nsf/ArtNum/108801?OpenDocument

Haila, T. (2016). Häiriötallennin- ja suojareletietojen hyödyntäminen katkaisijoiden kunnonvalvonnassa. Fingrid Oy. Retrieved 26.9.2022 from https://www.fingrid.fi/globalassets/dokumentit/fi/yhtio/tki-toiminta/raportit/haila_tuomas.pdf

Khan Academy, (n.d.). Voltage divider. Retrieved 11.12.2022 from https://www.khanacademy.org/science/electrical-engineering/ee-circuit-analysis-topic/ee-resistor-circuits/a/ee-voltage-divider

Kasanen, E., Lukka, K. & Siitonen, A. (1993). The Constructive Approach in Management Accounting Research. Retrieved 3.8.2022 from https://mycourses.aalto.fi/pluginfile.php/183797/mod_resource/content/1/Kasanen%20et%20al%201993.pdf

Lackovic, V. (n.d). Protection Relay Testing and Commissioning. Retrieved 17.7.2022 from https://www.cedengineering.com/userfiles/Protection%20Relay%20Testing%20and%20Commissioning-R1.pdf

Lindblad, P. (2008). Prolonging Periodical Testing Intervals of Numerical Protection Relays. Retrieved 19.7.2022 from https://ieeexplore.ieee.org/document/4497030

Lukka, K. (2003). The Constructive Research Approach. Retrieved 2.8.2022 from https://www.researchgate.net/publication/247817908_The_Constructive_Research_Approach

Lukka, K. (2001). Konstruktiivinen tutkimusote. Retrieved 24.7.2022 from https://metodix.fi/2014/05/19/lukka-konstruktiivinen-tutkimusote/

Microsoft. (2022a). What is Azure Internet of Things (IoT)?. Retrieved 26.10.2022 from https://learn.microsoft.com/en-us/azure/iot-fundamentals/iot-introduction

Microsoft. (2022b). Configure schedules for pipelines. Retrieved 17.11.2022 from https://learn.microsoft.com/en-us/azure/devops/pipelines/process/scheduled-triggers?view=azure-devops&tabs=yaml

Microsoft. (2022c). Quotas and limits. Retrieved 29.11.2022 from https://learn.microsoft.com/en-us/azure/iot-central/core/concepts-quotas-limits

Microsoft. (2022d). What is DevOps?. Retrieved 15.12.2022 from https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-devops/

Microsoft. (2022e). Add widgets to a dashboard. Retrieved 12.12.2022 from https://learn.microsoft.com/en-us/azure/devops/report/dashboards/add-widget-to-dashboard?view=azure-devops

Ma, K., Huang, S. Chen, J. & Yang, Y. (2010). Discussion of the Testing Technology for Protection Device Used in Digital Substation. Retrieved 22.7.2022 from https://ieeexplore.ieee.org/document/5736034

Nguyen, G. N., Chatterjee, J. M., Khari, M., Kumar, R., Le, D. & Mishra, B. K. (2019). Cyber Security in Parallel and Distributed Computing. Retrieved 26.10.2022 from https://tritonia.finna.fi/uva/Record/nelli07.4100000007816832

Na, Q. & Huaichang, D. (2015). Extension and application based on robot testing framework. Retrieved 20.6.2022 from https://ieeexplore.ieee.org/document/7339089

Ouadi, A., Chafal, M., Bentarzi, H. & Zitouni, A. (2011). Microcontroller based protective relay testing system. Retrieved 20.7.2022 from https://www.researchgate.net/publication/262257924_Microcontroller_based_protective_relay_testing_system

Postgresql. (2022). About PostgreSQL. Retrieved 26.10.2022 from https://www.postgresql.org/about/

PgAdmin. (2022). pgAdmin. Retrieved 30.9.2022 from https://www.pgadmin.org/

Pico Technology. (2021). PicoLog Cloud, TC-08. Retrieved 23.8.2022 from https://www.picotech.com/download/datasheets/usb-tc-08-thermocouple-data-logger-data-sheet.pdf

Patki, S., Kamin, D. & Suyash, K. (2018). Numerical Relay Testing and Validation in Relay Life Cycle. Retrieved 27.7.2022 from https://www.researchgate.net/publication/329338067_NUMERICAL_RELAY_TESTING_AND_VALIDATION_IN_RELAY_LIFE_CYCLE

Robot Framework. (2022). Robot Framework User Guide – Version 5.0. Retrieved 16.8.2022 from https://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html#getting-started

Rahebi, J. & Al-Shalah, M. M. S. (2020). Design, Modeling and Implementation of Multi-Function Protective Relay with Digital Logic Algorithm. Retrieved 1.9.2022 from https://dergipark.org.tr/en/download/article-file/1155506

Sravanthi, S., Dheenadhayalan, R., Vinod, G., Madhusoodanan, K. & Devan, K. (2016). Reliability Model of a Relay Output Card with Diagnostic Circuirty for Safety

Instrumented System. Retrieved 8.12.2022 from https://ieeexplore.ieee.org/document/7815851

Sklar, M. (2012). Analog Inputs for Raspberry Pi Using the MCP3008. Retrieved 19.12.2022 from https://learn.adafruit.com/reading-a-analog-in-and-controlling-audio-volume-with-the-raspberry-pi?view=all

Salminen, A. (2011). Mikä kirjallisuuskatsaus? Johdatus kirjallisuuskatsauksen tyyppeihin ja hallintotieteellisiin sovelluksiin. Retrieved 15.8.2022 from https://osuva.uwasa.fi/bitstream/handle/10024/7961/isbn_978-952-476-349-3.pdf?sequence=1&isAllowed=y

Thummalapenta, S., Sinha, S., Singhania, N. & Chandra, S. (2012). Automating Test Automation. Retrieved 27.9.2022 from https://ieeexplore.ieee.org/document/6227131

Turner, S. (2012). Using COMTRADE Records to Test Protective Relays. Retrieved 25.7.2022 from https://ieeexplore.ieee.org/abstract/document/6201248

Thermo Fisher Scientific. (2010). DT80 Range User's Manual. Retrieved 12.9.2022 from https://assets.thermofisher.com/TFS-Assets/ANZ/manuals/datataker-dt8x-users-manual-b8.pdf

Wang, Y., Pyhäjärvi, M. & Mäntylä, M. V. (2020). Test Automation Process Improvement in a DevOps Team: Experience Report. Retrieved 19.10.2022 from https://ieeexplore.ieee.org/document/9155991

Welton, D. & Knapek, W. (2017). Important Considerations in Testing and Commissioning Digital Protective Relays. Retrieved 26.9.2022 from https://ieeexplore.ieee.org/document/7967004

Wester, C., Smith, T. & Fahey, T. (2013). Commissioning and periodic maintenance of microprocessor-based protection relays at industrial facilities. Retrieved 20.6.2022 from https://ieeexplore.ieee.org/document/6525284

Warrington, A. R. van. C. (2012). Protective Relays: Their Theory and Practice Volume One. Retrieved 27.9.2022 from https://books.google.fi/books?id=zIb-gBwAAQBAJ&dq=protection+relay+closes+connectors&lr=&hl=fi&source=gbs_navlinks_s

Werstiuk, C. (2012). The Relay Testing Handbook. Retrieved 19.6.2022 from https://books.google.fi/books/about/The_Relay_Testing_Handbook_5D.html?id=MHQ-lDkW0TAC&redir_esc=y

Weiss Technik, (2010). Perfection in performance, equipment and design. Retrieved 4.12.2022 from https://issuu.com/weissgallenkamp/docs/wt3_wk3_-e-

Walliman, N. (2011). Research Methods, The basics. Retrieved 16.8.2022 from https://books.google.fi/books?id=mSFJngEACAAJ&hl=fi&source=gbs_navlinks_s

Xia, Y., Zou, Z. Li, W., Zhang, X., Cui, S. & Liu, H. (2021). Research on Remote Operation Automatic Test of Relay Protection Based on MMS Communication Protocol Engine Technology. Retrieved 28.9.2022 from https://ieeexplore.ieee.org/document/9633833

Zhao, L., Li, Z., Ni, M. & Cheng, Y. (2019). Review and prospect of hidden failure: protection system and security and stability control system. Retrieved 13.12.2022 from https://link.springer.com/content/pdf/10.1007/s40565-015-0128-9.pdf?pdf=button%20sticky

Appendices

Appendix 1. Questionnaire

1. What would be the best method for collecting temperature measurements from the protection relay?
2. We are interested in obtaining system-level data. Considering OS, is the operating layer blocked? Is it possible to read e.g., random access memory (RAM) or central processing unit (CPU) usage or process-specific data?

3. What other requirements are needed to access data surfaces that can be collected from the device?
4. What kinds of communication functionality may be needed?
5. Do you have any recommendations regarding the protection functionality or measurement methods of the project?
6. Based on the data available directly from the application on a relay, what data could potentially be good to monitor in a long-term testing setup?
7. Do you have any suggestions regarding to simulating or generating error situations that could occur in the field? What kind of data tentatively might be suitable for finding out such error situations?
8. Do you have any suggestions or recommendations that should be considered immediately regarding the IEC61850 standard?

Appendix 2. Import data from Raspberry Pi

```python
class Sensor_DS18b20:
    def __init__(self, db_cfg='/longtermdb.txt'):
        os.system('modprobe w1-gpio')
        os.system('modprobe w1-therm')
        self.db_cfg = db_cfg
        self.base_dir = r'/sys/bus/w1/devices/28*'
        self.sensor_path = []
        self.sensor_name = []
        self.rows = []

    def _find_sensors(self):
        self.sensor_path = glob.glob(self.base_dir)
        self.sensor_name = [path.split('/')[-1]
            for path in self.sensor_path]

    def _parse_temperature(self, temp_str):
        i = temp_str.index('t=')
        if i != -1:
            t = temp_str[i+2:]
            temp_celsius = float(t)/1000.0
            temp_fahrenheit = temp_celsius * (9.0/5.0) + 32.0
        return temp_celsius, temp_fahrenheit

    def _read_temp(self):
        timestamp = datetime.datetime.now()
        for sensors, path in zip(self.sensor, self.sensor_path):
            # open the file which include sensors and read temp
            with open(path + '/w1_slave', 'r') as f:
                valid, temp = f.readlines()
            # check if the data is valid by comparing
            if 'YES' in valid:
                self.rows.append((timestamp, sensors) +
                    self._parse_temperature(temp))
if __name__ == '__main__':
    t_sensor = Sensor_DS18b20()
    while True:
        t_sensor._connect_to_db() #connection to PostgreSQL DB
        t_sensor._find_sensors()
        t_sensor._read_temp()
        sleep(30)
```

Appendix 3. Read data from DT80 and insert into pgAdmin4 table

```
class DT80Measurements(LibraryComponent):
    """Connect to the dataTaker dt80 using `host`and `port`
    Inherits LibraryComponent to expand LibraryComponent class.
    Used for Robot Framework.
    """
    def __init__(self, host='', port=):
        self.host = host
        self.port = int(port)
        self.data = ""
        self.values = ""
        self.valid = ""

    @keyword(tags=['Configure DT80 to CSV Mode'])
    def configure_dt80_to_csv_mode(self):
        """Configures DT80 to send CSV values
        """
        with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
            s.connect((self.host, self.port))
            s.send(b"/E/M/R\r\n")
            s.send(b"/n/c/u/t P22=44 P24=13 P33=0\r\n")
            s.send(b"/e/m/r\r\n")
            sleep(1)

    @keyword(tags=['Read DT80 Values'])
    def read_dt80_values(self) -> str:
        """Reading all measured data from dt80
        by utilizing DeTransfer commands:
        Connect to dt80: /E/M/R
        recv: reads all logged data from dt80
        Disable connecting to dt80: /e/m/r
        """
        self.configure_dt80_to_csv_mode()

        with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
            s.connect((self.host, self.port))
            s.send(b"/E/M/R\r\n")
            sleep(1)
            self.data = s.recv(1024).decode("utf-8")     # read data
            s.send(b"/e/m/r\r\n")                        # disconnect
        return str(self.data)

```

```python
    @keyword(tags=['Read DT80 Values Multiple Times'])
    def read_dt80_values_multiple_times(self, times) -> str:
        """Retries reading of dt80 values
        :param times, specify how many times to retry
        :type int

        :return dt80values if data is valid, else False
        :type string/boolean
        """
        for i in range(int(times)):
            self.values = self.read_dt80_values()
            logger.console(self.values)
            if self.is_dt80_values_valid(self.values):
                return self.values
            sleep(1)
        return False

    @keyword(types={'dt80_values':str},tags=['Is DT80 Values Valid'])
    def is_dt80_values_valid(self, dt80_values: str) -> str:
        """Checks that dt80 is returning values
        :param dt80_values: value read from DT80 device
        :type string

        :return True or False if matching criteria
        :type boolean
        """

        return False if len(dt80_values.split(",")) <= 1 \
               or "E/M" in dt80_values \
               or "DT80>" in dt80_values \
               else True

 @keyword(types={'dt80_values':str},tags=['Insert DT80 Values To DB'])
    def insert_dt80_values_to_db(self, dt80_values: str, db_cfg: str):
        """Read data from dt80 data logger and insert them into
        dt_data table in postgresql database

        :param dt80_values: comma separated
        value read from DT80 device
        :type dt80_values: [str]

        Connect to the database using a `db_cfg`
        to load the configuration.
        :param db_cfg:config file for connection data,defaults to None
        :type db_cfg: [str].
        """
        con = DBConnection(db_cfg=db_cfg) # parametrizing db_cfg
```

```
     d = dt80_values.split(",")

# Must be in utc format,Grafana applies local time to the timestamp
     timestamp = datetime.now(timezone.utc)

     with con as conn:
        cur = conn.cursor()
        cur.execute(f"INSERT INTO dt_data(date_time, temperature, \
            temperature1, temperature2, \
            temperature3, temperature4, temperature5, \
            temperature6, temperature7, \
            current, voltage, IRF, ambient) VALUES('{timestamp}', \
            '{d[8]}', '{d[9]}', \
            '{d[2]}', '{d[3]}', '{d[4]}', \
            '{d[5]}', '{d[6]}', '{d[7]}', '{d[0]}', \
            '{d[1]}', '{d[10]}', '{d[11]}')"
                     )
         conn.commit()  #updating and saving changes in PostgreSQL
```

Appendix 4. Connect Azure DevOps and log data

```
*** Settings ***
Library             datacollectionlibrary
host=${DEVICE.DT80_IP}         port=${DEVICE.DT80_PORT}

*** Test Cases ***
Read DT80 Data and Log into Database
    FOR    ${i}    IN RANGE    11
        ${DT80 data}=          Read DT80 Values Multiple
                               Times    times=5
        ${DT80 data valid}=    Is DT80 Values
        Valid               ${DT80 data}
        IF     ${DT80 data valid}
               Insert DT80 Values To DB        ${DT80 data}
               db_cfg=${PATH.USER_HOME}${/}longtermdb.cfg
        END
        Sleep    58s
    END

*** Keywords ***
```