ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ & ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

Διπλωματική Εργασία

ΜΕΛΕΤΗ ΚΑΙ ΑΝΑΠΤΥΞΗ ΠΡΩΤΟΤΥΠΟΥ ΣΥΣΤΗΜΑΤΟΣ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ ΛΕΙΤΟΥΡΓΙΚΩΝ ΠΑΡΑΜΕΤΡΩΝ ΓΙΑ ΔΙΑΚΟΠΤΕΣ ΥΨΗΛΗΣ ΤΑΣΗ ΜΕ ΧΡΗΣΗ ΤΕΧΝΟΛΟΓΙΩΝ IoT

Φοιτητής: Εμμανουήλ Δ. Φυλλαδιτάκης
ΑΜ: 18387321

Επιβλέπων Καθηγητής

Αντώνιος Μορώνης
Καθηγητής

ΑΘΗΝΑ-ΑΙΓΑΛΕΩ, Ιούνιος 2022

UNIVERSITY OF WEST ATTICA
FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

Diploma Thesis

DESIGN AND DEVELOPMENT OF A PROTOTYPE OPERATING PARAMETERS MONITORING SYSTEM FOR HIGH VOLTAGE SWITCHES USING IoT TECHNOLOGIES

Student: Emmanouil D. Fylladitakis
Registration Number: 18387321

Supervisor

Antonios Moronis
Professor

ATHENS-EGALEO, June 2022

Η Διπλωματική Εργασία έγινε αποδεκτή και βαθμολογήθηκε από την εξής τριμελή επιτροπή:

Α. Μορώνης, Καθηγητής	Κ. Ψωμόπουλος, Καθηγητής	Δ. Μετάφας, Επίκουρος Καθηγητής
 Digitally signed by Antonios Moronis Date: 2022.07.18 10:10:14 +03'00'	 Digitally signed by Konstantinos Psomopoulos Date: 2022.07.18 13:10:06 +03'00'	Dimitrios Metafas Digitally signed by Dimitrios Metafas Date: 2022.07.19 18:45:41 +03'00'

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο/η κάτωθι υπογεγραμμένος Εμμανουήλ Φυλλαδιτάκης του Δημητρίου με αριθμό μητρώου 18387321, φοιτητής/τρια του Πανεπιστημίου Δυτικής Αττικής της Σχολής ΜΗΧΑΝΙΚΩΝ του Τμήματος ΗΛΕΚΤΡΟΛΟΓΩΝ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ,

δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του διπλώματός μου.

Επιθυμώ την απαγόρευση πρόσβασης στο πλήρες κείμενο της εργασίας μου μέχρι τις 31/12/2024 και έπειτα από αίτησή μου στη Βιβλιοθήκη και έγκριση του επιβλέποντος/ουσας καθηγητή/ήτριας.»

Ο Δηλών

Ε. Φυλλαδιτάκης

Digitally signed by Antonios Moronis
Date: 2022.07.18 10:10:51 +03'00'

Αφιέρωση

Η διπλωματική αυτή αφιερώνεται στους γονείς μου, για τους οποίους η αγάπη και ευγνωμοσύνη που νιώθω αδυνατεί να αποτυπωθεί με λέξεις.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω των επιβλέπων Καθηγητή μου, Δρ. Μορώνη, για την υπομονετική καθοδήγησή του όλα αυτά τα χρόνια.

Επίσης, θα ήθελα να ευχαριστήσω τους προϊσταμένους μου στον ΑΔΜΗΕ / ΔΣΣΜ, κ. Νιφόρο Ν., κ. Διονυσόπουλο Δ., και κ. Κατεμλιάδη Σ., για την ανανταπόδοτη συμπαράστασή τους.

Περίληψη

Επί του παρόντος, σχεδόν όλοι οι διακόπτες υψηλής τάσης και τα συστήματα με μόνωση αερίου (GIS) είναι γεμάτα με SF6 υπό πίεση. Η ακούσια διαρροή του SF_6 έχει δραματικές επιπτώσεις τόσο στο περιβάλλον όσο και στον ίδιο τον εξοπλισμό. Μια μεγάλη διαρροή επιβαρύνει το περιβάλλον με εκατοντάδες τόνους ισοδύναμου CO_2 και αποτελεί σημαντικό κίνδυνο ασφάλειας για τον εξοπλισμό του δικτύου. Ως εκ τούτου, απαιτείται πρακτικά στενή παρακολούθηση του SF_6 σε συσκευές διανομής υψηλής τάσης προκειμένου να διασφαλιστεί η σταθερότητα του ηλεκτρικού δικτύου. Πολύ λίγοι υποσταθμοί διαθέτουν επί του παρόντος την υποδομή για την απομακρυσμένη παρακολούθηση των περιουσιακών τους στοιχείων και οι λύσεις μετασκευής για παλαιότερους υποσταθμούς είναι είτε μη οικονομικές ή και εντελώς ανέφικτες. Αυτή η εργασία επικεντρώνεται στην ανάπτυξη ενός νέου, ανοιχτού συστήματος παρακολούθησης που θα είναι φθηνό, ελάχιστα παρεμβατικό, και απείρως επεκτάσιμο, τόσο από άποψη υλικού όσο και λογισμικού. Το αναπτυγμένο σύστημα παρακολούθησης θα πρέπει να επιτρέπει στους διαχειριστές συστημάτων μεταφοράς να παρακολουθούν εξ αποστάσεως τις βασικές παραμέτρους των πιο σημαντικών περιουσιακών στοιχείων τους, κάτι που θα τους προσφέρει επίσης χρόνο αντίδρασης για την αποφυγή κρίσιμων βλαβών.

Λέξεις-κλειδιά

Διακόπτες Υψηλής Τάσης
Διακοπτικό Υλικό Υψηλής Τάσης
Εξαφθοριούχο θείο - SF_6
Διαδίκτυο των Πραγμάτων (IoT)
Μετάδοση δεδομένων LoRa
Παρακολούθηση Υποσταθμού

Abstract

Currently, almost all of the high voltage circuit breakers and gas insulated systems (GIS) are filled with SF_6 under pressure. The unintended leakage of SF_6 has dramatic impacts on both the environment and on the equipment itself. A large leak burdens the environment with hundreds of tones of CO_2 equivalent and poses a major safety risk for the grid equipment. Therefore, close monitoring of the SF_6 in high voltage switchgear is practically required in order to ensure the stability of the electric grid. Very few substations currently have the infrastructure for the remote monitoring of their assets and retrofit solutions for older substations are either uneconomic or entirely unfeasible. This work is focused on the development of a new, open monitoring system that will be inexpensive, minimally intrusive, and infinitely expandable in terms of both hardware and software. The developed monitoring system should allow transmission system operators to remotely monitor the essential values of their most important assets, which will also offer them a reaction time to avoid critical failures.

Keywords

High Voltage Circuit Breakers

High Voltage Switchgear

Sulfur hexafluoride - SF_6

IoT

LoRa data transmission

Substation Monitoring

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Introduction

Sulfur hexafluoride, commonly known as SF_6, is by far the most widely used in the electric industry as a gaseous dielectric medium. Currently, almost all of the high voltage circuit breakers and gas insulated systems (GIS) are filled with SF_6 under pressure. The unintended leakage of SF_6 has dramatic impacts on both the environment and on the equipment itself. A large leak burdens the environment with hundreds of tones of CO_2 equivalent and poses a major safety risk for the grid equipment - not only for the circuit breaker itself but linked equipment as well, meaning that monetary figures in case of a catastrophic failure can reach up to several million Euros. It is therefore understood that close monitoring of the SF_6 in high voltage switchgear is practically required in order to ensure the stability of the electric grid.

Aims & Objectives

Having acknowledged the issue, both the manufacturers of high voltage equipment and third parties have developed monitoring systems for high voltage circuit breakers. The disadvantages of these solutions currently are several, with the most prominent being:

1 Incompatibility between different manufacturers
2 Requirement to purchase expensive equipment and software packages
3 Intrusive installation
4 Complete dependance on their manufacturer

These make it exceptionally difficult, and costly, to retrofit SF_6 monitoring systems in substations without infrastructure for it.

This dissertation is focused on the development of a monitoring system that will:

1 Be compatible with virtually any SF_6 circuit breaker
2 Have a minimal equipment cost and be based on widely available parts
3 Will be based exclusively on open software
4 Can be installed with minimal interruption, or even no interruption at all, as well as being capable to operate even if there is no power source present
5 Offer practically infinite scalability, expandability, and hardware/software development possibilities

Methodology

Initially, an extensive literature and market review was performed. The review's objective was to identify the most suitable technologies and parts for use in an electric substation, then choose those

that offered the best balance between configurability, availability, and cost. Then, a pilot system was assembled and installed in an electric substation, allowing for the field test of the chosen parts. Once the pilot monitoring system proved capable of reliable operation, a web server and software was developed.

Innovation

The dissertation delivers a very low cost, minimally intrusive monitoring system for SF_6 circuit breakers that can be easily retrofitted to any substation, regardless of its location. The parts of the system are widely available and the final firmware/software are provided to allow for virtually immediate deployment. The system is developed so as to encourage further additions to its hardware and software, which can be easily performed depending on the equipment and the needs of the equipment's operator.

Structure

The structure of the dissertation is as follows:

1. A two-pronged literature review, with the first part focused on IoT and its current technological status, and the second part based on high voltage circuit breakers and SF_6.
2. The initial selection, assembly, and configuration of the pilot monitoring system
3. The employment of the pilot monitoring system, database handling, and advanced software development
4. Appendices containing the developed firmware and software code

Chapter 1 : Literature Review

1.1 IoT and LPWAN

1.1.1 What is IoT?

Nowadays, the number of electronic devices that are connected to the internet is continuously increasing. These devices are part of what we call the Internet of Things (IoT). The term IoT refers to the communication network of any object that integrates electronic means, software, and sensors, to enable an internet connection and exchange data [1, 2].

To support IoT device communication, we can choose from a variety of technologies. Wi-Fi is widespread but it requires a lot of power and transmits large amounts of data. Its most efficient protocol, IEEE 802.11ah or Low-Power Wi-Fi, is still not wholly suitable for low-cost, long range applications [3]. Bluetooth allows devices to communicate efficiently, but it has a very limited range [4, 5]. Zigbee's single channel limitation makes it unreliable for applications with multiple nodes and it requires the receivers to be constantly active, wasting energy [2, 6].

There are several network technologies designed primarily for IoT applications that require a wide area coverage and minimal power consumption. These technologies are known as Low Power Wide Area Networks (LPWAN) [7]. Some of these are the Wi-Fi HaLow™ [8], Weightless [9], Random Phase Multiple Access [10], and Long Range Wide Area Network (LoRaWAN) [11]. While some rely on the cellular spectrum to transmit data, in most cases these operate in unlicenced sub - 1 GHz spectrums.

The present work will use LoRa technology, a physical layer developed and patented by Semtech [12]. It is a wireless protocol specifically designed for long-distance, low-cost communications, with low power consumption. LoRa stands for "Long Range" and has been developed to transmit data over great distances while using very little power [13]. Albeit it has limited bandwidth, the very long range and the ability to penetrate indoor environments make LoRa exceptionally useful in cases were the goal is to transmit low volumes of data over great distances while using very low amounts of power [14]. LoRa and LoRaWAN have a proven track record on wireless sensor networks, which are the design focus of this project [12, 15].

1.1.2 LoRa Physical Layer

LoRa refers to the lower physical layer of the network. It is based on Chirp Spread Spectrum (CSS) modulation technology, producing a chirp signal where all chirps will have practically the same time

duration [16]. A chirp is characterized by a time profile of the instantaneous frequency that changes over the time interval T from a frequency f_0 to f_1.

LoRa defines two different types of chirps [11]. The first is the base chirp and its frequency time profile starts with the minimal frequency:

$$f_{min} = -\frac{BW}{2}$$

then ends with the maximal frequency:

$$f_{max} = \frac{BW}{2}$$

, where BW is the spreading bandwidth of the signal in Hz.

The complex conjugate of the base chirp that starts with frequency $f_1 = \frac{BW}{2}$ and ends with $f_2 = -\frac{BW}{2}$ is referred to as a down-chirp. For different digital inputs, a modulator will produce different chirps that will have certain time shift compared to base chirp.

The time shift of each chirp at the receiver side can be determined after the alignment of the time reference between receiver and transmitter by means of preamble detection. The time shift of each chirp is determined by multiplying the chirp itself with a down-chirp and finding the Fast Fourier Transform (FFT) of the multiplication. The maximum of the FFT output will indicate the time shift of the transmitted chirp that consequently will determine the transmitted digital symbol. If N is the chirp length of a symbol, then there can be N possible different cyclic shifts of the base chirp, and the value of the cyclic shift can be coded using log2N bits that gives the spreading factor (SF) of the communication.

To further improve the robustness against noise and burst interference, LoRa uses diagonal interleaving as well as forward error correction (FEC) codes with code rates from 4/5 to 4/8. The symbol rate and data rate depend on the spreading factor and the bandwidth used. The symbol rate is given by the formula:

$$R_s = SF\frac{BW}{2^{SF}}$$

where SF stands for spreading factor.

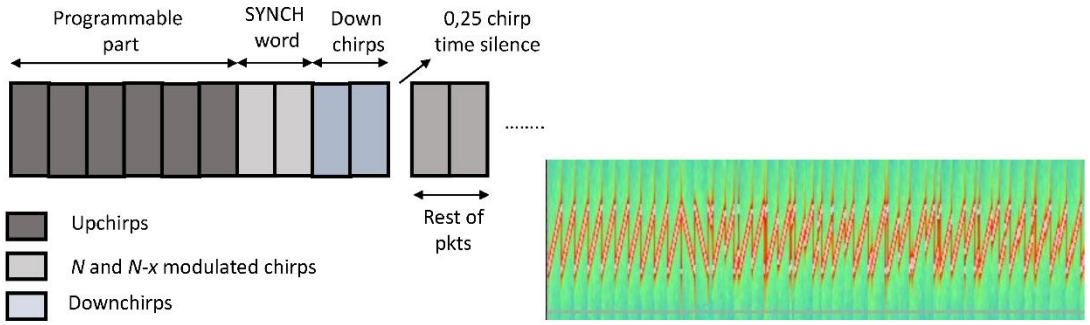
The data rate is given by:

$$R_b = SF \frac{\frac{4}{4+CR}}{\frac{2^{SF}}{BW}} 10^6 \ bps$$

where CR is the code rate.

The packet structure at the physical layer consists of a preamble, an optional header, and the payload. The preamble is used to synchronize the receiver with the transmitter and can have a length from 10 up to 65.536 symbols in total [17]. The fixed part of the preamble consists of four symbols, and the rest is programmable with a minimal length of six symbols and a maximal length of 65.532. The preamble starts with a sequence of constant upchirp symbols that is programmable and helps to detect the start of the frame. The programmable part is followed by two chirp symbols encoding the sync word that is used for frame synchronization. When the receiver detects three consecutive chirps, with the first one unmodulated (upchirp), the second one modulated with the first value, and the third one modulated with the opposite value, a new frame is compiled. The sync word can also be used to distinguish between devices from different networks by using different values for each network. Next to this, the preamble ends with two downchirp symbols that are used for frequency synchronization. After the last two symbols, a 0.25 symbol time represents a silence time used to let the receiver align the time. Optionally, the end of preamble can include another two unmodulated base chirps that will be used for fine time and frequency synchronization.

The structure of the preamble is shown in Figure 1a, while Figure 1b shows the spectrum capture of a LoRa packet where upchirps at the beginning of preamble are easily noticeable. Note that transmitter and receiver should know the SF in advance in order to detect the preamble, as preamble size scales with SF and there is no single preamble for all SFs.



(**a**) Preamble structure. (**b**) Spectrum capture of a LoRa packet.

Figure 1 - LoRa Preamble [11].

Optionally, the preamble is followed by a physical header. In such a case, the header contains the payload length in bytes, the FEC code rate of the payload and the header CRC. The header is always

protected with FEC with highest code rate of 4/8. If these three parameters are known in advance, the header can be removed completely. This decreases the time on air of the packet. In this case, the implicit header mechanism is applied, where the header parameters are fixed beforehand at both ends, the receiver and the transmitter side. The payload contains either LoRaWAN MAC layer control packets or data packets. Optionally, the payload can be followed by a payload CRC. The frame structure is shown in Figure 2.

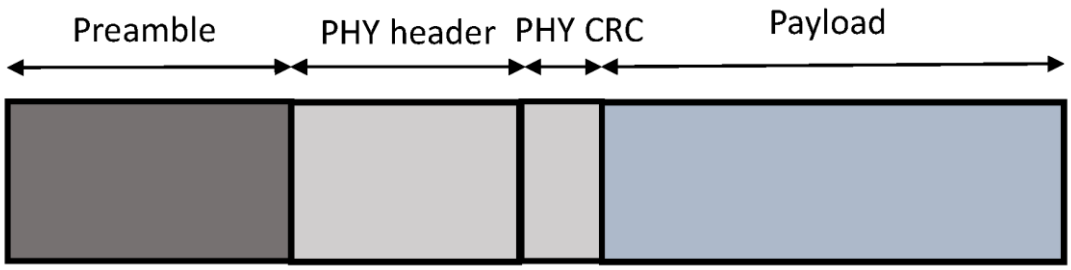


Figure 2 - LoRa frame structure [11].

The LoRa modulation mechanism has been patented and its implementation is never made public. The first mathematical description on how LoRa modulation works is presented in [16]. The authors described the LoRa modulation as Frequency Shift Chirp Modulation (FSCM) and give the mathematical model for it.

The LoRa physical layer operates in the 433, 868, or 915 MHz frequency bands. In Europe, only the 868 and 433 MHz bands can be used. There are eight 125 kHz channel in the 868 MHz band, three mandatory and five optional. In Europe due to transmission regulations, each transmission in any of the 868 MHz and 867 MHz sub-bands should comply with a 1% radio duty cycle, or implement a listen-before-talk or adaptive frequency agility mechanism [18]. When the duty cycle regulation is followed, it means that, if the radio transmitted for 1 s, it cannot transmit for the next 99 s.

1.1.3 LoRaWAN Network Layer

The LoRaWAN medium access control (MAC) protocol is an open-source protocol standardized by the LoRa Alliance [19] that runs on top of LoRa physical layer. The LoRaWAN network layer provides the medium access control (MAC) mechanism that enables communication between multiple devices and network gateway(s).

The LoRaWAN network architecture has a star topology where the end devices can only communicate with LoRaWAN gateways and not directly with each other. Multiple gateways are connected to a central network server. The LoRaWAN gateways are only responsible for forwarding raw data packets from end nodes towards the network server encapsulating them in UDP/IP packets.

When required, the network server can send downlink packets and MAC commands towards the end devices. The communication terminates at the application servers that can be owned by third parties. Multiple application layers can be connected to a single network server. The resulting LoRaWAN network architecture is shown in Figure 3.

The LoraWAN standard defines three classes of end devices: Class A, Class B, and Class C. Class A features devices are basic sets of options that every end device needs to implement in order to join a LoRaWAN network. To enable bidirectional communication, each uplink transmission of a Class A device is followed by two short downlink receive windows during which the end device will listen for possible downlink traffic. Consequently, the downlink communication is triggered by the end device, meaning that each downlink frame needs to wait for an uplink communication. The first and second downlink receive windows start 1 and 2 s, respectively, after the end of the uplink transmission. If the downlink transmission happens during the first window, then the same channel that is used for uplink is used for downlink too, while SF is determined based on the *RX1DROffset* parameter [20]. Class A end devices consume the least power since most of the time they are asleep.

To increase the downlink possibilities, Class B end devices will open additional receive windows at scheduled times. Gateways will transmit beacons in downlink for Class B end devices to get synchronized and for the network server to know when a certain end device will listen for downlink traffic. Class B devices consume higher power compared to Class A devices as they need to open more receive windows, even if those windows might not be used for downlink traffic at all. Class C devices will open continuous receive windows, practically being all the time available for downlink traffic except the time when they are transmitting [19].

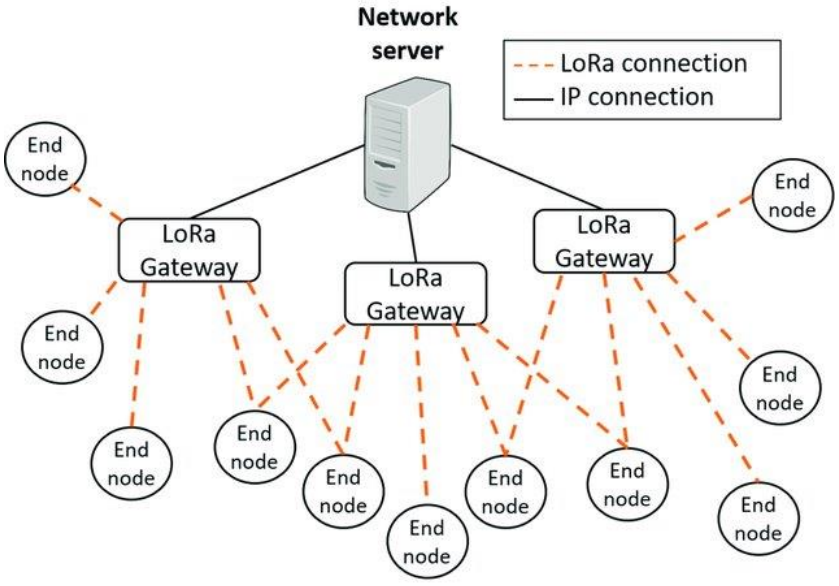


Figure 3 - The LoRaWAN network topology [21].

LoRaWAN protocol implement a number of mechanisms to ensure reliable and secure communication. The most highlighted mechanism is the adaptive data rate (ADR) mechanism, which is built into LoRaWAN to dynamically manage a node's link parameters. The ADR mechanism manages the data rate and transmission power of end devices, allowing it to increase the packet delivery ratio and maximize the efficiency of the network. Devices that support ADR can notify the gateway that they allow for transmission parameter modifications. Then, the gateway or the network server can manage the transmission parameters of the end devices, when required. Some end devices implement ADR mechanisms as well, managing their transmission power and data rate on their own, without or in synergy with the gateway and/or the network server. The latest LoRaWAN standard also allows compliant end devices to increase their transmit power in order to achieve connectivity. Failing that, then the devices will start decreasing their data transmission rates until connectivity has been achieved [22].

The LoRaWAN standard specifies how an end node can join the network. Each end device that joins the network needs to be personalized and activated. End devices can be activated over the air (OTAA) or by personalization (ABP). Several keys and identifiers are needed for the joining procedure, as well as for all communication during one session. All the required keys and identifiers are shown in Table 1.

Table 1 - LoRaWAN keys [11].

Key	Description	Required in Joining Type		Generated from or Stored Beforehand
		OTAA	ABP	
LoRaWAN v1.1				
Keys needed before activation				
NwkKey	Is used to calculate MIC for join-request packets, encrypt join-accept packets, and derive all NTW session keys.	Yes	No	Stored beforehand
AppKey	Is used to derive AppSKey	Yes	No	Stored beforehand
JSIntKey	Is used for MIC of rejoin-request and join accept packets	Yes	No	Generated from NwkKey and DevEUI
JSEncKey	Is used to encrypt join-accept triggered by rejoin-request	Yes	No	Generated from NwkKey and DevEUI
Keys needed after activation				
FNwkSIntKey	Is used for calculate MIC of part of it of all uplink data packets	Yes	Yes	Generated from NwkKey and join-accept message
SNwkSIntKey	Is used to verify MIC of all downlink data packets and calculate part of MIC of uplink packets	Yes	Yes	Generated from NwkKey and join-accept message
NwkSEncKey	Is used to encrypt all downlink and uplink MAC packets	Yes	Yes	Generated from NwkKey and join-accept message
AppSKey	Is used to encrypt/decrypt payload of data packets	Yes	Yes	Generated from AppKey and join-accept message
Identifiers				
JoinEUI	64-bit globally unique application ID that identifies the join server	Yes	No	Stored beforehand
DevEUI	64-bit globally unique device ID by the network server	Yes	No	Stored beforehand
DevAddr	32-bit unique device address in the current network	Yes	Yes	Received by join-accept message
LoRaWAN v1.0				
Keys needed before activation				
AppKey	Is used to derive AppSKey and NwkSKey and to calculate MIC for join-request message.	Yes	No	Stored beforehand

When using the OTAA procedure, an end device is personalized with keys and identifiers that are saved before the procedure starts, whereas with ABP all the session keys are saved beforehand and there is no need for an activation process that will require the exchange of join-request and join-accept packets.

1.2 High Voltage Switchgear & Equipment Monitoring

1.2.1 High Voltage Circuit Breakers

High voltage circuit breakers are a critical part of every electric substation, playing a vital role on safeguarding the stability and reliability of electrical systems [23]. High voltage circuit breakers operate similarly to common electrical switches - when they are operated, the contacts inside the extinguishing chamber open and the electrical circuit breaks. The only major difference is that an extinguishing medium is present inside the chamber, necessary to cope with the very high energy of the high voltage electrical arc [24].

A high voltage circuit breaker consists of the following basic parts [25]:

1. The interrupting chamber. The chamber is filled with an insulative medium in order to achieve proper electrical discontinuity between the contacts when they are open, as well as between the live parts of the breaker and its outer shell. The insulative medium is also expected to effectively quench the electric arc that draws during operations. Modern chamber designs use the properties of the insulative gas to aid the driving mechanisms.
2. The contacts. Both primary and, if applicable, the auxiliary contacts are present inside the interrupting chamber. The contacts are meant to conduct the rated current of the circuit breaker without overheating and sustain fault currents for limited periods of time. They are also designed so as to withstand the stress and critical temperatures caused by the electric arc. Still, contaminant build-up generated by the electric arcs can reduce the electrical characteristics of the contacts and eventually destroy them. This is the primary reason why all circuit breakers have a finite number of operations before requiring a complete overhaul, which also depends on the current they need to interrupt [26].
3. The driving mechanism. The driving mechanism is expected to quickly and securely move the contacts during operations. It usually is based on some form of mechanical system, such as a powerful spring, hydraulic piston, or compressed air. Modern driving mechanism designs are unfussy, with the minimal possible amount of moving parts, ensuring long-term reliability and ease of installation.

4. The control panel. The control panel can usually be found on the scaffold supporting the circuit breaker itself, or nearby. The control panel frequently includes the driving mechanism and every electrical contact/device associated with the safe use of the high voltage circuit breaker. Some of the items that can be usually found inside the control panel, except from the driving mechanism, are:
    a. Instrumentation regarding the pressure and/or density of the insulating gas
    b. Control, safety, and alarm relays
    c. A mechanical operations counter
    d. Anti-condensation heaters and a thermostat
    e. Manual control parts (buttons, switches, etc.)

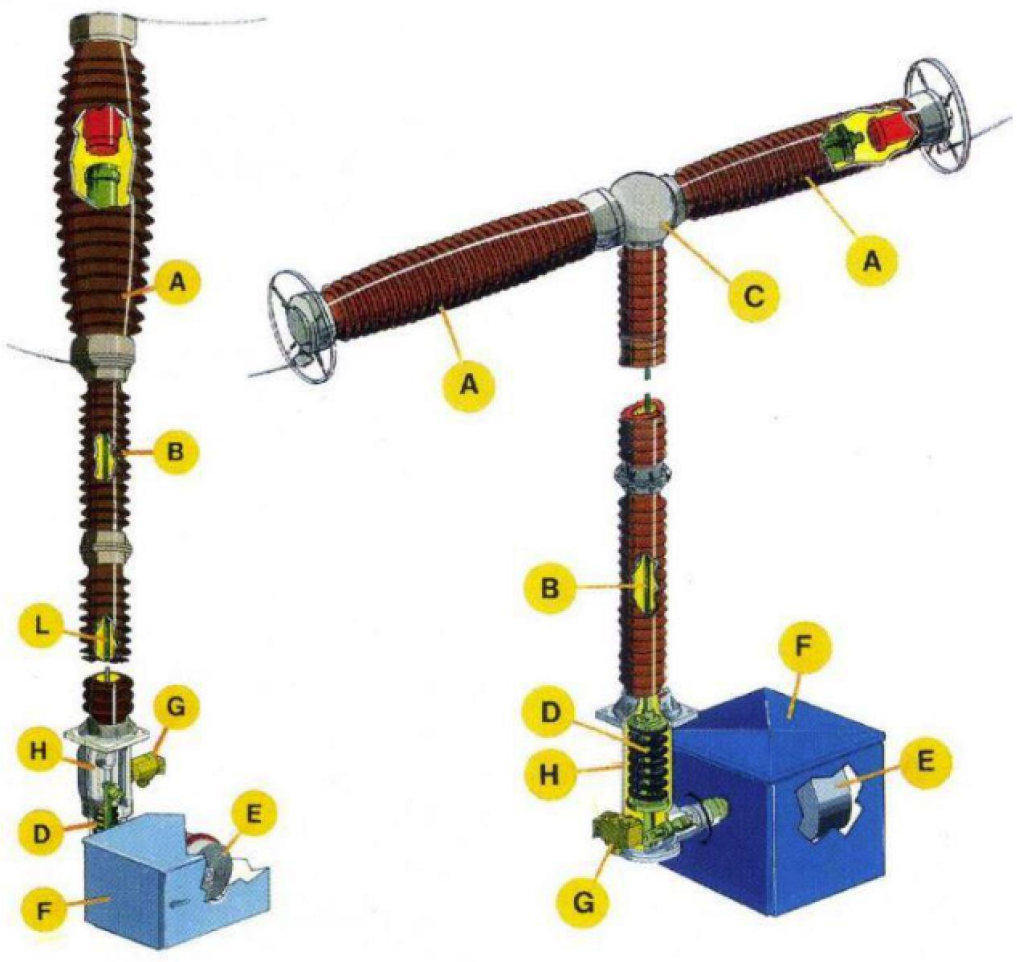


Figure 4 - High Voltage Circuit Breaker Parts:
(A) - Interrupting Chamber & Contacts, (B) - External Insulator
(C) - Upper Driving Mechanism, (D) - Lower Driving Mechanism
(E) - Mechanical Driver, (F) - Control Panel, (G) - Gas Density Supervisor
(H) - Lower Chamber & Filling Port, (L) - Contacts Driving Rod

1.2.2 Sulphur hexafluoride (SF6)

In their majority, high voltage circuit breakers today use sulphur hexafluoride (SF_6) as a medium of quenching and insulation [27]. SF_6 is a very heavy gas, with a molecular weight of 146.06 g/mol and a standard density of 6.07 g/L [28]. This makes SF_6 five times heavier than air and will amass at lower heights in sealed spaces, requiring proper evacuation measures wherever SF_6 is used indoors to avoid human inhalation in case of a leak. The dielectric strength of SF_6 is three times greater than that of dry air and tends to significantly improve at higher pressures. In the following figure, the dielectric strength of SF_6 is being compared to that of atmospheric air, as well as to itself at higher pressure [29].

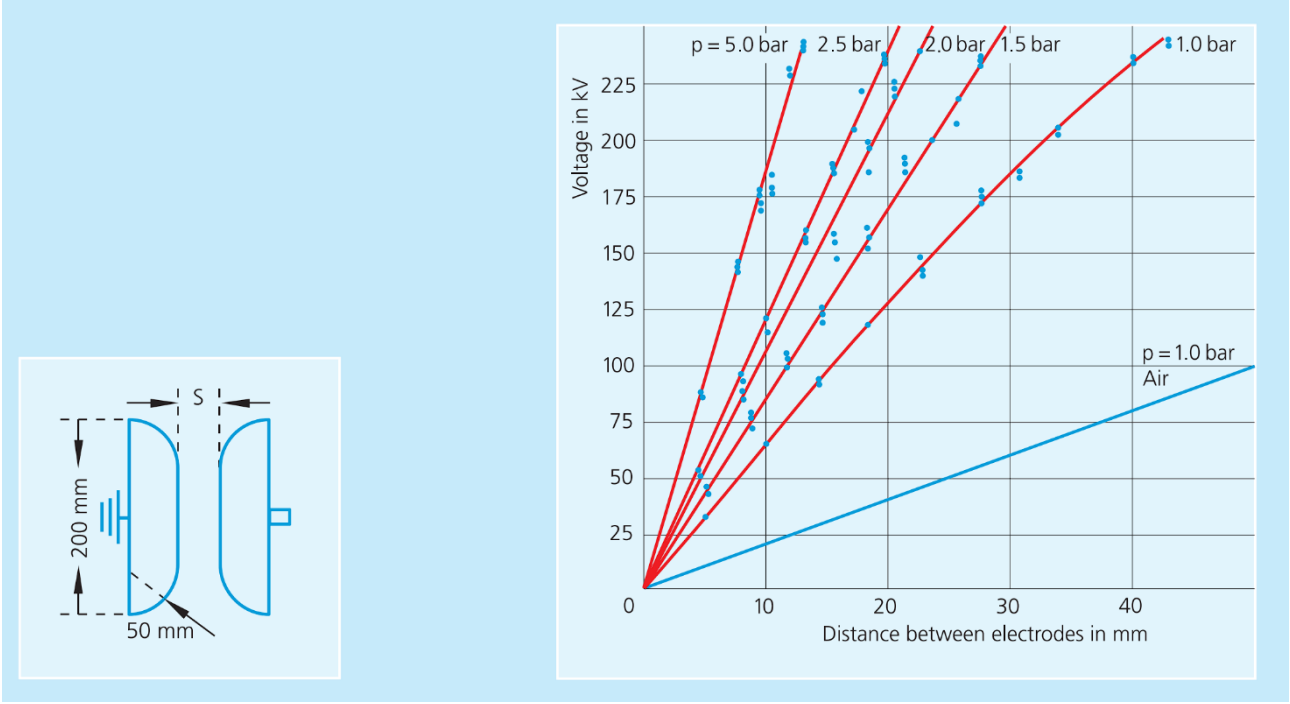


Figure 5 - Breakdown Voltage of SF_6 using 50 Hz and 200 mm mushroom electrodes (homogeneous field).

SF_6 is an electronegative gas, tending to easily capture free electrons and forming heavy, cumbersome negative ions. Therefore, it is extremely difficult to ionize SF_6, as the gas itself does not allow a considerable number of electrons to remain free.

The thermal conductivity of SF_6 is not always better than that of air, as it depends on temperature. However, it is significantly better than that of air at arcing temperatures, which typically are at around 2000 °C, resulting to exceptional arc quenching performance [30-32]. This is caused by the chemical disassociation of the gas, which absorbs considerable thermal energy. When the gas reassembles at the edge of the arcing region, this thermal energy is released, effectively transferred away from the

core of the arc. The process causes the centre of the arc to remain cooler, resulting to the faster quenching of the arc.

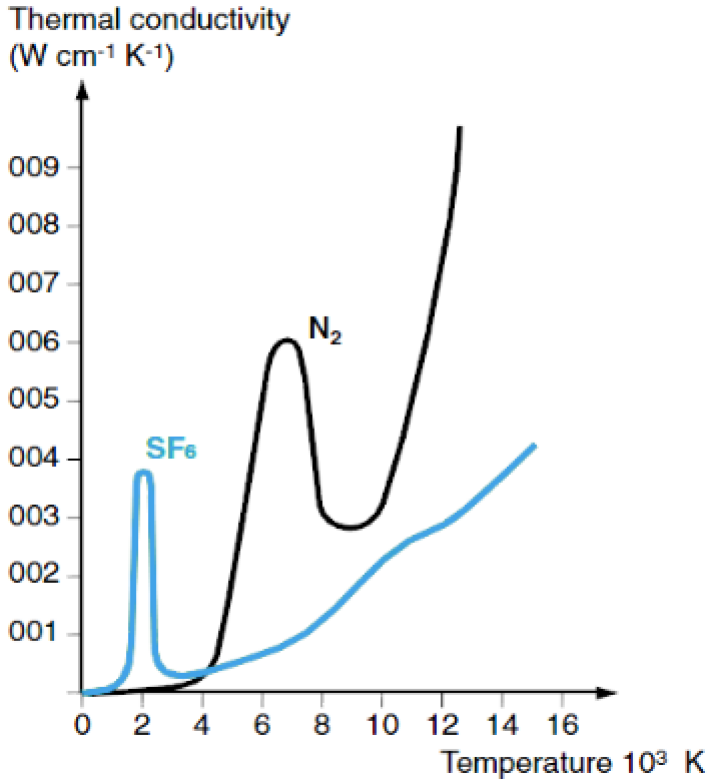


Figure 6 - Thermal conductivity of SF_6 and N_2 in relation to temperature.

Furthermore, SF_6 has a volumetric specific heat capacity 3.7 times greater than air as well [33]. This allows the gas to absorb greater amounts of thermal energy without substantial increase of its temperature. This characteristic allows the SF_6 to generally maintain lower temperatures during electric arcs and further improves its arc quenching capabilities.

In summary, the beforementioned characteristics of the SF_6 gas result to an arc quenching capacity that is about a hundred times faster and equally more efficient than that of atmospheric air [34].

The only practical disadvantage of SF_6 is that it can easily liquify if the pressure is high, even at ambient temperatures. It is known that higher pressures increase the dielectric strength of insulative mediums and is a common practice with gaseous insulators, employed by every SF_6 circuit breaker manufacturer to decrease the proportions of the devices. The following vapor pressure curve displays the state of SF_6 depending on the temperature, density, and pressure.

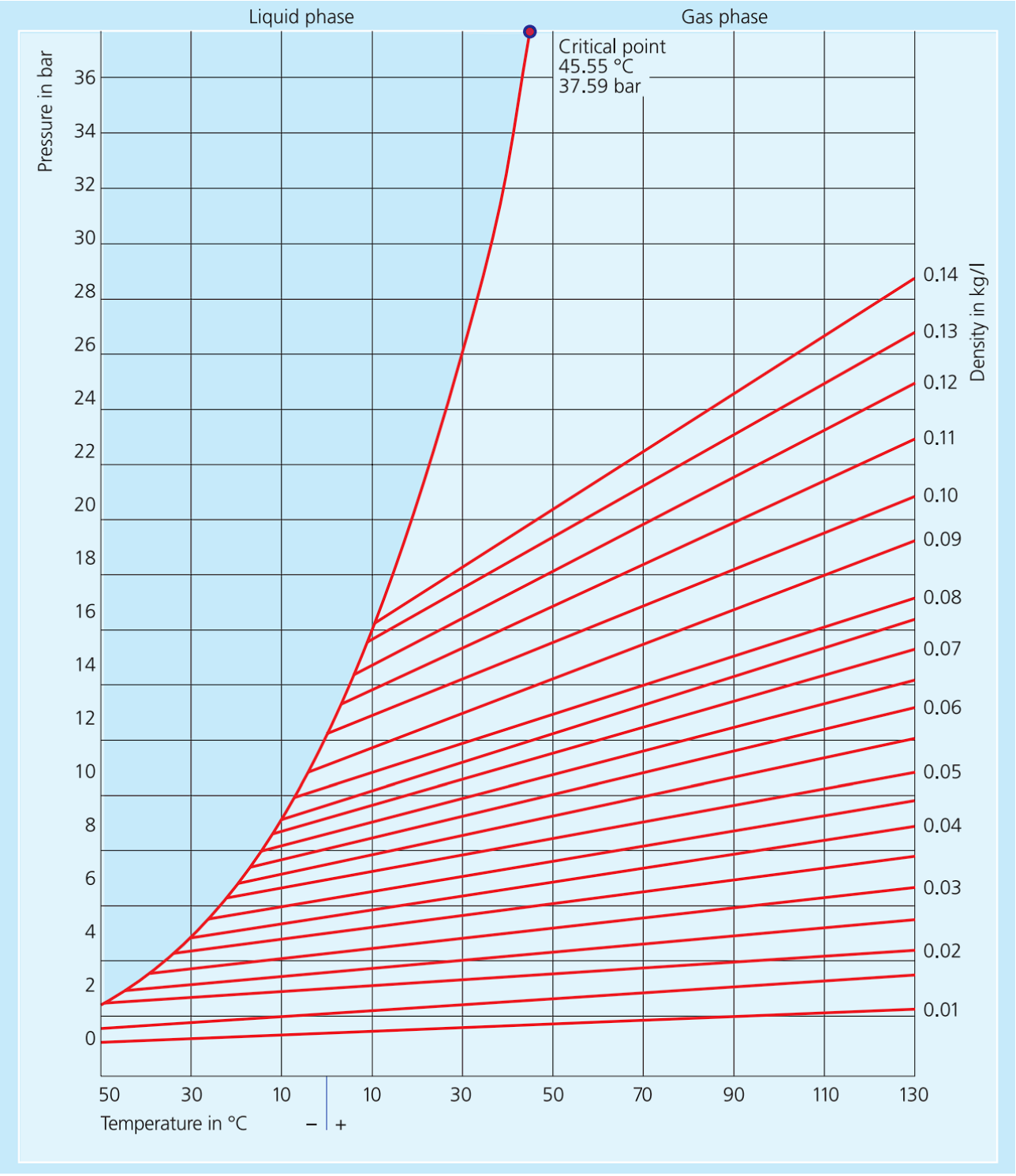


Figure 7 - SF_6 vapour pressure curve [35].

With a working pressure of 7.2 bar, a common working pressure in high voltage circuit breakers, contaminant-free SF_6 will liquify at about -26°C, making it dangerous to use it directly into cold areas and climates. If the pressure is increased to 10 bar, the gas can be liquified at temperatures higher than -10°C. In order to combat that, manufacturers need to develop equipment that can operate at lower SF_6 pressures, or mix SF_6 with other gases, such as N_2 or He [36, 37].

1.2.3 SF_6 Alternatives

SF_6 currently has unrivalled insulation and arc extinguishing properties but it is also included in the greenhouse gas panel with an extremely large carbon equivalent footprint. One kg of SF_6 equals to 22.800 kg CO_2 according to the EU Regulation 517/2014 [38], with researchers advocating that it can be up to 23.900 kg CO_2 [39], and has a lifetime of 3.200 years. Even though scientists are actively researching alternatives, SF_6 currently remains unrivalled for use in high voltage switchgear [40-45].

With the SF_6 losses increasing linearly every year and due to its massive environmental impact, significant research and funding is taking place in search for alternatives [46, 47]. CO_2 itself was explored as a viable alternative to SF_6, mixed with O_2 to improve the electrical properties of the gas [48, 49]. Although the dielectric strength of CO_2 is relatively low, presenting technical difficulties regarding the size and/or working pressure of equipment, CO_2 has very low global warming potential, is chemically very stable, and safe for both humans and the environment. ABB had working prototypes of 145 kV CO_2 circuit breakers available in 2018 that, albeit their increased size, showed promising results and the company was working towards the development of a 420 kV CO_2 circuit breaker [50]. With Hitachi's acquisition of ABB's Power Grids in 2021 and the almost immediate agreement of the company with GE's Grid Solutions for the use of GE's proprietary insulative mixture, it would seem that the development of 420 kV CO_2 circuit breakers is now unlikely.

With other gases proving insufficient for use in high voltage switchgear, mixtures based on SF_6 itself were explored as an alternative for lower greenhouse gas emissions. The most promising mixture was that of SF_6 with Nitrogen. A mixture of 20% SF_6 with 80% N_2 per volume retains about 80% of the dielectric strength that pure SF_6 has and also decreases the liquification temperature by 10°C, making it ideal for use in very cold climates [37, 51, 52]. Still, the global warming potential of the mixture remains unacceptably high. One kg of a 10%-90% SF_6 / N_2 mixture is equivalent to 8.650 kg CO_2, all while the dielectric properties of the mixture fall below 60% those of pure SF_6. Therefore, it was concluded that SF_6 / N_2 mixtures cannot be a permanent solution to the issue. The use of fluorinated gases (F-Gases) met the same end, as they did little to reduce greenhouse gas emissions while they had reduced electrical performance and proved to be a health hazard [44, 53-55].

Although it is still facing technical difficulties, the use of fluoroketones and fluoroketone-based mixtures is currently proving to be the most promising alternative to SF_6. The two most prominent mixtures are the Novec 4710 and the Novec 5110, developed by GE Grid Solutions and ABB respectively in collaboration with 3M [43, 44]. The primary difference between the two is that the former is using CO_2 and the latter Nitrogen as the buffer gas. GE Grid Solutions currently markets all relevant mixtures under the commercial name g^3 or g-cubed, including Nitrogen-based mixtures

after the agreement they signed with Hitachi ABB Power Grids in 2021. The g^3 is supplied premixed and any alteration of the initial mixture is dangerous, as even slight composition changes can result to the formation of graphite, with catastrophic results for the equipment [56]. g^3 has a global warming potential of 2.100, although GE Grid Solutions suggests that it reduces the greenhouse gas emissions by 99.95% compared to SF_6, because the CO_2 or O_2 evacuate first in case of a minor leak [57, 58]. There currently is no g^3-based equipment available that can operate at voltage levels greater than 170 kV but GE Grid Solutions has undertaken an EU-funded project to deliver 420 kV g^3-based circuit breakers, both GIS and AIS, by 2025.

1.2.4 Switchgear online monitoring systems

An SF_6 circuit breaker has an operational lifetime of at least twenty years, with equipment commonly staying in service for up to 40 years [23]. According to a CIGRE study, the most common defect that circuit breakers are susceptible to SF_6 gas leaks, which accounts for at least 40% of all minor failures [23]. Possible causes of SF_6 leaks are usually caused by minor items, such as hardened gaskets, corrosion, and O-ring deterioration [59]. SF_6 leaks are the main limiter to a high voltage circuit breakers lifetime and account for the majority of failures on equipment that is less than thirty years old, with issues sometimes observed mere months after the equipment's installation [60]. It is crucial to continuously monitor the pressure of SF_6, both for operational and environmental reasons, as SF_6 is included in the greenhouse gas panel with an extremely large carbon equivalent footprint (1 kg SF_6 = 23.900 kg CO_2) [39].

Online monitoring of medium and high voltage equipment entered practical applications as early as 1980 [61]. However, the rapid development of sensor and computer technologies in the new millennium, as well as the need to switch from time-based to condition-based maintenance, was what sparked the research and application interest on online monitoring [62].

Research on real-time monitoring systems for high voltage switchgear was extensive during 2006 to 2010, especially in China, due to the exponentially rising electricity demand in the country at the time [63-69]. However, there are very few publications after 2010, when KONČAR [70] and ABB [71, 72], which is now owned by Hitachi, introduced their commercial online monitoring systems. Other companies, like Grid Solutions by General Electric and Alstom, moved directly towards completely digital substations [73].

Chapter 2 : Remote Monitoring System Development

2.1 Introduction

KONČAR's Bay Monitoring System (BMS) and ABB's Modular Switchgear Monitoring (MSM) offer a multitude of real-time readings but they rely on wired connections and it is closed source software and hardware. As such, retrofitting aged substations with them can be an intrusive and expensive endeavour. GE's digital substations require specific equipment and site configurations, meaning that it is either impossible or prohibitively expensive to retrofit an aged substation to become a digital substation.

According to the above, there is a need for a system of continuous monitoring of low cost and easy adaptation to existing facilities, compatible with any type of equipment and using open-source software.

The focus of this work is to design a very low cost, open source, wireless monitoring system for high voltage switchgear. The system will be designed specifically for minimally intrusive installation, allowing the seamless and inexpensive installation in aged substations, allowing the basic monitoring of standard medium and high voltage switchgear. It will be monitoring the temperature and pressure of the SF_6 gas, as well as number of operations. These three physical figures are amongst the most vital data that can be acquired from high-voltage switchgear without getting the equipment out of service, with the sole exception being the installation of continuous gas quality sensors [74].

The implementation of the proposed system should significantly improve the ability to detect SF_6 leaks in time, before they lead to forced opening (trip) of the switch due to reduced SF_6 pressure. Furthermore, temperature monitoring can reveal issues before they reach catastrophic results. As a result, the availability and safety of the switching equipment will be improved, while the environmental footprint will be significantly improved via the reduced gas emissions.

2.2 Transmitter & Sensor Selection

Arduino Mega 2560 boards were selected to be the foundation of the transmitter nodes. Arduino boards are inexpensive, very flexible, widely available, and open-source, aligning with the requirements of this project [75]. Arduino has been the basis for data acquisition devices and sensor data forwarding for myriads of research and development projects [76-81], including electrical grids [82-84].

The Arduino Mega 2560 board has 54 digital input/output pins, 16 analogue input pins, four hardware serial ports (UART), and an USB connector. It can operate with an input voltage of 6 V to 20 V DC, allowing it to stay operative when the power source is not fully stable, as well as with 5 VDC via its USB port. The board can be seen in Figure 6.

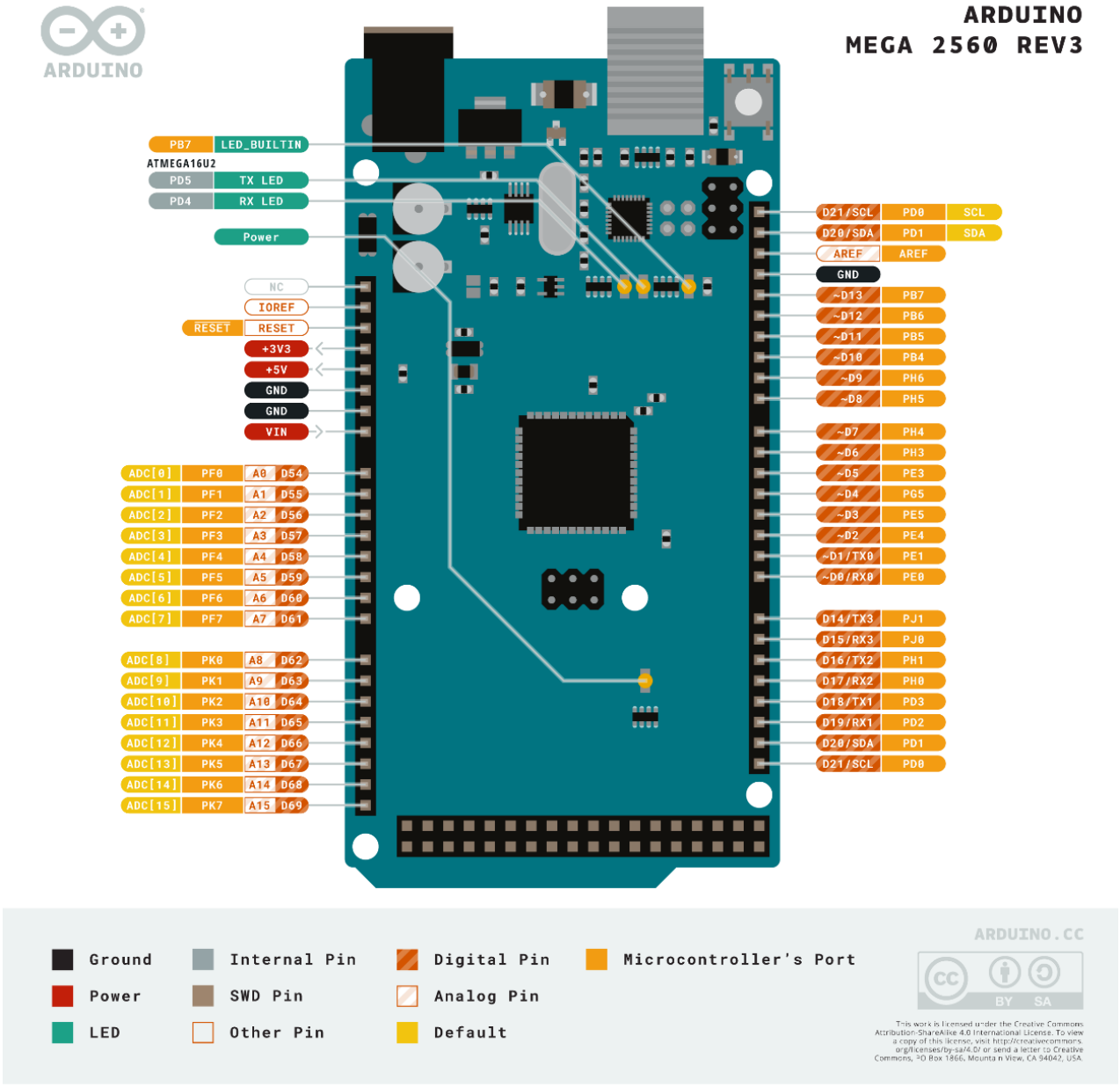


Figure 8 - Arduino Mega 2560 V3 Pinout Diagram

For communication, the SX127x Shield featuring LoRa technology has been selected due to its open-source library, low cost, and very low energy consumption. It is based on the Semtech SX1278 chip and designed for low data rates and very long transmission range, with the manufacturer claiming that it can reach a transmission range of up to 15 kms [85]. The LoRa transmission frequency is at

868 MHz, an unlicensed band that is virtually unaffected by the strong electromagnetic fields that are present inside electrical substations. European regulations limit only the duty cycle per sub-band when "listen before talk" mechanisms are applied [18], with the 1% duty cycle being ample in our application where only a few bytes are transmitted per transmission cycle. Studies has shown that the LoRa protocol is effective in noisy industrial environments, allowing the coverage of large sites with a single gateway present [86]. For the means of this project, which requires the transmission of just a few bytes of data reliably over a distance of up to 1.000 meters in an electrically noisy environment, the SX127x Shield is a dependable and cost-effective choice.

Figure 9 - Arduino Mega 2560 with SX127x Shield

The two most basic and important physical figures that require monitoring on an SF_6 circuit breaker is the pressure and temperature of the SF_6 gas. Monitoring the temperature of the SF_6 gas not only can be used as an early warning for anomalies that cause the circuit breaker to overheat, but temperature data can also be used to normalize the SF_6 gas pressure readings [87]. To that end, a temperature and a pressure sensor are implemented.

The temperature sensor is a DS18B20 digital thermometer with an included microprocessor and a 12-bit Celsius temperature measurement. It is effective between -55°C and +125°C, with a ±0.5°C accuracy from -10°C to +85°C, making it ideal for our application, as high voltage circuit breakers usually operate relatively close to the environment ambient temperature and temperatures above +80°C usually result to catastrophic events. The DS18B20 is ideal for the monitoring of thermally sensitive systems [88].

The pressure transducer is an analog 5 V DC transducer (SKU237545) with a working pressure range of 0 to 12 Bar (0.5 V - 4.5 V DC). It has been selected due to the 5 V DC output pins present on the Arduino Mega 2560 board and its ideal application range, as most high voltage circuit breakers have a working pressure of 5.5 to 8 Bar. The transducer is linear and has a full-scale error of only ±1.5 %, making it easy to calibrate.

One other significant factor that greatly affects the lifetime of high voltage circuit breakers is the number of operations [89, 90]. Typical high-voltage circuit breakers may last for up to 10.000 operations [91] but their actual lifetime depends on several factors, with the greatest being the amplitude and nature of the current they are required to interrupt [26].

In order to monitor the number of operations for each circuit breaker that we install a transmitter to, we implement two simple switches on each transmitter. One switch is mounted so as to be triggered by the mechanical parts of the circuit breaker with each operation and the other switch acts as a simple manual reset switch.

For demonstration purposes, an 1.8" TFT display has been added on the experimental transmitters, showing the temperature, pressure, and operations counter of the switch in real time. The display is inexpensive and could be useful to maintenance personnel visiting the switch but also adds a very significant constant power draw to the device, making its implementation impractical in cases where the transmitters are designed to operate completely autonomously.

Figure 10 - Arduino 1.8" TFT

Figure 11 - 1.8" TFT Module Application Example

With the communication shield requiring nine digital pins, the temperature sensor requiring one digital pin, the pressure sensor requiring one analogue input pin, the counter switches requiring two digital pins, and the LCD requiring five digital pins, the Arduino Mega 2560 still has 37 digital and 15 analogue input pins unoccupied, offering ample headroom for customization and future upgrades.

2.3 Gateway Selection

For our application, we selected the RAK7258 WisGate Edge Lite indoor gateway, which is using the MediaTek MT7628 SoC and the Semtech SX1301 Mini PCIe LoRa card. It has been selected as an inexpensive, widely available gateway with extensive implementation options and remote configuration capabilities, ideal for when data collection needs to be reliable without extensively increasing the cost [92]. The implementation of lower cost gateways, such as the Dragino LG01-P, has been explored but their single-channel capabilities and limited integration options suggest that they were not usable for a large-scale application [93].

The RAK7258 gateway has eight full LoRa channels, allowing the grouping of devices per channel in case numerous devices are present, greatly reducing transmission overlapping and flooding [94]. It can transmit data to most LoRaWAN networks but can also function as a basic stand-alone network

server, transmitting either MQTT or HTTP packets. From the IP connection side, the RAK7258 gateway can connect to local networks via LAN and Wi-Fi, or via LTE if the version with the optional Quectel EG95-E LTE module has been purchased. The LTE Cat 4 connection is critical for sites that are remote and have no landline internet connection.

2.4 Software

In order to use the data transmitted by the LoRa gateway, that data must be forwarded to a network server, which can be present either locally or remotely. When LoRa transmitters and gateways are used, the vast majority of real-world applications has them forwarding the data to LoRaWAN servers. LoRaWAN gateways operate entirely at the physical layer and essentially merely forward LoRa radio messages after checking the data integrity of each incoming packet. If the integrity of the packet fails the CRC check, the message will be dropped. If the CRC check succeeds, the gateway will forward it the packet to the server, alongside with some metadata, such as a timestamp [95].

LoRaWAN network servers manage the network and are responsible for the flow of traffic between the transmitters/end devices and the server but cannot see or access the application data, which is encrypted. The data are handled by the application server, which interprets and manages all the data. Application servers can be linked to databases, dashboards, and/or data portals. There are numerous LoRaWAN networks, open and commercial, with the most popular currently being the cloud-based TTN (The Things Network) [96] and the open-source Chirpstack project [97].

For the means of this project, proprietary design was chosen over reliance on commercial or all-in-a-pack solutions. The gateway has been configured to operate as a network server on its own and to transmit HTTP packets to a newly developed website. The website is developed to be hosted by a Web Hosting Service, ensuring practically zero downtime with a very low annual cost.

Chapter 3 : Installation & Data Acquisition

3.1 Transmitter Assembly & Configuration

For the means of this study, we will be using two transmitters attached to two 170 kV circuit breakers in a substation located in Athens, Greece. Both circuit breakers are made by Alstom (GL313) and have a working SF_6 pressure of six Bar.

The transmitters are attached inside the circuit breaker panels. To achieve that, a custom enclosure was designed and printed for the Arduino Mega 2560 and the LoRa communication shield, allowing the installation of the transmitter on the spare DIN rail inside the panel. The enclosure of the Arduino Mega 2560 and the LoRa shield is shown in the following figures.

Figure 12 - 3D Printed Enclosure

With the transmitter inside the circuit breaker's panel, it is possible to power it via either the 230 VAC or the 110 VDC circuits present there. The power requirements are extremely low, therefore a small DIN rail switching PSU or DC-DC converter would suffice. However, in order to avoid interference with the existing circuitry and ensuring complete autonomy, we opted for a stand-alone photovoltaic power system. A 20 W_p photovoltaic panel was installed beneath the circuit breaker's insulator and a Solar Power Management Module with a 3.7 V 14500 battery was installed inside the panel.

The Solar Power Management Module is developed by WaveShare specifically for similar IoT applications [98]. It has a regulated 5V/1A output and our testing revealed that it can easily sustain the power requirements of the transmitter when the LCD is not attached. The total cost of the components required for the stand-alone PV power source is lower than the cost of a quality DIN rain 12V PSU but they increase the installation time and periodical maintenance should be considered.

Figure 13 - PV Panel Installation

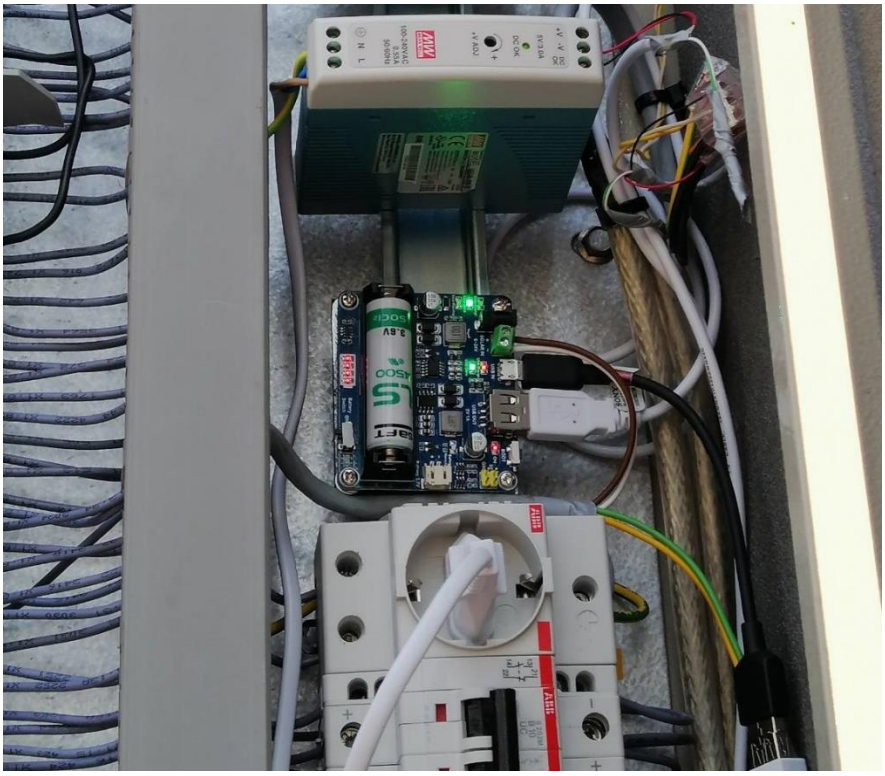


Figure 14 - The WaveShare Solar Power Management Module

The attachment of the sensors to the circuit breaker unobtrusively is of vital importance if any monitoring system is to be retrofitted to electrical substations. To achieve that, both sensors were

attached to the gas filling port of the circuit breaker using a custom milled adapter. The adapter simply attaches to the DN20 port of the circuit breaker and has the pressure transducer attached, with the temperature sensor in a pocket right next to it. The DN20 adapter can be seen in the following figure.

Figure 15 - DN20 to Transducer adapter

Our initial version of the adapter was developed for testing purposes only and blocks the filling port of the circuit breaker, forcing its removal should the circuit breaker needs to undergo maintenance. However, it is simple to manufacture the adapter so as to not block the filling port, as is simple to manufacture adapters for any type of gas filling port.

The Arduino Mega 2560 with the SX127x Shield transmitters are flashed with a customized sketch, an example of which can be found in Appendix A. For our tests, the transmission interval was set to sixty seconds and the spreading factor was set to 12. The high spreading factor severely limits transmission speeds but boosts reliability. During our field testing, the gateway dropped just two out of 12.310 packets. Both transmitters were set at 868.1 MHz and their transmissions were seemingly unaffected by the strong electromagnetic fields inside the electric substation.

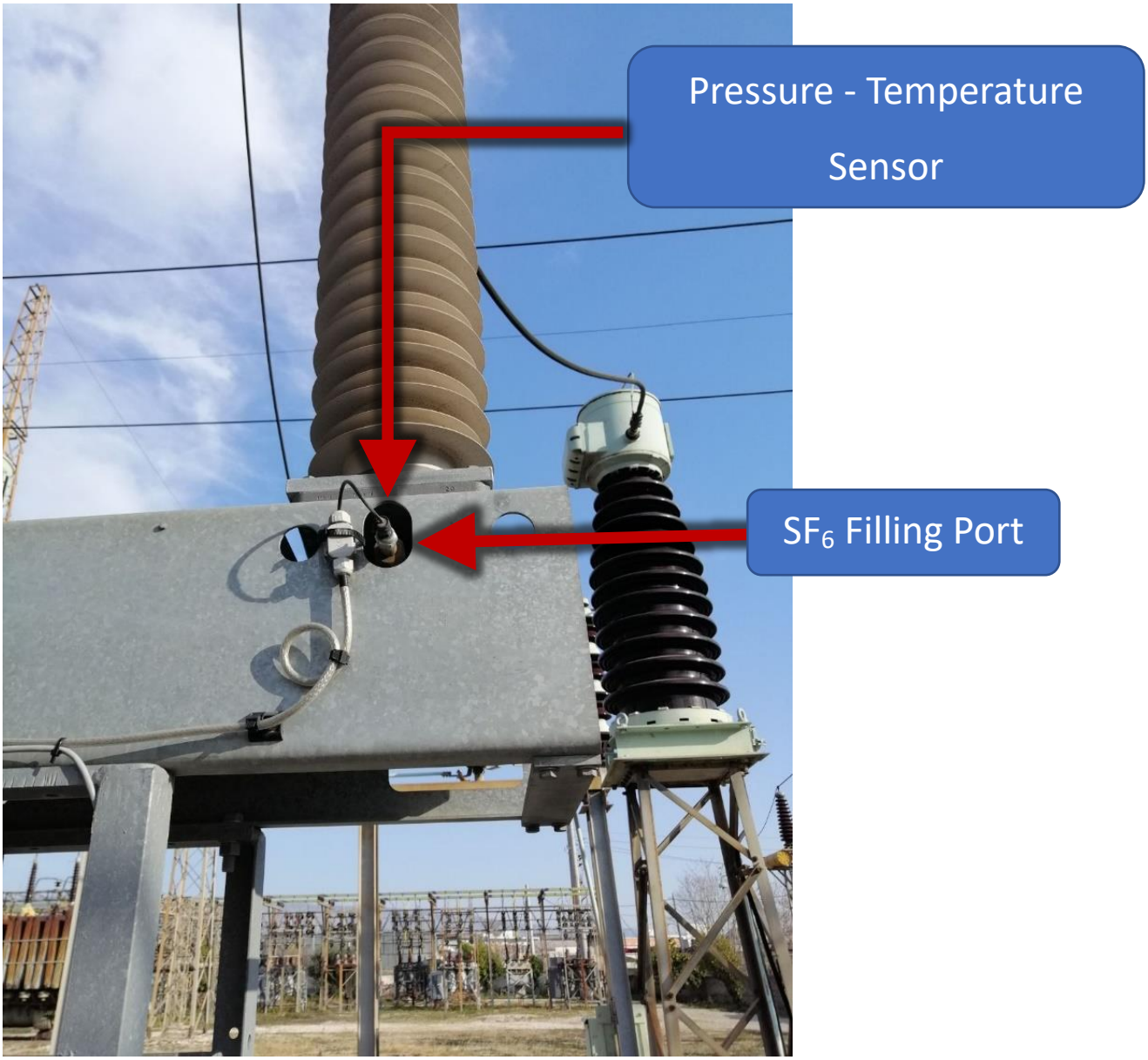


Figure 16 - Pressure Transducer attached to the DN20 port

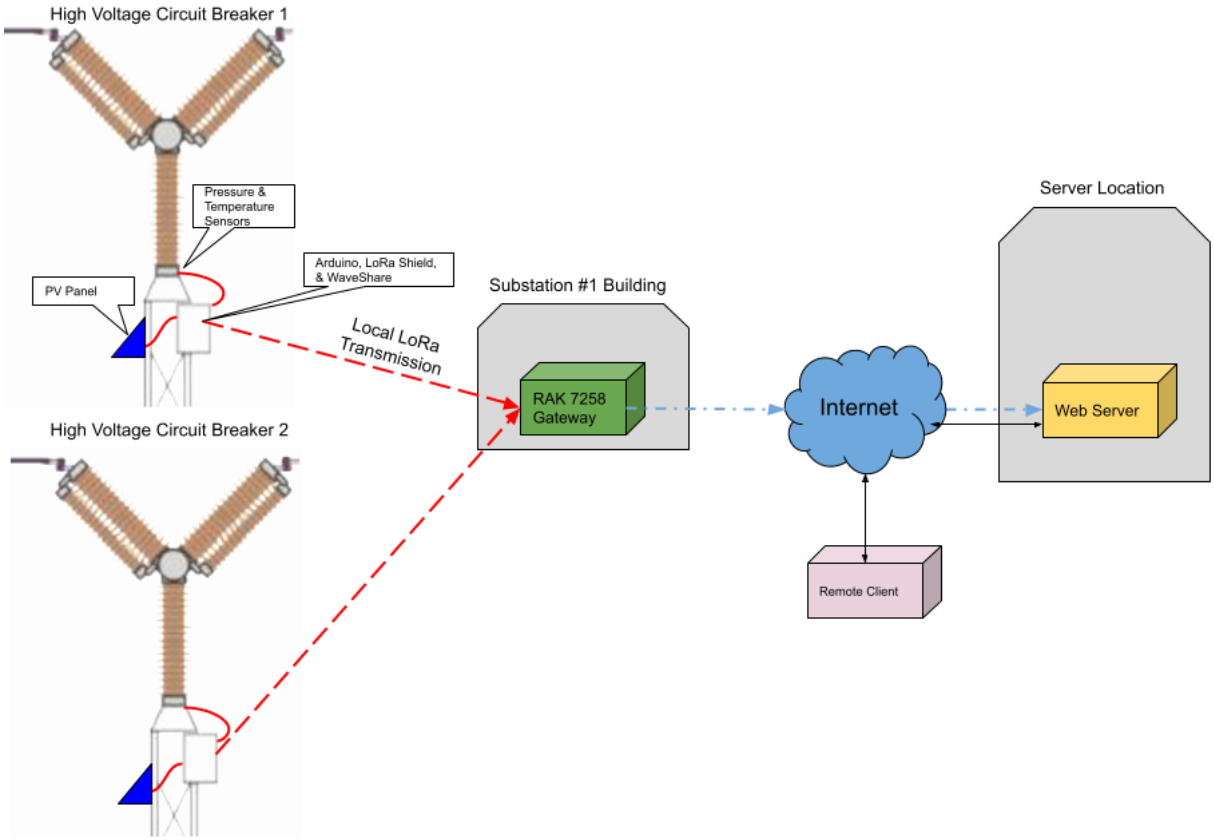


Figure 17 - Test System Configuration Diagram

As mentioned before, one of the prime advantages of the developed system is its practically infinite scalability. The Arduino-based data acquisition modules transfer the acquired data to the reachable LoRa gateway. While the application does not require rapid transmission repeat rates, it is possible to have hundreds of transmitters pinging only one gateway. Therefore, if there are no other technical limitations, such as the presence of devices that require very frequent data sampling, just one gateway will suffice for a single substation.

Once the LoRa gateway receives the packet, it will transmit the packet via the internet, or intranet, to the programmed Web Server for storage and manipulation. Similarly, assuming sufficient processing power, network capacity, and storage space, any number of gateways can be sending their data to a single Web Server, and any number of clients can be accessing the raw and processed data.

This approach also allows for maximum versatility as any number of parts, ranging from a single circuit breaker to a whole substation, can be added and/or modified without any downtime or impact on the rest of the system.

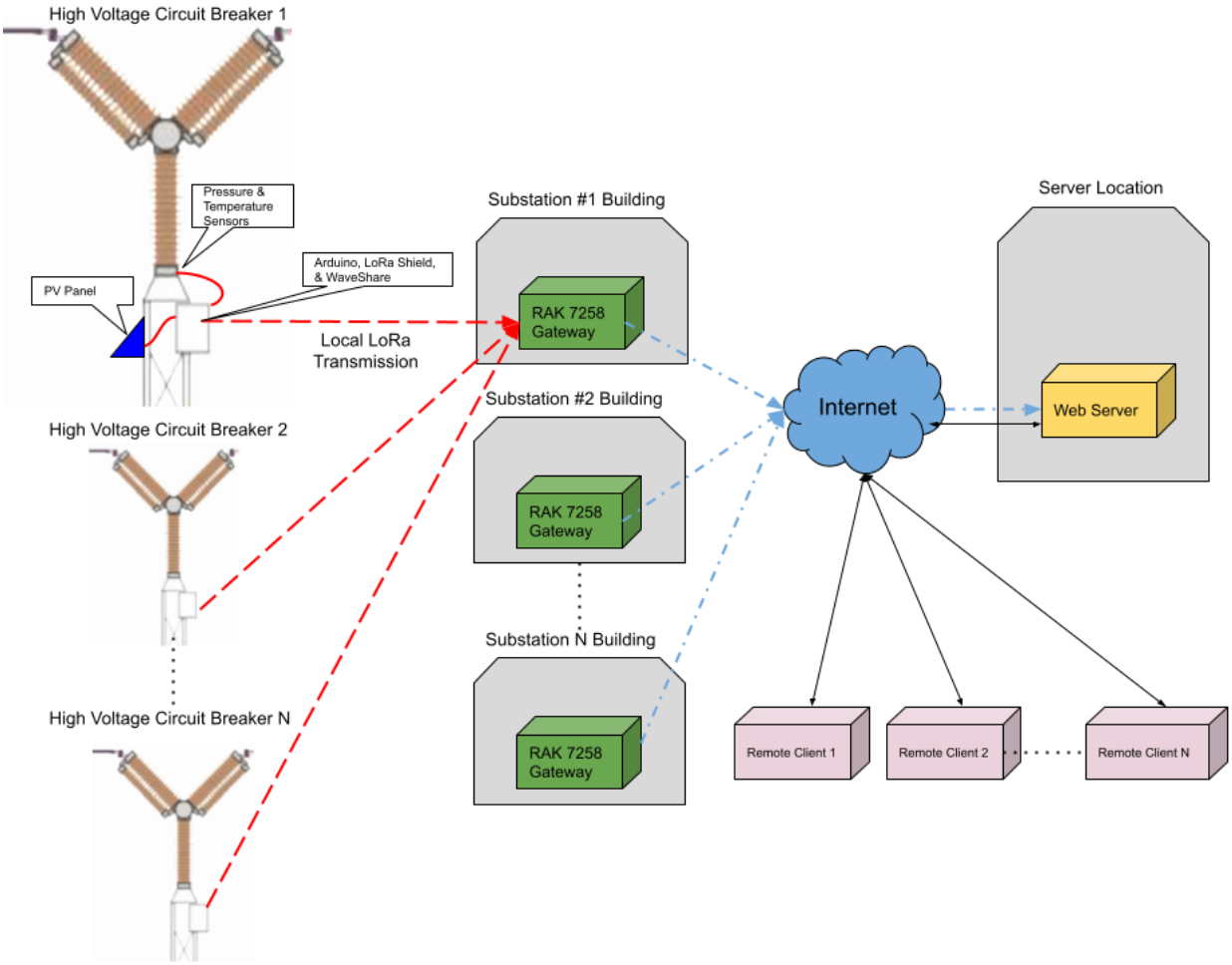


Figure 18 - System Scalability Example

3.2 Data Handling & Visualization

For the means of this project, a stand-alone website is developed and is hosted in a Windows 10 virtual machine, acting as our application server. Another approach would be to host the website using a web-based service, which approach offers visualization access from any device with internet access and ensures 100% uptime. In either case, it is easy to vary access and control per application and user. For our project, visualization is left accessible to any visitor but access to the mySQL database required administrator-only access.

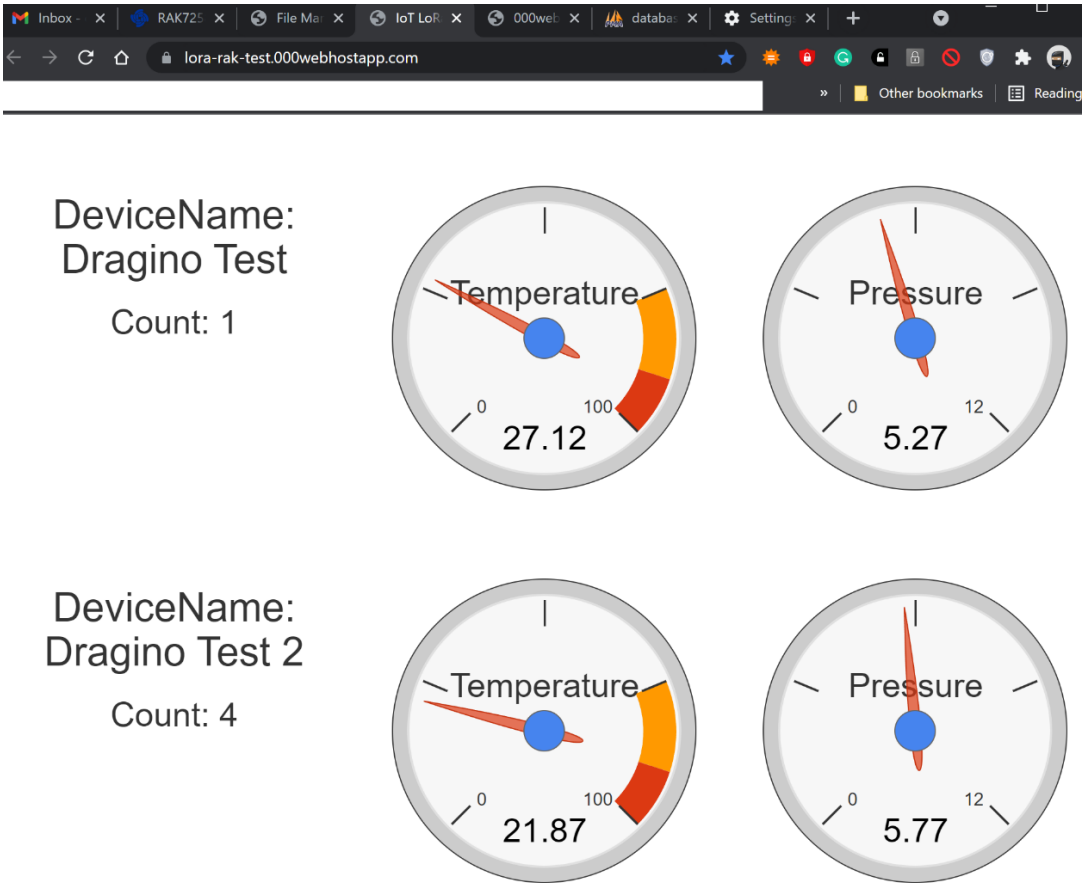


Figure 19 - Initial Visualization Interface (Web)

The incoming HTTP packets sent by the gateway are Base64 encrypted and the packet's timestamp is in Epoch format. An example of the transmitted packet can be seen in the following figure.

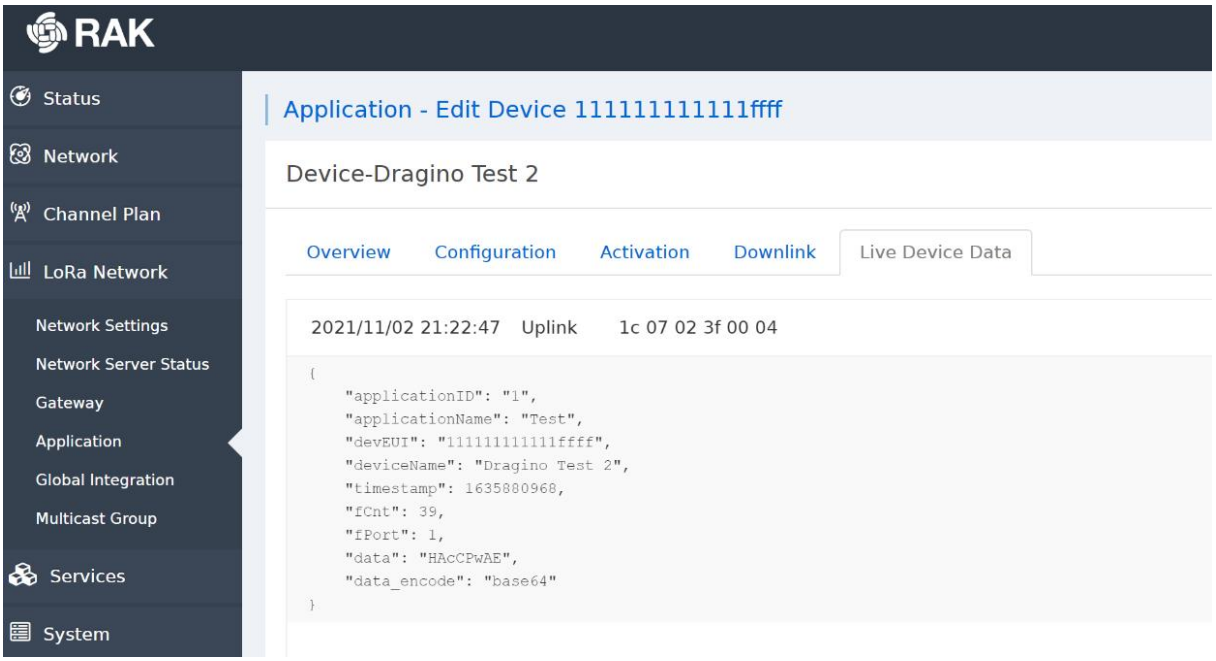


Figure 20 - End Node LoRa Packet

Once a packet arrives, it is fully decoded and the data are stored to a properly configured mySQL database, where they can receive any processing. The decoding algorithms for the Base64 encryption, as well as the conversion algorithms of the timestamp and raw data to usable records, can be found in Appendix B. The visualization façade of the website was initially configured to simply display the newest pressure, temperature, and counter data inserted into the mySQL database.

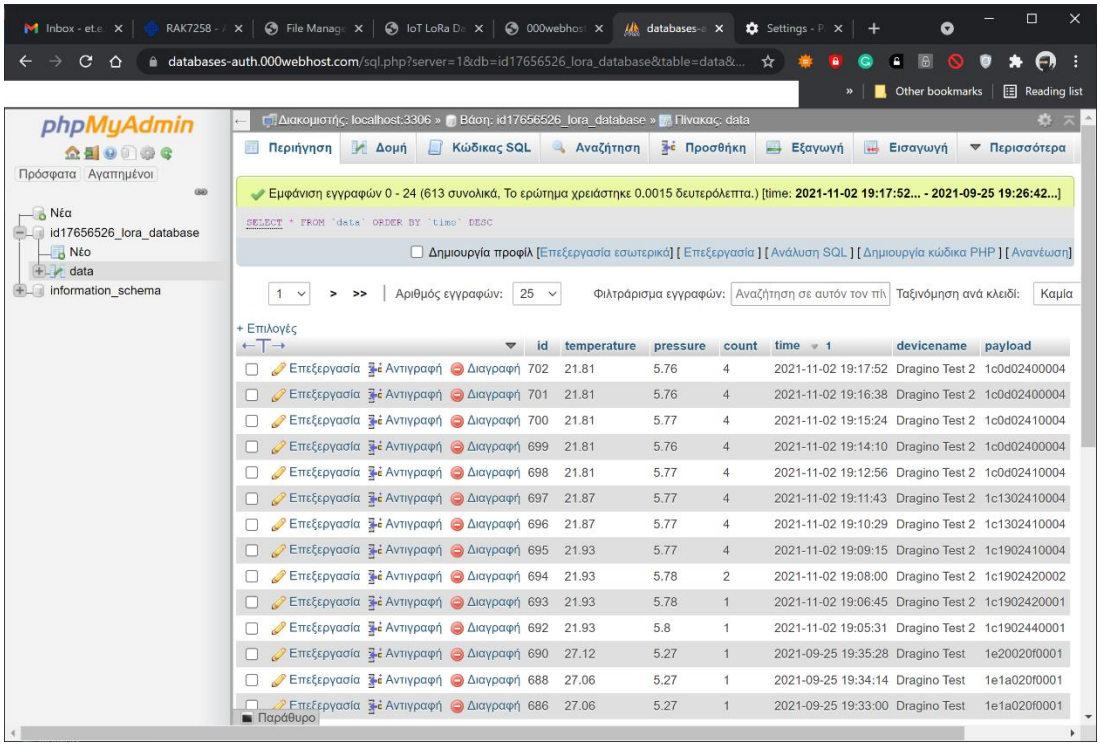


Figure 21 - MySQL Database Demo

3.3 Prognostics

The basis of any diagnostic procedure is the understanding of measurements. Not only expertise and experience affect the construal of such a task but physical parameters, such as the environment and the materials of the equipment itself, play a vital role on the actual condition of the switchgear. Establishing criteria that are valid require detailed knowledge of the equipment and deep understanding of the physical parameters in play.

Thus, diagnostics on switching equipment is far more involved than just collecting data. Simple data sets of basic physical values generally are not useful; at least not before the physical value reaches a critical point, at which point it usually is far too late to save the asset. As in most other aspects of handling technical equipment, the user must know what they are doing and why it is being done [99]. Assuming that the manufacturer of the equipment and/or an expert does establish the criteria

determining the health of a specific device, it is possible to use computing methods to derive the current - and future - health of an asset from sets of typically basic data.

The best form of condition monitoring is prognostics, which predicts the future health and remaining lifetime of an asset [100]. Condition monitoring combined with accurate forecasting allows for maintenance planning ahead of failures. With the data stored in a mySQL database, it is possible to manipulate that data to applicate any level of early warning mechanisms and/or prognostics.

3.3.1 Pressure & Temperature to Density Conversion

The primary requirement of prognostics is a stable data set. Pressure and temperature readings can vary due to environmental conditions and data manipulation could, in the long term, prove unreliable. This issue was instantly highlighted during our preliminary tests with the LoRa gateway, as it can be seen in the figure below, which displays one of the first sets of data that was acquired using our prototype transmitters.

However, it is possible to convert pressure and temperature readings directly to density by using the Beattie-Bridgeman equation [59, 87]:

$$\rho^3(RTF - D) + \rho^2(C - RTE) - \rho\, RT + p = 0$$

where T is the temperature, p is the absolute pressure, ρ is the density, and:

$$R = 569.502 \times 10^{-6} \frac{bar\ m^3}{kg\ K}\ , C = 738.22 \times 10^{-6} \frac{bar\ m^6}{kg^2}$$

$$D = 513.2105 \times 10^{-9} \frac{bar\ m^9}{kg^3}\ , E = 2.507 \times 10^{-3} \frac{m^3}{kg}\ , F = 2.1224 \times 10^{-6} \frac{m^6}{kg^2}$$

Implementing the above equation and converting the pressure to density is vital in order to receive a steady data output, as the pressure changes in correlation with the temperature. The result of the equation is in kg/m^3, the preferred density unit format suggested by CIGRE [101].

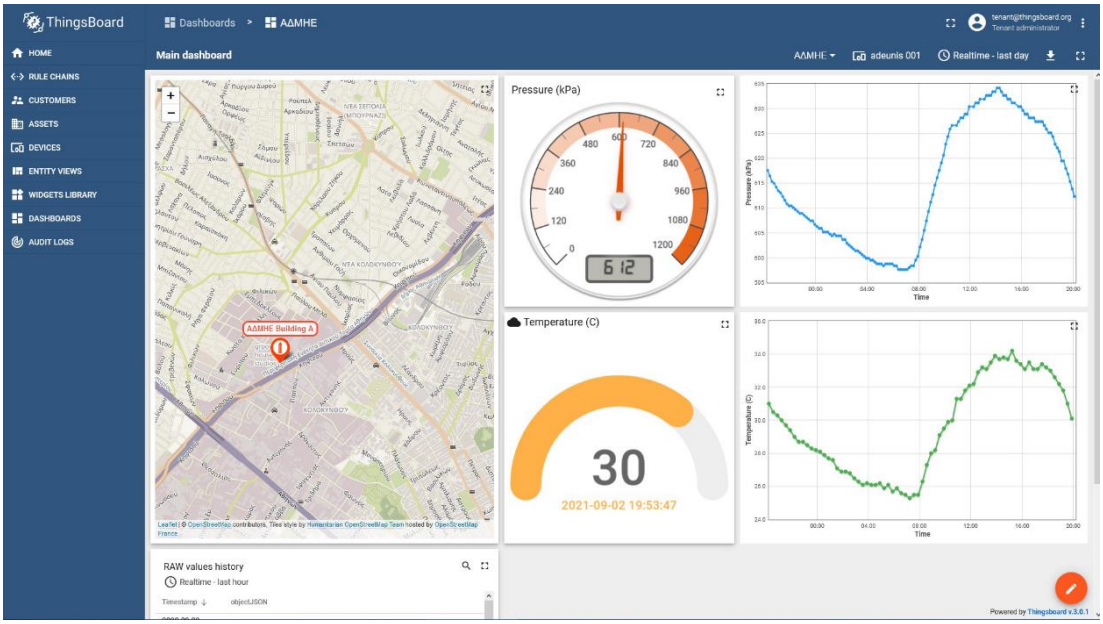


Figure 22 - Pressure & Temperature readings over 24h, uncorrected.

Using a MySQL database trigger, shown in Appendix C, we automatically implemented the Beattie-Bridgeman equation into the database itself. Every time a new row is inserted, or even updated, it triggers the calculation of the eighth database column, which corresponds to the density. The calculation of the density allows for stable results regardless of the environmental conditions.

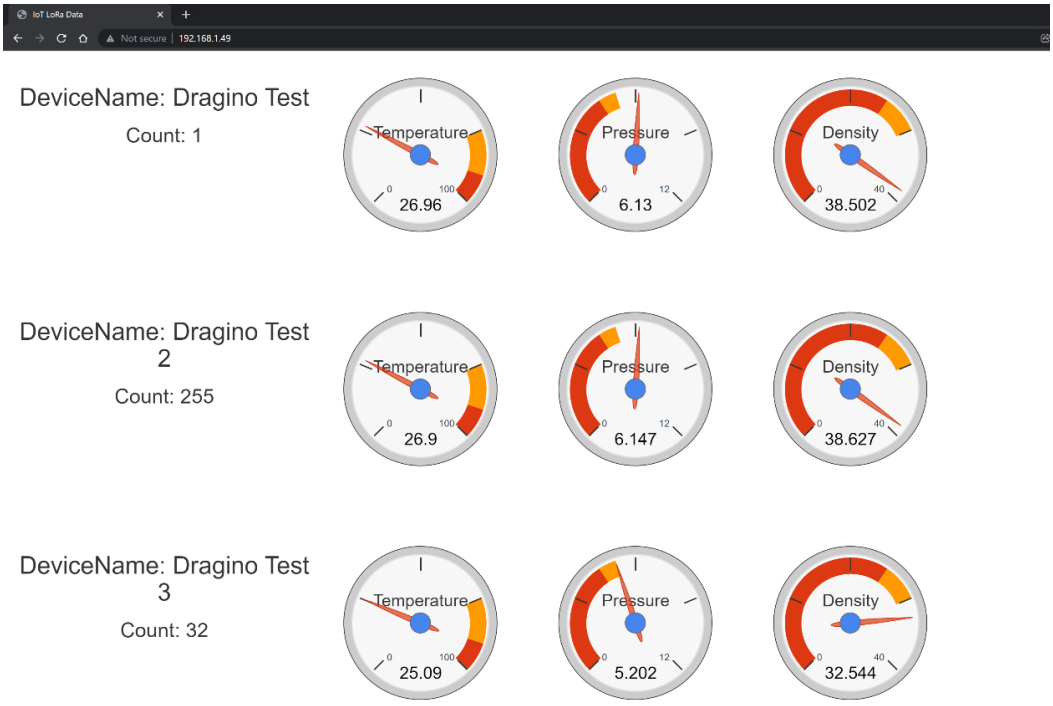


Figure 23 - Web Server Interface (Including Density Dials)

3.3.2 Data Visualization & Critical Forecasting

With the density known, it is mathematically simple to implement leak rate analysis to each circuit breaker that can serve as early warning systems, which can be used to both identify minor leaks and to project when the assumed leak will force the circuit breaker's protective mechanisms to trip, allowing for advanced condition-based maintenance planning [102].

As the data is stored in a MySQL database, there are numerous ways to access, process, and manipulate the data in real-time, regardless whether the web server is running locally or via a hosting service [103]. The most common approaches include Javascript [104], Python [105], and Open Database Connectivity (ODBC) [106]. Our test server is a standard Windows 10 WAMP 3.2.6 installation (Apache + PHP + MySQL), therefore we opted for a PHP/Javascript approach for advanced visualization and data handling.

A density graphing and forecasting page has been added into our test server, configured to automatically visualize the last 50 entries per device. For the means of data forecasting, we implemented a simple moving average forecasting algorithm. The algorithm can be seen as part of the code available in Appendix D. More advanced forecasting algorithms can be easily implemented if necessary.

In our test version, the user can manually enter a critical density value and the code will assess whether that value can be reached within a number of hours. This limit was set to 1.500 hours for our testing purposes but can be increased indefinitely, as long as the processing resources of the server allow it.

With the forecasting algorithm in place, it is easy to automate this procedure and have the server automatically run the forecasting algorithm every time a new row is inserted by using a trigger similar to that of paragraph 3.3.1, notifying certain users if the number of hours is below a certain threshold. The notification can be via any means the hosting server has access to, such as via e-mail, cellular text, or even a Modbus alert.

A screenshot of the desktop graphing sub-page running in our test server can be seen in the following figure.

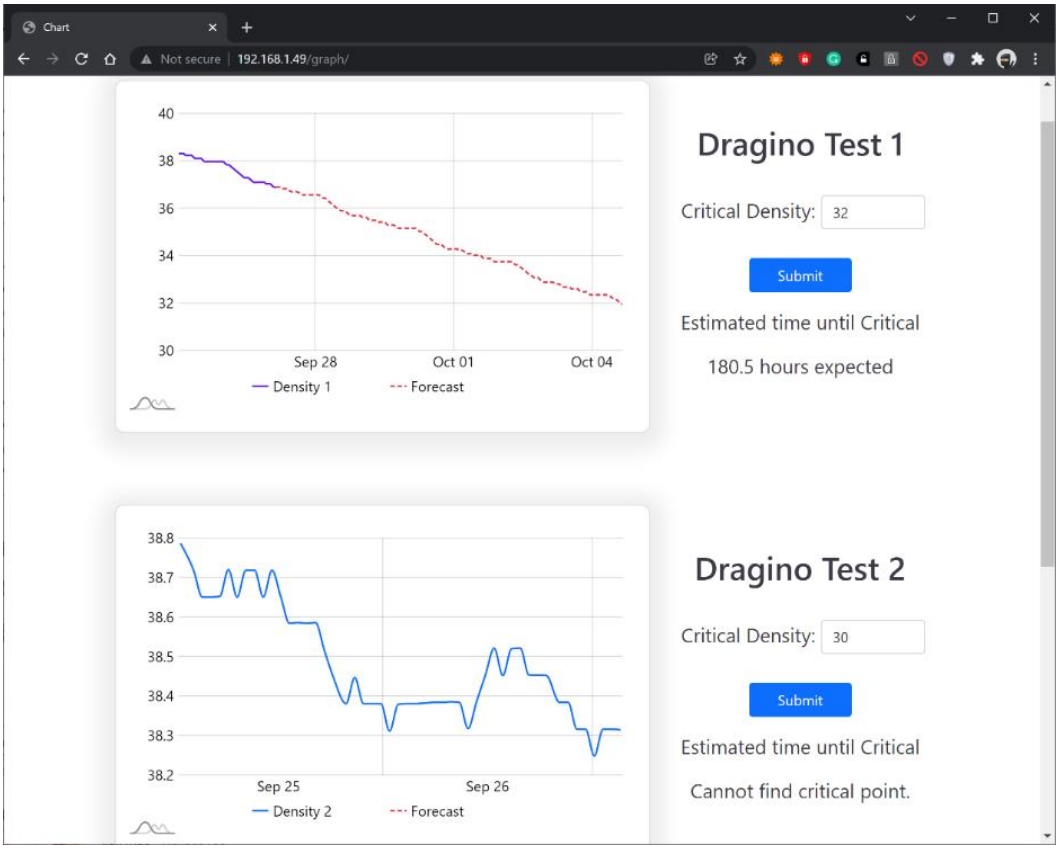


Figure 24 - Density Graphing & Forecasting

With long-term records, it is possible for an expert to manually implement a condition assessment algorithm for each different circuit breaker, similar to that proposed by CIGRE Guide 858 [107]. The condition assessment will have to take into account not only the maker and model of the circuit breaker but empirical factors as well, such as the position of the circuit breaker into the system.

Scale code log	Scale code linear	Description	Expected fault free life
1	1	Very good condition	>25 years with Normal test and inspect schedule
3	2	Good condition	15-25 years with Normal test and inspect schedule
10	3	Fair condition	5 - 15 years with More regular test and inspect schedule
30	4	Poor condition	<5 years Advanced test and inspect, plan remediation or replacement.
100	5	Critical condition	0-6 months Immediate remediation or replacement.

Figure 25 - Example of the condition assessment index proposed by CIGRE Guide 858.

Conclusion & Future Work

This work proves that, in summary, it is fairly easy to retrofit modern technology monitoring equipment into transmission equipment. The cost of the physical equipment is minimal. Although the development of the software requires a significant number of working hours, that typically needs to be done just once for the core setup, as every device can be running the same firmware and there is one server per system. All things considered, the cost of developing and retrofitting even a basic SF_6 monitoring system is likely to be offset if during its operational time prevents even one critical equipment failure. Further development can significantly expand the capabilities of the monitoring system and allow it to form the basis of a condition-based maintenance system.

The assessment of the accumulated data and their weighting to form a health index rating can be performed using empirical parameters, the recommendations of CIGRE Guide 858, or a mixture of both. For example, circuit breakers connecting/disconnecting low voltage reactors are called to interrupt very high inductive currents, causing a massive accumulation of SF_6 byproducts [108, 109], forcing Transmission System Operators to replace the gas on a frequent basis. Alerts can be programmed in the systems per device; therefore, it would be simple to have the software alert the operator when a set number of operations is reached, enabling condition-based maintenance as per the previous example. Finally, depending on the level of allowed system intrusiveness and the requirements of the operator, the system can be expanded to monitor a wider number of condition indicators mentioned in CIGRE Guide 858, such as, for example, the interrupting current (requires connection to the Current Transformer), Motor & Coil Current (requires ammeters), the velocity of the contacts (requires linear/rotary transducers), and even indicative vibration readings (requires acceleration sensors). Secondary equipment, such as the heaters and/or the formation of condensation inside the cabinet, can also be monitored, if critical for the particular asset.

References

[1] D. Singh, G. Tripathi, and A. J. Jara, "A survey of Internet-of-Things: Future vision, architecture, challenges and services," in *2014 IEEE World Forum on Internet of Things, WF-IoT 2014*, 2014, pp. 287-292.

[2] M. H. Miraz, M. Ali, P. S. Excell, and R. Picking, "A review on Internet of Things (IoT), Internet of Everything (IoE) and Internet of Nano Things (IoNT)," in *2015 Internet Technologies and Applications (ITA)*, 2015, pp. 219-224.

[3] V. Baños-Gonzalez, M. S. Afaqui, E. Lopez-Aguilera, and E. Garcia-Villegas, "IEEE 802.11 ah: A technology to face the IoT challenge," *Sensors,* vol. 16, p. 1960, 2016.

[4] M. Terán, J. Aranda, H. Carrillo, D. Mendez, and C. Parra, "IoT-based system for indoor location using bluetooth low energy," in *2017 IEEE Colombian Conference on Communications and Computing (COLCOM)*, 2017, pp. 1-6.

[5] M. Collotta, G. Pau, T. Talty, and O. K. Tonguz, "Bluetooth 5: A concrete step forward toward the IoT," *IEEE Communications Magazine,* vol. 56, pp. 125-131, 2018.

[6] M. R. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L. A. Grieco, G. Boggia*, et al.*, "Standardized protocol stack for the internet of (important) things," *IEEE communications surveys & tutorials,* vol. 15, pp. 1389-1406, 2012.

[7] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: An overview," *ieee communications surveys & tutorials,* vol. 19, pp. 855-873, 2017.

[8] I.-G. Lee, D. B. Kim, J. Choi, H. Park, S.-K. Lee, J. Cho*, et al.*, "WiFi HaLow for Long-Range and Low-Power Internet of Things: System on Chip Development and Performance Evaluation," *IEEE Communications Magazine,* vol. 59, pp. 101-107, 2021.

[9] B. Despatis-Paquette, L. Rivest, and R. Pellerin, "Connectivity validation for indoor IoT applications with Weightless protocol," in *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2019, pp. 393-399.

[10] L. Vangelista, A. Zanella, and M. Zorzi, "Long-range IoT technologies: The dawn of LoRa™," in *Future access enablers of ubiquitous and intelligent infrastructures*, 2015, pp. 51-58.

[11] J. Haxhibeqiri, E. De Poorter, I. Moerman, and J. Hoebeke, "A Survey of LoRaWAN for IoT: From Technology to Application," *Sensors,* vol. 18, p. 3995, 2018.

[12] A. Raychowdhury and A. Pramanik, "Survey on LoRa technology: solution for internet of things," *Intelligent Systems, Technologies and Applications,* pp. 259-271, 2020.

[13] M. Aref and A. Sikora, "Free space range measurements with Semtech LoRa™ technology," in *2014 2nd International Symposium on Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems*, 2014, pp. 19-23.

[14] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, "A study of LoRa: Long range & low power networks for the internet of things," *Sensors,* vol. 16, p. 1466, 2016.

[15] T. Petrić, M. Goessens, L. Nuaymi, L. Toutain, and A. Pelov, "Measurements, performance and analysis of LoRa FABIAN, a real-world implementation of LPWAN," in *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2016, pp. 1-7.

[16] L. Vangelista, "Frequency shift chirp modulation: The LoRa modulation," *IEEE Signal Processing Letters,* vol. 24, pp. 1818-1821, 2017.

[17] Semtech Corporation, "SX1272/73 860 MHz to 1020 MHz Low Power Long Range Transceiver," ed: Semtech Corporation Camarillo, CA, USA, 2017.

[18] E. T. ETSI, "Electromagnetic compatibility and radio spectrum matters (erm); short range devices (srd); radio equipment to be used in the 25 mhz to 1 000 mhz frequency range with

power levels ranging up to 500 mw," *European harmonized standard EN,* vol. 300, p. v2, 2012.
[19] N. Sornin, M. Luis, T. Eirich, T. Kramp, and O. Hersent, "Lorawan specification," *LoRa alliance,* 2015.
[20] L. Alliance, "Lorawantm 1.1 regional parameters," *LoRa Alliance: Fremont, CA, USA,* 2018.
[21] L. Leonardi, L. Lo Bello, F. Battaglia, and G. Patti, "Comparative Assessment of the LoRaWAN Medium Access Control Protocols for IoT: Does Listen before Talk Perform Better than ALOHA?," *Electronics,* vol. 9, p. 553, 2020.
[22] S. Li, U. Raza, and A. Khan, "How agile is the adaptive data rate mechanism of LoRaWAN?," in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 206-212.
[23] International Conference on Large High Voltage Electric Systems. Study Committee 13. Working Group 06 and A. Janssen, *Final report of the second international enquiry on high voltage circuit-breaker failures and defects in service*: CIGRE, 1994.
[24] J. Bessede, A. Buescher, R. Marshall, G. Montillet, A. Stelter, and A. D, "Limiting SF6 Gas emissions by Optimization of Design and Handling over the life cycle of HV switchgear," *EPA, San Antonio,* 2006.
[25] R. D. Garzon, *High voltage circuit breakers: design and applications*: CRC Press, 2002.
[26] M. Žarković and Z. Stojković, "Artificial intelligence SF6 circuit breaker health assessment," *Electric Power Systems Research,* vol. 175, p. 105912, 2019.
[27] J.-M. Biasse, E. Otegui, and B. Tilwitz-von Keiser, "Benefits of proper SF6 handling to reduce SF6 emissions for sustainable Electricity Transmission and Distribution," in *CICED 2010 Proceedings*, 2010, pp. 1-6.
[28] D. Lauzon and H. Koch, "Sulfur hexafluoride sf6," in *2006 IEEE Power Engineering Society General Meeting*, 2006, pp. 1-3.
[29] L. Kline, D. Davies, C. Chen, and P. Chantry, "Dielectric properties for SF6 and SF6 mixtures predicted from basic data," *Journal of Applied Physics,* vol. 50, pp. 6789-6796, 1979.
[30] F. M. Clark, *Insulating materials for design and engineering practice*: Wiley, 1962.
[31] A. Pedersen, "On the assessment of new gaseous dielectrics for GIS," *IEEE transactions on power apparatus and systems,* pp. 2232-2237, 1985.
[32] H. J. Koch, *Gas insulated substations*: John Wiley & Sons, 2014.
[33] D. Koch, "SF6 properties, and use in MV and HV switchgear," *Cahier technique,* vol. 188, 2003.
[34] G. Frind, "THE DECAY OF ELECTRIC ARCS. I. THEORETICAL CONSIDERATIONS," *Z. agnew. Phys.,* vol. 12, 1960.
[35] Solvay Special Chemicals. (2020, Sulphur Hexafluoride v.7. Available: https://www.solvay.com/en/brands/sulfur-hexafluoride-sf6-technical-grade
[36] W. Wang, X. Tu, D. Mei, and M. Rong, "Dielectric breakdown properties of hot SF6/He mixtures predicted from basic data," *Physics of Plasmas,* vol. 20, p. 113503, 2013.
[37] Y. Wang, J. Liang, C. Yu, L. Li, H. Zhang, and G. Li, "Research on the synergy of SF6/N2 mixture gas in low temperature environment," in *IOP Conference Series: Materials Science and Engineering*, 2019, p. 012045.
[38] M. Schulz and D. Kourkoulas, "Regulation (EU) No 517/2014 of The European Parliament and of the council of 16 April 2014 on fluorinated greenhouse gases and repealing Regulation (EC) No 842/2006," *Off. J. Eur. Union,* vol. 2014, p. L150, 2014.
[39] C. Neumann, A. Baur, A. Buscher, A. Luxa, F. Ploger, A. Reimuller, *et al.*, "Electrical power supply using SF6 technology–an ecological life cycle assessment," *Proceedings of the Cigré,* 2004.
[40] Y. Li, X. Zhang, J. Zhang, S. Xiao, B. Xie, D. Chen, *et al.*, "Assessment on the toxicity and application risk of C4F7N: A new SF6 alternative gas," *Journal of hazardous materials,* vol. 368, pp. 653-660, 2019.

[41] Y. Wu, C. Wang, H. Sun, M. Rong, A. B. Murphy, T. Li, *et al.*, "Evaluation of SF6-alternative gas C5-PFK based on arc extinguishing performance and electric strength," *Journal of Physics D: Applied Physics*, vol. 50, p. 385202, 2017.

[42] X. Li, H. Zhao, and A. B. Murphy, "SF6-alternative gases for application in gas-insulated switchgear," *Journal of Physics D: Applied Physics*, vol. 51, p. 153001, 2018.

[43] Y. Kieffel, "Characteristics of g 3-an alternative to SF 6," in *2016 IEEE International Conference on Dielectrics (ICD)*, 2016, pp. 880-884.

[44] Y. Wang, D. Huang, J. Liu, Y. Zhang, and L. Zeng, "Alternative environmentally friendly insulating gases for SF6," *Processes*, vol. 7, p. 216, 2019.

[45] Y. Hoshina, M. Sato, M. Shiiki, M. Hanai, and E. Kaneko, "Lightning impulse breakdown characteristics of SF6 alternative gases for gas-insulated switchgear," *IEE Proceedings-Science, Measurement and Technology*, vol. 153, pp. 1-6, 2006.

[46] E. Dullni, T. Endre, Y. Kieffel, and R. Coccioni, "Reducing SF6 emissions from electrical switchgear," *Carbon Management*, vol. 6, pp. 77-87, 2015.

[47] B. Pan, G. Wang, H. Shi, J. Shen, H.-K. Ji, and G.-S. Kil, "Green gas for grid as an eco-friendly alternative insulation gas to SF6: A Review," *Applied Sciences*, vol. 10, p. 2526, 2020.

[48] Z. Guo, S. Liu, Y. Pu, B. Zhang, X. Li, F. Tang, *et al.*, "Study of the arc interruption performance of CO 2 gas in high-voltage circuit breaker," *IEEE Transactions on Plasma Science*, vol. 47, pp. 2742-2751, 2019.

[49] B. Xiang, L. Gao, J. Luo, C. Wang, Z. Nan, Z. Liu, *et al.*, "A CO 2/O 2 mixed gas DC circuit breaker with superconducting fault current-limiting technology," *IEEE Transactions on Power Delivery*, vol. 35, pp. 1960-1967, 2019.

[50] I. Fradinho Bastos, "Marketing Introduction Plan for the New Generation of Sustainable Circuit Breakers LTA 420 kV: A real-life case for implementation at Hitachi ABB Power Grids," ed, 2021.

[51] L. G. Christophorou, J. K. Olthoff, and R. J. Van Brunt, "Sulfur hexafluoride and the electric power industry," *IEEE Electrical Insulation Magazine*, vol. 13, pp. 20-24, 1997.

[52] M. S. Dincer, S. S. Tezcan, and H. Duzkaya, "Suppression of electron avalanches in ultra-dilute SF6-N2 mixtures subjected to time-invariant crossed fields," *Energies*, vol. 11, p. 3247, 2018.

[53] Y. Kieffel, F. Biquez, P. Ponchon, and T. Irwin, "SF 6 alternative development for high voltage switchgears," in *2015 IEEE power & energy society general meeting*, 2015, pp. 1-5.

[54] A. Beroual and A. M. Haddad, "Recent advances in the quest for a new insulation gas with a low impact on the environment to replace sulfur hexafluoride (SF6) gas in high-voltage power network applications," *Energies*, vol. 10, p. 1216, 2017.

[55] M. Seeger, R. Smeets, J. Yan, H. Ito, M. Claessens, E. Dullni, *et al.*, "Recent trends in development of high voltage circuit breakers with SF6 alternative gases," *Plasma Physics and Technology*, vol. 4, pp. 8-12, 2017.

[56] E. André-Maouhoub, P. André, S. Makhlouf, and S. Nichele, "Production of Graphite During the Extinguishing Arc with New SF6 Alternative Gases," *Plasma Chemistry and Plasma Processing*, vol. 40, pp. 795-808, 2020.

[57] P. Robin-Jouan, K. Bousoltane, Y. Kieffel, J. Trepanier, R. Camarero, S. Arabi, *et al.*, "Analysis of last development results for high voltage circuit-breakers using new g3 gas," *Plasma Physics and Technology*, vol. 4, pp. 157-160, 2017.

[58] L. Maksoud, Y. Kieffel, J. Dabin, P. Lieutenant, I. Iddrissu, and T. Chen, "Dynamitron® SF6-FREE Project: The Assessment of g3 Applicability," in *MATPOST*, ed, 2019, pp. 1-4.

[59] P. Cheetham, A. Hellany, and S. Jones, "Density monitoring of high-voltage SF 6 circuit breakers," *IEEE Electrical Insulation Magazine*, vol. 31, pp. 6-13, 2015.

[60] X. Zhang, E. Gockenbach, Z. Liu, H. Chen, and L. Yang, "Reliability estimation of high voltage SF6 circuit breakers by statistical analysis on the basis of the field data," *Electric power systems research*, vol. 103, pp. 105-113, 2013.

[61] Z. Wang, Y. Xu, and S. Liu, "Condition monitoring and fault diagnosis for power equipment," ed: Shanghai Jiaotong University Press Shang Hai, 2012.

[62] S. Li and J. Li, "Condition monitoring and diagnosis of power equipment: review and prospective," *High Voltage*, vol. 2, pp. 82-91, 2017.

[63] Y. Zhang, J. Tian, F.-c. YE, and C. LU, "Development of temperature real-time monitoring network for high voltage switch cabinet based on IR sensor," *High Voltage Apparatus*, vol. 2, pp. 91-94, 2005.

[64] Q. Xiang-zhong, "Design of on-line monitoring system of temperature for contact inside of the high voltage switchgear [J]," *Instrument Technique and sensor*, vol. 2, pp. 73-75, 2007.

[65] C.-t. CHEN, S.-z. CAI, W.-b. LI, and Y.-p. WU, "Design and Implementation of Temperature Real-time Monitoring System for High Voltage Switchgear [J]," *Journal of Fujian Normal University (Natural Science Edition)*, vol. 5, 2008.

[66] Y. Tian and X.-b. Huang, "On-line monitoring system of temperature inside HV switchgear cabinet," *High Voltage Apparatus*, vol. 3, pp. 64-67, 2010.

[67] H. Ling, C. Chengmao, and L. Quan, "Real-time temperature monitoring system for High Voltage Switchgear based on infrared wireless transmission," in *The 2nd international conference on information science and engineering*, 2010, pp. 2265-2268.

[68] G. Qiang, G. Weichun, W. Chenggang, Z. Junyang, G. Baohong, and J. Xiuchen, "High voltage equipment online monitoring system of smart substation," in *IEEE PES Innovative Smart Grid Technologies*, 2012, pp. 1-5.

[69] R. Irawan, G. Scelsi, and G. Woolsey, "Continuous monitoring of SF/sub 6/degradation in high voltage switchgear using Raman scattering," *IEEE transactions on dielectrics and electrical insulation*, vol. 12, pp. 815-820, 2005.

[70] K. Meštrović, M. Poljak, M. Vidović, M. Furčić, M. Lončar, I. Maras, *et al.*, "New concept of high-voltage switchgear on-line monitoring system," in *16th International Symposium on high-voltage engineering–ISH*, 2009.

[71] L. Heinemann, J. Glock, C. Rehers, and T. Schulz, "Modular online condition monitoring system for gas insulated switchgear," in *IEEE PES T&D 2010*, 2010, pp. 1-6.

[72] K. Perdon, M. Scarpellini, S. Magoni, and L. Cavalli, "Modular online monitoring system to allow condition-based maintenance for medium voltage switchgear," *CIRED-Open Access Proceedings Journal*, vol. 2017, pp. 346-349, 2017.

[73] S. Richards, D. Chatrefou, and A. Procopiou, "Movement to the full digital substation for primary distribution," in *22nd International Conference and Exhibition on Electricity Distribution (CIRED 2013)*, 2013, pp. 1-4.

[74] N. Gariboldi, "Non-intrusive condition assessment methods for transmission and distribution switchgear—CIGRE/CIRED A3. 32 working group," in *2018 12th International Conference on the Properties and Applications of Dielectric Materials (ICPADM)*, 2018, pp. 218-222.

[75] H. K. Kondaveeti, N. K. Kumaravelu, S. D. Vanambathina, S. E. Mathe, and S. Vappangi, "A systematic literature review on prototyping with Arduino: Applications, challenges, advantages, and limitations," *Computer Science Review*, vol. 40, p. 100364, 2021.

[76] J. Busquets, J. V. Busquets, D. Tudela, F. Pérez, J. Busquets-Carbonell, A. Barberá, *et al.*, "Low-cost AUV based on Arduino open source microcontroller board for oceanographic research applications in a collaborative long term deployment missions and suitable for combining with an USV as autonomous automatic recharging platform," in *2012 IEEE/OES Autonomous Underwater Vehicles (AUV)*, 2012, pp. 1-10.

[77] B. Huang, "Open-source hardware–microcontrollers and physics education–integrating diy sensors and data acquisition with arduino," in *2015 ASEE Annual Conference & Exposition*, 2015, pp. 26.1205. 1-26.1205. 13.

[78] S. V. Hund, M. S. Johnson, and T. Keddie, "Developing a hydrologic monitoring network in data-scarce regions using open-source arduino dataloggers," *Agricultural & Environmental Letters*, vol. 1, p. 160011, 2016.

[79] J. P. Grinias, J. T. Whitfield, E. D. Guetschow, and R. T. Kennedy, "An inexpensive, open-source USB Arduino data acquisition device for chemical instrumentation," ed: ACS Publications, 2016.

[80] A. S. Ali, Z. Zanzinger, D. Debose, and B. Stephens, "Open Source Building Science Sensors (OSBSS): A low-cost Arduino-based platform for long-term indoor environmental data collection," *Building and Environment*, vol. 100, pp. 114-126, 2016.

[81] A. D. Wickert, C. T. Sandell, B. Schulz, and G.-H. C. Ng, "Open-source Arduino-compatible data loggers designed for field research," *Hydrology and Earth System Sciences*, vol. 23, pp. 2065-2076, 2019.

[82] M. Stusek, J. Pokorny, P. Masek, J. Hajny, and J. Hosek, "A non-invasive electricity measurement within the smart grid landscape: Arduino-based visualization platform for IoT," in *2017 9th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, 2017, pp. 423-429.

[83] I. González and A. J. Calderón, "Integration of open source hardware Arduino platform in automation systems applied to Smart Grids/Micro-Grids," *Sustainable Energy Technologies and Assessments*, vol. 36, p. 100557, 2019.

[84] M. Mozaffari Legha and E. Farjah, "IOT based load management of a micro-grid using Arduino and HMAS," *Iranian Journal of Electrical and Electronic Engineering*, vol. 16, pp. 228-234, 2020.

[85] V. H. L. Chalacan, *Performance Evaluation of Long Range (LoRa) Wireless Rf Technology for the Internet of Things (IoT) Using Dragino LoRa at 915 Mhz*: University of North Florida, 2020.

[86] J. Haxhibeqiri, A. Karaagac, F. Van den Abeele, W. Joseph, I. Moerman, and J. Hoebeke, "LoRa indoor coverage and performance in an industrial environment: Case study," in *2017 22nd IEEE international conference on emerging technologies and factory automation (ETFA)*, 2017, pp. 1-8.

[87] L. Fuhai, T. Zhan, D. Yongqiang, L. Yeyuan, and L. Shipeng, "Analysis and Discussion on the Current Situation of SF6 Relay Temperature Compensation Mode," in *IOP Conference Series: Earth and Environmental Science*, 2019, p. 032039.

[88] G. Li and Y. Zhao, "Principle and Application of 1Wire Bus Digital Thermometer DS18B20 [J]," *Modern Electronic Technique*, vol. 21, 2005.

[89] Y. Guan, M. Kezunovic, P. Dehghanian, and G. Gurrala, "Assessing circuit breaker life cycle using condition-based data," in *2013 IEEE Power & Energy Society General Meeting*, 2013, pp. 1-5.

[90] T. M. Lindquist, L. Bertling, and R. Eriksson. (2008, Circuit breaker failure data and reliability modelling. *IET Generation, Transmission & Distribution 2(6)*, 813-820. Available: https://digital-library.theiet.org/content/journals/10.1049/iet-gtd_20080127

[91] A. Siemens, "High-Voltage Circuit-Breakers: Trends and Recent Developments," *Order No. E50001-G630-M178-X-4A00*, 2011.

[92] M. Meli, E. Gatt, O. Casha, I. Grech, and J. Micallef, "A Low Cost LoRa-based IoT Big Data Capture and Analysis System for Indoor Air Quality Monitoring," in *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2020, pp. 376-381.

[93] N. J. B. Florita, A. N. M. Senatin, A. M. A. Zabala, and W. M. Tan, "Opportunistic LoRa-based gateways for delay-tolerant sensor data collection in urban settings," *Computer Communications*, vol. 154, pp. 410-432, 2020.

[94] Z. Wang, L. Kong, K. Xu, L. He, K. Wu, and G. Chen, "Online concurrent transmissions at LoRa gateway," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, 2020, pp. 2331-2340.
[95] Semtech Corporation. (2021, 30/8). *What are LoRa® and LoRaWAN®?* Available: https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan/
[96] N. Blenn and F. Kuipers, "LoRaWAN in the wild: Measurements from the things network," *arXiv preprint arXiv:1706.03086*, 2017.
[97] (2021, 30/10). *ChirpStack, open-source LoRaWAN® Network Server stack*. Available: https://www.chirpstack.io/
[98] WaveShare. (2021, 26/10). *Solar_Power_Manager*. Available: https://www.waveshare.com/solar-power-manager.htm
[99] C. Jones, "CIGRE working group 13.09-monitoring and diagnostic techniques for switching equipment," in *2001 IEEE/PES Transmission and Distribution Conference and Exposition. Developing New Perspectives (Cat. No.01CH37294)*, 2001, pp. 1083-1087 vol.2.
[100] J. B. Coble, "Merging data sources to predict remaining useful life–an automated method to identify prognostic parameters," 2010.
[101] E. Duggan, J. Potter, I. Ring, T. Schwarze, R. Kurte, A. Minotti, *et al.*, "Working Group B3.40 - 723," in *SF6 Measurement Guide*, ed: CIGRE, 2018.
[102] C. Deligi, "On-line SF6 monitoring—Breaking paradigms," in *2014 IEEE PES T&D Conference and Exposition*, 2014, pp. 1-4.
[103] L. Welling and L. Thomson, *PHP and MySQL Web development*: Sams Publishing, 2003.
[104] R. Nixon, *Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5*: " O'Reilly Media, Inc.", 2014.
[105] J. W. Krogh, G. Krogh, and Gennick, *MySQL Connector/Python Revealed*: Springer, 2018.
[106] W. Gilmore, "MySQL and ODBC," ed, 2001.
[107] J. Bednařík, J. Smit, R. Clerc, L. Darian, N. Kaiser, A. Livshitz, *et al.*, "Working Group B3.48 - 858," in *Asset health indices for equipment in existing substations*, ed: CIGRE, 2021.
[108] F. Chu, "SF6 decomposition in gas-insulated equipment," *IEEE Transactions on Electrical Insulation*, pp. 693-725, 1986.
[109] C. T. Dervos and P. Vassiliou, "Sulfur hexafluoride (SF6): global environmental effects and toxic byproduct formation," *Journal of the Air & Waste Management Association*, vol. 50, pp. 137-141, 2000.

Appendix A : Arduino Mega2560 & Dragino LoRa Shield Sketch

```
#include <lmic.h>
#include <EEPROM.h>
#include <hal/hal.h>
#include <SPI.h>
#include <Adafruit_GFX.h>
#include <Adafruit_ST7735.h>
#include <OneWire.h>
#include <DallasTemperature.h>

// LoRaWAN NwkSKey, network session key
static const PROGMEM u1_t NWKSKEY[16] = { 0x61, 0x9A, 0xA9, 0x7E, 0x6F, 0x46, 0xD3, 0xB
6, 0xF8, 0xE6, 0xB6, 0x87, 0xFE, 0x67, 0x29, 0x32 };

// LoRaWAN AppSKey, application session key
static const u1_t PROGMEM APPSKEY[16] = { 0x65, 0x13, 0xA0, 0x32, 0x15, 0x9A, 0xE2, 0x0
F, 0x62, 0xCD, 0xF7, 0xC8, 0xC0, 0x89, 0xB7, 0x90 };

// LoRaWAN end-device address (DevAddr)
static const u4_t DEVADDR = 0x260130E3 ; // <- Change this address for every node!

// These callbacks are only used in over-the-air activation, so they are
left empty here (we cannot leave them out completely unless
DISABLE_JOIN is set in config.h, otherwise the linker will complain).
void os_getArtEui (u1_t* buf) { }
void os_getDevEui (u1_t* buf) { }
void os_getDevKey (u1_t* buf) { }

static uint8_t mydata[] = "Hello, world!";
static osjob_t sendjob;

// Schedule TX every this many seconds (might become longer due to duty
// cycle limitations).
const unsigned TX_INTERVAL = 60;

// Pin mapping
const lmic_pinmap lmic_pins = {
    .nss = 10,
    .rxtx = LMIC_UNUSED_PIN,
    .rst = 9,
    .dio = {2, 6, 7},
};

int Pressure_sensor_val=A8;
float voltage;
float Pressure_pascal;
float Pressure_bar;
```

```
#define Set_counter 14
#define Reset_counter 15

int Counter=EEPROM.read(0);
int Activate=0;
int Set_counter_state = 0;
int Last_set_counter_state = 0;
int Reset_counter_state = 0;
int Last_reset_counter_state = 0;

#define TFT_CS 18
#define TFT_RST 17
#define TFT_DC 19
#define TFT_SCLK 21
#define TFT_MOSI 20
#define ONE_WIRE_BUS 16

// Setup a oneWire instance to communicate with any OneWire devices (not just Maxim/Dal
las temperature ICs)
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
float Celsius=0;
float Fahrenheit=0;

// LCD Setup

Adafruit_ST7735 tft = Adafruit_ST7735(TFT_CS, TFT_DC, TFT_MOSI, TFT_SCLK, TFT_RST);

// Transmission

void onEvent (ev_t ev) {
    Serial.print(os_getTime());
    Serial.print(": ");
    switch(ev) {
        case EV_SCAN_TIMEOUT:
            Serial.println(F("EV_SCAN_TIMEOUT"));
            break;
        case EV_BEACON_FOUND:
            Serial.println(F("EV_BEACON_FOUND"));
            break;
        case EV_BEACON_MISSED:
            Serial.println(F("EV_BEACON_MISSED"));
            break;
        case EV_BEACON_TRACKED:
            Serial.println(F("EV_BEACON_TRACKED"));
            break;
        case EV_JOINING:
            Serial.println(F("EV_JOINING"));
            break;
```

```
        case EV_JOINED:
            Serial.println(F("EV_JOINED"));
            break;
        case EV_RFU1:
            Serial.println(F("EV_RFU1"));
            break;
        case EV_JOIN_FAILED:
            Serial.println(F("EV_JOIN_FAILED"));
            break;
        case EV_REJOIN_FAILED:
            Serial.println(F("EV_REJOIN_FAILED"));
            break;
            break;
        case EV_TXCOMPLETE:
            Serial.println(F("EV_TXCOMPLETE (includes waiting for RX windows)"));
            if(LMIC.dataLen) {
                // data received in rx slot after tx
              Serial.print(F("Received "));
              Serial.print(LMIC.dataLen);
              Serial.println(F(" bytes of payload"));
            }
            // Schedule next transmission
            os_setTimedCallback(&sendjob, os_getTime()+sec2osticks(TX_INTERVAL), do_sen
d);
            break;
        case EV_LOST_TSYNC:
            Serial.println(F("EV_LOST_TSYNC"));
            break;
        case EV_RESET:
            Serial.println(F("EV_RESET"));
            break;
        case EV_RXCOMPLETE:
            // data received in ping slot
            Serial.println(F("EV_RXCOMPLETE"));
            break;
        case EV_LINK_DEAD:
            Serial.println(F("EV_LINK_DEAD"));
            break;
        case EV_LINK_ALIVE:
            Serial.println(F("EV_LINK_ALIVE"));
            break;
         default:
            Serial.println(F("Unknown event"));
            break;
    }
}

void do_send(osjob_t* j){
    // Check if there is not a current TX/RX job running
```

```
    if (LMIC.opmode & OP_TXRXPEND) {
        Serial.println(F("OP_TXRXPEND, not sending"));
    } else {

        float pressure = Pressure_pascal;
        float temperature = Celsius;
        int cnt=Counter;
        int temp=((int)(temperature*100))+5000;
        int pre=(int)(pressure/1000);

        uint8_t payload [ 6 ];

           payload [ 0 ] = temp >> 8 ;
           payload [ 1 ] = temp;
           payload [ 2 ] = pre >> 8 ;
           payload [ 3 ] = pre;
           payload [ 4 ] = cnt >> 8;
           payload [ 5 ] = cnt;

          Serial.println(temp);
          Serial.println(pre);
          Serial.println(cnt);

        // Prepare upstream data transmission at the next possible time.
        LMIC_setTxData2(1, payload, sizeof(payload), 0);
        Serial.println(F("Packet queued"));
    }
    // Next TX is scheduled after TX_COMPLETE event.
}

void setup() {
    Serial.begin(9600);
    Serial.println(F("Starting"));


  tft.initR(INITR_BLACKTAB);
  tft.fillScreen(ST7735_BLACK);

    //------------------------------

    pinMode(Pressure_sensor_val,INPUT);
    pinMode(Set_counter,INPUT_PULLUP);
    pinMode(Reset_counter,INPUT_PULLUP);
    sensors.begin();


    #ifdef VCC_ENABLE
    // For Pinoccio Scout boards
    pinMode(VCC_ENABLE, OUTPUT);
```

```
    digitalWrite(VCC_ENABLE, HIGH);
    delay(1000);
    #endif

    // LMIC init
    os_init();
    LMIC_reset();

    #ifdef PROGMEM
    uint8_t appskey[sizeof(APPSKEY)];
    uint8_t nwkskey[sizeof(NWKSKEY)];
    memcpy_P(appskey, APPSKEY, sizeof(APPSKEY));
    memcpy_P(nwkskey, NWKSKEY, sizeof(NWKSKEY));
    LMIC_setSession (0x1, DEVADDR, nwkskey, appskey);
    #else

    LMIC_setSession (0x1, DEVADDR, NWKSKEY, APPSKEY);
    #endif

    // Set up the channels

    LMIC_setupChannel(0, 868100000, DR_RANGE_MAP(DR_SF12, DR_SF7),  BAND_CENTI);

    // Disable link check validation
    LMIC_setLinkCheckMode(0);

    // Set data rate and transmit power (note: txpow seems to be ignored by the library
)
    LMIC_setDrTxpow(DR_SF7,14);

    // Start job
    do_send(&sendjob);
}

void loop() {

  Pressure_sensor_val=analogRead(A8);
  voltage=(Pressure_sensor_val)/1024.0;
  Pressure_pascal=(((voltage))*1200000.0);
  Pressure_bar=Pressure_pascal/100000;

//the user can select pressure unit

     char string[10];  // Create a character array of 10 characters

  tft.setCursor(15, 4);  // Set position (x,y)
  tft.setTextColor(ST7735_BLUE, ST7735_BLACK);  // Set color of text.
  tft.setTextSize(1);  // Set text size.
  tft.println("Pressure (bar):");  // Print a text or value
```

```
  // Convert float to a string:
  dtostrf(Pressure_bar, 3, 0, string);  // (<variable>,<amount of digits we are going t
o use>,<amount of decimal digits>,<string name>)

  tft.setCursor(15, 16);  // Set position (x,y)
  tft.setTextColor(ST7735_BLUE, ST7735_BLACK);  // Set color of text.
  tft.setTextSize(1);  // Set text size.
  tft.println(Pressure_bar);  // Print a text or value

  sensors.requestTemperatures();
  Celsius=sensors.getTempCByIndex(0);
  Fahrenheit=sensors.toFahrenheit(Celsius);

  char string1[10];
  dtostrf(Celsius, 3, 0,string1);
  tft.setCursor(15, 50);  // Set position (x,y)
  tft.setTextColor(ST7735_YELLOW, ST7735_BLACK);  // Set color of text.
  tft.setTextSize(1);  // Set text size.
  tft.println("Temperature (C)");  // Print a text or value
  tft.setCursor(15, 62);
  tft.setTextColor(ST7735_YELLOW, ST7735_BLACK);
  tft.setTextSize(1);
  tft.println(Celsius);  //the user have the option to display the temperature either i
n Celsius or Fahrenheit

  //read the state of the button value:

  Set_counter_state=digitalRead(Set_counter);
  delay(5);
  Reset_counter_state=digitalRead(Reset_counter);
  delay(5);

  // compare the Set_counter to its previous state

      if (Set_counter_state!= Last_set_counter_state){
      if (Set_counter_state == LOW) {

             Counter++;
             Serial.print( "Switch count: " ); Serial.println( Counter );
             char string3[10];
             dtostrf(Counter, 3, 0,string3);
             tft.setCursor(15, 96);
             tft.setTextColor(ST7735_RED, ST7735_BLACK);
             tft.setTextSize(1);
             tft.println("Counter:");
             tft.setTextColor(ST7735_RED, ST7735_BLACK);
             tft.setTextSize(1);
             tft.setCursor(15, 108);
```

```
            tft.println(Counter);
            EEPROM.write(0, Counter);
      }
           else {
            Serial.print( "Switch count: " ); Serial.println( Counter );
            char string3[10];
            dtostrf(Counter, 3, 0,string3);
            tft.setCursor(15, 96);
            tft.setTextColor(ST7735_RED, ST7735_BLACK);
            tft.setTextSize(1);
            tft.println("Counter:");
            tft.setTextColor(ST7735_RED, ST7735_BLACK);
            tft.setTextSize(1);
            tft.setCursor(15, 108);
            tft.println(Counter);

 // if the current state is LOW then the button went from on to off:
                 }
        // Delay a little bit to avoid bouncing
    }
 // save the current state as the last state, for next time through the loop
    Last_set_counter_state= Set_counter_state;

   // compare the Reset_counter to its previous state
    if (Reset_counter_state!= Last_reset_counter_state){
          if (Reset_counter_state == LOW) {
            Counter=0;
            tft.setCursor(15, 96);
            tft.setTextColor(ST7735_RED, ST7735_BLACK);
            tft.setTextSize(1);
            tft.println("Counter:");
            tft.setTextColor(ST7735_RED, ST7735_BLACK);
            tft.setTextSize(1);
            tft.setCursor(15, 108);
            tft.println(Counter);
            EEPROM.write(0, Counter);
          }
            else {
// if the current state is LOW then the button went from on to off:
            Serial.print( "Switch count: " ); Serial.println( Counter );

            char string3[10];
            dtostrf(Counter, 3, 0,string3);

            tft.setCursor(15, 96);
            tft.setTextColor(ST7735_RED, ST7735_BLACK);
            tft.setTextSize(1);
            tft.println("Counter:");
```

```
            tft.setTextColor(ST7735_RED, ST7735_BLACK);
            tft.setTextSize(1);
            tft.setCursor(15, 108);
            tft.println(Counter);
                 }
        // Delay a little bit to avoid bouncing
    }
    // save the current state as the last state, for next time through the loop
    Last_reset_counter_state= Reset_counter_state;

os_runloop_once();
}
```

Appendix B : Online Website Code

1. Base64 Decoder

```
package main
import (
     "encoding/base64"
     "encoding/hex"
     "fmt"
)
func main() {
     b, _ := base64.StdEncoding.DecodeString("BgApIN3zt4hgAAA=")
     fmt.Println("Bytes:", b)
     fmt.Println("HEX:", hex.EncodeToString(b))
}
```

2. Payload Decoder

```
function Decoder ( bytes , port )  {
  var tmp =  ( bytes [ 0 ] << 8  | bytes [ 1 ]);
  var pre =  ( bytes [ 2 ] << 8  | bytes [ 3 ]);
  var cnt =  ( bytes [ 4 ] << 8  | bytes [ 5 ]);

    return  {
      'temperature':  ( tmp - 5000 ) / 100 ,
      'pressure': pre / 10 ,
      'counter': cnt,
    }
}
```

3. Website Files

i. Index.php

```
<!DOCTYPE html>
<html lang="en">
    <head>
    <meta charset="UTF-8">
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
      <title>IoT LoRa Data</title>
      <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.min.js"></script>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
     <script type="text/javascript"
src="https://www.gstatic.com/charts/loader.js"></script>
     <script type="text/javascript">

        google.charts.load('current', {'packages':['gauge']});
        google.charts.setOnLoadCallback(drawChart);

        function drawChart() {

          var data = google.visualization.arrayToDataTable([
            ['Label', 'Value'],
             ['Temperature', 0]

          ]);
          var data2 = google.visualization.arrayToDataTable([
            ['Label', 'Value'],
             ['Pressure', 0]

          ]);
          var data3 = google.visualization.arrayToDataTable([
            ['Label', 'Value'],
             ['Density', 0]

          ]);
          var data4 = google.visualization.arrayToDataTable([
            ['Label', 'Value'],
             ['Temperature', 0]

          ]);
          var data5 = google.visualization.arrayToDataTable([
            ['Label', 'Value'],
             ['Pressure', 0]

            ]);
```

```
          var data6 = google.visualization.arrayToDataTable([
            ['Label', 'Value'],
             ['Density', 0]

          ]);
          var data7 = google.visualization.arrayToDataTable([
            ['Label', 'Value'],
             ['Temperature', 0]

          ]);
          var data8 = google.visualization.arrayToDataTable([
            ['Label', 'Value'],
             ['Pressure', 0]

            ]);
          var data9 = google.visualization.arrayToDataTable([
            ['Label', 'Value'],
             ['Density', 0]

          ]);

          var options = {
            width: 250, height: 250,
            redFrom: 90, redTo: 100,
            yellowFrom:75, yellowTo: 90,
            minorTicks: 1,

          };
          var options2 = {
            width: 250, height: 250,
            redFrom: 0, redTo: 4.5,
            yellowFrom:4.5, yellowTo: 5.2,
            minorTicks: 1,
            max: 12
          };
            var options3 = {
            width: 250, height: 250,
            redFrom: 0, redTo: 25,
            yellowFrom:25, yellowTo: 30,
            minorTicks: 1,
            max: 40
          };

          var chart = new
google.visualization.Gauge(document.getElementById('chart1'));
          chart.draw(data, options);

          var chart2 = new
google.visualization.Gauge(document.getElementById('chart2'));
```

```
          chart2.draw(data2, options2);

          var chart3 = new
google.visualization.Gauge(document.getElementById('chart3'));
          chart3.draw(data3, options3);

          var chart4 = new
google.visualization.Gauge(document.getElementById('chart4'));
          chart4.draw(data4, options);

          var chart5 = new
google.visualization.Gauge(document.getElementById('chart5'));
          chart5.draw(data5, options2);

          var chart6 = new
google.visualization.Gauge(document.getElementById('chart6'));
          chart6.draw(data6, options3);

          var chart7 = new
google.visualization.Gauge(document.getElementById('chart7'));
          chart7.draw(data7, options);

          var chart8 = new
google.visualization.Gauge(document.getElementById('chart8'));
          chart8.draw(data8, options2);

          var chart9 = new
google.visualization.Gauge(document.getElementById('chart9'));
          chart9.draw(data9, options3);



        setInterval(function() {
        var xhttp = new XMLHttpRequest();
        xhttp.onreadystatechange = function() {
           if (xhttp.readyState == 4 && xhttp.status == 200) {
               data.setValue(0,1,xhttp.responseText);
               chart.draw(data, options);
           }
        };
        xhttp.open("GET", "/readtemp.php?devicename=Dragino Test", true);
        xhttp.send();
       }, 1000);

       setInterval(function() {
        var xhttp = new XMLHttpRequest();
        xhttp.onreadystatechange = function() {
           if (xhttp.readyState == 4 && xhttp.status == 200) {
               data2.setValue(0,1,xhttp.responseText);
               chart2.draw(data2, options2);
```

```
    }
 };
 xhttp.open("GET", "/readpressure.php?devicename=Dragino Test", true);
 xhttp.send();
}, 1000);

 setInterval(function() {
 var xhttp = new XMLHttpRequest();
 xhttp.onreadystatechange = function() {
    if (xhttp.readyState == 4 && xhttp.status == 200) {
        data3.setValue(0,1,xhttp.responseText);
        chart3.draw(data3, options3);
    }
 };
 xhttp.open("GET", "/readdensity.php?devicename=Dragino Test", true);
 xhttp.send();
}, 1000);

setInterval(function() {
 var xhttp = new XMLHttpRequest();
 xhttp.onreadystatechange = function() {
    if (xhttp.readyState == 4 && xhttp.status == 200) {
     document.getElementById("Count").innerHTML = this.responseText;
    }
 };
 xhttp.open("GET", "/readcount.php?devicename=Dragino Test", true);
 xhttp.send();
}, 1000);



setInterval(function() {
 var xhttp = new XMLHttpRequest();
 xhttp.onreadystatechange = function() {
    if (xhttp.readyState == 4 && xhttp.status == 200) {
        data4.setValue(0,1,xhttp.responseText);
        chart4.draw(data4, options);
    }
 };
 xhttp.open("GET", "/readtemp.php?devicename=Dragino Test 2", true);
 xhttp.send();
}, 1000);

setInterval(function() {
 var xhttp = new XMLHttpRequest();
 xhttp.onreadystatechange = function() {
    if (xhttp.readyState == 4 && xhttp.status == 200) {
        data5.setValue(0,1,xhttp.responseText);
        chart5.draw(data5, options2);
```

```
    }
 };
 xhttp.open("GET", "/readpressure.php?devicename=Dragino Test 2", true);
 xhttp.send();
}, 1000);

 setInterval(function() {
 var xhttp = new XMLHttpRequest();
 xhttp.onreadystatechange = function() {
    if (xhttp.readyState == 4 && xhttp.status == 200) {
        data6.setValue(0,1,xhttp.responseText);
        chart6.draw(data6, options3);
    }
 };
 xhttp.open("GET", "/readdensity.php?devicename=Dragino Test 2", true);
 xhttp.send();
}, 1000);

setInterval(function() {
 var xhttp = new XMLHttpRequest();
 xhttp.onreadystatechange = function() {
    if (xhttp.readyState == 4 && xhttp.status == 200) {
     document.getElementById("Count2").innerHTML = this.responseText;
    }
 };
 xhttp.open("GET", "/readcount.php?devicename=Dragino Test 2", true);
 xhttp.send();
}, 1000);

setInterval(function() {
 var xhttp = new XMLHttpRequest();
 xhttp.onreadystatechange = function() {
    if (xhttp.readyState == 4 && xhttp.status == 200) {
        data7.setValue(0,1,xhttp.responseText);
        chart7.draw(data7, options);
    }
 };
 xhttp.open("GET", "/readtemp.php?devicename=Dragino Test 3", true);
 xhttp.send();
}, 1000);
setInterval(function() {
 var xhttp = new XMLHttpRequest();
 xhttp.onreadystatechange = function() {
    if (xhttp.readyState == 4 && xhttp.status == 200) {
        data8.setValue(0,1,xhttp.responseText);
        chart8.draw(data8, options2);
    }
 };
 xhttp.open("GET", "/readpressure.php?devicename=Dragino Test 3", true);
```

```
        xhttp.send();
       }, 1000);

        setInterval(function() {
        var xhttp = new XMLHttpRequest();
        xhttp.onreadystatechange = function() {
           if (xhttp.readyState == 4 && xhttp.status == 200) {
               data9.setValue(0,1,xhttp.responseText);
               chart9.draw(data9, options3);
           }
        };
        xhttp.open("GET", "/readdensity.php?devicename=Dragino Test 3", true);
        xhttp.send();
       }, 1000);

       setInterval(function() {
        var xhttp = new XMLHttpRequest();
        xhttp.onreadystatechange = function() {
           if (xhttp.readyState == 4 && xhttp.status == 200) {
            document.getElementById("Count3").innerHTML = this.responseText;
           }
        };
        xhttp.open("GET", "/readcount.php?devicename=Dragino Test 3", true);
        xhttp.send();
       }, 1000);

      //  setInterval(function() {
      //   var xhttp = new XMLHttpRequest();
      //   xhttp.onreadystatechange = function() {
      //      if (xhttp.readyState == 4 && xhttp.status == 200) {
      //       document.getElementById("deviceName").innerHTML = this.responseText;
      //      }
      //   };
      //   xhttp.open("GET", "/readdevice.php", true);
      //   xhttp.send();
      //  }, 1000);

        }
      </script>
    </head>
    <body>
        <div class="container-fluid" style="margin-top: 5%;">

        <div class="row" >
        <div class="col-sm-3" style="text-align: center;">

        <button onClick="document.location.href = 'graph/'" style="padding: 16px;
background-color: darkblue; color:aliceblue; border-radius: 4px;font-size:
20px">Calculate Density Trends</button>
```

```
        </div>
        </div>
          <div class="row" >
            <div class="col-sm-3" style="text-align: center;">

              <h1>
                <span class="ds-labels">DeviceName: </span>
                <span id="deviceName">Dragino Test</span>
                </h1>
                <h2>
                 <span class="ds-labels">Count:</span>
                 <span id="Count">0</span>
                 </h2>
               </div>
              <div class="col-sm-2" id="chart1">

              </div>
              <div class="col-sm-2" id="chart2">

              </div>
              <div class="col-sm-2" id="chart3">

              </div>

           </div>
        </div>
    </div>
        <div class="container-fluid" style="margin-top: 5%;">

          <div class="row" >
            <div class="col-sm-3" style="text-align: center;">

              <h1>
                <span class="ds-labels">DeviceName: </span>
                <span id="deviceName2">Dragino Test 2</span>
                </h1>
                <h2>
                 <span class="ds-labels">Count:</span>
                 <span id="Count2">0</span>
                </h2>
               </div>
              <div class="col-sm-2" id="chart4">

              </div>
              <div class="col-sm-2" id="chart5">

              </div>
               <div class="col-sm-2" id="chart6">
```

```
            </div>


         </div>
      </div>
      <div class="container-fluid" style="margin-top: 5%;">

        <div class="row" >
             <div class="col-sm-3" style="text-align: center;">

          <h1>
            <span class="ds-labels">DeviceName: </span>
            <span id="deviceName3">Dragino Test 3</span>
            </h1>
            <h2>
             <span class="ds-labels">Count:</span>
             <span id="Count3">0</span>
             </h2>
          </div>
            <div class="col-sm-2" id="chart7">

            </div>
            <div class="col-sm-2" id="chart8">

            </div>
            <div class="col-sm-2" id="chart9">

            </div>

         </div>
      </div>

   </body>
   </html>
```

ii. readcount.php

```
<?php
//Creating Array for JSON response
$response = array();
$servername = "localhost";
$username = "database_username";
$password = " database_password";
$dbname = "lora_database";
if(isset($_GET['devicename'])){
    $device=$_GET['devicename'];

 $link = mysqli_connect($servername, $username, $password, $dbname);
 // Fire SQL query to get all data from data
$result = mysqli_query($link,"SELECT *FROM data where devicename='$device' ORDER BY id
DESC") or die(mysql_error());

// Check for succesfull execution of query and no results found
if (mysqli_num_rows($result) > 0) {

    // Storing the returned array in response
    $response["data"] = array();

    // While loop to store all the returned response in variable
    if ($row = mysqli_fetch_array($result)) {

        echo $row["count"];
    }
}
else
      {

      }
}
?>
```

iii.readpressure.php

```
<?php


//Creating Array for JSON response
$response = array();
$servername = "localhost";
$username = "database_username";
$password = " database_password";
$dbname = "lora_database";
if(isset($_GET['devicename'])){
    $device=$_GET['devicename'];
```

```
 $link = mysqli_connect($servername, $username, $password, $dbname);
 // Fire SQL query to get all data from data
$result = mysqli_query($link,"SELECT *FROM data where devicename='$device' ORDER BY id
DESC") or die(mysql_error());

// Check for succesfull execution of query and no results found
if (mysqli_num_rows($result) > 0) {

    // Storing the returned array in response
    $response["data"] = array();

    // While loop to store all the returned response in variable
    if ($row = mysqli_fetch_array($result)) {


        echo $row["pressure"];
    }

}
else
{

}
}
?>
```

iv.readtemp.php

```
<?php


//Creating Array for JSON response
$response = array();
$servername = "localhost";
$username = "database_username";
$password = " database_password";
$dbname = "lora_database";
if(isset($_GET['devicename'])){
    $device=$_GET['devicename'];

 $link = mysqli_connect($servername, $username, $password, $dbname);
 // Fire SQL query to get all data from data
$result = mysqli_query($link,"SELECT *FROM data where devicename='$device' ORDER BY id
DESC") or die(mysql_error());

// Check for succesfull execution of query and no results found
if (mysqli_num_rows($result) > 0) {

    // Storing the returned array in response
```

```
    $response["data"] = array();

    // While loop to store all the returned response in variable
    if ($row = mysqli_fetch_array($result)) {

        echo $row["temperature"];
    }

}
else
{
}
}
?>
```

v.readdensity.php

```
<?php
//Creating Array for JSON response
$response = array();
$servername = "localhost";
$username = "database_username";
$password = " database_password";
$dbname = "lora_database";
if(isset($_GET['devicename'])){
    $device=$_GET['devicename'];

 $link = mysqli_connect($servername, $username, $password, $dbname);
 // Fire SQL query to get all data from data
$result = mysqli_query($link,"SELECT *FROM data where devicename='$device' ORDER BY id
DESC") or die(mysql_error());

// Check for succesfull execution of query and no results found
if (mysqli_num_rows($result) > 0) {

    // Storing the returned array in response
    $response["data"] = array();

    // While loop to store all the returned response in variable
    if ($row = mysqli_fetch_array($result)) {

        echo $row["Density"];
    }

}
else
{
}
}
?>
```

vi.recv.php

```
<?php
header("Content-type: application/json");
$json = file_get_contents("php://input");
$obj = json_decode($json);
$data=($obj->data);
$devName=($obj->deviceName);
$decoded= bin2hex(base64_decode($data));
$temperature=substr($decoded,0,2);
$temperature=hexdec($temperature);
$pressure=substr($decoded,2,2);
$pressure=hexdec($pressure);
$count=substr($decoded,4,2);
$count=hexdec($count);

$servername = "localhost";
$username = "database_username";
$password = " database_password";
$dbname = "lora_database";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
  die("Connection failed: " . mysqli_connect_error());
}

$sql = "INSERT INTO data (temperature, pressure, count,time,devicename)
VALUES ('$temperature', '$pressure', '$count',NOW(),'$devName')";

if (mysqli_query($conn, $sql)) {
  echo "New record created successfully";
} else {
  echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}

mysqli_close($conn);
?>
```

Appendix C : mySQL Database Beattie Bridgeman Equation Solver Trigger

```
USE test;

SELECT * FROM data;

-- create function to calculate density cubic equation:

DROP FUNCTION IF EXISTS FN_DENSITY_CUBIC_EQUATION;

DELIMITER //

CREATE FUNCTION FN_DENSITY_CUBIC_EQUATION (TEMPERATURE FLOAT, PRESSURE FLOAT)
RETURNS FLOAT
DETERMINISTIC
BEGIN
    DECLARE a float;
    DECLARE b float;
    DECLARE c float;
    DECLARE d float;

    DECLARE Q float;
    DECLARE R float;

    DECLARE Temp_S float;
    DECLARE Temp_T float;

    DECLARE S float;
    DECLARE T float;

    DECLARE x float;

    SELECT (569.502 / POWER(10, 6)) * (2.1224 / POWER(10, 6)) * (TEMPERATURE + 273.15) -
(513.2105 / POWER(10, 9)) INTO a;
    SELECT (738.22 / POWER(10, 6)) - (569.502 / POWER(10, 6)) * (2.507 / POWER(10, 3)) *
(TEMPERATURE + 273.15) INTO b;
    SELECT -1 * (569.502 / POWER(10, 6)) * (TEMPERATURE + 273.15) INTO c;
    SELECT PRESSURE INTO d;

    -- Cardano's formula:

    SELECT (3 * a * c - POWER(b, 2)) / (9 * POWER(a, 2)) INTO Q; -- Q = (3ac - b²) / (9a²)
    SELECT (9 * a * b * c - 27 * POWER(a, 2) * d - 2 * POWER(b, 3)) / (54 * POWER(a, 3)) INTO
R;  -- R = (9abc - 27a²d - 2b³) / (54a³)

    SELECT (R + SQRT(POWER(Q, 3) + POWER(R, 2))) INTO Temp_S; -- R + √(Q³ + R²)
    SELECT (R - SQRT(POWER(Q, 3) + POWER(R, 2))) INTO Temp_T; -- R - √(Q³ + R²)

    SELECT POWER(ABS(Temp_S), (1/3.0)) * (CASE WHEN Temp_S < 0 THEN -1 ELSE 1 END) INTO S;
```

```
    SELECT POWER(ABS(Temp_T), (1/3.0)) * (CASE WHEN Temp_T < 0 THEN -1 ELSE 1 END) INTO T;

    SELECT S + T - b / (3 * a) INTO x; -- x1 = S + T - b/(3a)

    RETURN x;
END; //

DELIMITER ;

-- create before insert trigger:

DROP TRIGGER IF EXISTS TR_DENSITY_BEFORE_INSERT;

DELIMITER $$

CREATE TRIGGER TR_DENSITY_BEFORE_INSERT
BEFORE INSERT
ON data FOR EACH ROW
BEGIN
    SET new.Density = (SELECT FN_DENSITY_CUBIC_EQUATION(new.temperature, new.pressure));
END$$

DELIMITER ;

-- create before update trigger:

DROP TRIGGER IF EXISTS TR_DENSITY_BEFORE_UPDATE;

DELIMITER $$

CREATE TRIGGER TR_DENSITY_BEFORE_UPDATE
BEFORE UPDATE
ON data FOR EACH ROW
BEGIN
    SET new.Density = (SELECT FN_DENSITY_CUBIC_EQUATION(new.temperature, new.pressure));
END$$

DELIMITER ;
```

Appendix D : Graphing & Forecasting Algorithm

i. Index.php

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>Chart</title>
        <meta charset="UTF-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1.0" />

        <link rel="stylesheet" href="./css/bootstrap.min.css" />
        <script src="./js/bootstrap.min.js"></script>

        <script src="./xy/animated.js"></script>
        <script src="./xy/index.js"></script>
        <script src="./xy/xy.js"></script>
        <script src = "./js/control.js"></script>

    </head>
    <style>
        body{
            background-color:white;
        }
        .chart-container {
            width: 100%;
            height: 350px;
        }
        .card{
            border-radius:10px;
            box-shadow:0 5px 20px 15px #00000012;
            padding:20px 10px;
        }
        p, span, button, input{
            font-size:22px ;
            font-weight:400 !important;
            color:#3b3944;
        }
        button{
            width:40%
        }
        input:focus, button:focus{
            box-shadow:none !important;
        }


    </style>
    <body onload = 'Control.init()'>
        <div class = "container" >
            <div class ="row mt-5 p-3">
```

```
                <div class="col-lg-8 col-md-12 text-center">
                    <div class="card ">
                        <div id="chart_1" class="chart-container"></div>
                    </div>
                </div>
                <div class="col-lg-4 col-md-12 text-center m-auto">
                    <p style="font-size:36px ; font-weight:500 !important ">Dragino Test 1</p>
                    <div class='row p-3 mb-3 align-items-cneter'>
                        <div class='col-7  p-0'>
                            <span class=''>Critical Density:</span>
                        </div>
                        <div class='col-5 p-0'>
                            <input type="number" class='form-
control'  id="critical_density_1"></input>
                        </div>
                    </div>
                    <button class='mb-3 btn btn-primary btn-md' id='submit_1' onclick =
"Control.initEvent(1)">Submit</button>
                    <p class='mb-3'>Estimated time until Critical</p>
                    <p class='' id="critical_time_1" > </p>
                </div>
            </div>
            <div class ="row mt-5 p-3">
                <div class="col-lg-8 col-md-12 text-center">
                    <div class="card">
                        <div id="chart_2" class="chart-container"></div>
                    </div>
                </div>
                <div class="col-lg-4 col-md-12 text-center m-auto">
                <p style="font-size:36px ; font-weight:500 !important ">Dragino Test  2</p>
                    <div class='row p-3 mb-3 align-items-cneter'>
                        <div class='col-7 p-0'>
                            <span class=''>Critical Density:</span>
                        </div>
                        <div class='col-5 p-0'>
                            <input type="number" class='form-control
'  id="critical_density_2"></input>
                        </div>
                    </div>
                    <button class='mb-3 btn btn-primary btn-md' id='submit_2' onclick =
"Control.initEvent(2)">Submit</button>
                    <p class='mb-3'>Estimated time until Critical</p>
                    <p class='' id="critical_time_2"></p>
                </div>
            </div>
            <div class ="row mt-5 p-3">
                <div class="col-lg-8 col-md-12 text-center">
                    <div class="card">
                        <div id="chart_3" class="chart-container"></div>
```

```
                    </div>
                </div>
                <div class="col-lg-4 col-md-12 text-center m-auto">
                    <p style="font-size:36px ; font-weight:500 !important ">Dragino Test  3</p>
                    <div class='row p-3 mb-3 align-items-cneter'>
                        <div class='col-7  p-0'>
                            <span class=''>Critical Density:</span>
                        </div>
                        <div class='col-5    p-0'>
                            <input type="number" class='form-control
'  id="critical_density_3"></input>
                        </div>
                    </div>
                    <button class='mb-3 btn btn-primary btn-md' id = 'submit_3' onclick =
"Control.initEvent(3)">Submit</button>
                    <p class='mb-3'>Estimated time until Critical</p>
                    <p class='' id="critical_time_3"></p>
                </div>
            </div>
        </div>

    </body>

        <script >

        </script>



</html>
```

ii. fetch-data.php

```
<?php

    header("Content-Type: application/json");
    // build a PHP variable from JSON sent using POST method
    $request = json_decode(stripslashes(file_get_contents("php://input")));
      $servername = "localhost";
        $username = "database_username";
        $password = " database_password";
        $dbname = "lora_database";
    $mysqli = new mysqli($dbhost, $dbuser, $dbpass,$db) or die("Connect failed: %s\n". $conn -> error);
    $data1 = array();
    if ($result = $mysqli -> query("SELECT * FROM data WHERE devicename ='Dragino Test' ORDER BY id DESC LIMIT 50")) {

        while ($row = $result->fetch_row()) {
            array_push($data1,$row);
        }

        $result -> free_result();
    }
    $data2 = array();
    if ($result = $mysqli -> query("SELECT * FROM data WHERE devicename ='Dragino Test 2' ORDER BY id DESC LIMIT 50")) {

        while ($row = $result->fetch_row()) {
            array_push($data2,$row);
        }

        $result -> free_result();
    }
    $data3 = array();
    if ($result = $mysqli -> query("SELECT * FROM data WHERE devicename ='Dragino Test 3' ORDER BY id DESC LIMIT 50")) {

        while ($row = $result->fetch_row()) {
            array_push($data3,$row);
        }

        $result -> free_result();
    }
    $mysqli -> close();
    echo json_encode(array('status'=>'success','data1'=>$data1,'data2'=>$data2,'data3'=>$data3));
?>
```

iii. Control.js

```
var Control = function () {

var chart_data1 = [];
var chart_data2 = [];
var chart_data3 = [];
var offset_value1 = [], offset_value2 = [], offset_value3 = [];
var offset_time1 = [], offset_time2 = [], offset_time3 = [];


var dot_chart1_series,dot_chart2_series,dot_chart3_series;

var _load_data = function () {
  const data = { username: 'example' };

  fetch('fetch-data.php', {
    method: 'POST', // or 'PUT'
    headers: {
      'Content-Type': 'application/json',
    },

    body: JSON.stringify(data),
  })
    .then(response => response.json())
    .then(data => {

      chart_data1 = data_complate(data.data1, 1);
      chart_data2 = data_complate(data.data2, 2);
      chart_data3 = data_complate(data.data3, 3);
      _init_chart([chart_data1.main_data, chart_data2.main_data, chart_data3.main_data]);

    });
}


var _init_event = function (value) {
  if(value == 1){
    dot_chart1_series.data.setAll([]);
    chart1.data.setAll(chart_data1.main_data);
  }
  if(value == 2){
    dot_chart2_series.data.setAll([]);
    chart2.data.setAll(chart_data2.main_data);
  }
  if(value == 3){
    dot_chart3_series.data.setAll([]);


    chart3.data.setAll(chart_data3.main_data);
```

```
  }



  var chart_data = [];
  var critical_value = document.getElementById('critical_density_' + value).value;
  if (value == 1) {
    chart_data = chart_data1;
  }
  else if (value == 2) {
    chart_data = chart_data2;
  }
  else {
    chart_data = chart_data3;
  }
  limitLoof = 0;
  data_critical(chart_data, eval(critical_value), value);
}

var data_complate = function (data, opt) {
  data = data.reverse();
  var chart_data = [];
  var value = 0, min_value = 0, max_value = 0;
  var start_value = eval(data[0][7]);
  var temp_value = 0, temp_time = 0;
  var index = 0;
  var last_value = 0, last_time = 0;
  data.forEach((item) => {
    console.log(item);
    var time = date_trans(item[4]);
    value = eval(item[7]);
    chart_data.push({
      time: time,
      value: value,
    });

    if (index == 0) {
      temp_value = value;
      temp_time = time;
    }
    index++;
    if (opt == 1) {
      offset_value1.push(value - temp_value);
      offset_time1.push(time - temp_time);
    }
    if (opt == 2) {
      offset_value2.push(value - temp_value);
      offset_time2.push(time - temp_time);
    }
```

```
    if (opt == 3) {
      offset_value3.push(value - temp_value);
      offset_time3.push(time - temp_time);
    }
    last_value = value;
    last_time = time;
  });


  return { 'main_data': chart_data, 'max_value': max_value, 'min_value': min_value,
'start_value': start_value, 'end_value': value, 'last_time': last_time, 'last_value': last_value
};
}

var date_trans = function (time) {

  var darr = time.split(" ");
  const [d, m, y] = darr[0].split("/");
  const [h, M] = darr[1].split(":");
  return new Date(y + "-" + m + "-" + d + " " + h + ":" + M + ":00").getTime()
}

// find critical value in decrease graph
var findArr1 = [], findArr2 = [], findArr3 = [];
var limitLoof = 0;
var limitLoofCount = 30;
var check_critical = function (critical_value, last_value, last_time, opt, once, grapType) {
  var findArr = [];
  var new_arr = [];
  var offset_value = (opt == 1 ? offset_value1 : (opt == 2 ? offset_value2 : offset_value3));
  var offset_time = (opt == 1 ? offset_time1 : (opt == 2 ? offset_time2 : offset_time3));
  var find = false, value = 0, time = 0;
  for (var i = 0; i < offset_value.length; i++) {

    value = eval(offset_value[i] + last_value);
    time = eval(offset_time[i] + last_time);

    if (opt == 1){
      findArr1.push({
        virtual: value,
        time: time,
      })
      findArr = findArr1;
    }

    else if (opt == 2){
      findArr2.push({
        virtual: value,
        time: time,
```

```
      })
      findArr = findArr2;
    }

    else if (opt == 3){
      findArr3.push({
        virtual: value,
        time: time,
      })
      findArr = findArr3;
    }


    if (once) {
      if (grapType == "decrease") {
        if (value >= critical_value) {

          find = true;
          break;
        }
      }
      else {
        if (value <= critical_value) {

          find = true;
          break;
        }
      }

    }
    else {
      if (grapType == "decrease") {
        if (value <= critical_value) {
          find = true;
          break;
        }
      }
      else {
        if (value >= critical_value) {
          find = true;
          break;
        }
      }
    }


  }
  limitLoof++;
  if (find == false) {
    if (once) {
```

```
      document.getElementById("critical_time_"+opt).innerText = ("Cannot find critical point.");
      console.log("not found");

    }
    else {
      if (limitLoof < limitLoofCount)
        check_critical(critical_value, value, time, opt, once, grapType);
      else {
        document.getElementById("critical_time_"+opt).innerText = ("Cannot find critical
point.");

      }
    }
  }
  else{
    if(findArr.length> 0){
      var estimated = findArr[findArr.length-1].time - findArr[0].time;
      document.getElementById("critical_time_"+opt).innerText = ((estimated / 3600 / 1000)+"
hours expected");
    }
  }
  if(find){
    if(opt == 1){
      dot_chart1_series.data.pushAll(findArr1);

    }
    if(opt == 2){
      dot_chart2_series.data.pushAll(findArr2);

    }
    if(opt == 3 ){
      dot_chart3_series.data.pushAll(findArr3);

    }
  }
  findArr1 = [];
  findArr2 = [];
  findArr3 = [];
}

var data_critical = function (chart_data, critical_value, opt) {

  var end_value = chart_data.end_value;
  var start_value = chart_data.start_value;
  // decrease or equal
  if (start_value >= end_value) {
    check_critical(critical_value, chart_data.last_value, chart_data.last_time, opt, (end_value
<= critical_value), "decrease");
  }
```

```
  // increase
  else if (start_value < end_value) {
    check_critical(critical_value, chart_data.last_value, chart_data.last_time, opt, (end_value > critical_value), "increase");
  }
}


var chart1, chart2,chart3;
var _draw_chart = function (index, data) {

  var root = am5.Root.new("chart_" + eval(index + 1));
  var chart = root.container.children.push(am5xy.XYChart.new(root, {
    panX: true,
    panY: true,
    wheelX: "panX",
    wheelY: "zoomX",
    layout: root.verticalLayout
  }));

  var cursor = chart.set("cursor", am5xy.XYCursor.new(root, {}));
  cursor.lineY.set("visible", false);


  var xAxis = chart.xAxes.push(am5xy.DateAxis.new(root, {
    maxDeviation: 1,
    baseInterval: {
      timeUnit: "minute",
      count: 30
    },
    renderer: am5xy.AxisRendererX.new(root, {}),
    tooltip: am5.Tooltip.new(root, {})
  }));

  var yAxis = chart.yAxes.push(am5xy.ValueAxis.new(root, {
    maxDeviation: 0.1,
    renderer: am5xy.AxisRendererY.new(root, {})
  }));

  function createSeries(field, name, color, dashed) {
    var series = chart.series.push(am5xy.SmoothedXLineSeries.new(root, {
      name: name,
      xAxis: xAxis,
      yAxis: yAxis,
      valueYField: field,
      valueXField: "time",
      stroke: color,
      tooltip: am5.Tooltip.new(root, {
        pointerOrientation: "horizontal",
        getFillFromSprite: false,
```

```
        labelText: "[bold]{name}[/]\n{valueX}: [bold]{valueY}[/]"
      })
    }));

    series.get("tooltip").get("background").setAll({
      fillOpacity: 0.7,
      fill: color,
      pointerBaseWidth: 0
    });

    series.strokes.template.setAll({
      strokeWidth: 2
    });

    if (dashed) {
      series.strokes.template.set("strokeDasharray", [5, 3]);
    }

    series.data.setAll(data);
    series.appear(10);

    return series;
  }


  if(index == 0){
    chart1 = createSeries("value", "Density " + eval(index + 1), am5.color(0x6610f2));;
     dot_chart1_series = createSeries("virtual", "Forecast", am5.color(0xdc3545), true);
  }

  if(index == 1){
    chart2 = createSeries("value", "Density " + eval(index + 1), am5.color(0x0d6efd));;
    dot_chart2_series = createSeries("virtual", "Forecast", am5.color(0xdc3545), true);
  }

  if(index == 2){
    chart3 = createSeries("value", "Density " + eval(index + 1), am5.color(0x198754));;
    dot_chart3_series = createSeries("virtual", "Forecast", am5.color(0xdc3545), true);
  }


  root.dateFormatter.setAll({
    dateFormat: "MM/dd HH:mm",
    dateFields: ["valueX"]
  });


  // Add legend
  // https://www.amcharts.com/docs/v5/charts/xy-chart/legend-xy-series/
  var legend = chart.children.push(
```

```
    am5.Legend.new(root, {
      centerX: am5.p50,
      x: am5.p50
    })
  );

  legend.data.setAll(chart.series.values);
  chart.appear(1000, 100);

}

var _init_chart = function (dataArray) {
  am5.ready(function () {

    dataArray.forEach((data, index) => {
      _draw_chart(index, data);
    })

  }); // end am5.ready()

}
return {
  init: function () {

    _load_data();
  },
  initEvent: function (ind) {
    _init_event(ind);
  }
}
}();
```