2023 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm) | 978-1-6654-5556-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/SmartGridComm57358.2023.10333923

Microcomputer Prototyping of IEC61850-9-2 with Performance Analysis

Nikolai Galkin*, Chen-Wei Yang*, Nicholas Etherden†‡, Valeriy Vyatkin*§
*Department of Computer Science, Computer and Space Engineering, Luleå Tekniska Universitet, Sweden
‡Department of Engineering Sciences and Mathematics, Luleå Tekniska Universitet, Sweden
†Vattenfall R&D, SE-169 79 Solna, Sweden
§Department of Electrical Engineering and Automation, Aalto University, Espoo, Finland

Email: nikolai.galkin@ltu.se, chen-wei.yang@ltu.se, nicholas.etherden@vattenfall.com, vyatkin@ieee.org

***Abstract*—The reliability of an electrical system depends on layered Protection, Automation, and Control (PAC) functions in which differential relays are considered a fundamental method to isolate the problem. By having a PAC architecture with functionality independent of hardware we hypothesize that the life cycle of PAC devices can be extended by using modern, powerful, and highly reconfigurable IoT devices. This can be achieved by applying modularity to hardware architecture, where an IoT device can be considered a mezzanine module. The Raspberry Pi and BeagleBone manufacturers are keeping the hardware the same size while constantly increasing its power. However, it is not clear how powerful the IoT device should be to meet some of the IEC 61850 requirements (which are usually included in PAC).**

In this article, we provide a step-by-step process for designing a system that mimics a single-phase transformer differential protection system. First, we provide a requirements analysis (Sampled Values rate, traffic synchronization, reaction time, etc.). Then we analyze the *BeagleBone Black* device (BBB). We give an example of an electrical system and finally analyze, in terms of IEC 61850 requirements, the performance of two software applications developed for the BBB (based on the operating system and on bare metal).

***Index Terms*—IEC 61850, SV, Merging Unit, BeagleBone Black, Bare Metal Programming.**

I. Introduction

The working group B5.60 of Cigre Study Committee B5 (Power System Protection and Local Control) describes architectures for Protection, Automation, and Control (PAC) based on application functions being independent of hardware (FIH). According to CIGRE technical brochure 891, the current life cycle of the Protection, Automation, and Control (PAC) hardware is typically estimated at 10-15 years, while the primary equipment (like, for example, protection relays) life cycle is approximately 40-60 years [1]. The core functions of PACs remain the same but require more flexibility throughout their entire life cycle, even today when the use of the power grid is becoming more flexible due to the widespread integration of renewable energy sources that affect the directions of energy flows. PAC's upgrade may be performed using software applications that are likely to have a longer lifespan than the underlying hardware. However, users were unable to maintain their PAC applications for the lifetime of their underlying hardware due to hardware dependency. The development of new PACS designs and new methods of building, commissioning and maintaining them, including upgrades and end-of-life replacement, will be made possible by separating the functions of applications and equipment.

In the current work, we hypothesize that this problem can be solved by applying a modular hardware design based on modern IoT devices that can be considered as a mezzanine module that can be replaced by another, more powerful device, that has the same hardware outline. Here we mainly refer to popular one-board microcomputers like Raspberry Pi and BeagleBone, where a lot of models have the same size (of a credit card).

A differential relay (DR) is a device that protects a dedicated zone of an electrical network from faults. The classical example of the current balance differential protection for a transformer is represented in Figure 1. The basic principle of operation is based on Kirchhoff's first circuit law, which states that the amount of current flowing into an electrical node is always equal to the amount of current flowing out of the node. If this statement is not true, the electrical system is considered unsafe and the faulty component must be isolated.

This principle lies in the basis of differential protection relays. It was invented at the beginning of the 20th century (Merz and Price, proposed longitudinal current comparison protection in 1903 [2]) and its basic operation principles did not change even now. What is changing is the requirements for the reliability of electrical systems and the safety of their operation. Rapid progress in the field of electronic design means also communication needs from PAC devices are increasing (see Figure 2), forcing us to reconsider the existing requirements for electrical networks. In particular, the IEC 61850 standard in part 9-2LE [3] defines the format (syntax and semantics) for transmission of digital current and voltage samples, while part 5 [4] together with the user group implementation guideline specifies the requirements for the sampling and transmission of electrical system parameters. Some of them we have presented below:

1) Analog signal sample rate: 80 Samples Per Cycle (SPC) for protection applications and 256 SPC for metering and monitoring applications (*per cycle* means @50Hz or @60Hz power line frequency).

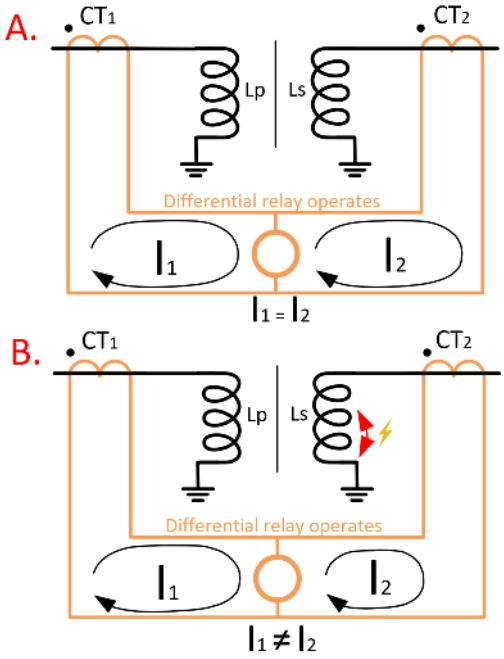

Fig. 1. Ideal transformer example to understand principles of differential relay operation: A) normal conditions, B) Short circuit in the secondary coil.

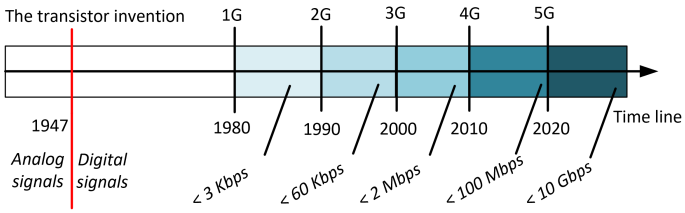

Fig. 2. Evolution of the information transfer rate.

2) Time synchronization: less than 1 µs with respect to the grand-master clock.
3) Number of analog signals: up to 8 (4 voltage signals + 4 current signals in case of a three-phase system plus neutral).

The implementation of the IEC 61850 stack is well-known[1]. What we lacked was the performance analysis of the IoT device in terms of IEC 61850 standard requirements. Because of that, the main focus of our work is to study the hardware requirements for implementing IEC 61850 SVs on IoT devices. Additionally, we believe, that the work results can potentially help other researchers to become more familiar with IEC 61850.

A. Scope of the article

In this article, we demonstrate the design process of the device that mimics the behaviour of a single-phase merging unit (MU). MU was defined in the standard in section IEC 61850-9-2 and further specified for stand-alone units in the instrument transformer standard IEC 61869-13 [5]. The MU

[1]https://libiec61850.com

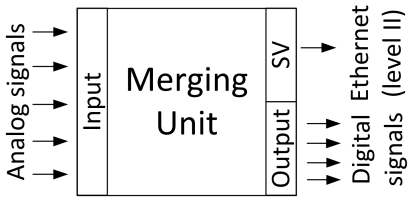

Fig. 3. IEC 61850-9-2 Merging Unit structure.

is a device that converts analogue signals from the primary equipment (process level) into IEC 61850 Sampled Values (SV) and transfers them to the bay/unit level (see Figure 3). Providing other researchers and engineers with a free and open-source tool that could help them validate the design scheme of a future IEC 61850 system is considered to be the main practical result of such work.

B. Outline of the article

The rest of the article is organized as follows: Section II describes the related works, Section III presents the properties of the BBB and the updates that need to be made to safely integrate it into a specific electrical system, Section IV presents the electrical circuit of the developed fusion unit, Section V briefly describes the software application that was used during the tests. Section VI summarizes the experiment and draws a conclusion and finally, Section VII demonstrates the conclusion and further work.

II. Related works

First, we would like to highlight the sustained interest in using IoT devices in IEC 61850 applications. S. Lu *et al.* demonstrate the advantages of using Raspberry Pi hardware as an aggregator operating according to IEC61850-related communication standard, known as *Generic Object Oriented Substation Event* or GOOSE [6]. Additionally, to that, Y. Lin *et al.* propose a reduced IEC 61850 SV message structure with the optimized *Application Service Data Unit* (or ASDU) length in order to provide a reliable IOT communication protocol [7].

The work of Mariana *et al.* investigated the applicability of Raspberry Pi IoT Device as a Back-Up Protection Device in Medium Voltage Power Systems [8]. But unlike our use case, the work of Mariana *et al.* is considering a Raspberry Pi as an IEC 61850 Unit-Level device. It means that Raspberry Pi plays the role of interoperability gateway between Merging Unit and SCADA (e.g., between Proess and Station Levels). Additionally, in [8] there is no clarification on how many signals they were analyzing simultaneously. Furthermore, the authors claim that in the Linux OS of the IoT device, they scheduled some of the application instructions to the highest priority (with the help of *pThread* library). In our research, in Section VI we are going to demonstrate that neither task rescheduling nor the use of *Programmable Real-Time Units* (or PRU) can give the level of signal time synchronization accuracy required by the IEC61850-5.

The work of Pereira *et al.* demonstrates the ability of Rasberry Pi integration into the Processing Level by reading analogue data and producing data packets into the cloud-based system [9]. That work shows the system where Raspberry Pi plays the role of a low-frequency analogue converter into HTTP-based data packages (supported by the remote cloud system). This work does not take into account the protection requirements of IEC 61850 and the minimum system requirements that IEC 61850 implied.

Finally, Burbano *et al.* presented their research where they tested Raspberry Pi 2B hardware in a roll of IEC 61850 Bay Level device where they considered the photovoltaic panel as a Process Level equipment [10]. However, the authors do not give any specific characteristics about their system but only say in conclusion that *"... article has provided enough functionality ..." and that "... the data exchange rate was high enough for a second layer control level..."*. Therefore, for us, it was still not clear how probably the most strict demand of the IEC 61850-5 (signal processing latency in order of 1 ms) can be fitted into the IoT device such as *Raspberry Pi* or *BeagleBone*.

III. Analysis of initial conditions

A. Electrical system (input conditions)

The test that we present was carried out Luleå University of Technology (Sweden). Sweden uses a frequency of 50 Hz to transmit electricity through lines. The network consists of about 17,000 km of 400 kV and 220 kV lines with stations and foreign lines [11]. The final RMS voltage that reaches consumers has an amplitude of 400 volts phase-to-phase (230 volts phase-to-ground). Forming a low voltage grid typically for some tens of customers from the one of the 200 000 20/0.4 or 10/0.4 kV distribution transformer in Sweden. The total length of medium (10 to 130 kV) and low voltage lines (0.4 kV) is about 25 times the length of the transmission line with the low voltage network forming the majority of the line length. The initial electric conditions of the designed system are presented in Table I

TABLE I
INITIAL ELECTRICAL PARAMETERS

N	Parameter	Value
1	Voltage (phase to ground), V	± 230 (+10% operational range)
2	Power line frequency, Hz	50

B. IEC 61850-9-2 demands to the Merging Units (output conditions)

SV message is a set of data designed to support data streaming from sensors (information source) to Bay-level devices (where this information is used). SV streaming is always one-way, which means the MU is never in line listening mode (unlike another type of IEC 61850 message known as GOOSE, which works in two ways). SV messages are usually implemented as Layer II Ethernet packets (although UDP and TCP are also an option) to reduce communication latency. IEC 61850-9-2 implementation guideline specifies the sampling rate in units of sample per period (cycle) for protection (80 samples/cycle) and power quality (256 samples/cycle). Thus, at @50 Hz and 80 samples per cycle, the required throughput of the communication line should be a minimum of 5.3 Mbps to provide reliability. If data from different devices have to be processed together, they should be synchronized with reasonably high precision. The requirement classes (see Table II) for time synchronization are given in 61850-5 (section 11.1.3.3).

TABLE II
CLASSES FOR TRANSFER TIMES DEFINED IN IEC 61850-5

Transfer time class	Transfer time [ms]	Application examples:
TT0	>1 000	Files, events, log contents
TT1	1 000	Events, alarms
TT2	500	Operator commands
TT3	100	Slow automatic interactions
TT4	20	Fast automatic interactions
TT5	10	Releases, status changes
TT6	3	Trips, blockings

In the case of our article, since we are dealing with the recording and transfer of analogue signals, class TT6 is requested. This was defined to allow multiple MUs (up to 6) to be connected to a single Bay-level device (see Figure 4). To avoid false interference, all devices on the same bus must operate in the same period. The synchronization accuracy between master and slave must be less than 1 µs. Synchronization of a master with a slave is usually implemented through the Protection Time Protocol (PTP) which potentially can provide accuracy up to several nanoseconds (ideal case). A summary of demands is presented in Table III.

TABLE III
INITIAL IEC 61850-9-2 DEMANDS

N	Parameter	Value
1	Minimum sampling rate for analogue signals, Hz	4000
2	Number of analogue inputs, (min. - max.)	2 - 8
4	Minimum accepted communication speed, Mbps	10

C. BeagleBone Black hardware characteristics

BBB is a low-cost, community-supported platform developed by developers and for developers. A short summary of this device is presented in Table IV.

So far, we have found one limitation of the target device, namely the number of analogue inputs. One BBB cannot be used to monitor a 3-phase electrical system (with 8 sensors). Therefore, with the idea of simplifying the use case, we deliberately decided to reduce the number of controlled AC lines from three to one.

IV. Design of the MU prototype

Summarizing data in Tables I - IV the electrical scheme presented in Figure 5 was designed. Here, the voltage transducer

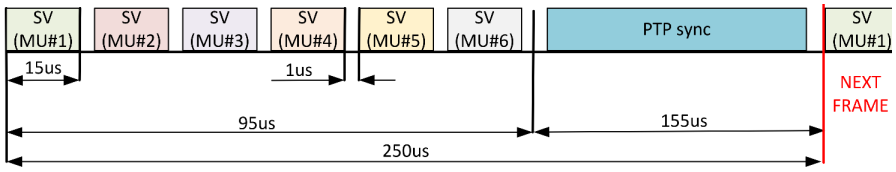

Fig. 4. An example of communication traffic between Process and Bay levels.

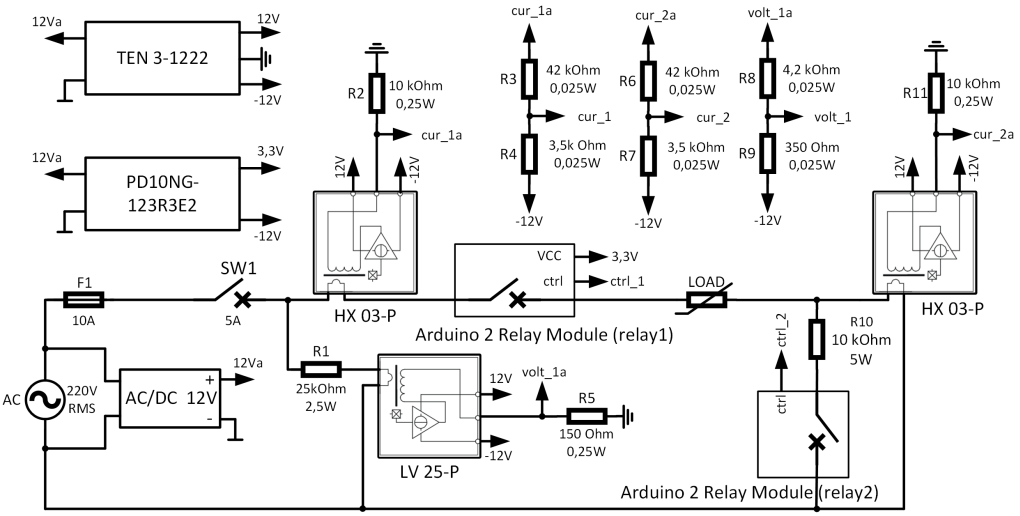

Fig. 5. Electrical scheme of the one-phase differential protection system.

TABLE IV
BEAGLEBONE BLACK HARDWARE PROPERTIES

N	Parameter	Value
1	Microprocessor	AM3358BZCZ100 1GHz, 2000 MIPS ARM Cortex-A8 32-Bit
5	Analog inputs	7, 12-bit resolution
6	The maximum sample rate of the ADC	200 kSPS
7	Maximum amplitude of the analog signal	1.8 V
8	Re-configurable DIO	65

LV 25-P and the current transducer *HX 03-P* were chosen because they have galvanic isolation between the primary and secondary circuits, they meet initial electrical parameters defined in Table I, and they have a relatively small form factor. Arduino's 2 Relay module is used for two purposes. One relay is used to disconnect a load from the electrical source (power outlet). The designed prototype of the differential protection is galvanically isolated both by power and signal parts. Resistors R3 - R9 were used to bring analogue signals to a 1.8V maximum level (as it is specified in Table IV). Finally, in Figure 5 the *LOAD* component is specified as a variable resistor. In our case, we were using an AC/DC power source (MDR-100-24) and an electronic load (EA-EL 3200-25 B).

V. SOFTWARE APPLICATION DESIGN

BBB is hardware where two different code implementation approaches can be applied. One of them is to run some code on top of the operating system (OS). In addition to the first approach, we investigate a modified version of the Linux kernel in which the real-time programmable (PRU) modules are loaded by the developed application tasks. Another approach is to deploy execution instructions directly on the chip without any OS (so-called *bare metal* design). The advantage of the first method is that it makes it possible to use libraries with a high level of abstraction (without being tied to the hardware configuration). The price to pay for this is that all applications on top of the OS work slower than those on top of the chip.

Since the main contribution of this work is to provide analysis of device performance, below we will briefly review the main functions and capabilities of the developed application. In addition, to make the results more presentable, all applications presented in this paper are publicly available (see footnotes 2 and 3).

A. OS-based application design

For OS-based implementation of the application, we used the IEC 61850 library developed at the Luleå University of Technology[2]. First, we need to create a new IEC 61850-9-2 *SV* object (SV structure is presented in Figure 6). This object provides access to the main methods related to building and sending a new SV message. This is done by following:

sv_header sv;

[2]https://github.com/mrv-king/BBB_SV-GOOSE

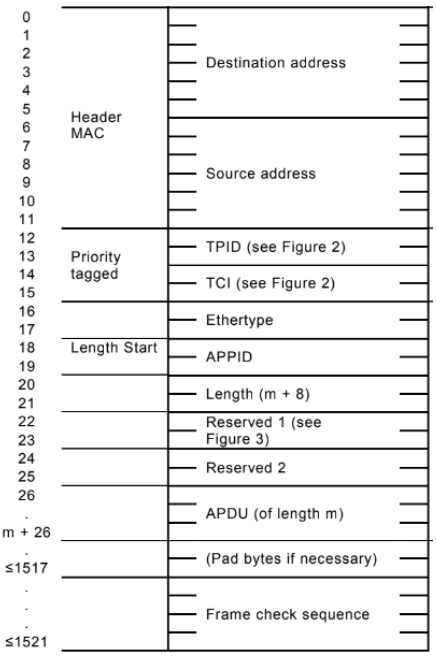

Fig. 6. SV frame structure defined by IEC 61850-9-2 standard.

After that, we need to specify which BBB interface and what socket we are going to use (*eth0* in our case). The code snippet below gives an example:

```
sockaddr_ll sockaddr;
typedef boost::asio::generic::raw_protocol raw_protocol_t;
/* Some code is missing for shortage */
raw_protocol_t (PF_PACKET, SOCK_RAW));
d_socket.bind(raw_endpoint_t (&sockaddr,
sizeof(sockaddr)));
```

Next, we need to build a new SV dataset that we are going to send. To do this, we need to specify four main parts, namely: source and destination MAC addresses, application ID, and analogue values. The code snippet below gives an example:

```
std::string src_mac_addr="01:23:45:67:89:AB";
std::string dst_mac_addr = "01:23:45:67:89:CD";
int APPID = 16384;
std::string sv_data = "3e2000003e200001";
sv.compose_frame(src_mac_addr, dst_mac_addr,
APPID, sv_data);
sv_frame = sv.get_ethernetII_frame();
d_socket.send (sv_frame);
```

B. Bare metal application design

Contrary to the high-level abstraction presented above, where we were working with defined objects and functions, the bare metal coding approach demands more detailed knowledge about the hardware. In this work, we used a toolset called *StarterWare* (designed by *Texas Instruments*). By using *StarterWare* toolset we implemented the same functionality that we did in the case of the OS-based application described in the previous subsection. The designed code is publically available on GitHub[3]

VI. Experiments. Performance analysis.

A. Application comparison. Performance analysis.

The designed device is aiming to implement the following functions:

1) *Reading*. Read three sensors.
2) *Data quality*. Determine the validity and detailed quality of the sampled value according to IEC 61850-7-3. (The detailed quality requires identification and tagging of sampled value in case of data overflow, *outOfRange*, *badReference*, oscillatory input data, failure of supervision function, *oldData* and inconsistent or inaccurate data).
3) *Analysing*. Compare recorded values against some limits.
4) *Transmit*. Send a new SV message.
5) *Acting*. Change the state of the output.
6) *Reconfiguration*. Accompanied by a configuration file to allow off-line configuration of the protection and control system where the sampled value streams can be pre-allocated to IEDs during the engineering phase of the system

A pseudo-code of the designed application work is presented in Algorithm 1

Algorithm 1 An application, mimicking the MU work

Require: GPIO, ADC, and Ethernet initialization
Ensure: Initialization is successful
  GPIO = HIGH;
  float current1, current2, voltage = 0;
  while $True$ **do**
    read ADC_ch0, ADC_ch1, ADC_ch2;
    copy values to current1, current2, voltage;
    if $|current1 - current2| >$ threshold1 **then**
      fault_flag = True;
    end if
    send SV frame;
    if fault_flag = True AND $|voltage| <$ threshold2 **then**
      GPIO = LOW;
    end if
  end while

To implement the performance analysis we have recorded the time that BBB needs to implement these functions above. We started by changing the state of the digital output only (and wrote down the time delta). Then we added analogue signals reading functionality (and wrote down the new time delta). The main idea is to analyze how long each functionality takes separately and estimate how much time it takes to implement them all. Table V demonstrates the results.

[3]https://github.com/mrv-king/BeagleBoneBlack-Ethernet-ADC-GPIO

TABLE V
PERFOMANCE ANALYSIS OF THE DESIGNED APPLICATIONS RUNNING ON BBB

Functionality	Application type		
	OS-based	*OS+PRU*	*Bare metal*
Change one DO, (δ_d), µs	80	70	1
Read 3 analog signals (δ_a, µs	800	580	15
Send one SV message (δ_s), µs	150	120	50
Summ (δ_d + δ_a + δ_s), µs	≈ 1000	≈ 800	≈ 65

The results shown in Table V show that the bare metal implementation is 70-80 times faster at changing the output state and overall 40-50 times faster at converting the analogue signal. Based on that information we have come to the conclusion that BBB is not powerful enough to implement IEC 61850-related functions on top of the OS, but it is powerful enough to implement the same functionality on top of the chip (as a bare metal application).

B. BBB mimicking MU functions. Performance analysis.

Finally, the BBB device with the deployed application was integrated with the test bed whose scheme is presented in Figure 5. To produce the difference between two current sensor values the relay #2 should be activated (normally closed). To do this, we used a button connected to the *ctrl_2* input. In our test, we pressed the button manually, so the resulting current difference occurred at random times relative to the time period of the utility grid. If the difference in current is detected, the BBB is waiting for the moment when the voltage amplitude of the phase is close to 0 (with the error of about 10-30V) and sends the command for switching off the protection relay#1. Figure 7 below is demonstrating the proof of the designed system concept.

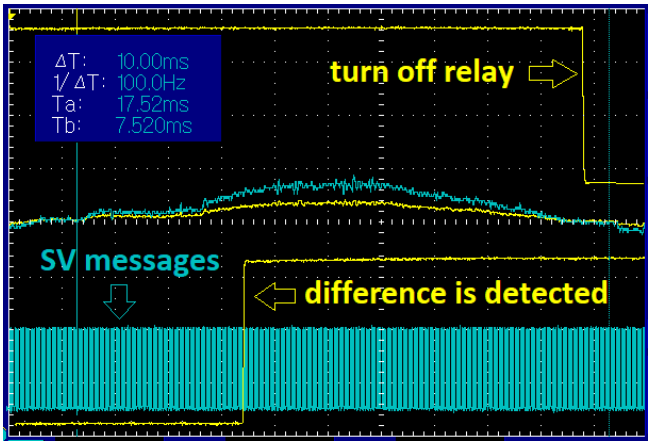

Fig. 7. Imitation of current difference and the reaction of the system.

The resulting structure of IEC 61850-9-2 related SV messages looks like in Figure 8. We want to emphasize once again that the main goal of our work is a mock-up design to compare bare metal and OS time delays on the application level (hardware + software). In our case, we encrypted all quality attributes to their default values (how it is specified in IEC 61850-7-3). Therefore it can not be fully considered a 61850-compatible device.

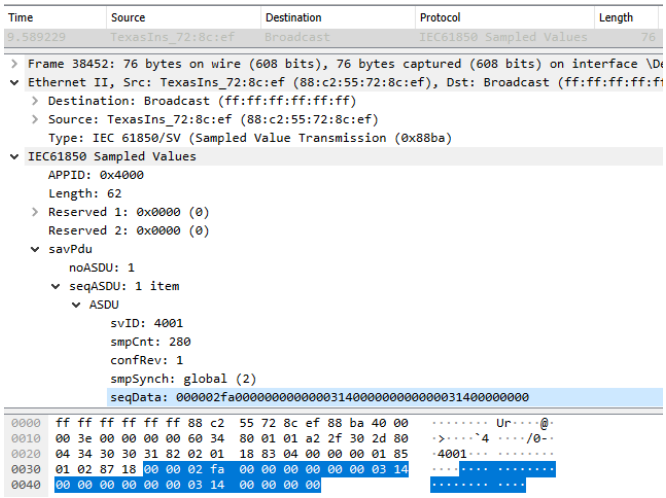

Fig. 8. SV message structure captured by the Wireshark.

VII. CONCLUSION AND FURTHER WORK

This work reports on the work of implementing the IEC61869-13 stand-alone Merging Unit sending IEC 61850-9-2 sampled values implemented on the BBB IoT device. The performance analysis of the developed device is based on two different types of software applications, namely one of them runs on top of the operating system and another is a bare metal (no operating system).

To test the performance of the developed system based on the BBB device we build a small test bed. Tests that we have made show that BBB's processor is not powerful enough to implement IEC 61850-9-2 specified functions on top of the OS (even upgraded with the real-time patch). The results shown in our article demonstrate that bare metal hardware applications is required on today's popular one-board microcomputers to meet the stringent requirements of IEC 61850-9-2 (because in this case, the processor is only busy executing the instructions specified in the developed software application).

The work done provides a better understanding of the hardware and software requirements that should be applied to solutions related to IEC 61850-9-2. Another requirement of the IEC 61850-9-2 standard - synchronization was discussed in this article, but was not implemented in the developed application. Therefore, Time Protection Protocol (PTP) integration and performance analysis of this feature is also considered a continuation of this work.

VIII. ACKNOWLEDGEMENT

This study has been supported, in part, by the Swedish Innovation Agency (VINNOVA) project VISA/5G, grant 2022-03002.

REFERENCES

[1] CIGRE TB-891 technical brochure on Protection, Automation, and Control Architectures with Functionality Independent of Hardware. [ONLINE] https://e-cigre.org/publication/wbn043-protection-automation-and-control-architectures-with-functionality-independent-of-hardware. Accessed: April 2023.

[2] History of protection engineering. *Electrical Engineering Academy*. [ONLINE] https://uploads-ssl.webflow.com/5ca3336faf774a29292763d4/5fd7ca4c758d5cb027e25bfb_2021.03.06.pdf. Accessed: April 2023.

[3] Implementation Guideline for Digital Interface to Instrument Transformers using IEC 61850-9-2.

[4] IEC 61850-5, Second edition, 2013

[5] IEC 61850-13, First edition, 2020

[6] S. Lu, S. Repo, J. Tjäder, A. Kjellström, M. Bollen and N. Etherden, "IEC 61850-based communication and aggregation solution for demand-response application," 2016 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), Ljubljana, Slovenia, 2016, pp. 1-6, doi: 10.1109/ISGTEurope.2016.7856283.

[7] Y. Lin and N. Xie, "IEC61850 Sample Value Service Based on Reduced Application Service Data Unit for Energy IOT," 2022 5th International Conference on Power and Energy Applications (ICPEA), Guangzhou, China, 2022, pp. 542-547, doi: 10.1109/ICPEA56363.2022.10052341.

[8] B. P. Kandamulla Arachchige, M. Mariana, C. Petrescu and C. Zafiu, "Applicability of Raspberry Pi IOT Device as Back Up Protection Device in Medium Voltage Power Systems," 2021 10th International Conference on ENERGY and ENVIRONMENT (CIEM), Bucharest, Romania, 2021, pp. 1-5, doi: 10.1109/CIEM52821.2021.9614777.

[9] R. Pereira, I. Dupont, P. Carvalho, S. Jucá, "IoT embedded Linux system based on Raspberry Pi applied to real-time cloud monitoring of a decentralized photovoltaic plant, " Measurement, Volume 114, 2018, Pages 286-297, ISSN 0263-2241, https://doi.org/10.1016/j.measurement.2017.09.033.

[10] R. A. G. Burbano, M. L. O. Gutierrez, J. A. Restrepo and F. G. Guerrero, "IED Design for a Small-Scale Microgrid Using IEC 61850," in IEEE Transactions on Industry Applications, vol. 55, no. 6, pp. 7113-7121, Nov.-Dec. 2019, doi: 10.1109/TIA.2019.2938734.

[11] Technology - Svenska kraftnät. [ONLINE] https://www.svk.se/en/grid-development/the-construction-process/technology/ Accessed: April 2023.